
Amazon Redshift

Cluster Management Guide

API Version 2012-12-01



Amazon Redshift: Cluster Management Guide

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Redshift?	1
Are You a First-Time Amazon Redshift User?	1
Amazon Redshift Management Overview	1
Cluster Management	2
Cluster Access and Security	2
Monitoring Clusters	3
Databases	4
Clusters	5
Overview	5
Clusters and Nodes	6
Migrating From DC1 Node Types to DC2 Node Types	7
Node Type Details	7
Determining the Number of Nodes	9
Resizing a Cluster	9
Supported Platforms to Launch Your Cluster	10
EC2-Classic Platform	11
EC2-VPC Platform	11
Choose a Platform	11
Regions and Availability Zone Considerations	11
Maintenance Windows	12
Default Disk Space Alarm	12
Renaming Clusters	13
Shutting Down and Deleting Clusters	14
Cluster Status	14
Managing Clusters Using the Console	15
Creating a Cluster	17
Modifying a Cluster	25
Deleting a Cluster	27
Rebooting a Cluster	28
Resizing a Cluster	29
Getting Information About Cluster Configuration	30
Getting an Overview of Cluster Status	30
Taking a Snapshot of a Cluster	31
Editing the Default Disk Space Alarm	32
Working with Cluster Performance Data	33
Managing Clusters Using the AWS SDK for Java	33
Manage Clusters Using the Amazon Redshift CLI and API	35
Managing Clusters in an Amazon Virtual Private Cloud (VPC)	36
Overview	36
Creating a Cluster in a VPC	37
Managing VPC Security Groups for a Cluster	39
Cluster Subnet Groups	40
Choosing Cluster Maintenance Tracks	45
Setting the Maintenance Track for a Cluster with the Console	46
Setting the Maintenance Track for a Cluster with the AWS CLI	47
Managing Cluster Versions	47
Cluster Version History	49
Versions 1.0.4349	49
Versions 1.0.3945, 1.0.4081, 1.0.4222	50
Versions 1.0.3639, 1.0.3688	50
Versions 1.0.3324, 1.0.3359	50
Version 1.0.3025	51
Versions 1.0.2762, 1.0.2819, 10.2842	51
Versions 1.0.2524, 1.0.2557, 1.02610, 1.0.2679, 1.02719	52

Version 1.0.2294, 1.0.2369	53
Version 1.0.2058	53
Version 1.0.2044	54
Versions 1.0.1792, 1.0.1793, 1.0.1808, 1.0.1865	54
Versions 1.0.1691 and 1.0.1703	55
Version 1.0.1636	55
Querying a Database	57
Querying a Database Using the Query Editor	57
Considerations	57
Enabling Access	58
Using the Query Editor	58
Connecting to a Cluster Using SQL Client Tools	60
Configuring Connections in Amazon Redshift	61
Configure a JDBC Connection	62
Configure an ODBC Connection	77
Configure Security Options for Connections	95
Connecting to Clusters from Client Tools and Code	100
Troubleshooting Connection Issues in Amazon Redshift	109
Enhanced VPC Routing	115
Working with VPC Endpoints	116
Enabling Enhanced VPC Routing	116
Using Amazon Redshift Spectrum with Enhanced VPC Routing	118
Considerations for Using Enhanced VPC Routing for Redshift Spectrum	119
Parameter Groups	122
Overview	122
About Parameter Groups	122
Default Parameter Values	122
Configuring Parameter Values Using the AWS CLI	123
Configuring Workload Management	124
WLM Dynamic and Static Properties	125
Properties in the <code>wlm_json_configuration</code> Parameter	125
Configuring the <code>wlm_json_configuration</code> Parameter Using the AWS CLI	128
Managing Parameter Groups Using the Console	133
Creating a Parameter Group	134
Modifying a Parameter Group	134
Creating or Modifying a Query Monitoring Rule Using the Console	136
Deleting a Parameter Group	140
Associating a Parameter Group with a Cluster	141
Managing Parameter Groups Using the AWS SDK for Java	141
Managing Parameter Groups Using the Amazon Redshift CLI and API	144
Snapshots	145
Overview	145
Automated Snapshots	145
Manual Snapshots	146
Excluding Tables From Snapshots	146
Copying Snapshots to Another Region	146
Restoring a Cluster from a Snapshot	147
Restoring a Table from a Snapshot	147
Sharing Snapshots	150
Managing Snapshots Using the Console	151
Creating a Manual Snapshot	152
Deleting a Manual Snapshot	153
Copying an Automated Snapshot	153
Restoring a Cluster from a Snapshot	154
Sharing a Cluster Snapshot	156
Configuring Cross-Region Snapshot Copy for a Non-Encrypted Cluster	157
Configure Cross-Region Snapshot Copy for an AWS KMS-Encrypted Cluster	157

Modifying the Retention Period for Cross-Region Snapshot Copy	158
Disabling Cross-Region Snapshot Copy	159
Managing Snapshots Using the AWS SDK for Java	159
Managing Snapshots Using the Amazon Redshift CLI and API	161
Database Encryption	163
Database Encryption for Amazon Redshift Using AWS KMS	163
Copying AWS KMS-Encrypted Snapshots to Another Region	164
Encryption for Amazon Redshift Using Hardware Security Modules	165
Configuring a Trusted Connection Between Amazon Redshift and an HSM	165
About Rotating Encryption Keys in Amazon Redshift	166
Migrating to an Encrypted Cluster	167
Configuring Database Encryption Using the Console	168
Configuring Amazon Redshift to Use an HSM Using the Amazon Redshift console	169
Rotating Encryption Keys Using the Amazon Redshift console	173
Configuring Database Encryption Using the Amazon Redshift API and AWS CLI	174
Configuring Amazon Redshift to Use AWS KMS Encryption Keys Using the Amazon Redshift API and AWS CLI	174
Configuring Amazon Redshift to use an HSM Using the Amazon Redshift API and AWS CLI	174
Rotating Encryption Keys Using the Amazon Redshift API and AWS CLI	175
Purchasing Reserved Nodes	176
Overview	176
About Reserved Node Offerings	176
Comparing Pricing Among Reserved Node Offerings	177
How Reserved Nodes Work	177
Reserved Nodes and Consolidated Billing	178
Reserved Node Examples	178
Purchasing a Reserved Node Offering with the Console	179
Upgrade a Reserved Node	181
Purchasing a Reserved Node Offering Using Java	184
Purchasing a Reserved Node Offering Using the AWS CLI and Amazon Redshift API	186
Security	187
Authentication and Access Control	187
Authentication	187
Access Control	188
Overview of Managing Access	188
Using Identity-Based Policies (IAM Policies)	193
Amazon Redshift API Permissions Reference	201
Using Service-Linked Roles	201
Security Groups	203
Overview	204
Managing Cluster Security Groups Using the Console	204
Managing Cluster Security Groups Using the AWS SDK for Java	213
Manage Cluster Security Groups Using the Amazon Redshift CLI and API	216
Using IAM Authentication to Generate Database User Credentials	217
Overview	217
Creating Temporary IAM User Credentials	218
Create an IAM Role for IAM Single Sign-On (SSO) Access	218
Configure SAML Assertions for Your IdP	218
Create an IAM Role or User With Permissions to Call GetClusterCredentials	220
Create a Database User and Database Groups	221
Configure a JDBC or ODBC Connection to Use IAM Credentials	222
Options for Providing IAM Credentials	228
JDBC and ODBC Options for Providing IAM Credentials	229
Using a Credentials Provider Plugin	229
Using a Configuration Profile	230
JDBC and ODBC Options for Creating Database User Credentials	231
Generating IAM Database Credentials Using the Amazon Redshift CLI or API	232

Authorizing Amazon Redshift to Access AWS Services	235
Creating an IAM Role to Allow Your Amazon Redshift Cluster to Access AWS Services	235
Restricting Access to IAM Roles	236
Restricting an IAM Role to an AWS Region	237
Chaining IAM Roles	238
Related Topics	239
Authorizing COPY, UNLOAD, and CREATE EXTERNAL SCHEMA Operations Using IAM Roles	240
Associating IAM Roles with Clusters	240
Using the Amazon Redshift Management Interfaces	245
Using the AWS SDK for Java	246
Running Java Examples Using Eclipse	246
Running Java Examples from the Command Line	247
Setting the Endpoint	247
Signing an HTTP Request	248
Example Signature Calculation	249
Setting Up the Amazon Redshift CLI	250
Installation Instructions	250
Getting Started with the AWS Command Line Interface	251
Monitoring Cluster Performance	255
Overview	255
Performance Data	256
Amazon Redshift Metrics	256
Amazon Redshift Query/Load Performance Data	260
Working with Performance Data	261
Viewing Cluster Performance Data	261
Viewing Query Performance Data	264
Viewing Cluster Metrics During Load Operations	272
Analyzing Workload Performance	273
Creating an Alarm	276
Working with Performance Metrics in the Amazon CloudWatch Console	278
Events	280
Overview	280
Viewing Events Using the Console	280
Filtering Events	280
Viewing Events Using the AWS SDK for Java	281
View Events Using the Amazon Redshift CLI and API	282
Event Notifications	283
Overview	283
Amazon Redshift Event Categories and Event Messages	284
Managing Event Notifications Using the Amazon Redshift Console	292
Managing Event Notifications Using the Amazon Redshift CLI and API	297
Database Audit Logging	298
Overview	298
Amazon Redshift Logs	298
Connection Log	299
User Log	299
User Activity Log	300
Enabling Logging	300
Managing Log Files	301
Bucket Permissions for Amazon Redshift Audit Logging	301
Bucket Structure for Amazon Redshift Audit Logging	303
Troubleshooting Amazon Redshift Audit Logging	303
Logging Amazon Redshift API Calls with AWS CloudTrail	304
Amazon Redshift Information in CloudTrail	304
Understanding Amazon Redshift Log File Entries	305
Amazon Redshift Account IDs in AWS CloudTrail Logs	307
Configuring Auditing Using the Console	308

Enabling Audit Logging Using the Console	308
Modifying the Bucket for Audit Logging	310
Disabling Audit Logging Using the Console	310
Configuring Logging by Using the Amazon Redshift CLI and API	310
Resizing Clusters	311
Overview	311
Resize Operation Overview	311
Snapshot, Restore, and Resize Operation Overview	312
Tutorial: Using the Resize Operation to Resize a Cluster	313
Prerequisites	313
Step 1: Resize the Cluster	313
Step 2: Delete the Sample Cluster	314
Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster	315
Prerequisites	315
Step 1: Take a Snapshot	315
Step 2: Restore the Snapshot into the Target Cluster	316
Step 3: Verify Data in the Target Cluster	318
Step 4: Resize the Target Cluster	318
Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster	319
Step 6: Rename the Source and Target Clusters	320
Step 7: Delete the Source Cluster	321
Step 8: Clean Up Your Environment	322
Limits	323
Quotas and Limits	323
Spectrum limits	323
Naming Constraints	324
Tagging	326
Tagging Overview	326
Tagging Requirements	327
Managing Resource Tags Using the Console	327
How To Open the Manage Tags Window	328
How to Manage Tags in the Amazon Redshift Console	329
Managing Tags Using the Amazon Redshift API	329
Document History	331

What Is Amazon Redshift?

Welcome to the *Amazon Redshift Cluster Management Guide*. Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. You can start with just a few hundred gigabytes of data and scale to a petabyte or more. This enables you to use your data to acquire new insights for your business and customers.

The first step to create a data warehouse is to launch a set of nodes, called an Amazon Redshift cluster. After you provision your cluster, you can upload your data set and then perform data analysis queries. Regardless of the size of the data set, Amazon Redshift offers fast query performance using the same SQL-based tools and business intelligence applications that you use today.

Are You a First-Time Amazon Redshift User?

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

- [Amazon Redshift Management Overview \(p. 1\)](#) – 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 walks you through the process of creating a cluster, creating database tables, uploading data, and testing queries.
- [Amazon Redshift Cluster Management Guide \(this guide\)](#) – This guide shows you how to create and manage Amazon Redshift clusters.
- [Amazon Redshift Database Developer Guide](#) – If you are a database developer, this guide explains how to design, build, query, and maintain the databases that make up your data warehouse.

There are several ways to manage clusters. If you prefer a more interactive way of managing clusters, you can use the Amazon Redshift console or the AWS Command Line Interface (AWS CLI). If you are an application developer, you can use the Amazon Redshift Query API or the AWS Software Development Kit (SDK) libraries to manage clusters programmatically. If you use the Amazon Redshift Query API, you must authenticate every HTTP or HTTPS request to the API by signing it. For more information about signing requests, go to [Signing an HTTP Request \(p. 248\)](#).

For information about the CLI, API, and SDKs, go to the following links:

- [AWS CLI Command Reference](#)
- [Amazon Redshift API Reference](#)
- SDK References in [Tools for Amazon Web Services](#).

Amazon Redshift Management Overview

The Amazon Redshift service manages all of the work of setting up, operating, and scaling a data warehouse. These tasks include provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.

Cluster Management

An Amazon Redshift cluster is a set of nodes, which consists of a leader node and one or more compute nodes. The type and number of compute nodes that you need depends on the size of your data, the number of queries you will execute, and the query execution performance that you need.

Creating and Managing Clusters

Depending on your data warehousing needs, you can start with a small, single-node cluster and easily scale up to a larger, multi-node cluster as your requirements change. You can add or remove compute nodes to the cluster without any interruption to the service. For more information, see [Amazon Redshift Clusters \(p. 5\)](#).

Reserving Compute Nodes

If you intend to keep your cluster running for a year or longer, you can save money by reserving compute nodes for a one-year or three-year period. Reserving compute nodes offers significant savings compared to the hourly rates that you pay when you provision compute nodes on demand. For more information, see [Purchasing Amazon Redshift Reserved Nodes \(p. 176\)](#).

Creating Cluster Snapshots

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: automated and manual. Amazon Redshift stores these snapshots internally in Amazon Simple Storage Service (Amazon S3) by using an encrypted Secure Sockets Layer (SSL) connection. If you need to restore from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot that you specify. For more information about snapshots, see [Amazon Redshift Snapshots \(p. 145\)](#).

Cluster Access and Security

There are several features related to cluster access and security in Amazon Redshift. These features help you to control access to your cluster, define connectivity rules, and encrypt data and connections. These features are in addition to features related to database access and security in Amazon Redshift. For more information about database security, see [Managing Database Security](#) in the *Amazon Redshift Database Developer Guide*.

AWS Accounts and IAM Credentials

By default, an Amazon Redshift cluster is only accessible to the AWS account that creates the cluster. The cluster is locked down so that no one else has access. Within your AWS account, you use the AWS Identity and Access Management (IAM) service to create user accounts and manage permissions for those accounts to control cluster operations. For more information, see [Security \(p. 187\)](#).

Security Groups

By default, any cluster that you create is closed to everyone. IAM credentials only control access to the Amazon Redshift API-related resources: the Amazon Redshift console, command line interface (CLI), API, and SDK. To enable access to the cluster from SQL client tools via JDBC or ODBC, you use security groups:

- If you are using the EC2-Classic platform for your Amazon Redshift cluster, you must use Amazon Redshift security groups.
- If you are using the EC2-VPC platform for your Amazon Redshift cluster, you must use VPC security groups.

In either case, you add rules to the security group to grant explicit inbound access to a specific range of CIDR/IP addresses or to an Amazon Elastic Compute Cloud (Amazon EC2) security group if your SQL client runs on an Amazon EC2 instance. For more information, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

In addition to the inbound access rules, you create database users to provide credentials to authenticate to the database within the cluster itself. For more information, see [Databases \(p. 4\)](#) in this topic.

Encryption

When you provision the cluster, you can optionally choose to encrypt the cluster for additional security. When you enable encryption, Amazon Redshift stores all data in user-created tables in an encrypted format. You can use AWS Key Management Service (AWS KMS) to manage your Amazon Redshift encryption keys.

Encryption is an immutable property of the cluster. The only way to switch from an encrypted cluster to a nonencrypted cluster is to unload the data and reload it into a new cluster. Encryption applies to the cluster and any backups. When you restore a cluster from an encrypted snapshot, the new cluster is encrypted as well.

For more information about encryption, keys, and hardware security modules, see [Amazon Redshift Database Encryption \(p. 163\)](#).

SSL Connections

You can use Secure Sockets Layer (SSL) encryption to encrypt the connection between your SQL client and your cluster. For more information, see [Configure Security Options for Connections \(p. 95\)](#).

Monitoring Clusters

There are several features related to monitoring in Amazon Redshift. You can use database audit logging to generate activity logs, configure events and notification subscriptions to track information of interest, and use the metrics in Amazon Redshift and Amazon CloudWatch to learn about the health and performance of your clusters and databases.

Database Audit Logging

You can use the database audit logging feature to track information about authentication attempts, connections, disconnections, changes to database user definitions, and queries run in the database. This information is useful for security and troubleshooting purposes in Amazon Redshift. The logs are stored in Amazon S3 buckets. For more information, see [Database Audit Logging \(p. 298\)](#).

Events and Notifications

Amazon Redshift tracks events and retains information about them for a period of several weeks in your AWS account. For each event, Amazon Redshift reports information such as the date the event occurred, a description, the event source (for example, a cluster, a parameter group, or a snapshot), and the source ID. You can create Amazon Redshift event notification subscriptions that specify a set of event filters. When an event occurs that matches the filter criteria, Amazon Redshift uses Amazon Simple Notification Service to actively inform you that the event has occurred. For more information about events and notifications, see [Amazon Redshift Events \(p. 280\)](#).

Performance

Amazon Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. Amazon Redshift uses Amazon CloudWatch metrics to

monitor the physical aspects of the cluster, such as CPU utilization, latency, and throughput. Amazon Redshift also provides query and load performance data to help you monitor the database activity in your cluster. For more information about performance metrics and monitoring, see [Monitoring Amazon Redshift Cluster Performance \(p. 255\)](#).

Databases

Amazon Redshift creates one database when you provision a cluster. This is the database you use to load data and run queries on your data. You can create additional databases as needed by running a SQL command. For more information about creating additional databases, go to [Step 1: Create a database](#) in the *Amazon Redshift Database Developer Guide*.

When you provision a cluster, you specify a master user who has access to all of the databases that are created within the cluster. This master user is a superuser who is the only user with access to the database initially, though this user can create additional superusers and users. For more information, go to [Superusers](#) and [Users](#) in the *Amazon Redshift Database Developer Guide*.

Amazon Redshift uses parameter groups to define the behavior of all databases in a cluster, such as date presentation style and floating-point precision. If you don't specify a parameter group when you provision your cluster, Amazon Redshift associates a default parameter group with the cluster. For more information, see [Amazon Redshift Parameter Groups \(p. 122\)](#).

For more information about databases in Amazon Redshift, go to the [Amazon Redshift Database Developer Guide](#).

Amazon Redshift Clusters

In the following sections, you can learn the basics of creating a data warehouse by launching a set of compute nodes, called an Amazon Redshift *cluster*.

Topics

- [Overview \(p. 5\)](#)
- [Clusters and Nodes in Amazon Redshift \(p. 6\)](#)
- [Resizing a Cluster \(p. 9\)](#)
- [Supported Platforms to Launch Your Cluster \(p. 10\)](#)
- [Regions and Availability Zone Considerations \(p. 11\)](#)
- [Maintenance Windows \(p. 12\)](#)
- [Default Disk Space Alarm \(p. 12\)](#)
- [Renaming Clusters \(p. 13\)](#)
- [Shutting Down and Deleting Clusters \(p. 14\)](#)
- [Cluster Status \(p. 14\)](#)
- [Managing Clusters Using the Console \(p. 15\)](#)
- [Managing Clusters Using the AWS SDK for Java \(p. 33\)](#)
- [Manage Clusters Using the Amazon Redshift CLI and API \(p. 35\)](#)
- [Managing Clusters in an Amazon Virtual Private Cloud \(VPC\) \(p. 36\)](#)
- [Choosing Cluster Maintenance Tracks \(p. 45\)](#)
- [Managing Cluster Versions \(p. 47\)](#)
- [Cluster Version History \(p. 49\)](#)

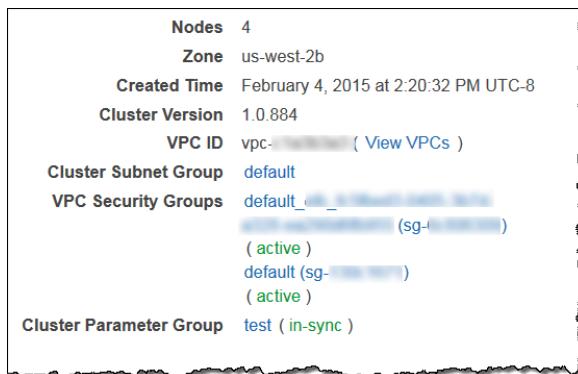
Overview

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.

Note

At this time, Amazon Redshift version 1.0 engine is available. However, as the engine is updated, multiple Amazon Redshift engine versions might be available for selection.

You can determine the Amazon Redshift engine and database versions for your cluster in the **Cluster Version** field in the console. The first two sections of the number are the cluster version, and the last section is the specific revision number of the database in the cluster. In the following example, the cluster version is 1.0 and the database revision number is 884.



Note

Although the console displays this information in one field, it is two parameters in the Amazon Redshift API: `ClusterVersion` and `ClusterRevisionNumber`. For more information, go to [Cluster](#) in the *Amazon Redshift API Reference*.

Amazon Redshift provides a setting, **Allow Version Upgrade**, to specify whether to automatically upgrade the Amazon Redshift engine in your cluster if a new version of the engine becomes available. This setting doesn't affect the database version upgrades, which are applied during the maintenance window that you specify for your cluster. Amazon Redshift engine upgrades are *major version upgrades*, and Amazon Redshift database upgrades are *minor version upgrades*. You can disable automatic version upgrades for major versions only. For more information about maintenance windows for minor version upgrades, see [Maintenance Windows \(p. 12\)](#).

Clusters and Nodes in Amazon Redshift

An Amazon Redshift cluster consists of nodes. Each cluster has a leader node and one or more compute nodes. The *leader node* receives queries from client applications, parses the queries, and develops query execution plans. The leader node then coordinates the parallel execution of these plans with the compute nodes and aggregates the intermediate results from these nodes. It then finally returns the results back to the client applications.

Compute nodes execute the query execution plans and transmit data among themselves to serve these queries. The intermediate results are sent to the leader node for aggregation before being sent back to the client applications. For more information about leader nodes and compute nodes, see [Data Warehouse System Architecture](#) in the *Amazon Redshift Database Developer Guide*.

When you launch a cluster, one option you specify is the node type. The node type determines the CPU, RAM, storage capacity, and storage drive type for each node. The *dense storage* (DS) node types are storage optimized. The *dense compute* (DC) node types are compute optimized.

DS2 node types are optimized for large data workloads and use hard disk drive (HDD) storage.

DC2 node types, and the earlier generation DC1 node types, are optimized for performance-intensive workloads. Because they use solid state drive (SSD) storage, the DC node types deliver much faster I/O compared to DS node types, but provide less storage space.

You launch clusters that use the DC2 node types in a virtual private cloud (VPC). You can't launch DC2 clusters in EC2 classic mode. For more information, see [Creating a Cluster in a VPC \(p. 37\)](#).

The node type that you choose depends heavily on three things:

- The amount of data you import into Amazon Redshift

- The complexity of the queries and operations that you run in the database
- The needs of downstream systems that depend on the results from those queries and operations

Node types are available in different sizes. DS2 nodes are available in `xlarge` and `8xlarge` sizes. DC2 nodes are available in `large` and `8xlarge` sizes. Node size and the number of nodes determine the total storage for a cluster.

Some node types allow one node (single-node) or two or more nodes (multi-node). The minimum for `8xlarge` clusters is two nodes. On a single-node cluster, the node is shared for leader and compute functionality. On a multi-node cluster, the leader node is separate from the compute nodes.

Amazon Redshift applies quotas to resources for each AWS account in each region. A *quota* restricts the number of resources that your account can create for a given resource type, such as nodes or snapshots, within a region. For more information about the default quotas that apply to Amazon Redshift resources, go to [Amazon Redshift Limits](#) in the *Amazon Web Services General Reference*. To request an increase, submit an [Amazon Redshift Limit Increase Form](#).

The cost of your cluster depends on the region, node type, number of nodes, and whether the nodes are reserved in advance. For more information about the cost of nodes, go to the [Amazon Redshift pricing](#) page.

Migrating From DC1 Node Types to DC2 Node Types

To take advantage of performance improvements, you can migrate your DC1 cluster to the newer DC2 node types.

Clusters that use the DC2 node types must be launched in a virtual private cloud (EC2-VPC). If your cluster is not in a VPC (EC2-CLASSIC), first create a snapshot of your cluster, and then choose one of the following options:

- From a `dc1.large` cluster, restore directly to a `dc2.large` cluster in a VPC.
- From a `dc1.8xlarge` cluster in EC2-CLASSIC, first restore to a `dc1.8xlarge` cluster in a VPC, and then resize your `dc1.8xlarge` cluster to a `dc2.8xlarge` cluster. You can't restore directly to a `dc2.8xl` cluster because the `dc2.8xlarge` node type has a different number of slices than the `dc1.8xlarge` node type.

If your cluster is in a VPC, choose one of the following options:

- From a `dc1.large` cluster, restore directly to a `dc2.large` cluster in a VPC.
- From a `dc1.8xlarge` cluster, resize your `dc1.8xl` cluster to a `dc2.8xlarge` cluster. You can't restore directly to a `dc2.8xlarge` cluster because the `dc2.8xlarge` node type has a different number of slices than the `dc1.8xlarge` node type.

For more information, see [Amazon Redshift Snapshots](#) (p. 145) and [Resizing Clusters](#) (p. 311).

Node Type Details

The following tables summarize the node specifications for each node type and size. In the two tables following, these headings have the given meaning:

- *vCPU* is the number of virtual CPUs for each node.
- *ECU* is the number of Amazon EC2 compute units for each node.
- *RAM* is the amount of memory in gibibytes (GiB) for each node.

- *Slices per Node* is the number of slices into which a compute node is partitioned.
- *Storage* is the capacity and type of storage for each node.
- *Node Range* is the minimum and maximum number of nodes that Amazon Redshift supports for the node type and size.

Note

You might be restricted to fewer nodes depending on the quota that is applied to your AWS account in the selected region, as discussed preceding.

- *Total Capacity* is the total storage capacity for the cluster if you deploy the maximum number of nodes that is specified in the node range.

Important

The DS1 node types are deprecated. The new DS2 node types provide higher performance than DS1 at no extra cost. If you have purchased DS1 reserved nodes, contact redshift-pm@amazon.com for assistance transitioning to DS2 node types.

Dense Storage Node Types

Node Size	vCPU	ECU	RAM (GiB)	Slices Per Node	Storage Per Node	Node Range	Total Capacity
ds2.xlarge	4	13	31	2	2 TB HDD	1–32	64 TB
ds2.8xlarge	36	119	244	16	16 TB HDD	2–128	2 PB

Dense Compute Node Types

Node Size	vCPU	ECU	RAM (GiB)	Slices Per Node	Storage Per Node	Node Range	Total Capacity
dc1.large	2	7	15	2	160 GB SSD	1–32	5.12 TB
dc1.8xlarge	32	104	244	32	2.56 TB SSD	2–128	326 TB
dc2.large	2	7	15.25	2	160 GB NVMe-SSD	1–32	5.12 TB
dc2.8xlarge	32	99	244	16	2.56 TB NVMe-SSD	2–128	326 TB

Previous Node Type Names

In previous releases of Amazon Redshift, the node types had different names. You can use the old names in the Amazon Redshift API and AWS Command Line Interface (AWS CLI). However, we recommend that you update any scripts that reference those names to use the current names instead. The current and previous names are as follows.

Previous Node Type Names

Current Name	Previous Name(s)
ds2.xlarge	ds1.xlarge, dw.hs1.xlarge, dw1.xlarge
ds2.8xlarge	ds1.8xlarge, dw.hs1.8xlarge, dw1.8xlarge
dc1.large	dw2.large

Current Name	Previous Name(s)
dc1.8xlarge	dw2.8xlarge

Determining the Number of Nodes

The number of nodes that you choose depends on the size of your dataset and your desired query performance. Using the dense storage node types as an example, if you have 32 TB of data, you can choose either 16 ds2.xlarge nodes or 2 ds2.8xlarge nodes. If your data grows in small increments, choosing the ds2.xlarge node size allows you to scale in increments of 2 TB. If you typically see data growth in larger increments, a ds2.8xlarge node size might be a better choice.

Because Amazon Redshift distributes and executes queries in parallel across all of a cluster's compute nodes, you can increase query performance by adding nodes to your cluster. Amazon Redshift also distributes your data across all compute nodes in a cluster. When you run a cluster with at least two compute nodes, data on each node will always be mirrored on disks on another node and you reduce the risk of incurring data loss.

Regardless of the choice you make, you can monitor query performance in the Amazon Redshift console and with Amazon CloudWatch metrics. You can also add or remove nodes as needed to achieve the balance between storage and performance that works best for you. When you request an additional node, Amazon Redshift takes care of all the details of deployment, load balancing, and data maintenance. For more information about cluster performance, see [Monitoring Amazon Redshift Cluster Performance \(p. 255\)](#).

If you intend to keep your cluster running continuously for a prolonged period, say, one year or more, you can pay considerably less by reserving the compute nodes for a one-year or three-year period. To reserve compute nodes, you purchase what are called reserved node offerings. You purchase one offering for each compute node that you want to reserve. When you reserve a compute node, you pay a fixed up-front charge and then an hourly recurring charge, whether your cluster is running or not. The hourly charges, however, are significantly lower than those for on-demand usage. For more information, see [Purchasing Amazon Redshift Reserved Nodes \(p. 176\)](#).

Resizing a Cluster

If your storage and performance needs change after you initially provision your cluster, you can resize your cluster. You can scale the cluster in or out by adding or removing nodes. Additionally, you can scale the cluster up or down by specifying a different node type.

For example, you can add more nodes, change node types, change a single-node cluster to a multinode cluster, or change a multinode cluster to a single-node cluster. However, you must ensure that the resulting cluster is large enough to hold the data that you currently have or else the resize will fail. When using the API, you have to specify the node type, node size, and the number of nodes even if you only change one of the two.

The following describes the resize process:

1. When you initiate the resize process, Amazon Redshift sends an event notification that acknowledges the resize request and starts to provision the new (target) cluster.
2. When the new (target) cluster is provisioned, Amazon Redshift sends an event notification that the resize has started, then restarts your existing (source) cluster in read-only mode. The restart terminates all existing connections to the cluster. All uncommitted transactions (including COPY) are rolled back. While the cluster is in read-only mode, you can run read queries but not write queries.
3. Amazon Redshift starts to copy data from the source cluster to the target cluster.

4. When the resize process nears completion, Amazon Redshift updates the endpoint of the target cluster and all connections to the source cluster are terminated.
5. After the resize completes, Amazon Redshift sends an event notification that the resize has completed. You can connect to the target cluster and resume running read and write queries.

When you resize your cluster, it will remain in read-only mode until the resize completes. You can view the resize progress on the cluster's **Status** tab in the Amazon Redshift console. The time it takes to resize a cluster depends on the amount of data in each node. Typically, the resize process varies from a couple of hours to a day, although clusters with larger amounts of data might take even longer. This is because the data is copied in parallel from each node on the source cluster to the nodes in the target cluster. For more information about resizing clusters, see [Resizing Clusters in Amazon Redshift \(p. 311\)](#) and [Resizing a Cluster \(p. 29\)](#).

Amazon Redshift doesn't sort tables during a resize operation. When you resize a cluster, Amazon Redshift distributes the database tables to the new compute nodes based on their distribution styles and runs an ANALYZE to update statistics. Rows that are marked for deletion are not transferred, so you will only need to run a VACUUM if your tables need to be resorted. For more information, see [Vacuuming tables](#) in the *Amazon Redshift Database Developer Guide*.

If your cluster is public and is in a VPC, it keeps the same elastic IP address (EIP) for the leader node after resizing. If your cluster is private and is in a VPC, it keeps the same private IP address for the leader node after resizing. If your cluster is not in a VPC, a new public IP address is assigned for the leader node as part of the resize operation.

To get the leader node IP address for a cluster, use the dig utility, as shown following:

```
dig mycluster.abcd1234.us-west-2.redshift.amazonaws.com
```

The leader node IP address is at the end of the ANSWER SECTION in the results, as shown following:

```
; <>> DiG 9.10.1-P1 <>> [REDACTED].us-west-2.redshift.amazonaws.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55520
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 13, ADDITIONAL: 6
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;[REDACTED].us-west-2.redshift.amazonaws.com. IN A
;;
;; ANSWER SECTION:
[REDACTED].us-west-2.redshift.amazonaws.com. 60 IN CNAME ec2-54-21
[REDACTED].compute.amazonaws.com.
[REDACTED].compute.amazonaws.com. 4239 IN A 54.123.456.789
;;
;; AUTHORITY SECTION:
[REDACTED]
```

Supported Platforms to Launch Your Cluster

Amazon Redshift clusters run in Amazon Elastic Compute Cloud (Amazon EC2) instances that are configured for the Amazon Redshift node type and size that you select. You can launch an Amazon Redshift cluster in one of two platforms: EC2-Classic or EC2-VPC, which are the supported platforms for Amazon EC2 instances. For more information about these platforms, go to [Supported Platforms](#) in the *Amazon EC2 User Guide for Linux Instances*. The platform or platforms available to you depend on your AWS account settings.

Note

To prevent connection issues between SQL client tools and the Amazon Redshift database, we recommend doing one of two things. You can configure an inbound rule that enables the hosts to negotiate packet size. Alternatively, you can disable TCP/IP jumbo frames by setting the maximum transmission unit (MTU) to 1500 on the network interface (NIC) of your Amazon EC2 instances. For more information about these approaches, see [Queries Appear to Hang and Sometimes Fail to Reach the Cluster \(p. 112\)](#).

EC2-Classic Platform

In the EC2-Classic platform, your cluster runs in a single, flat network that you share with other AWS customers. If you provision your cluster in the EC2-Classic platform, you control access to your cluster by associating one or more Amazon Redshift cluster security groups with the cluster. For more information, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

EC2-VPC Platform

In the EC2-VPC platform, your cluster runs in a virtual private cloud (VPC) that is logically isolated to your AWS account. If you provision your cluster in the EC2-VPC platform, you control access to your cluster by associating one or more VPC security groups with the cluster. For more information, go to [Security Groups for Your VPC](#) in the *Amazon VPC User Guide*.

To create a cluster in a VPC, you must first create an Amazon Redshift cluster subnet group by providing subnet information of your VPC, and then provide the subnet group when launching the cluster. For more information, see [Amazon Redshift Cluster Subnet Groups \(p. 40\)](#).

For more information about Amazon Virtual Private Cloud (Amazon VPC), go to the [Amazon VPC product detail page](#).

Choose a Platform

Your AWS account is capable of launching instances either into both platforms, or only into EC2-VPC, on a region-by-region basis. To determine which platform your account supports, and then launch a cluster, do the following:

1. Decide on the AWS region in which you want to deploy a cluster. For a list of AWS regions in which Amazon Redshift is available, go to [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.
2. Find out which Amazon EC2 platforms your account supports in the chosen AWS region. You can find this information in the Amazon EC2 console. For step-by-step instructions, go to [Supported Platforms](#) in the *Amazon EC2 User Guide for Linux Instances*.
3. If your account supports both of the platforms, choose the one on which you want to deploy your Amazon Redshift cluster. If your account supports only EC2-VPC, you must deploy your cluster in VPC.
4. Deploy your Amazon Redshift cluster. You can deploy a cluster by using the Amazon Redshift console, or programmatically by using the Amazon Redshift API, CLI, or SDK libraries. For more information about these options and links to the related documentation, see [What Is Amazon Redshift? \(p. 1\)](#).

Regions and Availability Zone Considerations

Amazon Redshift is available in several AWS regions. By default, Amazon Redshift provisions your cluster in a randomly selected Availability Zone (AZ) within the AWS region that you select. All the cluster nodes are provisioned in the same AZ.

You can optionally request a specific AZ if Amazon Redshift is available in that AZ. For example, if you already have an Amazon EC2 instance running in one AZ, you might want to create your Amazon Redshift cluster in the same AZ to reduce latency. On the other hand, you might want to choose another AZ for higher availability. Amazon Redshift might not be available in all AZs within a region.

For a list of supported AWS regions where you can provision an Amazon Redshift cluster, go to [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

Maintenance Windows

Amazon Redshift periodically performs maintenance to apply upgrades to your cluster. During these updates, your Amazon Redshift cluster is not available for normal operations.

Amazon Redshift assigns a 30-minute maintenance window at random from an 8-hour block of time per region, occurring on a random day of the week (Monday through Sunday, inclusive). The following list shows the time blocks for each region from which the default maintenance windows are assigned:

- US East (N. Virginia) region: 03:00–11:00 UTC
- US East (Ohio) region: 03:00–11:00 UTC
- US West (N. California) region: 06:00–14:00 UTC
- US West (Oregon) region: 06:00–14:00 UTC
- Canada (Central) region: 03:00–11:00 UTC
- Asia Pacific (Mumbai) region: 16:30–00:30 UTC
- Asia Pacific (Seoul) region: 13:00–21:00 UTC
- Asia Pacific (Singapore) region: 14:00–22:00 UTC
- Asia Pacific (Sydney) region: 12:00–20:00 UTC
- Asia Pacific (Tokyo) region: 13:00–21:00 UTC
- EU (Frankfurt) region: 06:00–14:00 UTC
- China (Beijing) region: 13:00–21:00 UTC
- EU (Ireland) region: 22:00–06:00 UTC
- EU (London) region: 22:00–06:00 UTC
- South America (São Paulo) region: 19:00–03:00 UTC

If a maintenance event is scheduled for a given week, it will start during the assigned 30 minute maintenance window. While Amazon Redshift is performing maintenance, it terminates any queries or other operations that are in progress. Most maintenance completes during the 30 minute maintenance window, but some maintenance tasks might continue running after the window closes. If there are no maintenance tasks to perform during the scheduled maintenance window, your cluster continues to operate normally until the next scheduled maintenance window.

You can change the scheduled maintenance window by modifying the cluster, either programmatically or by using the Amazon Redshift console. The window must be at least 30 minutes and not longer than 24 hours. For more information, see [Managing Clusters Using the Console \(p. 15\)](#).

Default Disk Space Alarm

When you create an Amazon Redshift cluster, you can optionally configure an Amazon CloudWatch alarm to monitor the average percentage of disk space that is used across all of the nodes in your cluster. We'll refer to this alarm as the *default disk space alarm*.

The purpose of default disk space alarm is to help you monitor the storage capacity of your cluster. You can configure this alarm based on the needs of your data warehouse. For example, you can use the warning as an indicator that you might need to resize your cluster. You might resize either to a different node type or to add nodes, or perhaps to purchase reserved nodes for future expansion.

The default disk space alarm triggers when disk usage reaches or exceeds a specified percentage for a certain number of times and at a specified duration. By default, this alarm triggers when the percentage that you specify is reached, and stays at or above that percentage for five minutes or longer. You can edit the default values after you launch the cluster.

When the CloudWatch alarm triggers, Amazon Simple Notification Service (Amazon SNS) sends a notification to specified recipients to warn them that the percentage threshold is reached. Amazon SNS uses a topic to specify the recipients and message that are sent in a notification. You can use an existing Amazon SNS topic; otherwise, a topic is created based on the settings that you specify when you launch the cluster. You can edit the topic for this alarm after you launch the cluster. For more information about creating Amazon SNS topics, see [Getting Started with Amazon Simple Notification Service](#).

After you launch the cluster, you can view and edit the alarm from the cluster's **Status** window under **CloudWatch Alarms**. The name is **percentage-disk-space-used-default-<string>**. You can open the alarm to view the Amazon SNS topic that it is associated with and edit alarm settings. If you did not select an existing Amazon SNS topic to use, the one created for you is named **<clustername>-default-alarms (<recipient>)**; for example, **examplecluster-default-alarms (notify@example.com)**.

For more information about configuring and editing the default disk space alarm, see [Creating a Cluster \(p. 17\)](#) and [Editing the Default Disk Space Alarm \(p. 32\)](#).

Note

If you delete your cluster, the alarm associated with the cluster will not be deleted but it will not trigger. You can delete the alarm from the CloudWatch console if you no longer need it.

Renaming Clusters

You can rename a cluster if you want the cluster to use a different name. Because the endpoint to your cluster includes the cluster name (also referred to as the *cluster identifier*), the endpoint will change to use the new name after the rename finishes. For example, if you have a cluster named `examplecluster` and rename it to `newcluster`, the endpoint will change to use the `newcluster` identifier. Any applications that connect to the cluster must be updated with the new endpoint.

You might rename a cluster if you want to change the cluster to which your applications connect without having to change the endpoint in those applications. In this case, you must first rename the original cluster and then change the second cluster to reuse the name of the original cluster before the rename. Doing this is necessary because the cluster identifier must be unique within your account and region, so the original cluster and second cluster cannot have the same name. You might do this if you restore a cluster from a snapshot and don't want to change the connection properties of any dependent applications.

Note

If you delete the original cluster, you are responsible for deleting any unwanted cluster snapshots.

When you rename a cluster, the cluster status changes to `renaming` until the process finishes. The old DNS name that was used by the cluster is immediately deleted, although it could remain cached for a few minutes. The new DNS name for the renamed cluster becomes effective within about 10 minutes. The renamed cluster is not available until the new name becomes effective. The cluster will be rebooted and any existing connections to the cluster will be dropped. After this completes, the endpoint will change to use the new name. For this reason, you should stop queries from running before you start the rename and restart them after the rename finishes.

Cluster snapshots are retained, and all snapshots associated with a cluster remain associated with that cluster after it is renamed. For example, suppose you have a cluster that serves your production database and the cluster has several snapshots. If you rename the cluster and then replace it in the production environment with a snapshot, the cluster that you renamed will still have those existing snapshots associated with it.

Amazon CloudWatch alarms and Amazon Simple Notification Service (Amazon SNS) event notifications are associated with the name of the cluster. If you rename the cluster, you need to update these accordingly. You can update the CloudWatch alarms in the CloudWatch console, and you can update the Amazon SNS event notifications in the Amazon Redshift console on the **Events** pane. The load and query data for the cluster continues to display data from before the rename and after the rename. However, performance data is reset after the rename process finishes.

For more information, see [Modifying a Cluster \(p. 25\)](#).

Shutting Down and Deleting Clusters

You can shut down your cluster if you want to stop it from running and incurring charges. When you shut it down, you can optionally create a final snapshot. If you create a final snapshot, Amazon Redshift will create a manual snapshot of your cluster before shutting it down. You can later restore that snapshot if you want to resume running the cluster and querying data.

If you no longer need your cluster and its data, you can shut it down without creating a final snapshot. In this case, the cluster and data are deleted permanently. For more information about shutting down and deleting clusters, see [Deleting a Cluster \(p. 27\)](#).

Regardless of whether you shut down your cluster with a final manual snapshot, all automated snapshots associated with the cluster will be deleted after the cluster is shut down. Any manual snapshots associated with the cluster are retained. Any manual snapshots that are retained, including the optional final snapshot, are charged at the Amazon Simple Storage Service storage rate if you have no other clusters running when you shut down the cluster, or if you exceed the available free storage that is provided for your running Amazon Redshift clusters. For more information about snapshot storage charges, go to the [Amazon Redshift pricing page](#).

Cluster Status

The cluster status displays the current state of the cluster. The following table provides a description for each cluster status.

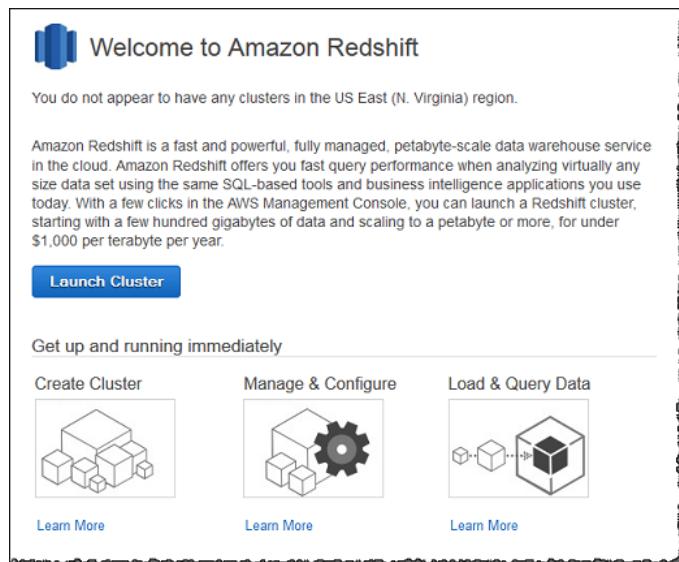
Status	Description
available	The cluster is running and available.
creating	Amazon Redshift is creating the cluster. For more information, see Creating a Cluster (p. 17) .
deleting	Amazon Redshift is deleting the cluster. For more information, see Deleting a Cluster (p. 27) .
final-snapshot	Amazon Redshift is taking a final snapshot of the cluster before deleting it. For more information, see Deleting a Cluster (p. 27) .
hardware-failure	The cluster suffered a hardware failure.

Status	Description
	If you have a single-node cluster, the node cannot be replaced. To recover your cluster, restore a snapshot. For more information, see Amazon Redshift Snapshots (p. 145) .
incompatible-hsm	Amazon Redshift cannot connect to the hardware security module (HSM). Check the HSM configuration between the cluster and HSM. For more information, see Encryption for Amazon Redshift Using Hardware Security Modules (p. 165) .
incompatible-network	There is an issue with the underlying network configuration. Make sure that the VPC in which you launched the cluster exists and its settings are correct. For more information, see Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 36) .
incompatible-parameters	There is an issue with one or more parameter values in the associated parameter group, and the parameter value or values cannot be applied. Modify the parameter group and update any invalid values. For more information, see Amazon Redshift Parameter Groups (p. 122) .
incompatible-restore	There was an issue restoring the cluster from the snapshot. Try restoring the cluster again with a different snapshot. For more information, see Amazon Redshift Snapshots (p. 145) .
modifying	Amazon Redshift is applying changes to the cluster. For more information, see Modifying a Cluster (p. 25) .
rebooting	Amazon Redshift is rebooting the cluster. For more information, see Rebooting a Cluster (p. 28) .
renaming	Amazon Redshift is applying a new name to the cluster. For more information, see Renaming Clusters (p. 13) .
resizing	Amazon Redshift is resizing the cluster. For more information, see Resizing a Cluster (p. 29) .
rotating-keys	Amazon Redshift is rotating encryption keys for the cluster. For more information, see About Rotating Encryption Keys in Amazon Redshift (p. 166) .
storage-full	The cluster has reached its storage capacity. Resize the cluster to add nodes or to choose a different node size. For more information, see Resizing a Cluster (p. 29) .
updating-hsm	Amazon Redshift is updating the HSM configuration. .

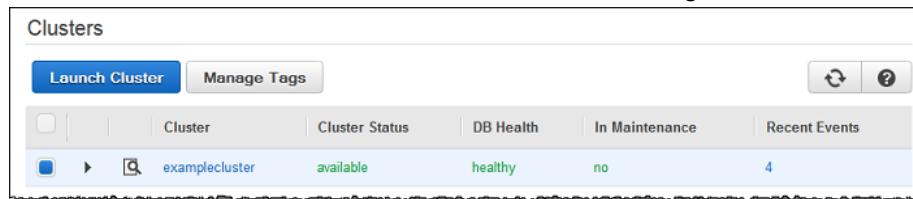
Managing Clusters Using the Console

To create, modify, resize, delete, reboot, and back up clusters, use the **Clusters** section in the Amazon Redshift console.

When you don't have any clusters in an AWS Region and you open the **Clusters** page, you have the option to launch a cluster. In the following screenshot, the AWS Region is the US East (N. Virginia) Region and there are no clusters for this account.



When you have at least one cluster in an AWS Region, the **Clusters** section displays a subset of information about all the clusters for the account in that AWS Region. In the following screenshot, there is one cluster created for this account in the selected AWS Region.



You can expand the cluster to view more information about the cluster, such as the endpoint details, cluster and database properties, tags, and so on. In the following screenshot, *examplecluster* is expanded to show a summary of information about the cluster.

The screenshot shows the 'Clusters' page in the AWS Management Console. At the top, there are three buttons: 'Launch Cluster', 'Manage Tags', and 'Manage IAM Roles'. Below the buttons is a table with one row for the cluster 'examplecluster'. The table columns are: Cluster, Cluster Status, DB Health, In Maintenance, and Recent Events. The cluster details are as follows:

Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
examplecluster	available	healthy	no	1

Below the table, the 'Endpoint' is listed as examplecluster.us-west-1.redshift.amazonaws.com:5439 (authorized). The 'Cluster Properties' section includes:

Cluster Name	examplecluster
Node Type	dc1.large
Nodes	1
Zone	us-west-1a
Cluster Parameter Group	default.redshift-1.0 (in-sync)
Enhanced VPC Routing	Yes

The 'Cluster Status' section includes:

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

The 'Cluster Database Properties' section includes:

Port	5439
Database Name	dev
Master Username	masteruser
Encrypted	No

The 'Backup, Audit Logging, and Maintenance' section includes:

Automated Snapshot Retention Period	1
Cross-Region Snapshots Enabled	No
Audit Logging Enabled	No
Maintenance Window	thu:07:30-thu:08:00
Allow Version Upgrade	Yes

The 'Tags' section indicates no tags have been created yet.

Topics

- [Creating a Cluster \(p. 17\)](#)
- [Modifying a Cluster \(p. 25\)](#)
- [Deleting a Cluster \(p. 27\)](#)
- [Rebooting a Cluster \(p. 28\)](#)
- [Resizing a Cluster \(p. 29\)](#)
- [Getting Information About Cluster Configuration \(p. 30\)](#)
- [Getting an Overview of Cluster Status \(p. 30\)](#)
- [Taking a Snapshot of a Cluster \(p. 31\)](#)
- [Editing the Default Disk Space Alarm \(p. 32\)](#)
- [Working with Cluster Performance Data \(p. 33\)](#)

Creating a Cluster

Before you create a cluster, make sure to read the clusters topics [Overview \(p. 5\)](#) and [Clusters and Nodes in Amazon Redshift \(p. 6\)](#).

You can create a cluster in the AWS Management Console in two ways:

- If you're new to Amazon Redshift or just need a basic cluster, use **Quick launch cluster**. With this approach, you specify only the node type, number of nodes, user name, password, and AWS Identity

and Access Management (IAM) role to use for access. For more information, see [Creating a Cluster by Using Quick Launch Cluster \(p. 18\)](#).

- If you're an existing user or want to customize your cluster, use **Launch cluster**. For example, use **Launch cluster** to use a specific virtual private cloud (VPC) or encrypt data in your cluster. For more information, see [Creating a Cluster by Using Launch Cluster \(p. 18\)](#).

Creating a Cluster by Using Quick Launch Cluster

If you're new to Amazon Redshift or just need a basic cluster, use this streamlined approach. If you're an existing user or want to customize your cluster, see [Creating a Cluster by Using Launch Cluster \(p. 18\)](#).

To create a cluster by using Quick launch 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, make sure 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. Choose the AWS Region where you want to create the cluster, for example **US West (Oregon)**.
3. On the Amazon Redshift Dashboard, choose **Quick launch cluster**.
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**.
 - **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**.

A confirmation page appears. The cluster takes a few minutes to be created. Choose **Close** to return to the list of clusters.

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

Clusters					
	Cluster	Cluster Status	DB Health	Release Status	In Maintenance
<input type="checkbox"/>	[REDACTED]	available	healthy	Up to date	no
<input type="checkbox"/>	redshift-cluster-1	available	healthy	Up to date	no

Creating a Cluster by Using Launch Cluster

If you're an existing Amazon Redshift user or want to customize your cluster, use the following procedure to launch your cluster. If you're new to Amazon Redshift or just need a basic cluster, see [Creating a Cluster by Using Quick Launch Cluster \(p. 18\)](#).

To create a cluster by using Launch cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. Choose **Launch Cluster**.
3. On the **Cluster Details** page, specify values for the following options, and then choose **Continue**.

Cluster Identifier

Type a unique name for your cluster.

Cluster identifiers must meet the following conditions:

- They must contain 1–63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- The first character must be a letter.
- They can't end with a hyphen or contain two consecutive hyphens.
- They must be unique for all clusters within an AWS account.

Database Name

Type a name if you want to create a database with a custom name (for example, `mydb`). This field is optional. A default database named `dev` is created for the cluster whether or not you specify a custom database name.

Database names must meet the following conditions:

- They must contain 1–64 alphanumeric characters.
- They must contain only lowercase letters.
- A database name can't be a reserved word. For more information, see [Reserved Words](#) in the *Amazon Redshift Database Developer Guide*.

Database Port

Type a port number through which you plan to connect from client applications to the database. The port number must be included in the connection string when opening JDBC or ODBC connections to the databases in the cluster.

The port number must meet the following conditions:

- It must contain only numeric characters.
- It must fall in the range 1150–65535. The default port is 5439.
- It must specify an open port that accepts inbound connections, if you are behind a firewall.

Master User Name

Type an account name for the master user of the database.

Master user names must meet the following conditions:

- They must contain 1–128 alphanumeric characters.
- The first character must be a letter.
- A master user name can't be a reserved word. For more information, see [Reserved Words](#) in the *Amazon Redshift Database Developer Guide*.

Master User Password and Confirm Password

Type a password for the master user account, and then retype it to confirm the password.

The password must meet the following conditions:
19

- It must be 8–64 characters in length.

- It must contain at least one uppercase letter.
- It must contain at least one lowercase letter.
- It must contain at least one number.
- It can be any printable ASCII character (ASCII code 33–126) except single quotation mark, double quotation mark, \, /, @, or space.

In the following screenshot, `examplecluster` is the cluster identifier, no custom database name is specified, 5439 is the port, and `masteruser` is the master user name.

CLUSTER DETAILS NODE CONFIGURATION ADDITIONAL CONFIGURATION REVIEW

Provide the details of your cluster. Fields marked with * are required.

Cluster Identifier* `examplecluster`

This is the unique key that identifies a cluster.
This parameter is stored as a lowercase string.
(e.g. my-dw-instance)

Database Name

Optional. A default database named dev is created for the cluster. Optionally, specify a custom database name (e.g. mydb) to create an additional database.

Database Port* `5439`

Port number on which the database accepts connections.

Master User Name* `masteruser`

Name of master user for your cluster. (e.g. awuser)

Master User Password*

Password must contain 8 to 64 printable ASCII characters excluding: /, ., \, and @. It must contain 1 uppercase letter, 1 lowercase letter, and 1 number.

Confirm Password*

Confirm Master User Password.

4. On the **Node Configuration** page, specify values for the following options, and then choose **Continue**.

Node Type

Choose a node type. When you choose a node type, the page displays information that corresponds to the selected node type, such as **CPU**, **Memory**, **Storage**, and **I/O Performance**.

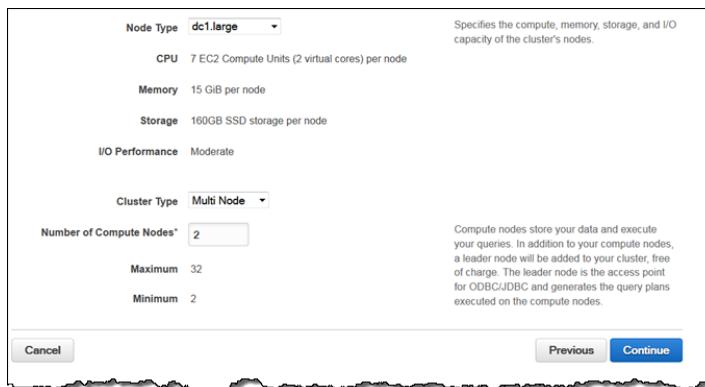
Cluster Type

Choose a cluster type. When you do, the maximum number of compute nodes for the selected node and cluster type appears for **Maximum**, and the minimum number appears for **Minimum**.

If you choose **Single Node**, you have one node that shares leader and compute functionality.

If you choose **Multi Node**, specify the number of compute nodes that you want for the cluster in **Number of Compute Nodes**.

In the following screenshot, the **dc1.large** node type is selected for a **Multi Node** cluster with two compute nodes.



5. On the **Additional Configuration** page, specify values for the following options, and then choose **Continue**.

- a. For **Provide the optional additional configuration details below**, configure the following options:

Cluster Parameter Group

Choose a cluster parameter group to associate with the cluster. If you don't choose one, the cluster uses the default parameter group.

Encrypt Database

Choose whether you want to encrypt all data within the cluster and its snapshots. If you leave the default setting, **None**, encryption is not enabled. If you want to enable encryption, choose whether you want to use AWS Key Management Service (AWS KMS) or a hardware security module (HSM), and then configure the related settings. For more information about encryption in Amazon Redshift, see [Amazon Redshift Database Encryption \(p. 163\)](#).

- **KMS**

Choose **KMS** if you want to enable encryption and use AWS KMS to manage your encryption key. In **Master Key**, choose **(default) aws/redshift** to use a default customer master key (CMK) or choose another key from your AWS account.

Note

If you want to use a key from another AWS account, choose **Enter a key ARN** from **Master Key**. Then type the Amazon Resource Name (ARN) for the key to use. You must have permission to use the key. For more information about access to keys in AWS KMS, see [Controlling Access to Your Keys in the AWS Key Management Service Developer Guide](#).

For more information about using AWS KMS encryption keys in Amazon Redshift, see [Database Encryption for Amazon Redshift Using AWS KMS \(p. 163\)](#).

- **HSM**

Choose **HSM** if you want to enable encryption and use a hardware security module (HSM) to manage your encryption key.

If you choose **HSM**, choose values from **HSM Connection** and **HSM Client Certificate**. These values are required for Amazon Redshift and the HSM to form a trusted connection over which the cluster key can be passed. The HSM connection and client certificate must be set up in Amazon Redshift before you launch a cluster. For more information about setting up HSM connections and client certificates, see [Encryption for Amazon Redshift Using Hardware Security Modules \(p. 165\)](#).

- b. For **Configure Networking Options**, you configure whether to launch your cluster in a virtual private cloud (VPC) or outside a VPC. The option you choose affects the additional options available in this section. Amazon Redshift uses the EC2-Classic and EC2-VPC platforms to launch clusters. Your AWS account determines which platform or platforms are available to you for your cluster. For more information, see [Supported Platforms](#) in the *Amazon EC2 User Guide for Linux Instances*.

Choose a VPC

To launch your cluster in a virtual private cloud (VPC), choose the VPC you want to use. You must have at least one Amazon Redshift subnet group set up to use VPCs. For more information, see [Amazon Redshift Cluster Subnet Groups \(p. 40\)](#).

Configure Networking Options:

Choose a VPC The identifier of the VPC in which you want to create your cluster

Cluster Subnet Group Selected Cluster Subnet Group may limit the choice of Availability Zones

Publicly Accessible Yes No Select Yes if you want the cluster to have a public IP address that can be accessed from the public internet, select No if you want the cluster to have a private IP address that can only be accessed from within the VPC.

Choose a Public IP Address Yes No Select Yes if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select No to have Amazon Redshift create an EIP for your instance.

Enhanced VPC Routing Yes No ⚠ Additional VPC configuration might be required. [Learn more](#).

Availability Zone The EC2 Availability Zone that the cluster will be created in.

To launch your cluster outside a VPC, choose **Not in VPC**. This option is available only to AWS accounts that support the EC2-Classic platform. Otherwise, you must launch your cluster in a VPC.

Configure Networking Options:

Choose a VPC Only VPCs with Cluster Subnet Groups are allowed

Availability Zone The EC2 Availability Zone that the cluster will be created in.

Cluster Subnet Group

Select the Amazon Redshift subnet group in which to launch the cluster.

Note

This option is available only for clusters in a VPC.

Publicly Accessible

Choose **Yes** to enable connections to the cluster from outside of the VPC in which you launch the cluster. Choose **No** if you want to limit connections to the cluster from only within the VPC.

Note

This option is available only for clusters in a VPC.

Choose a Public IP Address

If you set **Publicly Accessible** to **Yes**, choose **No** here to have Amazon Redshift to provide an Elastic IP (EIP) for the cluster. Alternatively, choose **Yes** if you want to use an EIP that

you have created and manage. If you have Amazon Redshift create the EIP, it is managed by Amazon Redshift.

Note

This option is available only for clusters in a VPC where **Publicly Accessible** is **Yes**.

Elastic IP

Select the EIP that you want to use to connect to the cluster from outside of the VPC.

Note

This option is available only for clusters in a VPC where **Publicly Accessible** and **Choose a Public IP Address** are **Yes**.

Availability Zone

Choose **No Preference** to have Amazon Redshift choose the Availability Zone that the cluster is created in. Otherwise, choose a specific Availability Zone.

Enhanced VPC Routing

Choose **Yes** to enable enhanced VPC routing. Enhanced VPC routing might require some additional configuration. For more information, see [Amazon Redshift Enhanced VPC Routing \(p. 115\)](#).

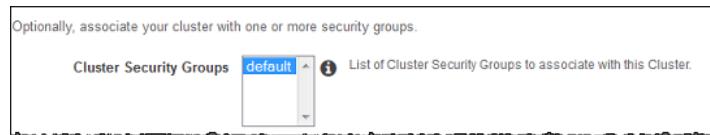
- c. For **Optionally, associate your cluster with one or more security groups**, specify values for the following options:

Cluster Security Groups

Choose an Amazon Redshift security group or groups for the cluster. By default, the chosen security group is the default security group. For more information about cluster security groups, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

Note

This option is only available if you launch your cluster in the EC2-Classic platform.

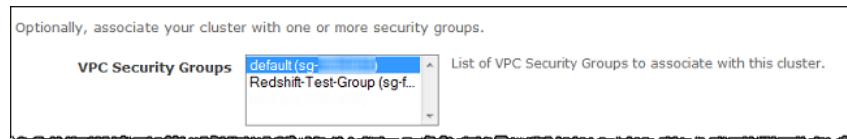


VPC Security Groups

Choose a VPC security group or groups for the cluster. By default, the chosen security group is the default VPC security group. For more information about VPC security groups, see [Security Groups for Your VPC](#) in the [Amazon VPC User Guide](#).

Note

This option is only available if you launch your cluster in the EC2-VPC platform.



- d. For **Optionally create a basic alarm for this cluster**, configure the following options, and then choose **Continue**:

Create CloudWatch Alarm

Choose **Yes** if you want to create an alarm that monitors the disk usage of your cluster, and then specify values for the corresponding options. Choose **No** if you don't want to create an alarm.

Disk Usage Threshold

Choose a percentage of average disk usage that has been reached or exceeded at which the alarm should trigger.

Use Existing Topic

Choose **No** if you want to create a new Amazon Simple Notification Service (Amazon SNS) topic for this alarm. In the **Topic** box, edit the default name if necessary. For **Recipients**, type the email addresses for any recipients who should receive the notification when the alarm triggers.

Optional, create a basic alarm for this cluster.

Create CloudWatch Alarm	<input checked="" type="button"/> Yes	Create a CloudWatch alarm to monitor the disk usage of your cluster.
Disk Usage Threshold	<input type="button"/> 80%	Threshold at which the alarm will trigger when disk usage across all nodes reaches this percentage.
Use Existing Topic	<input type="button"/> No	Use an existing SNS topic or create a new one. SNS is a Simple Notification Service which will send email notifications to the recipients of the SNS topic when the alarm triggers.
Topic	<input type="text"/> examplecluster-default-alarms	Name of the SNS topic that will be created.
Recipients	<input type="text"/> notify@example.com	Recipients of this SNS topic. If you have multiple recipients, separate the recipients with a comma.

Choose **Yes** if you want to choose an existing Amazon SNS topic for this alarm, and then in the **Topic** list, choose the topic that you want to use.

Optional, create a basic alarm for this cluster.

Create CloudWatch Alarm	<input checked="" type="button"/> Yes	Create a CloudWatch alarm to monitor the disk usage of your cluster.
Disk Usage Threshold	<input type="button"/> 80%	Threshold at which the alarm will trigger when disk usage across all nodes reaches this percentage.
Use Existing Topic	<input checked="" type="button"/> Yes	Use an existing SNS topic or create a new one. SNS is a Simple Notification Service which will send email notifications to the recipients of the SNS topic when the alarm triggers.
Topic	<input type="text"/> xyz123	SNS topic that will be used for the basic alarm.

- e. For **Optionally, select your maintenance track for this cluster**, choose **Current** or **Trailing**.

If you choose **Current**, your cluster is updated with the latest approved release during your maintenance window. If you choose **Trailing**, your cluster is updated with the release that was approved previously.

Optional, select your maintenance track for this cluster.

Maintenance Track	<input checked="" type="radio"/> Current	<input type="radio"/> Trailing	Select Current to apply the latest certified maintenance release including features and bug-fixes. Select Trailing to apply the previously certified maintenance release.
-------------------	--	--------------------------------	---

6. On the **Review** page, review the details of the cluster. If everything is satisfactory, choose **Launch Cluster** to start the creation process. Otherwise, choose **Back** to make any necessary changes, and then choose **Continue** to return to the **Review** page.

Note

Some cluster properties, such as the values for **Database Port** and **Master User Name**, cannot be modified later. If you need to change them, choose **Back** to change them now.

The following screenshot shows a summary of various options chosen during the cluster launch process.

The screenshot shows the 'Cluster Properties' tab selected. It displays the following configuration:

- Cluster Identifier:** examplecluster
- Node Type:** dc1.large
- Number of Compute Nodes:** 1 (leader and compute run on a single node)
- Availability Zone:** No Preference
- Database Configuration:**
 - Database Name:** A default database will be created (dev)
 - Database Port:** 5439
 - Master User Name:** masteruser
 - Cluster Parameter Group:** default.redshift-1.0

The 'Database Configuration' section also includes a note: "These properties specify the database name, port, and username you will use to connect to the database. The parameter group contains configuration values used by the database."

The 'Security, Access, and Encryption' tab is also visible, showing VPC settings like Virtual Private Cloud (vpc-ed1c0e8f), Cluster Subnet Group, Publicly Accessible (Yes), Elastic IP (Not used), VPC Security Groups (sg-6c4f540e), Enhanced VPC Routing (Yes), and Encrypt Database (No). The note for this tab states: "These settings control whether your cluster will be created in an existing VPC to allow for simpler integration with other AWS Services, and the security groups which define access rules to your cluster."

The 'CloudWatch Alarms' tab is shown with the note: "CloudWatch alarms are used to notify if metrics for your cluster are within a certain threshold. All recipients under the SNS topic specified for your alarm will receive notifications once an alarm is triggered. Basic alarms will not be created for this cluster."

- After you initiate the creation process, choose **Close**. The cluster might take several minutes to be ready to use.



You can monitor the status of the operation in the performance dashboard.

Modifying a Cluster

When you modify a cluster, changes to the following options are applied immediately:

- VPC Security Groups**
- Publicly Accessible**
- Master User Password**
- Automated Snapshot Retention Period**
- HSM Connection**
- HSM Client Certificate**
- Maintenance Window Start**

- **Maintenance Window End**

Changes to the following options take effect only after the cluster is restarted:

- **Cluster Identifier**

Amazon Redshift restarts the cluster automatically when you change **Cluster Identifier**.

- **Enhanced VPC Routing**

Amazon Redshift restarts the cluster automatically when you change **Enhanced VPC Routing**.

- **Cluster Parameter Group**

If you decrease the automated snapshot retention period, existing automated snapshots whose settings fall outside of the new retention period are deleted. For more information, see [Amazon Redshift Snapshots \(p. 145\)](#).

To modify a cluster

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 **Clusters**, and then choose the cluster that you want to modify.



3. On the **Configuration** tab of the cluster details page, choose **Cluster**, and then choose **Modify**.



4. In the **Modify Cluster** window, change your cluster, and then choose **Modify**.

The following screenshot shows the **Modify Cluster** options for a cluster in a VPC.

Deleting a Cluster

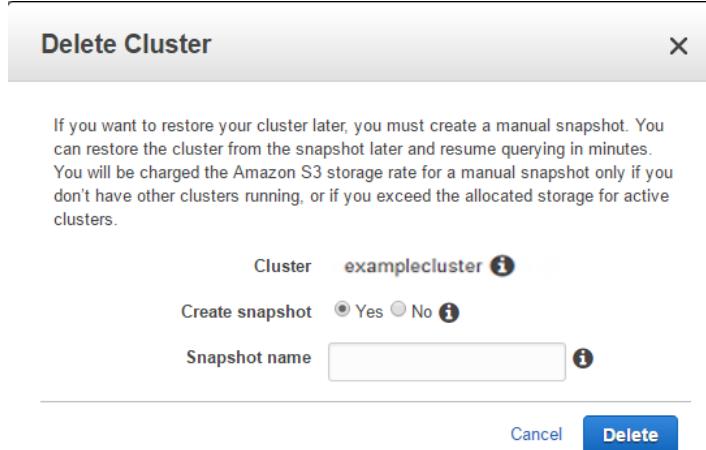
If you no longer need your cluster, you can delete it. If you plan to provision a new cluster with the same data and configuration as the one you are deleting, you need a manual snapshot. By using a manual snapshot, you can restore the snapshot later and resume using the cluster. If you delete your cluster but you don't create a final manual snapshot, the cluster data is deleted. In either case, automated snapshots are deleted after the cluster is deleted, but any manual snapshots are retained until you delete them. You might be charged Amazon Simple Storage Service storage rates for manual snapshots, depending on the amount of storage you have available for Amazon Redshift snapshots for your clusters. For more information, see [Shutting Down and Deleting Clusters \(p. 14\)](#).

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. In the navigation pane, choose **Clusters**, and then choose the cluster that you want to delete.

3. On the **Configuration** tab of the cluster details page, choose **Cluster**, and then choose **Delete**.
4. In the **Delete Cluster** dialog box, do one of the following:

- In **Create snapshot**, choose **Yes** to delete the cluster and take a final snapshot. In **Snapshot name**, type a name for the final snapshot, and then choose **Delete**.



- In **Create snapshot**, choose **No** to delete the cluster without taking a final snapshot, and then choose **Delete**.

After you initiate the delete of the cluster, it can take several minutes for the cluster to be deleted. You can monitor the status in the cluster list as shown in the following screenshots. If you requested a final snapshot, **Cluster Status** shows **final-snapshot** before deleting.

The following screenshot shows the cluster with a status of **final-snapshot** before it is deleted.

Clusters					
	Launch Cluster	Manage Tags			
	Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
	examplecluster	final-snapshot	healthy	no	5

The following screenshot shows the cluster with a status of **deleting**.

Clusters					
	Launch Cluster	Manage Tags			
	Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
	examplecluster	deleting	healthy	no	5

When the process has finished, you can verify that the cluster has been deleted because it no longer appears in the list of clusters on the **Clusters** page.

Rebooting a Cluster

When you reboot a cluster, the cluster status is set to **rebooting** and a cluster event is created when the reboot is completed. Any pending cluster modifications are applied at this reboot.

To reboot a cluster

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 **Clusters**, and then choose the cluster that you want to reboot.
3. On the **Configuration** tab of the cluster details page, choose **Cluster** and then choose **Reboot**.
4. In the **Reboot Clusters** window, confirm that you want to reboot this cluster, and then choose **Reboot**.

It can take several minutes for the cluster to be available. You can monitor the status of the reboot in the cluster list as shown in the following screenshot.

Clusters					
	Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
	examplecluster	rebooting	healthy	no	4

Resizing a Cluster

When you resize a cluster, you specify a number of nodes or node type that is different from the current configuration of the cluster. While the cluster is in the process of resizing, you cannot run any write or read/write queries on the cluster; you can run only read queries.

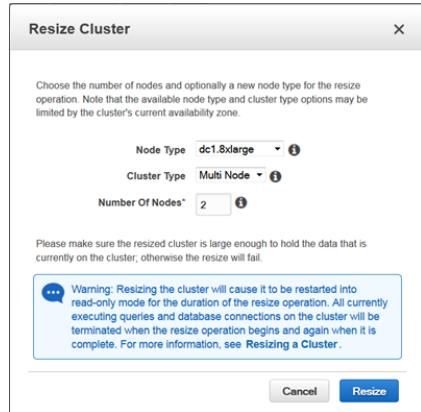
For more information about resizing clusters, including walking through the process of resizing clusters using different approaches, see [Resizing Clusters in Amazon Redshift \(p. 311\)](#).

To resize a cluster

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 **Clusters**, and then choose the cluster that you want to resize.

Clusters					
	Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
	examplecluster	available	healthy	no	4

3. On the **Configuration** tab of the cluster details page, choose **Cluster**, and then choose **Resize**.
4. In the **Resize Clusters** window, configure the resize parameters including the **Node Type**, **Cluster Type**, and **Number of Nodes**, and then choose **Resize**.



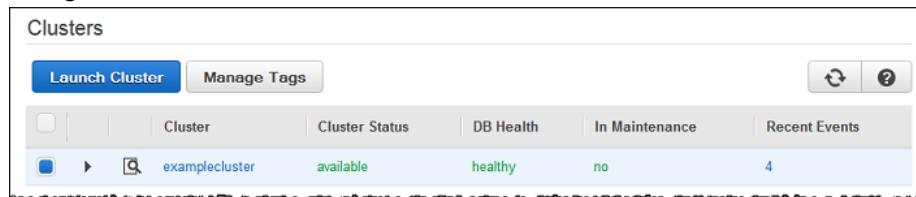
You can monitor the progress of the resize on the **Status** tab.



Getting Information About Cluster Configuration

To get cluster configuration details

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 **Clusters**, and then choose the cluster for which you want to view configuration information.



3. On the **Configuration** tab of the cluster details page, review the configuration information. You can view information about the cluster properties, status, database, capacity, backup, audit logging, maintenance, and SSH ingestion settings.

Getting an Overview of Cluster Status

The cluster **Status** tab provides a high-level overview of the status of a cluster and a summary of events related to the cluster. It also provides a list of Amazon CloudWatch alarms associated with the cluster.

To get an overview of cluster status

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 **Clusters**, and then choose the cluster for which you want to view status information.



3. Choose the **Status** tab.

The status summary page is displayed as shown in the following screenshot.

The screenshot shows the 'Status' tab of the cluster details page. It displays the following information:

- Cluster Status:** available, healthy, active, no, in-sync, None
- Recent Events:**

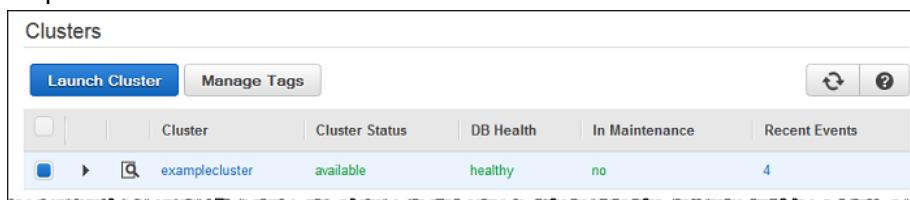
Time	Event	Source Identifier	Source Type
Jul 8 2:16 PM	Amazon Redshift cluster 'examplecluster' was rebooted at 2014-07-08 21:16 UTC.	examplecluster	cluster
Jul 3 3:41 PM	Amazon Redshift cluster 'examplecluster' has been created at 2014-07-03 22:41 UTC and is ready for use.	examplecluster	cluster
- CloudWatch Alarms:** (not visible in the screenshot)

Taking a Snapshot of a Cluster

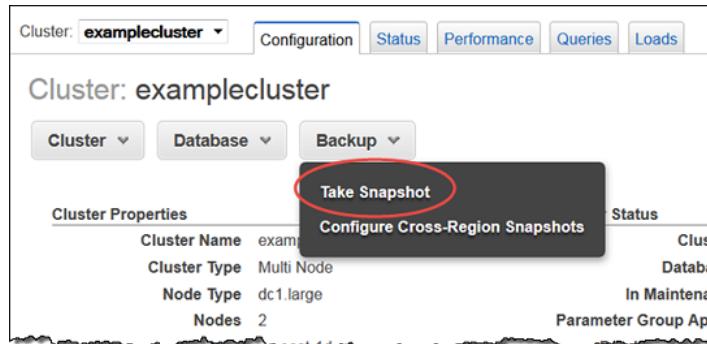
You can take a snapshot of your cluster from the **Configuration** tab of your cluster as shown following. You can also take a snapshot of your cluster from the snapshots part of the Amazon Redshift console. For more information, see [Managing Snapshots Using the Console \(p. 151\)](#).

To take a snapshot of a cluster

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 **Clusters**, and then choose the cluster for which you want to take a snapshot.

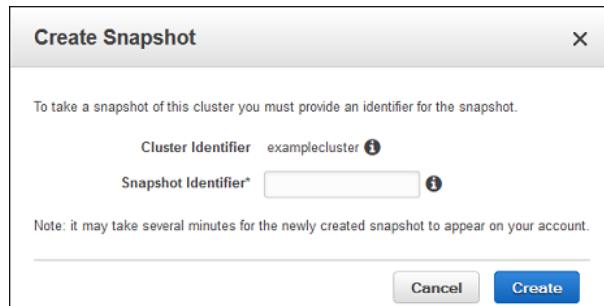


3. On the **Configuration** tab of the cluster details page, choose **Backup**, and then choose **Take Snapshot**.



4. In the **Create Snapshot** dialog box, do the following:

- a. In the **Cluster Identifier** box, choose the cluster that you want to take a snapshot of.
- b. In the **Snapshot Identifier** box, type a name for the snapshot.



5. Choose **Create**.

To view details about the snapshot taken and all other snapshots for your AWS account, see the snapshots section of the Amazon Redshift console. For more information, see [Managing Snapshots Using the Console \(p. 151\)](#).

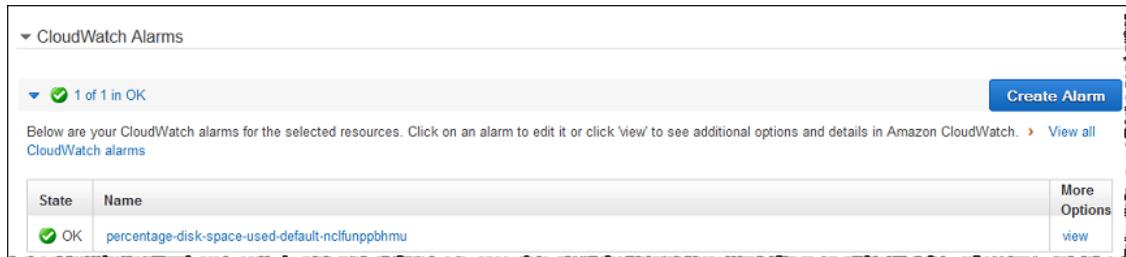
Editing the Default Disk Space Alarm

If you opted to create a default disk space alarm when you created your Amazon Redshift cluster, you can edit the alarm. For example, you might want to change the percentage at which the alarm triggers, or you might want to change the duration settings.

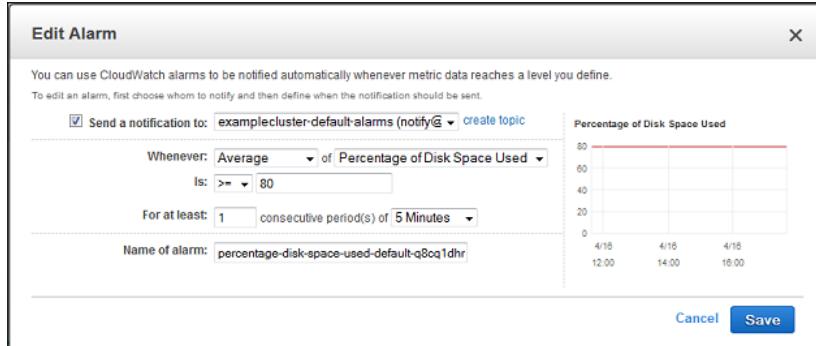
To edit the default disk space alarm

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 **Clusters**, and then choose the cluster associated with the alarm that you want to edit.
3. Choose the **Status** tab.
4. In the **CloudWatch Alarms** section, choose the alarm that you want to edit.

The default disk space alarm that was created when you launched your cluster is named **percentage-disk-space-used-default-<string>**. The **string** is randomly generated by Amazon Redshift.



- In the **Edit Alarm** window, edit any values that you want to change, such as the percentage or minutes.



- To change the Amazon SNS topic that the alarm is associated with, do one of the following:
 - If you want to choose another existing topic, choose a topic from the **Send a notification to** list.
 - If you want to create a new topic, choose **create topic** and specify a new topic name and the email addresses for recipients.
- Choose **Save**.

Working with Cluster Performance Data

You can work with cluster performance data using the **Performance**, **Queries**, and **Loads** tabs. For more information about working with cluster performance, see [Working with Performance Data in the Amazon Redshift Console \(p. 261\)](#).

Managing Clusters Using the AWS SDK for Java

The following Java code example demonstrates common cluster management operations including:

- Creating a cluster.
- Listing metadata about a cluster.
- Modifying configuration options.

After you initiate the request for the cluster to be created, you must wait until the cluster is in the available state before you can modify it. This example uses a loop to periodically check the status of the cluster using the `describeClusters` method. When the cluster is available, the preferred maintenance window for the cluster is changed.

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and specify a cluster identifier.

Example

```
import java.io.IOException;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyCluster {

    public static AmazonRedshiftClient client;
    public static String clusterIdentifier = "***provide a cluster identifier***";
    public static long sleepTime = 20;

    public static void main(String[] args) throws IOException {

        AWS Credentials credentials = new PropertiesCredentials(
            CreateAndModifyCluster.class
                .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
            createCluster();
            waitForClusterReady();
            describeClusters();
            modifyCluster();
            describeClusters();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createCluster() {
        CreateClusterRequest request = new CreateClusterRequest()
            .withClusterIdentifier(clusterIdentifier)
            .withMasterUsername("masteruser")
            .withMasterUserPassword("12345678Aa")
            .withNodeType("ds1.xlarge")
            .withNumberOfNodes(2);

        Cluster createResponse = client.createCluster(request);
        System.out.println("Created cluster " + createResponse.getClusterIdentifier());
    }

    private static void describeClusters() {
        DescribeClustersRequest request = new DescribeClustersRequest()
            .withClusterIdentifier(clusterIdentifier);

        DescribeClustersResult result = client.describeClusters(request);
        printResult(result);
    }

    private static void modifyCluster() {
        ModifyClusterRequest request = new ModifyClusterRequest()
            .withClusterIdentifier(clusterIdentifier)
            .withPreferredMaintenanceWindow("wed:07:30-wed:08:00");

        client.modifyCluster(request);
        System.out.println("Modified cluster " + clusterIdentifier);
    }

    private static void printResult(DescribeClustersResult result)
```

```
{  
    if (result == null)  
    {  
        System.out.println("Describe clusters result is null.");  
        return;  
    }  
  
    System.out.println("Cluster property:");  
    System.out.format("Preferred Maintenance Window: %s\n",  
result.getClusters().get(0).getPreferredMaintenanceWindow());  
}  
  
private static void waitForClusterReady() throws InterruptedException {  
    Boolean clusterReady = false;  
    System.out.println("Waiting for cluster to become available.");  
    while (!clusterReady) {  
        DescribeClustersResult result = client.describeClusters(new  
DescribeClustersRequest()  
            .withClusterIdentifier(clusterIdentifier));  
        String status = (result.getClusters().get(0).getClusterStatus());  
        if (status.equalsIgnoreCase("available")) {  
            clusterReady = true;  
        }  
        else {  
            System.out.print(".");  
            Thread.sleep(sleepTime*1000);  
        }  
    }  
}  
}
```

Manage Clusters Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage clusters.

- [create-cluster](#)
- [delete-cluster](#)
- [describe-clusters](#)
- [describe-cluster-versions](#)
- [describe-orderable-cluster-options](#)
- [modify-cluster](#)
- [reboot-cluster](#)

You can use the following Amazon Redshift API operations to manage clusters.

- [CreateCluster](#)
- [DeleteCluster](#)
- [DescribeClusters](#)
- [DescribeClusterVersions](#)
- [DescribeOrderableClusterOptions](#)
- [ModifyCluster](#)

- [RebootCluster](#)

Managing Clusters in an Amazon Virtual Private Cloud (VPC)

Topics

- [Overview \(p. 36\)](#)
- [Creating a Cluster in a VPC \(p. 37\)](#)
- [Managing VPC Security Groups for a Cluster \(p. 39\)](#)
- [Amazon Redshift Cluster Subnet Groups \(p. 40\)](#)

Overview

Amazon Redshift supports both the EC2-VPC and EC2-Classic platforms to launch a cluster. For more information, see [Supported Platforms to Launch Your Cluster \(p. 10\)](#).

Note

Amazon Redshift supports launching clusters into dedicated tenancy VPCs. For more information, see [Dedicated Instances](#) in the *Amazon VPC User Guide*.

When provisioning a cluster in VPC, you need to do the following:

- **Provide VPC information.**

When you request Amazon Redshift to create a cluster in your VPC, you must provide your VPC information, such as the VPC ID, and a list of subnets in your VPC by first creating a cluster subnet group. When you launch a cluster you provide the cluster subnet group so that Amazon Redshift can provision your cluster in one of the subnets in the VPC. For more information about creating subnet groups in Amazon Redshift, see [Amazon Redshift Cluster Subnet Groups \(p. 40\)](#). For more information about setting up VPC, go to [Getting Started with Amazon VPC](#) in the *Amazon VPC Getting Started Guide*.

- **Optionally, configure the publicly accessible options.**

If you configure your cluster to be publicly accessible, you can optionally select an *elastic IP address (EIP)* to use for the external IP address. An EIP is a static IP address that is associated with your AWS account. You can use an EIP to connect to your cluster from outside the VPC. An EIP gives you the ability to change your underlying configuration without affecting the IP address that clients use to connect to your cluster. This approach can be helpful for situations such as recovery after a failure.

If you want to use an EIP associated with your own AWS account, you must create it in Amazon EC2 prior to launching your Amazon Redshift cluster. Otherwise, it will not be available during the launch process. You can also have Amazon Redshift configure an EIP to use for the VPC, but the assigned EIP will be managed by the Amazon Redshift service and will not be associated with your AWS account. For more information, go to [Elastic IP Addresses \(EIP\)](#) in the *Amazon EC2 User Guide for Linux Instances*.

If you have a publicly accessible cluster in a VPC, and you want to connect to it by using the private IP address from within the VPC, you must set the following VPC parameters to `true`:

- `DNS resolution`
- `DNS hostnames`

If you have a publicly accessible cluster in a VPC, but do not set those parameters to `true` in the VPC, connections made from within the VPC will resolve to the EIP of the cluster instead of the private IP

address. We recommend that you set these parameters to `true` and use the private IP address for a publicly accessible cluster when connecting from within the VPC. For more information, see [Using DNS with Your VPC](#) in the Amazon VPC User Guide.

Note

If you have an existing publicly accessible cluster in a VPC, connections from within the VPC will continue to use the EIP to connect to the cluster even with those parameters set until you resize the cluster. Any new clusters will follow the new behavior of using the private IP address when connecting to the publicly accessible cluster from within the same VPC.

Also, note that the *EIP* is an external IP address for accessing the cluster outside of a VPC, but it is not related to the *cluster node public IP addresses and private IP addresses* that are displayed in the Amazon Redshift console under **SSH Ingestion Settings**. The public and private cluster node IP addresses appear regardless of whether the cluster is publicly accessible or not. They are used only in certain circumstances to configure ingress rules on the remote host when you load data from an Amazon EC2 instance or other remote host using a Secure Shell (SSH) connection. For more information, see [Step 1: Retrieve the cluster public key and cluster node IP addresses](#) in the Amazon Redshift Database Developer Guide.

The option to associate a cluster with an EIP is available only when you create the cluster or restore the cluster from a snapshot. You can't attach an EIP after the cluster is created or restored. If you want to associate the cluster with an EIP or change an EIP that is associated with the cluster, you need to restore the cluster from a snapshot and specify the EIP at that time.

- **Associate a VPC security group.**

You then grant inbound access using a VPC security group. This VPC security group must allow access over the database port for the cluster so that you can connect by using SQL client tools. You can configure this in advance, or add rules to it after you launch the cluster. For more information, go to [Security in Your VPC](#) in the *Amazon VPC User Guide*. You cannot use the Amazon Redshift cluster security groups to grant inbound access to the cluster.

For more information about working with clusters in a VPC, see [Creating a Cluster in a VPC \(p. 37\)](#).

Restoring a Snapshot of a Cluster in VPC

A snapshot of a cluster in VPC can only be restored in a VPC, not outside the VPC. You can restore it in the same VPC or another VPC in your account. For more information about snapshots, see [Amazon Redshift Snapshots \(p. 145\)](#).

Creating a Cluster in a VPC

The following are the general steps how you can deploy a cluster in your VPC.

To create a cluster in a VPC

1. Set up a VPC.

You can create your cluster either in the default VPC for your account, if your account has one, or a VPC that you have created. For more information, see [Supported Platforms to Launch Your Cluster \(p. 10\)](#). To create a VPC, follow steps 2 and 3 in the Amazon Virtual Private Cloud [Getting Started Guide](#). Make a note of the VPC identifier, subnet, and subnet's availability zone. You will need this information when you launch your cluster.

Note

You must have at least one subnet defined in your VPC so you can add it to the cluster subnet group in the next step. If you use the VPC Wizard, a subnet for your VPC is automatically created for you. For more information about adding a subnet to your VPC, go to [Adding a Subnet to Your VPC](#).

2. Create an Amazon Redshift cluster subnet group that specifies which of the subnets in the VPC can be used by the Amazon Redshift cluster.

You can create cluster subnet group using either the Amazon Redshift console or programmatically. For more information, see [Amazon Redshift Cluster Subnet Groups \(p. 40\)](#).

3. Authorize access for inbound connections in a VPC security group that you will associate with the cluster.

To enable a client outside the VPC (on the public Internet) to connect to the cluster, you must associate the cluster with a VPC security group that grants inbound access to the port that you used when you launched the cluster. For examples of security group rules, go to [Security Group Rules](#) in the *Amazon VPC User Guide*.

4. Launch a cluster in your VPC.

You can use the procedure described in the Getting Started to launch the cluster in your VPC. For more information, see [Step 2: Launch a Cluster](#). As you follow the wizard, in the **Configure Network Options** of the **ADDITIONAL CONFIGURATION** page, specify the following information:

- **Choose a VPC** Select the VPC from the drop-down list.
- **Cluster Subnet Group** Select the cluster subnet group you created in step 2.
- **Publicly Accessible** Select **Yes** if you want the cluster to have a public IP address that can be accessed from the public internet, select **No** if you want the cluster to have a private IP addressed that can only be accessed from within the VPC. If your AWS account allows you to create EC2-Classic clusters, the default is no, otherwise the default is yes.
- **Choose a Public IP Address** Select **Yes** if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select **No** to have Amazon Redshift create an EIP for your instance.
- **Elastic IP** Select an EIP to use to connect to the cluster from outside of the VPC.
- **Availability Zone** Select **No Preference** to have Amazon Redshift select the availability zone that the cluster will be created in. Otherwise, select a specific availability zone.
- Select the VPC security group that grants authorized devices access to the cluster.

The following is an example screen shot of the **Configure Networking Options** section of the **ADDITIONAL CONFIGURATION** page.

Configure Networking Options:

Choose a VPC: vpc- [dropdown] The identifier of the VPC in which you want to create your cluster

Cluster Subnet Group: examplesubnet [dropdown] Selected Cluster Subnet Group may limit the choice of Availability Zones

Publicly Accessible: Yes [dropdown] Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network

Choose a Public IP Address: Yes [dropdown] Select Yes if you want to select your own public IP address from a list of elastic IP (EIP) addresses that are already configured for your cluster's VPC. Select No if you want Amazon Redshift to provide an EIP for you instead.

Elastic IP: [dropdown] Select the Elastic IP that you want the cluster to use for its public IP address.

Availability Zone: No Preference [dropdown] The EC2 Availability Zone that the cluster will be created in.

Now you are ready to use the cluster. You can follow the Getting Started steps to test the cluster by uploading sample data and trying example queries.

Managing VPC Security Groups for a Cluster

When you provision an Amazon Redshift cluster, it is locked down by default so nobody has access to it. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group. If you are on the EC2-VPC platform, you can either use an existing Amazon VPC security group or define a new one and then associate it with a cluster as described following. If you are on the EC2-Classic platform, you define a cluster security group and associate it with a cluster. For more information on using cluster security groups on the EC2-Classic platform, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

A VPC security group consists of a set of rules that control access to an instance on the VPC, such as your cluster. Individual rules set access based either on ranges of IP addresses or on other VPC security groups. When you associate a VPC security group with a cluster, the rules that are defined in the VPC security group control access to the cluster.

Each cluster you provision on the EC2-VPC platform has one or more Amazon VPC security groups associated with it. Amazon VPC provides a VPC security group called default, which is created automatically when you create the VPC. Each cluster that you launch in the VPC is automatically associated with the default VPC security group if you don't specify a different VPC security group when you create the cluster. You can associate a VPC security group with a cluster when you create the cluster, or you can associate a VPC security group later by modifying the cluster. For more information on associating a VPC security group with a cluster, see [Creating a Cluster by Using Launch Cluster \(p. 18\)](#) and [To modify a cluster \(p. 26\)](#).

The following table describes the default rules for the default VPC security group.

Inbound			
Source	Protocol	Port Range	Comments
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group
Outbound			
Destination	Protocol	Port Range	Comments
0.0.0.0/0	All	All	Allow all outbound traffic

You can change the rules for the default VPC security group as needed for your Amazon Redshift cluster.

If the default VPC security group is enough for you, you don't need to create more. However, you can optionally create additional VPC security groups to better manage inbound access to your cluster. For example, suppose that you are running a service on an Amazon Redshift cluster, and you have several different service levels you provide to your customers. If you don't want to provide the same access at all service levels, you might want to create separate VPC security groups, one for each service level. You can then associate these VPC security groups with your cluster.

Keep in mind that while you can create up to 100 VPC security groups for a VPC, and you can associate a VPC security group with many clusters, you can only associate up to 5 VPC security groups with a given cluster.

Amazon Redshift applies changes to a VPC security group immediately. So if you have associated the VPC security group with a cluster, inbound cluster access rules in the updated VPC security group apply immediately.

You can create and modify VPC security groups in the <https://console.aws.amazon.com/vpc/>. You can also manage VPC security groups programmatically by using the AWS CLI, the AWS EC2 CLI, and the AWS Tools for Windows PowerShell. For more information about working with VPC security groups, go to [Security Groups for Your VPC](#) in the [Amazon VPC User Guide](#).

Amazon Redshift Cluster Subnet Groups

Overview

You create a cluster subnet group if you are provisioning your cluster in your virtual private cloud (VPC). For more information about VPC, go to [Amazon Virtual Private Cloud \(Amazon VPC\) product detail page](#).

Your VPC can have one or more subnets, a subset of IP addresses within your VPC, that enable you to group your resources based on your security and operation needs. A cluster subnet group allows you to specify a set of subnets in your VPC. When provisioning a cluster you provide the subnet group and Amazon Redshift creates the cluster on one of the subnets in the group.

For more information about creating a VPC, go to [Amazon VPC User Guide](#) Documentation.

After creating a subnet group, you can remove subnets you previously added or add more subnets. Amazon Redshift provides APIs for you to create, modify or delete a cluster subnet group. You can also perform these operations in the console.

Managing Cluster Subnet Groups Using the Console

The section explains how to manage your cluster subnet groups using the Amazon Redshift console. You can create a cluster subnet group, manage an existing one, or delete one. All of these tasks start from the cluster subnet group list. You must select a cluster subnet group to manage it.

In the example cluster subnet group list below, there is one cluster subnet group. By default, there are no cluster subnet groups defined for your AWS account. Because **my-subnet-group** is selected (highlighted), you can edit or delete it. The details of the selected security group are shown under **Cluster Subnet Group Details**.

Cluster Subnet Group List			
Cluster Subnet Group Details			
Name	Description	Status	VPC ID
subnetgroup1	subnet group1	Complete	vpc-XXXXXX
subnetgroup2	subnet group 2	Complete	vpc-XXXXXX

Creating a Cluster Subnet Group

You must have at least one cluster subnet group defined to provision a cluster in a VPC.

To create a cluster subnet group

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, click **Security**.
3. On the **Subnet Groups** tab, click **Create Cluster Subnet Group**.

Create Cluster Subnet Group			
Cluster Subnet Group Details			
Name	Description	Status	VPC ID
subnetgroup1	subnet group1	Complete	vpc-3e0a8053

4. In the **Create Cluster Subnet Group** dialog box, add subnets to the group.

Availability Zone	Subnet ID	CIDR Block	Action
None added			

- a. Specify a **Name**, **Description**, and **VPC ID** for the cluster subnet group.
- b. Add subnets to the group by doing one of the following:
 - Click **add all the subnets** link. or
 - Use the **Availability Zone** and **Subnet ID** boxes to choose a specific subnet and then click **Add**.

The following example shows a cluster subnet group specified with one subnet group.

Availability Zone	Subnet ID	CIDR Block	Action
us-east-1d	subnet-XXXXXX	10.0.0.0/24	Remove

5. Click **Yes, Create**.

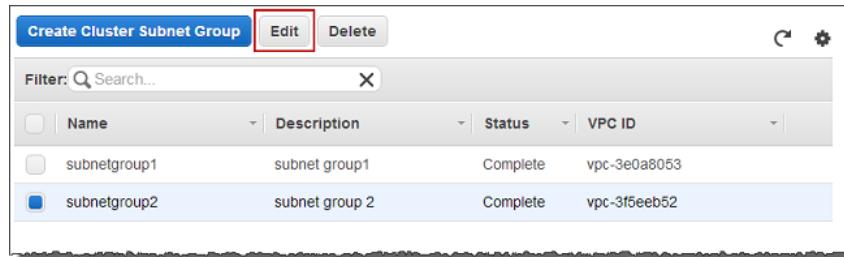
The new group will be displayed in the list of cluster subnet groups.

Modifying a Cluster Subnet Group

To modify a cluster subnet group

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, click **Security**.
3. On the **Subnet Groups** tab, in the cluster subnet group list, click the row of the group you want to modify, and then click **Edit**.

In the example below, **subnetgroup2** is the cluster subnet group we want to modify.



4. In the **Cluster Subnet Group Details**, take one of the following actions.

To...	Do this...
Add one or more subnets to the group.	<p>Select an individual subnet by using the Availability Zone and Subnet ID boxes or click add all the subnets.</p> <p>Click Save.</p>
Remove a subnet from the group.	<p>In the lists of subnets in use for the group, click Remove next to the subnet to remove.</p> <p>Click Save.</p>

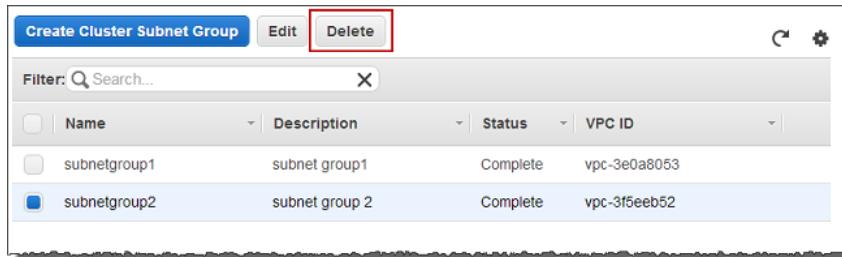
Deleting a Cluster Subnet Group

You cannot delete a cluster subnet group that is used by a cluster.

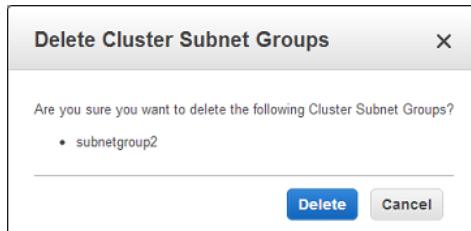
To delete a cluster subnet group

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, click **Security**.
3. On the **Subnet Groups** tab, in the cluster subnet group list, click the row of the group you want to delete.

In the example below, `my-subnet-group` is the cluster subnet group we want to delete.



4. In the Delete Cluster Subnet Group dialog box, click **Delete**.



Managing Cluster Subnet Groups Using the AWS SDK for Java

The following Java code example demonstrates common cluster subnet operations including:

- Creating a cluster subnet group.
- Listing metadata about a cluster subnet group.
- Modifying a cluster subnet group.

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and provide a cluster subnet group name and two subnet identifiers.

Example

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSubnetGroup {

    public static AmazonRedshiftClient client;
    public static String clusterSubnetGroupName = "***provide a cluster subnet group name***";
    // You can use the VPC console to find subnet IDs to use.
    public static String subnetId1 = "***provide a subnet ID***";
    public static String subnetId2 = "***provide a subnet ID***";

    public static void main(String[] args) throws IOException {
        AWS Credentials credentials = new PropertiesCredentials(
            CreateAndModifyClusterSubnetGroup.class
                .getResourceAsStream("AwsCredentials.properties"));
    }
}
```

```
client = new AmazonRedshiftClient(credentials);

try {
    createClusterSubnetGroup();
    describeClusterSubnetGroups();
    modifyClusterSubnetGroup();
} catch (Exception e) {
    System.err.println("Operation failed: " + e.getMessage());
}
}

private static void createClusterSubnetGroup() {
    CreateClusterSubnetGroupRequest request = new CreateClusterSubnetGroupRequest()
        .withClusterSubnetGroupName(clusterSubnetGroupName)
        .withDescription("my cluster subnet group")
        .withSubnetIds(subnetId1);
    client.createClusterSubnetGroup(request);
    System.out.println("Created cluster subnet group: " + clusterSubnetGroupName);
}

private static void modifyClusterSubnetGroup() {
    // Get existing subnet list.
    DescribeClusterSubnetGroupsRequest request1 = new
DescribeClusterSubnetGroupsRequest()
    .withClusterSubnetGroupName(clusterSubnetGroupName);
    DescribeClusterSubnetGroupsResult result1 =
client.describeClusterSubnetGroups(request1);
    List<String> subnetNames = new ArrayList<String>();
    // We can work with just the first group returned since we requested info about one
group.
    for (Subnet subnet : result1.getClusterSubnetGroups().get(0).getSubnets()) {
        subnetNames.add(subnet.getSubnetIdentifier());
    }
    // Add to existing subnet list.
    subnetNames.add(subnetId2);

    ModifyClusterSubnetGroupRequest request = new ModifyClusterSubnetGroupRequest()
        .withClusterSubnetGroupName(clusterSubnetGroupName)
        .withSubnetIds(subnetNames);
    ClusterSubnetGroup result2 = client.modifyClusterSubnetGroup(request);
    System.out.println("\nSubnet group modified.");
    printResultSubnetGroup(result2);
}

private static void describeClusterSubnetGroups() {
    DescribeClusterSubnetGroupsRequest request = new
DescribeClusterSubnetGroupsRequest()
    .withClusterSubnetGroupName(clusterSubnetGroupName);

    DescribeClusterSubnetGroupsResult result = client.describeClusterSubnetGroups(request);
    printResultSubnetGroups(result);
}

private static void printResultSubnetGroups(DescribeClusterSubnetGroupsResult result)
{
    if (result == null)
    {
        System.out.println("\nDescribe cluster subnet groups result is null.");
        return;
    }

    for (ClusterSubnetGroup group : result.getClusterSubnetGroups())
    {
        printResultSubnetGroup(group);
    }
}
```

```
        }
        private static void printResultSubnetGroup(ClusterSubnetGroup group) {
            System.out.format("Name: %s, Description: %s\n", group.getClusterSubnetGroupName(),
group.getDescription());
            for (Subnet subnet : group.getSubnets()) {
                System.out.format(" Subnet: %s, %s, %s\n", subnet.getSubnetIdentifier(),
subnet.getSubnetAvailabilityZone().getName(),
subnet.getSubnetStatus());
            }
        }
    }
```

Manage Cluster Subnet Groups Using Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage cluster subnet groups.

- [create-cluster-subnet-group](#)
- [delete-cluster-subnet-group](#)
- [describe-cluster-subnet-groups](#)
- [modify-cluster-subnet-group](#)

You can use the following Amazon Redshift APIs to manage cluster subnet groups.

- [CreateClusterSubnetGroup](#)
- [DeleteClusterSubnetGroup](#)
- [DescribeClusterSubnetGroups](#)
- [ModifyClusterSubnetGroup](#)

Choosing Cluster Maintenance Tracks

When a new version of Amazon Redshift becomes available, your cluster is updated during its maintenance window. You can control whether your cluster is updated to the most recent approved release or the previous release.

To control this, you set the maintenance track value for the cluster. The two possible values are the following:

- **Current** – The cluster is updated to the most current approved version of Amazon Redshift.
- **Trailing** – The cluster is updated to the approved version of Amazon Redshift before the current one.

For example, if your cluster is currently running version 1.0.2762 and you set the maintenance track value to **Current**, your cluster is updated to version 1.0.3072 (the next approved release) during the next maintenance window. If your cluster's maintenance track value is set to **Trailing**, your cluster isn't updated until there is a new release after 1.0.3072.

Switching between Maintenance Tracks

Changing tracks for a cluster is generally a one-time decision, and you should exercise caution in changing tracks. If you change the maintenance track from **Trailing** to **Current** we will put the cluster on the **Current** track during the next maintenance window. However, if you change the cluster's maintenance track from **Current** to **Trailing** we won't put the cluster on the **Trailing** track until the

Trailing track version has caught up to the release version the cluster was on at the time you made the change.

Working with Maintenance Tracks and Restore or Resize Operations

When you take a snapshot, the snapshot inherits the maintenance track setting from the cluster. Over time the snapshot might be on a different track than the cluster that was the source for the snapshot. For example, suppose that you take a snapshot of a cluster that is on the current track and then change the source cluster to be on the trailing track. In this case, the snapshot and the source cluster are on different tracks. When you restore from the snapshot to a new cluster, the new cluster will be on the maintenance track that was inherited from the source cluster. You can change the maintenance track after you have completed the restore operation.

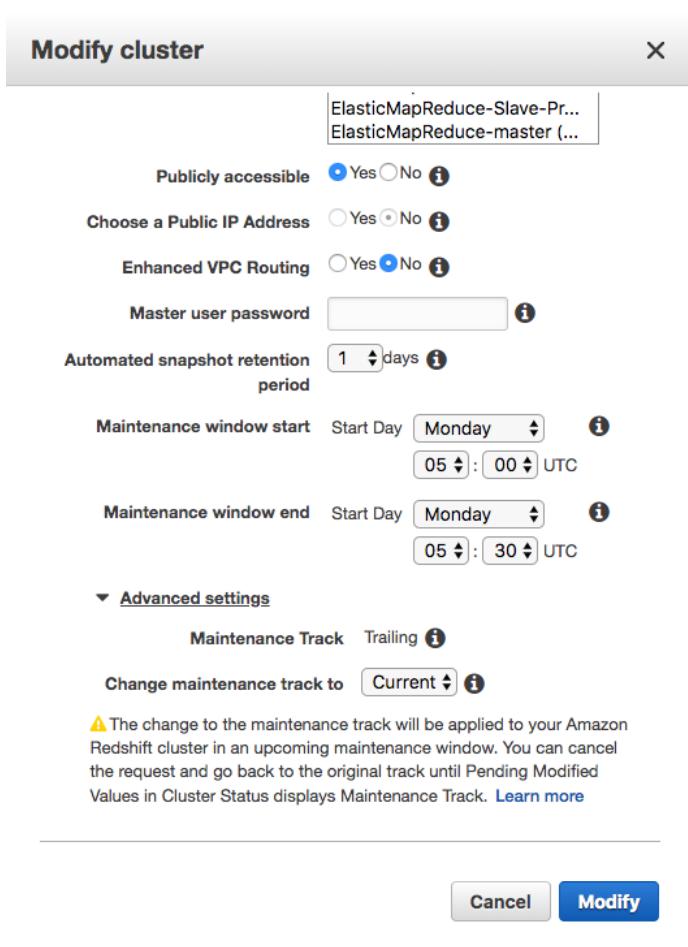
[Resizing a Cluster \(p. 9\)](#) doesn't have any impact on the cluster's maintenance track.

Setting the Maintenance Track for a Cluster with the Console

You can set the maintenance track for a cluster with the console.

To set a maintenance track 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. Choose the cluster that you want to modify.
3. Choose **Advanced settings**.
4. Choose either **Current** or **Trailing**.
5. Choose **Modify**.



Setting the Maintenance Track for a Cluster with the AWS CLI

You can set the maintenance track for a cluster with the AWS CLI.

To set the maintenance track for a cluster with the CLI

- Run the `modify-cluster` command with your cluster identifier and the `maintenance-track-name` option, as shown following.

```
aws redshift modify-cluster --cluster-identifier xxxxxxx --maintenance-track-name trailing
```

Managing Cluster Versions

When Amazon Redshift releases a new cluster version for a track (either Current or Trailing), you can choose to upgrade to that version immediately instead of waiting until your next maintenance window. You can also choose to roll back to a previous version within that track.

You can find which versions are available for upgrade or rollback in the Amazon Redshift console.

To upgrade to a new cluster version

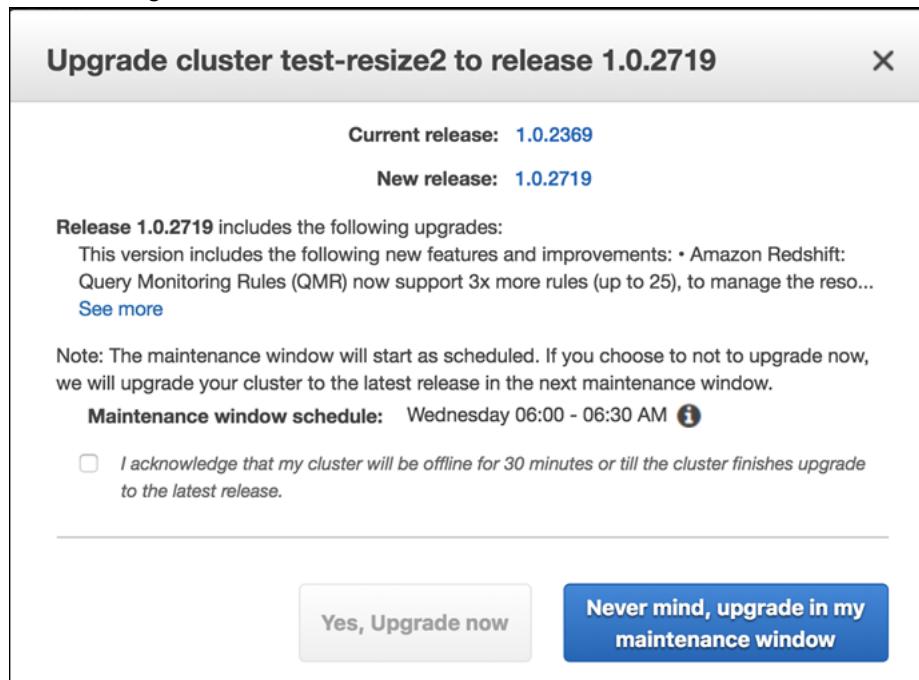
1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation pane, choose **Clusters**.

The **Release Status** column displays one of the following four states:

- **New release available**
- **Up to date**
- **Failed to update**
- **Update in progress**

3. If a new release is available for your cluster, choose **New release available** to get detailed information about what is included in the upgrade.

The following screenshot illustrates the notification of a new release version.



For detailed information about features and improvements included with each cluster version, see [Cluster Version History \(p. 49\)](#).

To roll back to a previous cluster version

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 **Clusters**.
3. Choose the cluster that you want to roll back.

If there is a version available to roll back to, it appears on the status tab of the details page.

The screenshot shows a web-based interface for managing an Amazon Redshift cluster. At the top, there is a dropdown menu labeled "Cluster:" followed by a "Configuration" button. Below this, the "Cluster Status" section is displayed, containing the following information:

Cluster Status	available
Database Health	healthy
In Maintenance Mode	no
Rollback release	Rollback to release [REDACTED]
Parameter Group Apply Status	in-sync
Pending Modified Values	None

4. Choose Rollback to release (release number).

Cluster Version History

Find notes about cluster versions for Amazon Redshift.

Versions 1.0.4349

Time period: October 10–October 31, 2018

Features and Improvements

- **Amazon Redshift:** You can now specify the VERBOSE option in the UNLOAD command to generate additional schema and metadata information inside the manifest.
- **Amazon Redshift:** You can now manage (both see and cancel) the components of a multi-part query in the AWS console.
- **Amazon Redshift:** Significant performance improvements by optimizing the data redistribution strategy during query planning
- **Redshift Spectrum:** You can now use Redshift Spectrum with an Amazon Redshift cluster that is configured to use Enhanced VPC Routing.
- **Redshift Spectrum:** Improved performance when retrieving metadata for late-binding views.
- **Redshift Spectrum:** Improved performance when using ILIKE and NOT ILIKE functions in a Redshift Spectrum query
- **Redshift Spectrum:** ALTER TABLE ADD PARTITION now supports creating multiple partitions at once. This makes managing tables with many partitions faster and more convenient.

Fixes

- Fix for an issue where queries on svv_columns could return an error if there were late-binding views on the cluster.

- Fix for an issue where certain conversion from TIMESTAMP to CHAR or VARCHAR could truncate milliseconds.
- Fix for an issue where some very large Redshift Spectrum queries could encounter out of memory errors.

Versions 1.0.3945, 1.0.4081, 1.0.4222

Time period: September 19–October 10, 2018

Features and Improvements

- **Amazon Redshift:** You can specify the HEADER option in the UNLOAD command to add a specific header row to each created output file. The header row contains the column names created by the unload query.
- **Amazon Redshift:** Significant performance improvements for queries with LIKE predicates.
- **Amazon Redshift:** Improved query performance for mixed read and write workloads.

Fixes

- Enhanced the performance of catalog access for clusters with highly demanding workloads.
- Fix for an issue when a lateral column alias reference is used with window functions.
- Fix for an issue with evaluating NULL values in LEAST() and GREATEST() functions.
- Fix for an issue for certain queries specifying LIMIT 0.
- Fix for an issue when terminating a session with pg_terminate_backend while the session was returning data.

Versions 1.0.3639, 1.0.3688

Time period: August 29–September 19, 2018

Features and Improvements

- **Amazon Redshift:** Significant performance improvements for complex queries that create internal temporary tables during query execution.
- **Amazon Redshift:** Significant performance improvements for some queries with rank functions: rank(), dense_rank() and row_number() in a subquery.
- **Redshift Spectrum:** Query support for nested data has been extended to support EXISTS and NOT EXISTS subqueries.
- **Redshift Spectrum:** Performance improvement of IN-list predicate processing in Redshift Spectrum scans.

Fixes

- Fix for an issue in queries with a UDF that keeps running beyond the requirements of the LIMIT clause.
- Fix for a permission issue when a regular view references a late binding view.

Versions 1.0.3324, 1.0.3359

Time period: August 9–August 28, 2018

Features and Improvements

- **Amazon Redshift:** Improved performance for write and update workloads by automatically enhancing the locality of system metadata during the maintenance window of your cluster.
- **Amazon Redshift:** Significant performance improvements for complex EXCEPT subqueries.
- **Amazon Redshift:** Performance improvements for joins involving large numbers of NULL values in a join key column.
- **Amazon Redshift:** Performance improvement for queries that refer to stable functions over constant expressions.
- **Amazon Redshift:** Performance improvements for queries with intermediate subquery results that can be distributed.
- **Redshift Spectrum:** Performance improvements for queries with expressions on the partition columns of external tables.
- **Redshift Spectrum:** You can now specify the root of an S3 bucket as the data source for an external table.

Fixes

- Fix for permission error when the definition of a regular view refers to a late-binding view.
- Fix for error 'variable not found in subplan target lists' that affected some queries.

Version 1.0.3025

Time period: July 19–August 8, 2018

Features and Improvements

- **Amazon Redshift:** Queries can now refer to a column alias within the same query immediately after it is declared, improving the readability of complex SQL queries.
- **Amazon Redshift:** DC1 reserved nodes can now be migrated for free to DC2 with no change in term length.
- **Amazon Redshift:** A new Amazon CloudWatch metric tracks the current number of waiting queries per workload management (WLM) queue.
- **Amazon Redshift:** Made significant performance improvements for single-row inserts into a table.
- **Amazon Redshift:** Made significant performance improvements for queries operating over CHAR and VARCHAR columns.
- **Amazon Redshift:** Improved performance for hash joins that have filtering dimensions in inner joins or right outer joins.
- **Amazon Redshift:** Improved performance for queries that refer to stable functions with constant expressions.
- **Redshift Spectrum:** Extended the SVV_COLUMNS catalog view to provide information about the columns of late-binding views.

Fixes

- Improved memory management for prefetching for wide table scans or aggregations.
- Improved behavior for long-running queries against Redshift Spectrum external tables with a large number of small files.

Versions 1.0.2762, 1.0.2819, 10.2842

Time period: June 27–July 18, 2018

Features and Improvements

- **Amazon Redshift:** Improved handling of short read queries leading to reduced contention and better system utilization.
- **Amazon Redshift:** Significant enhancement to the Redshift cluster resize feature, enabling you to create temporary tables during the resize operation.
- **Amazon Redshift:** Made further improvements to the compiled code cache, leading to an improvement in overall query performance by reducing the number of segments that have to be recompiled.
- **Amazon Redshift:** Performance improvements for the COPY operation when ingesting data from Parquet and ORC file formats.
- **Redshift Spectrum:** You can now query external columns during a resize operation.
- **Redshift Spectrum:** Made performance improvements for Redshift Spectrum queries with aggregations on partition columns.

Fixes

- Fix for an issue when reassigning a query from the short query queue to a user queue.
- Fix for an issue when pushing down predicates during a query rewrite.
- Fix for an issue with the DATE_TRUNC function in DST timezones.
- Fix for an issue when the LOWER function is evaluated on specific multibyte character sequences.
- Fix for a data type derivation issue in a CREATE TABLE AS (CTAS) statement.
- Enhanced error messaging when the Redshift table total column count limit is exceeded by the number of external tables.

Versions 1.0.2524, 1.0.2557, 1.02610, 1.0.2679, 1.02719

Time period: June 7–July 5, 2018

Features and Improvements

- **Amazon Redshift:** Query Monitoring Rules (QMR) now support three times as many rules (up to 25). You can use QMRs to manage the resource allocation of your Amazon Redshift cluster based on query execution boundaries for WLM queues and take action automatically when a query goes beyond those boundaries.
- **Amazon Redshift:** A new Amazon CloudWatch metric, QueryRuntimeBreakdown, is now available. You can use this metric to get details on the various query execution stages. For more information, see [Amazon Redshift Performance Data \(p. 256\)](#).
- **Redshift Spectrum:** You can now rename external columns.
- **Redshift Spectrum:** You can now specify the file compression type for external tables.
- **Amazon Redshift:** Enhanced performance to hash joins when queries involve large joins. Some complex queries now run three times as fast.
- **Amazon Redshift:** Substantial enhancements to the VACUUM DELETE command so that it frees up additional disk space and has improved performance.
- **Amazon Redshift:** Late materialization now supports DELETE and UPDATE operations, enhancing query performance.
- **Amazon Redshift:** Improved performance for cluster resize operations.

- **Redshift Spectrum:** Support for an increased number of add and drop operations on a single external Redshift Spectrum table.
- **Redshift Spectrum:** Enhanced predicate filtering efficiency when using the DATE_TRUNC function for timestamp columns.

Fixes

- Fix for an issue for queries based on certain views with constants.
- Fix for an issue when accessing certain late-binding views after a schema is renamed.
- Fix for an issue related to unsupported join type.
- Fix for an issue during query hopping as part of Amazon Redshift Workload Management.
- Fix for an issue when compiling certain very large queries.
- Fix for whitespace handling in the format string of the TO_DATE function.
- Fix for an issue when canceling some queries.
- Fix an issue when joining on a partition column of CHAR or DECIMAL type.

Version 1.0.2294, 1.0.2369

Time period: May 17–June 14, 2018

Features and Improvements

- **Amazon Redshift:** The COPY command now supports loading data in Parquet and ORC columnar formats to Amazon Redshift.
- **Amazon Redshift:** Short Query Acceleration now automatically and dynamically determines the maximum run time for short queries based on your workload to further enhance query performance.
- **Redshift Spectrum:** ALTER TABLE ADD/DROP COLUMN for external tables is now supported by using standard JDBC calls.
- **Redshift Spectrum:** Amazon Redshift can now push the LENGTH() string function to Redshift Spectrum, enhancing performance.
- **Amazon Redshift:** Significant improvements to the effectiveness of the compiled code cache leading to an improvement in overall query performance by reducing the number of segments that have to be recompiled.
- **Amazon Redshift:** Improved performance of certain UNION ALL queries.
- **Amazon Redshift:** Improved performance of the communication layer when dealing with out-of-order network packets.

Fixes

- Fix for an issue when NULL values are generated for certain queries.

Version 1.0.2058

Time period: April 19–May 8, 2018

Features and Improvements

- **Monitoring functionality and performance enhancements** – Amazon CloudWatch metrics for Amazon Redshift provide detailed information about your cluster's health and performance. You can now use the added Query Throughput and Query Latency metrics with Amazon Redshift.

- **Console performance for query plan execution details has been significantly improved** – For more information, see [Amazon Redshift Performance Data \(p. 256\)](#).
- **Amazon Redshift:** Enhanced performance for aggregation queries.
- **Amazon Redshift:** Added support for closing client connections in case keepalive hasn't been acknowledged.
- **Redshift Spectrum:** Support for an increased number of add and drop operations on a single external Redshift Spectrum table.
- **Redshift Spectrum:** Enhanced predicate filtering efficiency when using the DATE_TRUNC function for timestamp columns.

Fixes

- Fix for an issue during the optimization of certain complex queries having an EXISTS clause.
- Fix for a permission error when some late-binding views are nested within regular views.
- Fix for an issue during the creation of certain late-binding views that specify 'WITH NO SCHEMA BINDING', where the underlying view definition has a WITH clause.
- Fix for an issue when deriving data types for columns of certain CREATE TABLE AS SELECT statements involving UNION ALL queries.
- Fix for Redshift Spectrum issue for DROP TABLE IF EXISTS when the table doesn't exist.

Version 1.0.2044

Time period covered: March 16–April 18, 2018

Features and Improvements

- **Increase to the number of tables you can create to 20,000 for 8xlarge cluster types** – You now have greater control and granularity for your data with twice as many tables.
- **New search patterns for REGEXP_INSTR and REGEXP_SUBSTR** – You can now search for a specific occurrence of a regular expression pattern and specify case sensitivity.
- **Amazon Redshift:** Performance of window functions has been significantly improved for 8xlarge clusters.
- **Amazon Redshift:** Performance improvement for resize operations.
- **Redshift Spectrum:** Now available in Asia Pacific (Mumbai) and South America (São Paulo) regions.
- **Redshift Spectrum:** Support for ALTER TABLE ADD/DROP COLUMN for external tables.
- **Redshift Spectrum:** New column added to stl_s3query, sv1_s3query, and sv1_s3query_summary showing file formats for external tables.

Fixes

- Fix for an issue during query cancellation.
- Fix for an issue when dropping temp tables.
- Fix for an issue when altering a table.

Versions 1.0.1792, 1.0.1793, 1.0.1808, 1.0.1865

Time period covered: February 22–March 15, 2018

Features and Improvements

- **Amazon Redshift:** Start-time scheduling added to improve the response time of complex queries that are decomposed into multiple subqueries.
- **Redshift Spectrum** You can now process scalar JSON and ION file formats for external tables in Amazon S3.
- **Amazon Redshift:** Short Query Acceleration: Enhanced accuracy of the ML classifier for highly selective queries.
- **Amazon Redshift:** Improved hash join execution time.
- **Amazon Redshift:** Improved runtime of vacuum delete operations on DISTALL tables.
- **Redshift Spectrum:** Improved runtime for queries of external tables consisting of multiple small files.
- **Redshift Spectrum:** Improved performance for querying `svv_external_tables` and `svv_external_columns`.

Fixes

- Improved handling of Boolean predicates during query rewrites.
- Fix for an issue with inconsistent temp table states.
- Fix for a race condition between drop and select statements.
- Enhanced resiliency during cancellation of some queries with prepared statements.
- Enhanced resiliency for some query rewrites.

Versions 1.0.1691 and 1.0.1703

Time period: February 1–February 23, 2018

Features and Improvements

- **Amazon Redshift:** Enhanced support for regular expressions. `REGEXP_INSTR()` and `REGEXP_SUBSTR()` can now search for a specific occurrence of a regular expression pattern and specify case sensitivity when matching a pattern.
- **Amazon Redshift:** Enhanced result caching to support cursor queries and prepared statements.
- **Amazon Redshift:** Enhanced Short Query Acceleration to handle a burst of short queries.
- **Amazon Redshift:** Added support for IAM role chaining to assume roles that span accounts.
- **Redshift Spectrum:** Pushing to Redshift Spectrum the `F_TIMESTAMP_PL_INTERVAL`, `T_CoalesceExpr`, and `DATE-relates` functions.

Fixes

- Fixes for the calculation of timeouts for very small files on S3.
- PSQL-c was modified to respect the auto-commit behavior for TRUNCATE
- Fix for a query compilation failed error.
- Added support for a three-part column names in late-binding views.
- Fix for queries with Boolean predicates inside complex expressions.
- Redshift Spectrum: Increased parallelism using smaller split sizes.

Version 1.0.1636

Time period: January 11–January 25, 2018

Features and Improvements

- **Amazon Redshift:** Released Elastic Short Query Acceleration which improves system responsiveness when there is a burst of short queries in a workload.
- **Amazon Redshift:** Enabled result caching for read queries within a transaction block.
- **Redshift Spectrum:** Added support for DATE columns in Parquet files and for running CREATE EXTERNAL TABLE command with DATE columns.
- **Redshift Spectrum:** Added support for IF NOT EXISTS with the ADD PARTITION command.
- **Amazon Redshift:** Some short queries over tables with the DISTSTYLE attributes of either KEY or ALL now run three times as fast.

Fixes

-
- Fix for an intermittent assert error during query processing. The assert message is of the form 'p - id=380 not in rowsets list'
- Fix for a memory leak during exception handling of certain queries.
- Addressed inconsistent results between CTAS and INSERT.

Querying a Database

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 a SQL client application.
- Connect to your cluster through a SQL client tool, such as SQL Workbench/j.

Amazon Redshift supports SQL client tools connecting through Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC). Amazon Redshift doesn't provide or install any SQL client tools or libraries, so you must install them on your client computer or Amazon EC2 instance to use them. You can use most SQL client tools that support JDBC or ODBC drivers.

Topics

- [Querying a Database Using the Query Editor \(p. 57\)](#)
- [Connecting to an Amazon Redshift Cluster Using SQL Client Tools \(p. 60\)](#)

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 by using the Query Editor on the Amazon Redshift console.

The following cluster node types support the Query Editor:

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

Using the 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

Be aware of the following considerations when you use the Query Editor:

- Up to 50 users can connect to a cluster with the Query Editor at the same time.

- Up to 500 users can connect to a cluster simultaneously. This total includes the users connecting through the Query Editor.
- Up to 50 workload management (WLM) query slots can be active at the same time. For more information about query slots, see [Implementing Workload Management](#).
- Query Editor only runs short queries that can complete within two minutes.
- Query result sets are paginated with 100 rows per page.
- You can't use the Query Editor with Enhanced VPC Routing. For more information, see [Amazon Redshift Enhanced VPC Routing \(p. 115\)](#).
- You can't use transactions in the Query Editor. For more information about transactions, see [BEGIN](#) in the *Amazon Redshift Database Developer 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 account 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 `Amazon RedshiftQueryEditor` and `AmazonRedshiftReadOnlyAccess`.
7. Choose **Next: Review**.
8. Choose **Add permissions**.

Using the Query Editor

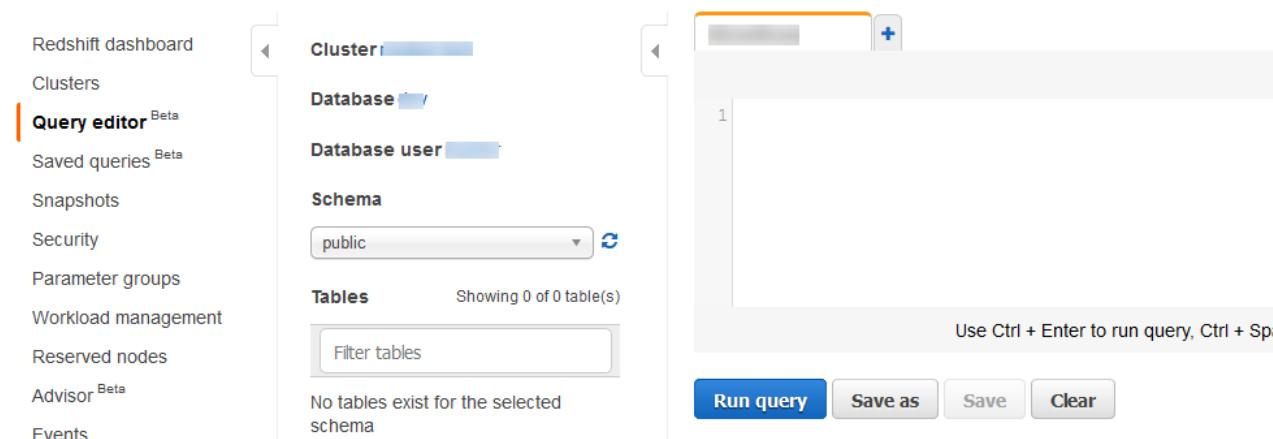
In the following example, you use the Query Editor to perform the following tasks:

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

To complete the following example, you need an existing Amazon Redshift cluster. If you don't have a cluster, create one by following the procedure described in [Creating a Cluster \(p. 17\)](#).

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. 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 query** 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 query** to query the new table.

```
select * from shoes;
```

You should see the following results.

The screenshot shows the Amazon Redshift Query Editor interface. At the top, there is a toolbar with a dropdown menu labeled 'showshoes' and a '+' button. Below the toolbar is a code editor containing the SQL query: 'select * from shoes'. A status message at the bottom of the code editor says 'Use Ctrl + Enter to run query, Ctrl + Space to autocomplete'. Below the code editor is a row of buttons: 'Run query' (highlighted in blue), 'Save as', 'Save', and 'Clear'. To the right of these buttons is a link 'Send feedback'. Underneath the buttons, there are two sections: 'Query results' and 'Showing row(s) 1 - 2'. The 'Query results' section includes a timestamp 'Query completed in 0.642 seconds'. To the right of this section are two buttons: 'View execution' (highlighted in blue) and 'Download CSV'. Below these sections is a table with two rows of data. The table has three columns: an index column (labeled with '1' and '2'), a 'shoetype' column (containing 'sandals' and 'loafers'), and a 'color' column (containing 'black' and 'brown'). There are navigation arrows on either side of the table.

9. Choose **View execution** to view the execution details.
10. Choose **Download CSV** to download the query results as a CSV file.

Connecting to an Amazon Redshift Cluster Using SQL Client Tools

You can connect to Amazon Redshift clusters from SQL client tools over Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) connections. Amazon Redshift does not provide or install any SQL client tools or libraries, so you must install them on your client computer or Amazon EC2 instance to use them to work with data in your clusters. You can use most SQL client tools that support JDBC or ODBC drivers.

You can use this section to walk through the process of configuring your client computer or Amazon EC2 instance to use a JDBC or ODBC connection, and related security options for the client connection to the server. Additionally, in this section you can find information about setting up and connecting from two example third-party SQL client tools, SQL Workbench/J and psql, if you don't have a business intelligence tool to use yet. You can also use this section to learn about connecting to your cluster programmatically. Finally, if you encounter issues when attempting to connect to your cluster, you can review the troubleshooting information in this section to identify possible solutions.

Topics

- [Configuring Connections in Amazon Redshift \(p. 61\)](#)
- [Configure a JDBC Connection \(p. 62\)](#)
- [Configure an ODBC Connection \(p. 77\)](#)
- [Configure Security Options for Connections \(p. 95\)](#)
- [Connecting to Clusters from Client Tools and Code \(p. 100\)](#)
- [Troubleshooting Connection Issues in Amazon Redshift \(p. 109\)](#)

Configuring Connections in Amazon Redshift

Use this section to learn how to configure JDBC and ODBC connections to connect to your cluster from SQL client tools. This section describes how to set up JDBC and ODBC connections and how to use Secure Sockets Layer (SSL) and server certificates to encrypt communication between the client and server.

JDBC and ODBC Drivers for Amazon Redshift

To work with data in your cluster, you need JDBC or ODBC drivers for connectivity from your client computer or instance. Code your applications to use JDBC or ODBC data access APIs, and use SQL client tools that support either JDBC or ODBC.

Amazon Redshift offers JDBC and ODBC drivers for download. Previously, Amazon Redshift recommended PostgreSQL drivers for JDBC and ODBC; if you are currently using those drivers, we recommend moving to the new Amazon Redshift-specific drivers going forward. For more information about how to download the JDBC and ODBC drivers and configure connections to your cluster, see [Configure a JDBC Connection \(p. 62\)](#) and [Configure an ODBC Connection \(p. 77\)](#).

Finding Your Cluster 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 configuration page.

To get your cluster connection string

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. On the **Clusters** page, click the name of the cluster for which you want to get the connection string.
3. On the cluster's **Configuration** tab, under **JDBC URL** or **ODBC URL**, copy the connection string.

The following example shows the connection strings of a cluster launched in the US West region. If you launched your cluster in a different region, the connection strings will be based on that region's endpoint.

Cluster Database Properties	
Port	5439
Publicly Accessible	Yes
Elastic IP	[REDACTED]
Database Name	dev
Master Username	masteruser
Encrypted	Yes
KMS Key ID	[REDACTED] [REDACTED]
JDBC URL	<code>jdbc:redshift://examplecluster.[REDACTED].us-west-2.redshift.amazonaws.com:5439/dev</code>
ODBC URL	<code>Driver={Amazon Redshift (x64)}; Server=examplecluster.[REDACTED].us-west-2.redshift.amazonaws.com; Database=dev; UID=masteruser; PWD=insert_your_master_user_password_here; Port=5439</code>

Configure a JDBC Connection

You can use a JDBC connection to connect to your Amazon Redshift cluster from many third-party SQL client tools. To do this, you need to download a JDBC driver. Follow the steps in this section if you want to use a JDBC connection.

Topics

- [Download the Amazon Redshift JDBC Driver \(p. 62\)](#)
- [Obtain the JDBC URL \(p. 63\)](#)
- [JDBC Driver Configuration Options \(p. 64\)](#)
- [Configure a JDBC Connection with Apache Maven \(p. 68\)](#)
- [Previous JDBC Driver Versions \(p. 71\)](#)

Download the Amazon Redshift JDBC Driver

Amazon Redshift offers drivers for tools that are compatible with either the JDBC 4.2 API, JDBC 4.1 API, or JDBC 4.0 API. For information about the functionality supported by these drivers, go to the [Amazon Redshift JDBC Driver Release Notes](#).

JDBC drivers version 1.2.8.1005 and later support database authentication using AWS Identity and Access Management (IAM) credentials or identity provider (IdP) credentials. For more information, see [Using IAM Authentication to Generate Database User Credentials \(p. 217\)](#).

Download one of the following, depending on the version of the JDBC API that your SQL client tool or application uses. If you're not sure, download the latest version of the JDBC 4.2 API driver.

Note

For driver class name, use either `com.amazon.redshift.jdbc.Driver` or the version-specific class name listed with the driver in the list following.

- JDBC 4.2-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.16.1027/RedshiftJDBC42-1.2.16.1027.jar>.

The class name for this driver is `1.2.16.1027/RedshiftJDBC41-1.2.16.1027.jar com.amazon.redshift.jdbc42.Driver`.

- JDBC 4.1-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1027/1.2.16.1027/RedshiftJDBC41-1.2.16.1027.jar>.

The class name for this driver is `com.amazon.redshift.jdbc41.Driver`.

- JDBC 4.0-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.16.1027/RedshiftJDBC4-1.2.16.1027.jar>.

The class name for this driver is `com.amazon.redshift.jdbc4.Driver`.

The standard Amazon Redshift JDBC drivers include the AWS SDK that is required to use IAM database authentication. We recommend using the standard drivers unless the size of the driver files is an issue for your application. If you need smaller driver files and you do not use IAM database authentication, or if you already have AWS SDK for Java 1.11.118 or later in your Java class path, then download one of the following drivers.

- JDBC 4.2-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.16.1027/RedshiftJDBC42-no-awssdk-1.2.16.1027.jar>.

The class name for this driver is `com.amazon.redshift.jdbc42.Driver`.

- JDBC 4.1-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.16.1027/RedshiftJDBC41-no-awssdk-1.2.16.1027.jar>.

The class name for this driver is `com.amazon.redshift.jdbc41.Driver`.

- JDBC 4.0-compatible driver: <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.16.1027/RedshiftJDBC4-no-awssdk-1.2.16.1027.jar>.

The class name for this driver is `com.amazon.redshift.jdbc4.Driver`.

Then download and review the [Amazon Redshift JDBC Driver License Agreement](#).

If your tool requires a specific previous version of a driver, see [Previous JDBC Driver Versions \(p. 71\)](#).

If you need to distribute these drivers to your customers or other third parties, please send email to redshift-pm@amazon.com to arrange an appropriate license.

Obtain the JDBC URL

Before you can connect to your Amazon Redshift cluster from a SQL client tool, you need to know the JDBC URL of your cluster. The JDBC URL has the following format:

`jdbc:redshift://endpoint:port/database`

Note

A JDBC URL specified with the former format of `jdbc:postgresql://endpoint:port/database` will still work.

Field	Value
<code>jdbc</code>	The protocol for the connection.
<code>redshift</code>	The subprotocol that specifies to use the Amazon Redshift driver to connect to the database.
<code>endpoint</code>	The endpoint of the Amazon Redshift cluster.
<code>port</code>	The port number that you specified when you launched the cluster. If you have a firewall, ensure that this port is open for you to use.
<code>database</code>	The database that you created for your cluster.

The following is an example JDBC URL: `jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

To obtain your JDBC URL

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

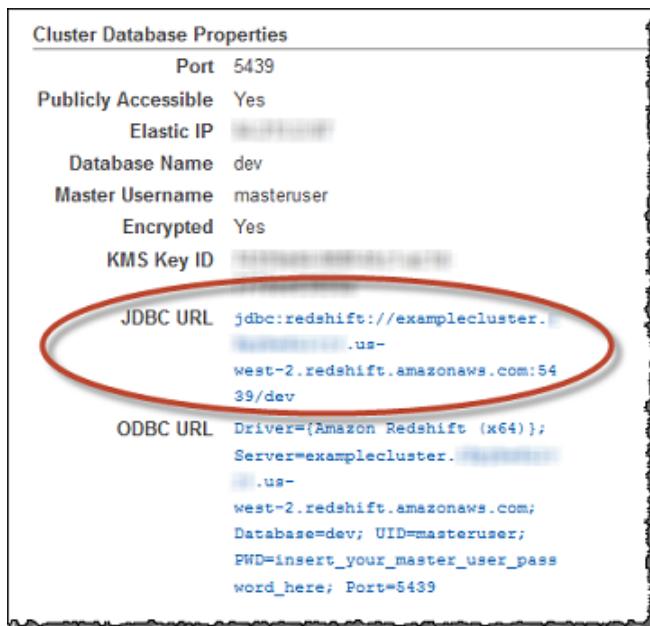
2. At top right, select the region in which you created your cluster.

If you followed the *Amazon Redshift Getting Started*, select **US West (Oregon)**.

3. In the left navigation pane, click **Clusters**, and then click your cluster.

If you followed the *Amazon Redshift Getting Started*, click `examplecluster`.

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



If the client computer fails to connect to the database, you can troubleshoot possible issues. For more information, see [Troubleshooting Connection Issues in Amazon Redshift \(p. 109\)](#).

JDBC Driver Configuration Options

To control the behavior of the Amazon Redshift JDBC driver, you can append the configuration options described in the following table to the JDBC URL. For example, the following JDBC URL connects to your cluster using Secure Socket Layer (SSL), user (UID), and password (PWD).

```
jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev?  
ssl=true&UID=your_username&PWD=your_password
```

For more information about SSL options, see [Connect Using SSL \(p. 95\)](#).

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
AccessKeyId	No	null	The access key ID for the IAM role or IAM user configured for IAM database authentication. The AccessKeyId option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229) . AccessKeyId and SecretAccessKey must be specified together.
AuthMech	No	DISABLE	Deprecated. By default, Amazon Redshift drivers use SSL. Use ssl and sslmode instead.
AutoCreate	No	false	Specify true to create a database user with the name specified for DbUser if one does not exist. The AutoCreate option is part of a set of options

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
			used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
BlockingRowsMode	No	0	The number of rows to hold in memory. After one row is discarded, another row is loaded in its place.
DbGroups	No	null	A comma-delimited list of the names of existing database groups the database user joins for the current session. The DbGroups option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
DbUser	No	null	The name of a database user. The DbUser option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
DisableIsValidQuery	No	false	By default, DisableIsValidQuery is set to False, which enables the java Connection.isValid() method to detect when the JDBC driver no longer has a valid database connection, even if the database connection has terminated unexpectedly. With Amazon Redshift JDBC drivers prior to version 1.2.1, isValid() did not reliably detect when a valid connection was lost. To restore the behavior observed in earlier drivers, set DisableIsValidQuery to true.
DSILogLevel	No	0	<p>Enable logging and specify the amount of detail included in log files. The following table lists the logging detail levels.</p> <ul style="list-style-type: none"> • 0 – Disable all logging. • 1 – Log severe error events that lead the driver to abort. • 2 – Log error events that might allow the driver to continue running. • 3 – Log potentially harmful situations. • 4 – Log general information that describes the progress of the driver. • 5 – Log detailed information that is useful for debugging the driver. • 6 – Log all driver activity.

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
FilterLevel	No	NOTICE	<p>The minimum severity level of a message that the client processes. The following values are possible, in order from lowest to highest severity:</p> <ul style="list-style-type: none"> • DEBUG • INFO • NOTICE • WARNING • LOG • ERROR
loginTimeout	Yes	0	<p>The number of seconds to wait before timing out when connecting to the server. If establishing the connection takes longer than this threshold, then the connection is aborted.</p> <p>When this property is set to the default value of 0, connections do not time out.</p>
loglevel	Yes	null	<p>The amount of logging information output by the driver. By default, no logging is performed. Information will be output to the location specified by the LogStream or LogValue option in the driver manager. The following values are possible:</p> <ul style="list-style-type: none"> • 2 (DEBUG)—Log a lot of detailed information. • 1 (INFO)—Log fewer details. <p>Note Use this property only if you are troubleshooting problems with a driver, because it can affect performance.</p>
OpenSourceSubProtocolOverride		false	<p>When enabled, this setting prevents potential conflicts between the Amazon Redshift JDBC driver and a PostgreSQL JDBC driver. In some cases, your application might simultaneously connect to your cluster using the Amazon Redshift JDBC driver and to other data sources using a PostgreSQL JDBC driver. In this case, append this connection attribute to the JDBC URL that you use to connect to your PostgreSQL data sources. The following values are possible:</p> <ul style="list-style-type: none"> • true—Enable OpenSourceSubProtocolOverride. • false—Disable OpenSourceSubProtocolOverride.

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
Plugin_Name	No	null	The fully qualified class name that implements a credentials provider. The Plugin_Name option is part of a set of options used to configure IAM database authentication. For more information, see Using a Credentials Provider Plugin (p. 229) .
Profile	No	null	The name of a profile in an AWS credentials or config file that contains values for the JDBC connection options. The Plugin_Name option is part of a set of options used to configure IAM database authentication. For more information, see Using a Configuration Profile (p. 230) .
PWD	Yes	null	The password to use to connect to the Amazon Redshift server.
SecretAccessKey	No	null	The secret access key for the IAM role or IAM user configured for IAM database authentication. The SecretAccessKey option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229) . AccessKeyID and SecretAccessKey must be specified together.
SessionToken	No	null	The temporary session token for the IAM role configured for IAM database authentication. SessionToken is not required when IAM database authentication is configured for an IAM user. The SessionToken option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229) . If SessionToken is used, AccessKeyID and SecretAccessKey must also be specified.
socketTimeout	Yes	0	<p>The number of seconds to wait during socket read operations before timing out. If an operation takes longer than this threshold, then the connection is closed.</p> <p>When this property is set to the default value of 0, connections do not time out.</p>
ssl	Yes	true	<p>A value that determines whether to use an SSL connection. The following values are possible:</p> <ul style="list-style-type: none"> • true—Use SSL. • false—Don't use SSL. <p>The driver sets ssl to true by default. If ssl is explicitly set to false with IAM authentication, the connection attempt fails.</p>

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
sslMode	Yes	null	<p>A setting that determines how to handle server certificate verification. The following values are possible:</p> <ul style="list-style-type: none"> • verify-ca—SSL must be used and the server certificate must be verified. • verify-full—SSL must be used. The server certificate must be verified and the server hostname must match the hostname attribute on the certificate. <p>If sslMode is not specified, a server certificate is not required.</p> <p>For more information about sslMode options, see Using SSL and Server Certificates in Java (p. 97).</p> <p>Important Amazon Redshift has changed the way that we manage SSL certificates. If you must use a driver version earlier than 1.2.8.1005, you might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see Transitioning to ACM Certificates for SSL Connections (p. 98).</p>
sslRootCert	No	null	The full path of a .pem or .crt file containing the trust root Certificate Authority (CA) certificate bundle for verifying the Amazon Redshift server certificate when using SSL.
tcpKeepAlive	Yes	true	<p>A value that determines whether TCP keepalives are enabled. The following values are possible:</p> <ul style="list-style-type: none"> • true—Enable TCP keepalives. • false—Disable TCP keepalives.
TCPKeepAliveMinutes	No	5	The threshold for minutes of inactivity before initiating a TCP keepalive transmission.
UID	Yes	null	The user name to use to connect to the Amazon Redshift server.

Configure a JDBC Connection with Apache Maven

Apache Maven is a software project management and comprehension tool. The AWS SDK for Java supports Apache Maven projects. For more information, see [Using the SDK with Apache Maven](#).

If you use Apache Maven, you can configure and build your projects to use an Amazon Redshift JDBC driver to connect to your Amazon Redshift cluster. To do this, you need to add the JDBC driver as a

dependency in your project's `pom.xml` file. Follow the steps in this section if you use Maven to build your project and want to use a JDBC connection.

Configuring the JDBC Driver as a Maven Dependency

To configure the JDBC Driver as a Maven dependency

1. Add the following repository to the repositories section of your `pom.xml` file.

Note

The URL in the following code will return an error if used in a browser. The URL is intended to be used only within the context of a Maven project.

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>http://redshift-maven-repository.s3-website-us-east-1.amazonaws.com/
release</url>
  </repository>
</repositories>
```

To connect using SSL, add the following repository to your `pom.xml` file.

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>https://s3.amazonaws.com/redshift-maven-repository/release</url>
  </repository>
</repositories>
```

2. Declare the version of the driver you want to use in the dependencies section of your `pom.xml` file.

Amazon Redshift offers drivers for tools that are compatible with either the JDBC 4.2 API, JDBC 4.1 API, or JDBC 4.0 API. For information about the functionality supported by these drivers, see the [Amazon Redshift JDBC Driver Release Notes](#).

Add a dependency for the driver from the following list.

Note

For version 1.2.1.1001 and later, you can use either the generic driver class name `com.amazon.redshift.jdbc.Driver` or the version-specific class name listed with the driver in the list following; for example `com.amazon.redshift.jdbc42.Driver`. For releases prior to 1.2.1001, only version specific class names are supported.

- JDBC 4.2-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42</artifactId>
  <version>1.2.10.1009</version>
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc42.Driver`.

- JDBC 4.1-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc41</artifactId>
  <version>1.2.10.1009</version>
```

```
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc41.Driver`.

- JDBC 4.0-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.2.10.1009</version>
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc4.Driver`.

3. Download and review the [Amazon Redshift JDBC License Agreement](#).

The standard Amazon Redshift JDBC drivers include the AWS SDK that is required to use IAM database authentication. We recommend using the standard drivers unless the size of the driver files is an issue for your application. If you need smaller driver files and you do not use IAM database authentication, or if you already have AWS SDK for Java 1.11.118 or later in your Java class path, then add a dependency for the driver from the following list.

- JDBC 4.2-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42-no-awssdk</artifactId>
  <version>1.2.10.1009</version>
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc42.Driver`.

- JDBC 4.1-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc41-no-awssdk</artifactId>
  <version>1.2.10.1009</version>
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc41.Driver`.

- JDBC 4.0-compatible driver:

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4-no-awssdk</artifactId>
  <version>1.2.10.1009</version>
</dependency>
```

The class name for this driver is `com.amazon.redshift.jdbc4.Driver`.

The Amazon Redshift Maven drivers with no SDKs include the following optional dependencies that you can include in your project as needed.

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-core</artifactId>
```

API Version 2012-12-01

```
<version>1.11.118</version>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
<dependency>
<groupId>com.amazonaws</groupId>
<artifactId>aws-java-sdk-redshift</artifactId>
<version>1.11.118</version>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
<dependency>
<groupId>com.amazonaws</groupId>
<artifactId>aws-java-sdk-sts</artifactId>
<version>1.11.118</version>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
```

If your tool requires a specific previous version of a driver, see [Previous JDBC Driver Versions Using Maven \(p. 73\)](#).

If you need to distribute these drivers to your customers or other third parties, please send email to redshift-pm@amazon.com to arrange an appropriate license.

Upgrading the Driver to the Latest Version

To upgrade or change the Amazon Redshift JDBC driver to the latest version, modify the version section of the dependency to the latest version of the driver and then clean your project with the Maven Clean Plugin, as shown following.

```
mvn clean
```

Previous JDBC Driver Versions

Download a previous version of the Amazon Redshift JDBC driver only if your tool requires a specific version of the driver. For information about the functionality supported in previous versions of the drivers, download [Amazon Redshift JDBC Driver Release Notes](#).

For authentication using AWS Identity and Access Management (IAM) credentials or identity provider (IdP) credentials, use Amazon Redshift JDBC driver version 1.2.8.1005 or later.

Important

Amazon Redshift has changed the way that we manage SSL certificates. If you must use a driver version earlier than 1.2.8.1005, you might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

These are previous JDBC 4.2-compatible drivers:

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC42-1.2.15.1025.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC42-no-awssdk-1.2.15.1025.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC42-1.2.12.1017.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC42-no-awssdk-1.2.12.1017.jar>

- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-1.2.10.1009.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-no-awssdk-1.2.10.1009.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-1.2.8.1005.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-no-awssdk-1.2.8.1005.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-1.2.7.1003.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-no-awssdk-1.2.7.1003.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-1.2.1.1001.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC42-1.1.17.1017.jar>.

These are previous JDBC 4.1-compatible drivers:

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/1.2.15.1025/RedshiftJDBC41-1.2.15.1025.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC41-no-awssdk-1.2.15.1025.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC41-1.2.12.1017.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC41-no-awssdk-1.2.12.1017.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.2.10.1009.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-no-awssdk-1.2.10.1009.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.2.8.1005.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-no-awssdk-1.2.8.1005.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.2.7.1003.jar>.

- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-no-awssdk-1.2.7.1003.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.2.1.1001.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.17.1017.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.10.1010.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.9.1009.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.7.1007.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.6.1006.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.2.0002.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.1.0001.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.0.0000.jar>

These are previous JDBC 4.0-compatible drivers:

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC4-1.2.15.1025.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC4-no-awssdk-1.2.15.1025.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC4-1.2.12.1017.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.12.1017/RedshiftJDBC4-no-awssdk-1.2.12.1017.jar>

- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.2.10.1009.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-no-awssdk-1.2.10.1009.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.2.8.1005.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-no-awssdk-1.2.8.1005.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.2.7.1003.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-no-awssdk-1.2.7.1003.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.2.1.1001.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.17.1017.jar>.
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.10.1010.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.9.1009.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.7.1007.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.6.1006.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.2.0002.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.1.0001.jar>
- <https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.0.0000.jar>

Previous JDBC Driver Versions Using Maven

Add a previous version of the Amazon Redshift JDBC driver to your project only if your tool requires a specific version of the driver. For information about the functionality supported in previous versions of the drivers, download [Amazon Redshift JDBC Driver Release Notes](#).

These are previous JDBC 4.2-compatible drivers:

- JDBC42-1.2.8.1005

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42</artifactId>
  <version>1.2.8.1005</version>
</dependency>
```

- JDBC42-1.2.8.1005 no SDK

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42-no-awssdk</artifactId>
  <version>1.2.8.1005</version>
</dependency>
```

- JDBC42-1.2.7.1003

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42</artifactId>
  <version>1.2.7.1003</version>
</dependency>
```

- JDBC42-1.2.1.1001

```
<dependency>
```

```
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc42</artifactId>
<version>1.2.1.1001</version>
</dependency>
```

- JDBC42-1.1.17.1017

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc42</artifactId>
<version>1.1.17.1017</version>
</dependency>
```

These are previous JDBC 4.1-compatible drivers:

- JDBC41-1.2.8.1005

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.2.8.1005</version>
</dependency>
```

- JDBC41-1.2.8.1005 no SDK

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41-no-awssdk</artifactId>
<version>1.2.8.1005</version>
</dependency>
```

- JDBC41-1.2.7.1003

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.2.7.1003</version>
</dependency>
```

- JDBC41-1.2.1.1001

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.2.1.1001</version>
</dependency>
```

- JDBC41-1.1.17.1017

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.17.1017</version>
</dependency>
```

- JDBC41-1.1.10.1010

```
<dependency>
<groupId>com.amazon.redshift</groupId>
```

```
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.10.1010</version>
</dependency>
```

- JDBC41-1.1.9.1009

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.9.1009</version>
</dependency>
```

- JDBC41-1.1.7.1007

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.7.1007</version>
</dependency>
```

- JDBC41-1.1.6.1006

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.6.1006</version>
</dependency>
```

- JDBC41-1.1.2.0002

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.2.0002</version>
</dependency>
```

- JDBC41-1.1.1.0001

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.1.0001</version>
</dependency>
```

- JDBC41-1.1.0.0000

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc41</artifactId>
<version>1.1.0.0000</version>
</dependency>
```

These are previous JDBC 4.0-compatible drivers:

- JDBC4-1.2.8.1005

```
<dependency>
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc4</artifactId>
```

```
<version>1.2.8.1005</version>
</dependency>
```

- JDBC4-1.2.8.1005 no SDK

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4-no-awssdk</artifactId>
  <version>1.2.8.1005</version>
</dependency>
```

- JDBC4-1.2.7.1003

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.2.7.1003</version>
</dependency>
```

- JDBC4-1.2.1.1001

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.2.1.1001</version>
</dependency>
```

- JDBC4-1.1.17.1017

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.17.1017</version>
</dependency>
```

- JDBC4-1.1.10.1010

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.10.1010</version>
</dependency>
```

- JDBC4-1.1.9.1009

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.9.1009</version>
</dependency>
```

- JDBC4-1.1.7.1007

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.7.1007</version>
</dependency>
```

- JDBC4-1.1.6.1006

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.6.1006</version>
</dependency>
```

- JDBC4-1.1.2.0002

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.2.0002</version>
</dependency>
```

- JDBC4-1.1.1.0001

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.1.0001</version>
</dependency>
```

- JDBC4-1.1.0.0000

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc4</artifactId>
  <version>1.1.0.0000</version>
</dependency>
```

Configure an ODBC Connection

You can use an ODBC connection to connect to your Amazon Redshift cluster from many third-party SQL client tools and applications. To do this, you need to set up the connection on your client computer or Amazon EC2 instance. If your client tool supports JDBC, you might choose to use that type of connection rather than ODBC due to the ease of configuration that JDBC provides. However, if your client tool doesn't support JDBC, follow the steps in this section to configure an ODBC connection.

Amazon Redshift provides ODBC drivers for Linux, Windows, and Mac OS X operating systems. Before you install an ODBC driver, you need to determine whether your SQL client tool is 32-bit or 64-bit. You should install the ODBC driver that matches the requirements of your SQL client tool; otherwise, the connection will not work. If you use more than one SQL client tool on the same computer or instance, make sure that you download the appropriate drivers. You might need to install both the 32-bit and the 64-bit drivers if the tools differ in their system architecture.

Topics

- [Obtain the ODBC URL for Your Cluster \(p. 78\)](#)
- [Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems \(p. 79\)](#)
- [Install the Amazon Redshift ODBC Driver on Linux Operating Systems \(p. 82\)](#)
- [Install the Amazon Redshift ODBC Driver on Mac OS X \(p. 84\)](#)
- [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#)
- [ODBC Driver Configuration Options \(p. 88\)](#)
- [Previous ODBC Driver Versions \(p. 93\)](#)

Obtain the ODBC URL for Your Cluster

Amazon Redshift displays the ODBC URL for your cluster in the Amazon Redshift console. This URL contains the information that you need to set up the connection between your client computer and the database.

An ODBC URL has the following format:

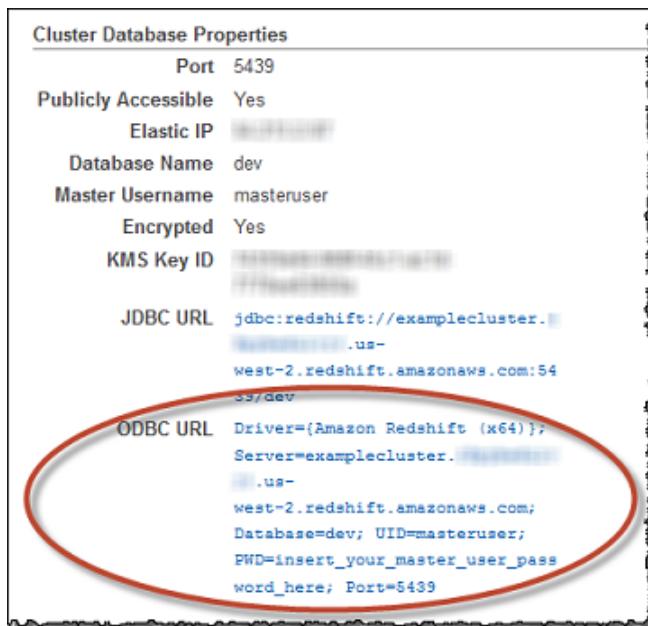
Driver={*driver*};Server=*endpoint*;Database=*database_name*;UID=*user_name*;PWD=*password*;Port=*port_number*

Field	Value
Driver	The name of the ODBC driver to use. Depending on the driver you download for your architecture, values will be Amazon Redshift (x86) (for the 32-bit driver) or Amazon Redshift (x64) (for the 64-bit driver).
Server	The endpoint of the Amazon Redshift cluster.
Database	The database that you created for your cluster.
UID	The user name of a user account that has permission to connect to the database. This value is a database permission, not an Amazon Redshift permission, although you can use the master user account that you set up when you launched the cluster.
PWD	The password for the user account to connect to the database.
Port	The port number that you specified when you launched the cluster. If you have a firewall, ensure that this port is open for you to use.

The following is an example ODBC URL: Driver={Amazon Redshift (x64)}; Server=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com; Database=dev; UID=masteruser; PWD=*insert_your_master_user_password_here*; Port=5439

To obtain your ODBC URL

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. At top right, select the region in which you created your cluster.
If you followed the Amazon Redshift Getting Started, select US West (Oregon).
3. In the left navigation pane, click **Clusters**, and then click your cluster.
If you followed the Amazon Redshift Getting Started, click examplecluster.
4. On the **Configuration** tab, under **Cluster Database Properties**, copy the ODBC URL of the cluster.



Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems

System Requirements

You install the Amazon Redshift ODBC driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- Microsoft Windows Vista operating system or later
- 55 MB of available disk space
- Administrator privileges on the client computer
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Windows Operating Systems

Use the steps in this section to download the Amazon Redshift ODBC drivers for Microsoft Windows operating systems. You should only use a driver other than these if you are running a third-party application that is certified for use with Amazon Redshift and that requires a specific driver for that application.

To install the ODBC driver

1. Download one of the following, depending on the system architecture of your SQL client tool or application:
 - 32-bit: <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.2.1010/AmazonRedshiftODBC32-1.4.2.1010.msi>

The name for this driver is Amazon Redshift (x86).

- 64-bit: <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.2.1010/AmazonRedshiftODBC64-1.4.2.1010.msi>

The name for this driver is Amazon Redshift (x64).

Note

Download the MSI package that corresponds to the system architecture of your SQL client tool or application. For example, if your SQL client tool is 64-bit, install the 64-bit driver.

Then download and review the [Amazon Redshift ODBC Driver License Agreement](#). If you need to distribute these drivers to your customers or other third parties, please email redshift-pm@amazon.com to arrange an appropriate license.

2. Double-click the .msi file, and then follow the steps in the wizard to install the driver.

[Creating a System DSN Entry for an ODBC Connection on Microsoft Windows](#)

After you download and install the ODBC driver, you need to add a data source name (DSN) entry to the client machine or Amazon EC2 instance. SQL client tools use this data source to connect to the Amazon Redshift database.

Note

For authentication using AWS Identity and Access Management (IAM) credentials or identity provider (IdP) credentials, additional steps are required. For more information, see [Configure a JDBC or ODBC Connection to Use IAM Credentials \(p. 222\)](#).

To create a system DSN entry

1. In the **Start** menu, in your list of programs, locate the driver folder or folders.

Note

If you installed the 32-bit driver, the folder is named **Amazon Redshift ODBC Driver (32-bit)**.

If you installed the 64-bit driver, the folder is named **Amazon Redshift ODBC Driver (64-bit)**.

If you installed both drivers, you'll have a folder for each driver.

2. Click **ODBC Administrator**, and then type your administrator credentials if you are prompted to do so.
3. Select the **System DSN** tab if you want to configure the driver for all users on the computer, or the **User DSN** tab if you want to configure the driver for your user account only.
4. Click **Add**. The **Create New Data Source** window opens.
5. Select the **Amazon Redshift** ODBC driver, and then click **Finish**. The **Amazon Redshift ODBC Driver DSN Setup** window opens.
6. Under **Connection Settings**, enter the following information:

Data Source Name

Type a name for the data source. You can use any name that you want to identify the data source later when you create the connection to the cluster. For example, if you followed the *Amazon Redshift Getting Started*, you might type `exampleclusterdsn` to make it easy to remember the cluster that you will associate with this DSN.

Server

Specify the endpoint for your Amazon Redshift cluster. You can find this information in the Amazon Redshift console on the cluster's details page. For more information, see [Configuring Connections in Amazon Redshift \(p. 61\)](#).

Port

Type the port number that the database uses. By default, Amazon Redshift uses 5439, but you should use the port that the cluster was configured to use when it was launched.

Database

Type the name of the Amazon Redshift database. If you launched your cluster without specifying a database name, type **dev**; otherwise, use the name that you chose during the launch process. If you followed the *Amazon Redshift Getting Started*, type **dev**.

7. Under **Credentials**, enter the following information:

User

Type the user name for the database user account that you want to use to access the database. If you followed the *Amazon Redshift Getting Started*, type **masteruser**.

Password

Type the password that corresponds to the database user account.

8. Under **SSL Settings**, specify a value for the following:

SSL Authentication

Select a mode for handling Secure Sockets Layer (SSL). In a test environment, you might use **prefer**, but for production environments and when secure data exchange is required, use **verify-ca** or **verify-full**. For more information about using SSL, see [Connect Using SSL \(p. 95\)](#).

9. Under **Additional Options**, select one of the following options to specify how to return query results to your SQL client tool or application:

- **Single Row Mode.** Select this option if you want query results to be returned one row at a time to the SQL client tool or application. Use this option if you plan to query for large result sets and don't want the entire result in memory. Disabling this option improves performance, but it can increase the number of out-of-memory errors.
- **Use Declare/Fetch.** Select this option if you want query results to be returned to the SQL client tool or application in a specified number of rows at a time. Specify the number of rows in **Cache Size**.
- **Use Multiple Statements.** Select this option to return results based on multiple SQL statements in a query.
- **Retrieve Entire Result Into Memory.** Select this option if you want query results to be returned all at once to the SQL client tool or application. The default is enabled.

10. In **Logging Options**, specify values for the following:

- **Log Level.** Select an option to specify whether to enable logging and the level of detail that you want captured in the logs.

Important

You should only enable logging when you need to capture information about an issue. Logging decreases performance, and it can consume a large amount of disk space.

- **Log Path.** Specify the full path to the folder where you want to save log files.

Then click **OK**.

11. In **Data Type Options**, specify values for the following:

- **Use Unicode.** Select this option to enable support for Unicode characters. The default is enabled.
- **Show Boolean Column As String.** Select this option if you want Boolean values to be displayed as string values instead of bit values. If you enable this, "1" and "0" display instead of **1** and **0**. The default is enabled.
- **Text as LongVarChar.** Select this option to enable showing text as LongVarChar. The default is enabled.
- **Max Varchar.** Specify the maximum value for the Varchar data type. A Varchar field with a value larger than the maximum specified will be promoted to LongVarChar. The default value is 255.
- **Max LongVarChar.** Specify the maximum value for the LongVarChar data type. A LongVarChar field value that is larger than the maximum specified will be truncated. The default value is 8190.

- **Max Bytea.** Specify the maximum value for the Bytea data type. A Bytea field value that is larger than the maximum specified will be truncated. The default value is 255.

Note

The Bytea data type is only used by Amazon Redshift system tables and views, and otherwise is not supported.

Then click **OK**.

12Click **Test**. If the client computer can connect to the Amazon Redshift database, you will see the following message: **Connection successful**.

If the client computer fails to connect to the database, you can troubleshoot possible issues. For more information, see [Troubleshooting Connection Issues in Amazon Redshift \(p. 109\)](#).

Install the Amazon Redshift ODBC Driver on Linux Operating Systems

System Requirements

You install the Amazon Redshift ODBC driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- One of the following Linux distributions (32- and 64-bit editions):
 - Red Hat Enterprise Linux (RHEL) 5.0/6.0/7.0
 - CentOS 5.0/6.0/7.0
 - Debian 7
 - SUSE Linux Enterprise Server (SLES) 11
- 75 MB of available disk space
- One of the following ODBC driver managers:
 - iODBC Driver Manager 3.52.7 or later. For more information about the iODBC driver manager and links to download it, go to the [Independent Open Database Connectivity website](#).
 - unixODBC 2.3.0 or later. For more information about the unixODBC driver manager and links to download it, go to the [unixODBC website](#).
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Linux Operating Systems

Use the steps in this section to download and install the Amazon Redshift ODBC drivers on a supported Linux distribution. The installation process will install the driver files in the following directories:

- /opt/amazon/redshiftodbc/lib/32 (for a 32-bit driver)
- /opt/amazon/redshiftodbc/lib/64 (for a 64-bit driver)
- /opt/amazon/redshiftodbc/ErrorMessages
- /opt/amazon/redshiftodbc/Setup

To install the Amazon Redshift ODBC driver

1. Download one of the following, depending on the system architecture of your SQL client tool or application:

- 32-bit .rpm: <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.2.1010/AmazonRedshiftODBC-32-bit-1.4.2.1010-1.i686.rpm>
- 64-bit .rpm: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64-bit-1.4.2.1010-1.x86_64.rpm
- Debian 32-bit .rpm: <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.2.1010/AmazonRedshiftODBC-32-bit-1.4.2.1010-1.i686.deb>
- Debian 64-bit .rpm: https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.2.1010/AmazonRedshiftODBC-64-bit-1.4.2.1010-1.x86_64.deb

The name for both of these drivers is Amazon Redshift ODBC Driver.

Note

Download the package that corresponds to the system architecture of your SQL client tool or application. For example, if your client tool is 64-bit, install a 64-bit driver.

Then download and review the [Amazon Redshift ODBC Driver License Agreement](#). If you need to distribute these drivers to your customers or other third parties, please email redshift-pm@amazon.com to arrange an appropriate license.

2. Navigate to the location where you downloaded the package, and then run one of the following commands. Use the command that corresponds to your Linux distribution.
 - On RHEL 5.0/6.0 and CentOS 5.0/6.0 operating systems, run this command:

```
yum --nogpgcheck localinstall RPMFileName
```

Replace *RPMFileName* with the RPM package file name. For example, the following command demonstrates installing the 32-bit driver:

```
yum --nogpgcheck localinstall AmazonRedshiftODBC-32bit-1.x.x.xxxx-x.x86_64.deb
```

- On SLES 11, run this command:

```
zypper install RPMFileName
```

Replace *RPMFileName* with the RPM package file name. For example, the following command demonstrates installing the 64-bit driver:

```
zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
```

- On Debian 7, run this command:

```
sudo apt install ./DEBFileName.deb
```

Replace *DEBFileName.deb* with the Debian package file name. For example, the following command demonstrates installing the 64-bit driver:

```
sudo apt install ./AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.deb
```

Important

When you have finished installing the drivers, configure them for use on your system. For more information on driver configuration, see [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#).

Install the Amazon Redshift ODBC Driver on Mac OS X

System Requirements

You install the driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- Mac OS X version 10.6.8 or later
- 215 MB of available disk space
- iODBC Driver Manager version 3.52.7 or later. For more information about the iODBC driver manager and links to download it, go to the [Independent Open Database Connectivity website](#).
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Mac OS X

Use the steps in this section to download and install the Amazon Redshift ODBC driver on a supported version of Mac OS X. The installation process will install the driver files in the following directories:

- /opt/amazon/redshift/lib/universal
- /opt/amazon/redshift/ErrorMessages
- /opt/amazon/redshift/Setup

To install the Amazon Redshift ODBC driver on Mac OS X

1. Download <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/> **AmazonRedshiftODBC-1.4.2.1010.dmg**. The name for this driver is Amazon Redshift ODBC Driver.
Then download and review the [Amazon Redshift ODBC Driver License Agreement](#). If you need to distribute these drivers to your customers or other third parties, please email *redshift-pm@amazon.com* to arrange an appropriate license.
2. Double-click **AmazonRedshiftODBC.dmg** to mount the disk image.
3. Double-click **AmazonRedshiftODBC.pkg** to run the installer.
4. Follow the steps in the installer to complete the driver installation process. You'll need to agree to the terms of the license agreement to perform the installation.

Important

When you have finished installing the driver, configure it for use on your system. For more information on driver configuration, see [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#).

Configure the ODBC Driver on Linux and Mac OS X Operating Systems

On Linux and Mac OS X operating systems, you use an ODBC driver manager to configure the ODBC connection settings. ODBC driver managers use configuration files to define and configure ODBC data sources and drivers. The ODBC driver manager that you use depends on the operating system that you use. For more information about the supported ODBC driver managers to configure the Amazon Redshift ODBC drivers, see [System Requirements \(p. 82\)](#) for Linux operating systems and [System Requirements \(p. 84\)](#) for Mac OS X operating systems.

Three files are required for configuring the Amazon Redshift ODBC driver: `amazon.redshiftodbc.ini`, `odbc.ini`, and `odbcinst.ini`.

If you installed to the default location, the `amazon.redshiftodbc.ini` configuration file is located in one of the following directories:

- `/opt/amazon/redshiftodbc/lib/32` (for the 32-bit driver on Linux operating systems)
- `/opt/amazon/redshiftodbc/lib/64` (for the 64-bit driver on Linux operating systems)
- `/opt/amazon/redshift/lib` (for the driver on Mac OS X)

Additionally, under `/opt/amazon/redshiftodbc/Setup` on Linux or `/opt/amazon/redshift/Setup` on Mac OS X, there are sample `odbc.ini` and `odbcinst.ini` files for you to use as examples for configuring the Amazon Redshift ODBC driver and the data source name (DSN).

We don't recommend using the Amazon Redshift ODBC driver installation directory for the configuration files. The sample files in the Setup directory are for example purposes only. If you reinstall the Amazon Redshift ODBC driver at a later time, or upgrade to a newer version, the installation directory is overwritten and you'll lose any changes you might have made to those files.

To avoid this, you should copy the `amazon.redshiftodbc.ini` file to a directory other than the installation directory. If you copy this file to the user's home directory, add a period (.) to the beginning of the file name to make it a hidden file.

For the `odbc.ini` and `odbcinst.ini` files, you should either use the configuration files in the user's home directory or create new versions in another directory. By default, your Linux or Mac OS X operating system should have an `.odbc.ini` file and an `.odbcinst.ini` file in the user's home directory (`/home/$USER` or `~/`). These default files are hidden files, which are indicated by the dot (.) in front of the file name, and they will only display when you use the `-a` flag to list the directory contents.

Whichever option you choose for the `odbc.ini` and `odbcinst.ini` files, you will need to modify them to add driver and DSN configuration information. If you chose to create new files, you also need to set environment variables to specify where these configuration files are located.

Configuring the `odbc.ini` File

You use the `odbc.ini` file to define data source names (DSNs).

Use the following format on Linux operating systems:

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

The following example shows the configuration for `odbc.ini` on Linux operating systems:

```
[ODBC Data Sources]
Amazon_Redshift_x32=Amazon Redshift (x86)
Amazon_Redshift_x64=Amazon Redshift (x64)

[Amazon Redshift (x86)]
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

```
[Amazon Redshift (x64)]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Use the following format on Mac OS X operating systems:

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/libamazonredshiftodbc.dylib

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

The following example shows the configuration for odbc.ini on Mac OS X operating systems:

```
[ODBC Data Sources]
Amazon_Redshift_dylib=Amazon Redshift DSN for Mac OS X

[Amazon Redshift DSN for Mac OS X]
Driver=/opt/amazon/redshift/libamazonredshiftodbc.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Configuring the odbcinst.ini File

You use the odbcinst.ini file to define ODBC drivers.

Use the following format on Linux operating systems:

```
[ODBC Drivers]
driver_name=Installed
...
[driver_name]
Description=driver_description
Driver=path/driver_file
...
...
```

The following example shows the odbcinst.ini configuration for both the 32-bit and 64-bit drivers installed in the default directories on Linux operating systems:

```
[ODBC Drivers]
Amazon Redshift (x86)=Installed
Amazon Redshift (x64)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so

[Amazon Redshift (x64)]
```

```
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

Use the following format on Mac OS X operating systems:

```
[ODBC Drivers]
driver_name=Installed
...
[driver_name]
Description=driver_description
Driver=path/libamazonredshiftodbc.dylib
...
...
```

The following example shows the odbcinst.ini configuration for the driver installed in the default directory on Mac OS X operating systems:

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for Mac OS X
Driver=/opt/amazon/redshift/libamazonredshiftodbc.dylib
```

Configuring Environment Variables for Driver Configuration Files

In order for the Amazon Redshift ODBC driver to function properly, you need to set a number of environmental variables, as described following.

Set an environment variable to specify the path to the driver manager libraries:

- On Linux, set `LD_LIBRARY_PATH` to point to the directory containing the driver manager libraries. For more information on supported driver managers, see [Install the Amazon Redshift ODBC Driver on Linux Operating Systems \(p. 82\)](#).
- On Mac OS X, set `DYLD_LIBRARY_PATH` to point to the directory containing the driver manager libraries. For more information on supported driver managers, see [Install the Amazon Redshift ODBC Driver on Mac OS X \(p. 84\)](#).

Optionally, set `AMAZONREDSHIFTODBCINI` to point to your `amazon.redshiftodbc.ini` file.

`AMAZONREDSHIFTODBCINI` must specify the full path, including the file name. You must either set this variable, or place this file in a location where the system will find it in a search. The following search order is used to locate the `amazon.redshiftodbc.ini` file:

1. If the `AMAZONREDSHIFTODBCINI` environment variable is defined, then the driver searches for the file specified by the environment variable.
2. If the `AMAZONREDSHIFTODBCINI` environment variable is not defined, then the driver searches in its own directory—that is, the directory that contains the driver binary.
3. If the `amazon.redshiftodbc.ini` file cannot be found, the driver tries to automatically determine the driver manager settings and connect. However, error messages won't display correctly in this case.

If you decide to use a directory other than the user's home directory for the `odbc.ini` and `odbcinst.ini` files, you also need to set environment variables to specify where the configuration files appear:

- Set `ODBCINI` to point to your `odbc.ini` file.
- Set `ODBCSYSINI` to point to the directory containing the `odbcinst.ini` file.

If you are on Linux, your driver manager libraries are located in the /usr/local/lib directory, your odbc.ini and amazon.redshiftodbc.ini files are located in the /etc directory, and your odbcinst.ini file is located in the /usr/local/odbc directory, then set the environment variables as shown in the following example:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export ODBCINI=/etc/odbc.ini
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
export ODBCSYSINI=/usr/local/odbc
```

If you are on Mac OS X, your driver manager libraries are located in the /usr/local/lib directory, your odbc.ini and amazon.redshiftodbc.ini files are located in the /etc directory, and your odbcinst.ini file is located in the /usr/local/odbc directory, then set the environment variables as shown in the following example:

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
export ODBCINI=/etc/odbc.ini
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
export ODBCSYSINI=/usr/local/odbc
```

ODBC Driver Configuration Options

You can use the configuration options described in the following table to control the behavior of the Amazon Redshift ODBC driver.

In Microsoft Windows, you typically set driver options when you configure a data source name (DSN). You can also set driver options in the connection string when you connect programmatically, or by adding or changing registry keys in HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\your_DSN. For more information about configuring a DSN, see [Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems \(p. 79\)](#). For an example of setting driver options in a connection string, see [Connect to Your Cluster Programmatically \(p. 106\)](#).

In Linux and Mac OS X, you set driver configuration options in your odbc.ini and amazon.redshiftodbc.ini files, as described in [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#). Configuration options set in an amazon.redshiftodbc.ini file apply to all connections. In contrast, configuration options set in an odbc.ini file are specific to a connection. Configuration options set in odbc.ini take precedence over configuration options set in amazon.redshiftodbc.ini.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
AccessKeyId	No	null	The access key ID for the IAM role or IAM user configured for IAM database authentication. The AccessKeyId option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229) . AccessKeyId and SecretAccessKey must be specified together.
AutoCreate	No	false	Specify true to create a database user with the name specified for DbUser if one does not exist. The AutoCreate option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
BoolsAsChar	Yes	0	<p>When this option is enabled (1), the driver exposes Boolean values as data type SQL_VARCHAR with a length of 5.</p> <p>When this option is disabled (0), the driver exposes Boolean values as data type SQL_BIT.</p>
Database	Yes		The name of the database to use when the connection is established.
DbGroups	No	null	A comma-delimited list of the names of existing database groups the database user joins for the current session. The DbGroups option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
DbUser	No	null	The name of a database user. The DbUser option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
Driver	Yes		The location of the Amazon Redshift ODBC driver shared object file.
Fetch	Yes	100	The number of rows that the driver returns when UseDeclareFetch is enabled.
Iam	Yes		Specify 1 to use IAM database authentication. For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 231) .
KeepAlive	No. If keepalives are disabled at the TCP/IP level, KeepAliveTime and KeepAliveInterval are set to 0.	1	<p>When this option is enabled (1), the driver uses TCP keepalives to prevent connections from timing out.</p> <p>When this option is disabled (0), the driver does not use TCP keepalives.</p>
KeepAliveCount	No	0	<p>The number of TCP keepalive packets that can be lost before the connection is considered broken.</p> <p>When this option is set to 0, the driver uses the TCP/IP system default for this setting.</p>

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
KeepAliveTime	Yes	0	The number of seconds of inactivity before the driver sends a TCP keepalives packet. When this option is set to 0, the driver uses the TCP/IP system default for this setting.
KeepAliveInterval	Yes	0	The number of seconds between each TCP keepalive retransmission.
Locale	No	en-US	The locale to use for error messages.
MaxBytea	Yes	255	The maximum data length for BYTEA columns, in bytes.
MaxLongVarChar	Yes	8190	The maximum data length for LONG VARCHAR columns, in UTF-8 code units.
MaxVarchar	Yes	255	The maximum data length for VARCHAR columns, in UTF-8 code units.
Port	Yes		The port to connect to on the Amazon Redshift server. Note By default, Amazon Redshift uses port 5439.
Profile	No	null	The name of a profile in an AWS credentials or config file that contains values for the JDBC connection options. The Plugin_Name option is part of a set of options used to configure IAM database authentication. For more information, see Using a Configuration Profile (p. 230) .
PWD or Password	Yes		The password to use to connect to the Amazon Redshift server.
SecretAccessKey	No	null	The secret access key for the IAM role or IAM user configured for IAM database authentication. The SecretAccessKey option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229) . AccessKeyId and SecretAccessKey must be specified together.
Server or Servername	Yes		The IP address or hostname of the Amazon Redshift server.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
SessionToken	No	null	<p>The temporary session token for the IAM role configured for IAM database authentication. SessionToken is not required when IAM database authentication is configured for an IAM user. The SessionToken option is part of a set of options used to configure IAM database authentication. For more information, see JDBC and ODBC Options for Providing IAM Credentials (p. 229). If SessionToken is used, AccessKeyId and SecretAccessKey must also be specified.</p>
SingleRowMode	No	0	<p>When this option is enabled (1), the driver returns query results one row at a time. Enable this option if you plan to query large results and don't want to retrieve the entire result into memory.</p> <p>When this option and UseDeclareFetch are both disabled (0), the driver retrieves the entire query result into memory.</p> <p>Note If UseDeclareFetch is enabled (1), then it takes precedence over SingleRowMode. If SingleRowMode is enabled (1) and UseDeclareFetch is disabled (0), then SingleRowMode takes precedence over UseMultipleStatements.</p>
SSLCertPath	Yes	The default file name is <code>root.crt</code> and the default path is the location of the driver DLL file.	<p>The full path of the file containing the certificate authority bundle for verifying the server certificate. If this option is not set, then the driver looks in the folder that contains the driver DLL file.</p>
SSLMode	Yes	require	<p>The SSL certificate verification mode to use when connecting. For more information about possible SSL modes to use, see Using SSL and Trust CA Certificates in ODBC (p. 96).</p> <p>Important Amazon Redshift has changed the way that we manage SSL certificates. If you must use an ODBC driver version earlier than 1.3.7.1000, you might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see Transitioning to ACM Certificates for SSL Connections (p. 98).</p>

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
TextAsLongVarchar	Yes	0	<p>When this option is enabled (1), the driver returns TEXT columns as LONG VARCHAR data.</p> <p>When this option is disabled (0), the driver returns TEXT columns as TEXT data.</p>
UID	Yes		The user name to use to connect to the Amazon Redshift server.
UseDeclareFetch	Yes	0	<p>When this option is enabled (1), the driver returns a specific number of rows at a time. To set the number of rows, use the Fetch option.</p> <p>When this option is disabled (0) and SingleRowMode is enabled (1), the driver returns query results one row at a time. If SingleRowMode is also disabled (0), then the driver retrieves the entire query result into memory.</p> <p>Note If UseDeclareFetch is enabled, then UseDeclareFetch takes precedence over SingleRowMode and UseMultipleStatements.</p>
UseMultipleStatements	No	0	<p>When this option is enabled (1), the driver can run queries that are split into separate statements.</p> <p>When this option is disabled (0), the driver runs queries as single statements.</p> <p>Note If UseDeclareFetch is enabled (1), then UseDeclareFetch takes precedence over UseMultipleStatements. If UseDeclareFetch is disabled (0) but SingleRowMode is enabled (1), then SingleRowMode takes precedence over UseMultipleStatements.</p>
Username	Yes		The same information as UID (the user name to use to connect to the Amazon Redshift server). If UID is defined, then UID takes precedence over Username.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
UseUnicode	No	0	<p>When this option is enabled (1), the driver returns data as Unicode character types:</p> <ul style="list-style-type: none"> CHAR is returned as SQL_WCHAR. VARCHAR is returned as SQL_WVARCHAR. TEXT is returned as SQL_WLONGVARCHAR. <p>When this option is disabled (0), the driver returns data as regular SQL types:</p> <ul style="list-style-type: none"> CHAR is returned as SQL_CHAR. VARCHAR is returned as SQL_VARCHAR. TEXT is returned as SQL_LONGVARCHAR.

Previous ODBC Driver Versions

Download a previous version of the Amazon Redshift ODBC driver only if your tool requires a specific version of the driver. For information about the functionality supported in previous versions of the drivers, go to the [Amazon Redshift ODBC Driver Release Notes](#).

For authentication using AWS Identity and Access Management (IAM) credentials or identity provider (IdP) credentials, use Amazon Redshift ODBC driver version 1.3.6.1000 or later.

Important

Amazon Redshift has changed the way that we manage SSL certificates. If you must use a driver version earlier than 1.3.7.1000, you might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

Previous ODBC Driver Versions for Windows

The following are the previous 32-bit drivers:

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.1.1001/>
AmazonRedshiftODBC32-1.4.1.1001.msi
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.3.7.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.3.6.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.3.1.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.2.7.1007.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.2.6.1006.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32-1.2.1.1001.msi>

The following are the previous 64-bit drivers:

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.1.1001/>
AmazonRedshiftODBC64-1.4.1.1001.msi
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.3.7.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.3.6.1000.msi>

- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.3.1.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.2.7.1007.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.2.6.1006.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64-1.2.1.1001.msi>

Previous ODBC Driver Versions for Linux

The following are the previous versions of the 32-bit driver:

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.1.1001/AmazonRedshiftODBC-32-bit-1.4.1.1001-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.3.7.1000-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.1.1001/AmazonRedshiftODBC-32-bit-1.4.1.1001-1.i686.deb>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.3.7.1000-1.i686.deb>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.3.6.1000-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.3.1.1000-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.2.7.1007-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.2.6.1006-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.2.1.1001-1.i686.rpm>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.1.0.0000-1.i686.rpm>

The following are the previous versions of the 64-bit driver:

- 64-bit .rpm: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64-bit-1.4.1.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.3.7.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.1.1001/AmazonRedshiftODBC-64-bit-1.4.1.1001-1.x86_64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.3.7.1000-1.x86_64.deb
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.3.6.1000-1.i686.rpm>
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.3.1.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.2.7.1007-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.2.6.1006-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.2.1.1001-1.x86_64.rpm

- <https://s3.amazonaws.com/redshift-downloads/drivers/>
[AmazonRedshiftODBC-64bit-1.1.0.0000-1.x86_64.rpm](https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.1.0.0000-1.x86_64.rpm)

Previous ODBC Driver Versions for Mac

The following are the previous versions of the Amazon Redshift ODBC driver for Mac OS X:

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/AmazonRedshiftODBC-1.4.1.1001.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-1.3.7.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-1.3.6.1000-1.i686.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-1.3.1.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-1.2.7.1007.dmg>
- https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC_1.2.6.1006.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-1.2.1.1001.dmg>

Configure Security Options for Connections

Amazon Redshift supports Secure Sockets Layer (SSL) connections to encrypt data and server certificates to validate the server certificate that the client connects to.

Connect Using SSL

To support SSL connections, Amazon Redshift creates and installs an [AWS Certificate Manager \(ACM\)](#) issued SSL certificate on each cluster. The set of Certificate Authorities that you must trust in order to properly support SSL connections can be found at <https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt>. If the certificate bundle doesn't download, right-click the previous link and choose **Save link as....**

Important

Amazon Redshift has changed the way that we manage SSL certificates. You might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

By default, cluster databases accept a connection whether it uses SSL or not. To configure your cluster to require an SSL connection, set the `require_SSL` parameter to `true` in the parameter group that is associated with the cluster.

Amazon Redshift supports an SSL mode that is compliant with Federal Information Processing Standard (FIPS) 140-2. FIPS-compliant SSL mode is disabled by default.

Important

Enable FIPS-compliant SSL mode only if your system is required to be FIPS compliant.

To enable FIPS-compliant SSL mode, set both the `use_fips_ssl` parameter and the `require_SSL` parameter to `true` in the parameter group that is associated with the cluster. For information about modifying a parameter group, see [Amazon Redshift Parameter Groups \(p. 122\)](#).

Amazon Redshift supports the Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) key agreement protocol. With ECDHE, the client and server each have an elliptic curve public-private key pair that is used to establish a shared secret over an insecure channel. You do not need to configure anything in Amazon Redshift to enable ECDHE; if you connect from a SQL client tool that uses ECDHE to encrypt communication between the client and server, Amazon Redshift will use the provided cipher list to make the appropriate connection. For more information, see [Elliptic Curve Diffie—Hellman](#) on Wikipedia and [Ciphers](#) on the OpenSSL website.

Using SSL and Trust CA Certificates in ODBC

If you connect using the latest Amazon Redshift ODBC drivers (version 1.3.7.1000 or later), you can skip this section. To download the latest drivers, see [Configure an ODBC Connection \(p. 77\)](#).

You might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

The Amazon Redshift certificate authority bundle is stored at <https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt>. If the certificate bundle doesn't download, right-click the previous link and choose **Save link as....** The expected MD5 checksum number is e7a76d62fc7775ac54fc4d21e89d36b. The sha256 checksum is e77daa6243a940eb2d144d26757135195b4bdefd345c32a064d4ebea02b9f8a1. You can use the Md5sum program (on Linux operating systems) or other tool (on Windows and Mac OS X operating systems) to verify that the certificate that you downloaded matches this expected MD5 checksum number.

ODBC DSNs contain an `sslmode` setting that determines how to handle encryption for client connections and server certificate verification. Amazon Redshift supports the following `sslmode` values from the client connection:

- `disable`

SSL is disabled and the connection is not encrypted.

- `allow`

SSL is used if the server requires it.

- `prefer`

SSL is used if the server supports it. Amazon Redshift supports SSL, so SSL is used when you set `sslmode` to `prefer`.

- `require`

SSL is required.

- `verify-ca`

SSL must be used and the server certificate must be verified.

- `verify-full`

SSL must be used. The server certificate must be verified and the server hostname must match the hostname attribute on the certificate.

To determine whether SSL is used and server certificates are verified in a connection between the client and the server, you need to review the `sslmode` setting for your ODBC DSN on the client and the `require_SSL` setting for the Amazon Redshift cluster on the server. The following table describes the encryption result for the various client and server setting combinations:

<code>sslmode (client)</code>	<code>require_SSL (server)</code>	Result
<code>disable</code>	<code>false</code>	The connection is not encrypted.
<code>disable</code>	<code>true</code>	The connection cannot be made because the server requires SSL and the client has SSL disabled for the connection.
<code>allow</code>	<code>true</code>	The connection is encrypted.

sslmode (client)	require_SSL (server)	Result
allow	false	The connection is not encrypted.
prefer or require	true	The connection is encrypted.
prefer or require	false	The connection is encrypted.
verify-ca	true	The connection is encrypted and the server certificate is verified.
verify-ca	false	The connection is encrypted and the server certificate is verified.
verify-full	true	The connection is encrypted and the server certificate and hostname are verified.
verify-full	false	The connection is encrypted and the server certificate and hostname are verified.

Connect Using the Server Certificate with ODBC on Microsoft Windows

If you want to connect to your cluster using SSL and the server certificate, you need to download the certificate to your client computer or Amazon EC2 instance, and then configure the ODBC DSN.

1. Download the [Amazon Redshift Certificate Authority Bundle](#) to your client computer at the lib folder in your driver installation directory, and save the file as root.crt.
2. Open **ODBC Data Source Administrator**, and add or edit the system DSN entry for your ODBC connection. For **SSL Mode**, select verify-full unless you use a DNS alias. If you use a DNS alias, select verify-ca. Then click **Save**.

For more information about configuring the ODBC DSN, see [Configure an ODBC Connection \(p. 77\)](#).

Using SSL and Server Certificates in Java

SSL provides one layer of security by encrypting data that moves between your client and cluster. Using a server certificate provides an extra layer of security by validating that the cluster is an Amazon Redshift cluster. It does so by checking the server certificate that is automatically installed on all clusters that you provision. For more information about using server certificates with JDBC, go to [Configuring the Client](#) in the PostgreSQL documentation.

Connect Using Trust CA Certificates in Java

If you connect using the latest Amazon Redshift JDBC drivers (version 1.2.8.1005 or later), you can skip this section. To download the latest drivers, see [Configure a JDBC Connection](#).

Important

Amazon Redshift has changed the way that we manage SSL certificates. You might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

To connect using trust CA certificates

You can use redshift-keytool.jar to import CA certificates in the Redshift Certificate Authority bundle into a Java TrustStore or your private TrustStore.

1. If you use the Java command line `-Djavax.net.ssl.trustStore` option, remove it from command line, if possible.
2. Download the [redshift-keytool.jar](#)
3. Do one of the following:
 - To import Redshift Certificate Authority bundle into a Java TrustStore, run the following command.

```
java -jar redshift-keytool.jar -s
```

- To import Redshift Certificate Authority bundle into your private TrustStore, run the following command:

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -p <keystore_password>
```

Transitioning to ACM Certificates for SSL Connections

Amazon Redshift is replacing the SSL certificates on your clusters with [AWS Certificate Manager \(ACM\)](#) issued certificates. ACM is a trusted public certificate authority (CA) that is trusted by most current systems. You might need to update your current trust root CA certificates to continue to connect to your clusters using SSL.

This change affects you only if all of the following apply:

- Your SQL clients or applications connect to Amazon Redshift clusters using SSL with the `sslMode` connection option set to `require`, `verify-ca`, or `verify-full` configuration option.
- Your clusters are in any AWS region except the AWS GovCloud (US) region, the China (Beijing) region, or the China (Ningxia) region.
- You aren't using the Amazon Redshift ODBC or JDBC drivers, or you use Amazon Redshift drivers prior to ODBC version 1.3.7.1000 or JDBC version 1.2.8.1005.

If this change affects you, then you must update your current trust root CA certificates before October 23, 2017. Amazon Redshift will transition your clusters to use ACM certificates between now and October 23, 2017. The change should have very little or no effect on your cluster's performance or availability.

To update your current trust root CA certificates, identify your use case and then follow the steps in that section.

- [Using the Latest Amazon Redshift ODBC or JDBC drivers \(p. 98\)](#)
- [Using Earlier Amazon Redshift ODBC or JDBC drivers \(p. 99\)](#)
- [Using Other SSL Connection Types \(p. 99\)](#)

Using the Latest Amazon Redshift ODBC or JDBC drivers

The preferred method is to use the latest Amazon Redshift ODBC or JDBC drivers. Amazon Redshift drivers beginning with ODBC version 1.3.7.1000 and JDBC version 1.2.8.1005 automatically manage the transition from an Amazon Redshift self-signed certificate to an ACM certificate. To download the latest drivers, see [Configure an ODBC Connection \(p. 77\)](#) or [Configure a JDBC Connection \(p. 62\)](#).

If you use the latest Amazon Redshift JDBC driver, it's best not to use `-Djavax.net.ssl.trustStore` in JVM options. If you must use `-Djavax.net.ssl.trustStore`, import the [Redshift Certificate](#)

Authority Bundle into the truststore it points to. For more information, see [Importing the Redshift Certificate Authority Bundle into a TrustStore \(p. 99\)](#).

Using Earlier Amazon Redshift ODBC or JDBC drivers

If you must use an Amazon Redshift ODBC driver prior to version 1.3.7.1000, then download the [Redshift Certificate Authority Bundle](#) and overwrite the old certificate file.

- If your ODBC DSN is configured with `SSLCertPath`, overwrite the certificate file in the specified path.
- If `SSLCertPath` is not set, then overwrite the certificate file named `root.crt` in the driver DLL location.

If you must use an Amazon Redshift JDBC driver prior to version 1.2.8.1005, then do one of the following:

- If your JDBC connection string uses the `sslCert` option, remove the `sslCert` option. Then import the [Redshift Certificate Authority Bundle](#) to your Java TrustStore. For more information, see [Importing the Redshift Certificate Authority Bundle into a TrustStore \(p. 99\)](#)
- If you use the Java command line `-Djavax.net.ssl.trustStore` option, remove it from command line, if possible. Then import the [Redshift Certificate Authority Bundle](#) to your Java TrustStore. For more information, see [Importing the Redshift Certificate Authority Bundle into a TrustStore \(p. 99\)](#)

Importing the Redshift Certificate Authority Bundle into a TrustStore

You can use the `redshift-keytool.jar` to import CA certificates in the Redshift Certificate Authority bundle into a Java TrustStore or your private truststore.

To import the Redshift Certificate Authority Bundle into a TrustStore:

1. Download the [redshift-keytool.jar](#)
2. Do one of the following:
 - To import Redshift Certificate Authority bundle into a Java TrustStore, run the following command.

```
java -jar redshift-keytool.jar -s
```

2. Do one of the following:
 - To import Redshift Certificate Authority bundle into your private TrustStore, run the following command:

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -p <keystore_password>
```

Using Other SSL Connection Types

Follow the steps in this section if you connect using any of the following:

- Open source ODBC driver
- Open source JDBC driver
- The `psql` command line interface
- Any language bindings based on `libpq`, such as `psycopg2` (Python) and `ruby-pg` (Ruby)

To use ACM certificates with other SSL connection types:

1. Download the [Redshift Certificate Authority Bundle](#).

2. Place the certificates from the bundle in your `root.crt` file.
 - On Linux and Mac OS X operating systems, the file is `~/.postgresql/root.crt`
 - On Microsoft Windows, the file is `%APPDATA%\postgresql\root.crt`

Connecting to Clusters from Client Tools and Code

This section provides some options for third-party tools to connect to the cluster if you do not already have a business intelligence tool to do so. Additionally, it describes how to connect to your cluster programmatically.

Topics

- [Connect to Your Cluster by Using SQL Workbench/J \(p. 100\)](#)
- [Connect to Your Cluster by Using the psql Tool \(p. 103\)](#)
- [Connect to Your Cluster Programmatically \(p. 106\)](#)

Connect to Your Cluster by Using SQL Workbench/J

Amazon Redshift does not provide or install any SQL client tools or libraries, so you must install any that you want to use with your clusters. If you already have a business intelligence application or any other application that can connect to your clusters using a standard PostgreSQL JDBC or ODBC driver, then you can skip this section. If you don't already have an application that can connect to your cluster, this section presents one option for doing so using SQL Workbench/J, a free, DBMS-independent, cross-platform SQL query tool.

Install SQL Workbench/J

The *Amazon Redshift Getting Started* uses SQL Workbench/J. In this section, we explain in detail how to connect to your cluster by using SQL Workbench/J.

To install SQL Workbench/J

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 on your client computer or Amazon EC2 instance.
3. Go to the [Installing and starting SQL Workbench/J](#) page. Follow the instructions for installing SQL Workbench/J on your system.

Note

SQL Workbench/J requires the Java Runtime Environment (JRE) be installed on your system. Ensure you are using the correct version of the JRE required by the SQL Workbench/J client. To determine which version of the Java Runtime Environment is running on your system, do one of the following:

- Mac: In the **System Preferences**, click the Java icon.
- Windows: In the **Control Panel**, click the Java icon.
- Any system: In a command shell, type `java -version`. You can also visit <https://www.java.com>, click the **Do I Have Java?** link, and click on the **Verify Java** button.

For information about installing and configuring the Java Runtime Environment, go to <https://www.java.com>.

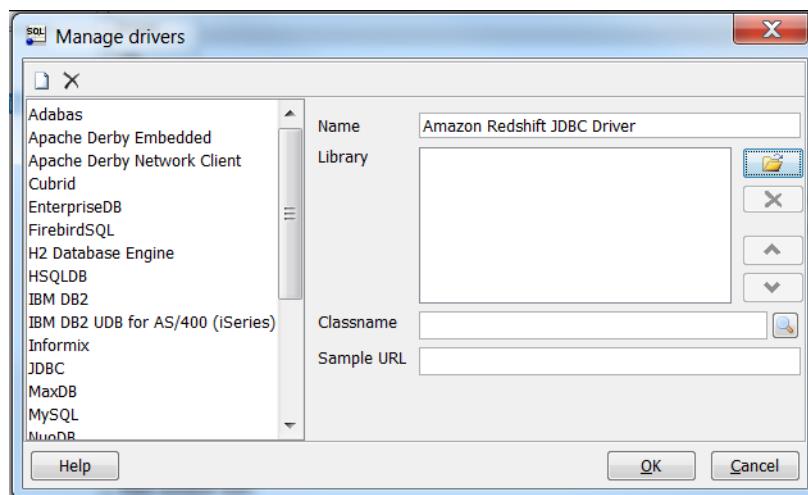
Connect to Your Cluster over a JDBC Connection in SQL Workbench/J

Important

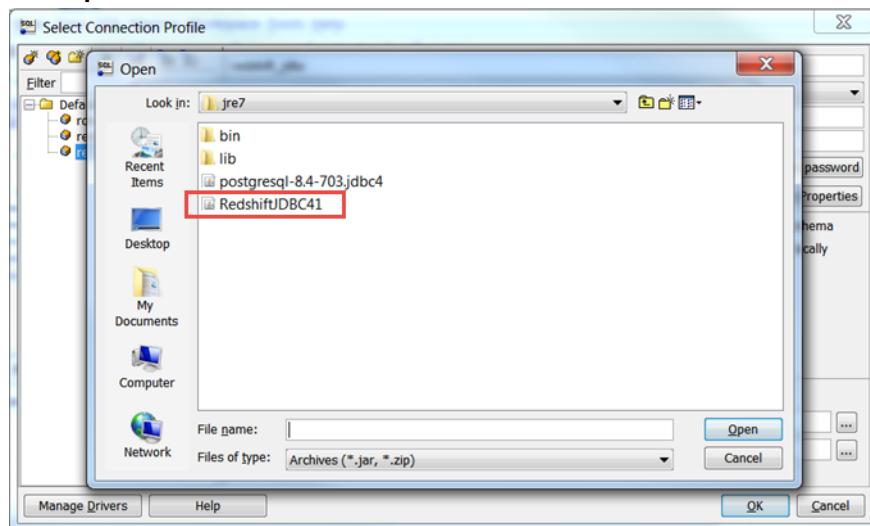
Before you perform the steps in this procedure, make sure that your client computer or Amazon EC2 instance has the recommended Amazon Redshift JDBC driver. For links to download the latest drivers, see [Download the Amazon Redshift JDBC Driver \(p. 62\)](#). Also, make sure you have configured firewall settings to allow access to your cluster. For more information, see [Step 4: Authorize Access to the Cluster](#).

To use a JDBC connection in SQL Workbench/J

1. Open SQL Workbench/J.
2. Click **File**, and then click **Connect window**.
3. Click **Create a new connection profile**.
4. In the **New profile** box, type a name for the profile. For example, examplecluster_jdbc.
5. Click **Manage Drivers**. The **Manage Drivers** dialog opens. In the **Name** box, type a name for the driver.



Click the folder icon next to the **Library** box, navigate to the location of the driver, click it, and then click **Open**.



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

6. In the **Driver** box, select the driver you just added.
7. In **URL**, copy the JDBC URL from the Amazon Redshift console and paste it here.

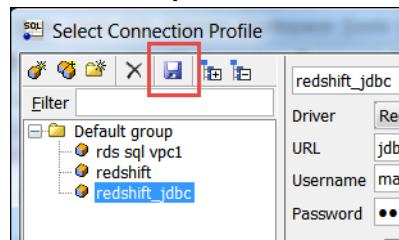
For more information about finding the JDBC URL, see [Configure a JDBC Connection \(p. 62\)](#).

8. In **Username**, type the name of the master user.

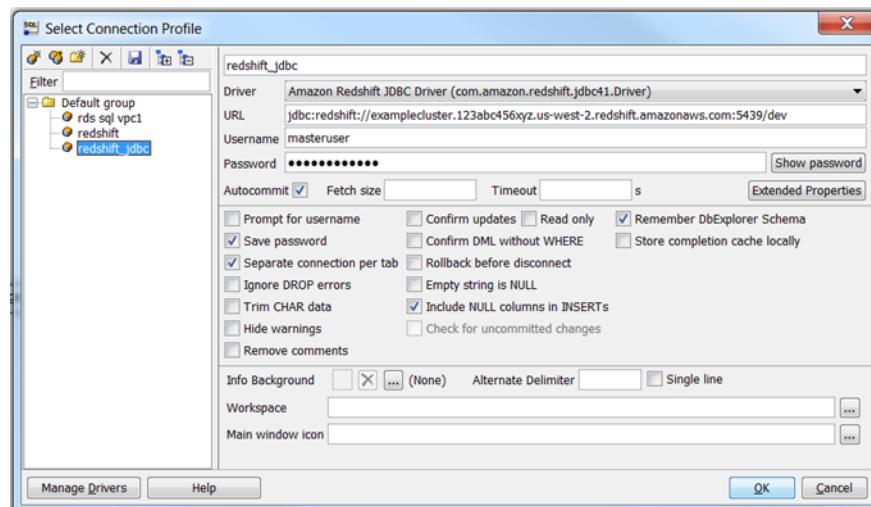
If you are following the *Amazon Redshift Getting Started*, type **masteruser**.

9. In **Password**, type the password associated with the master user account.
10. Select the **Autocommit** box.

11. Click the **Save profile list** icon, as shown below:



12. Click **OK**.



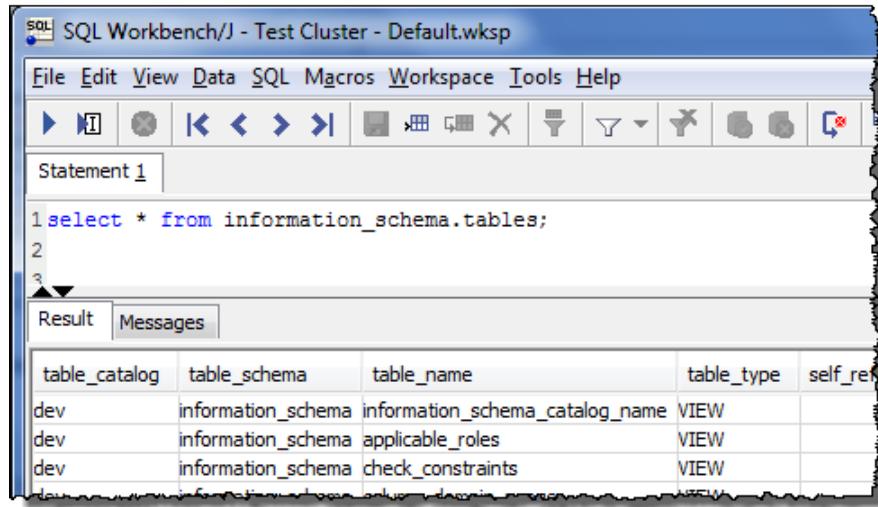
Test the SQL Workbench/J Connection

After you configure your JDBC or ODBC connection, you can test the connection by running an example query.

1. You can use the following query to test your connection.

```
select * from information_schema.tables;
```

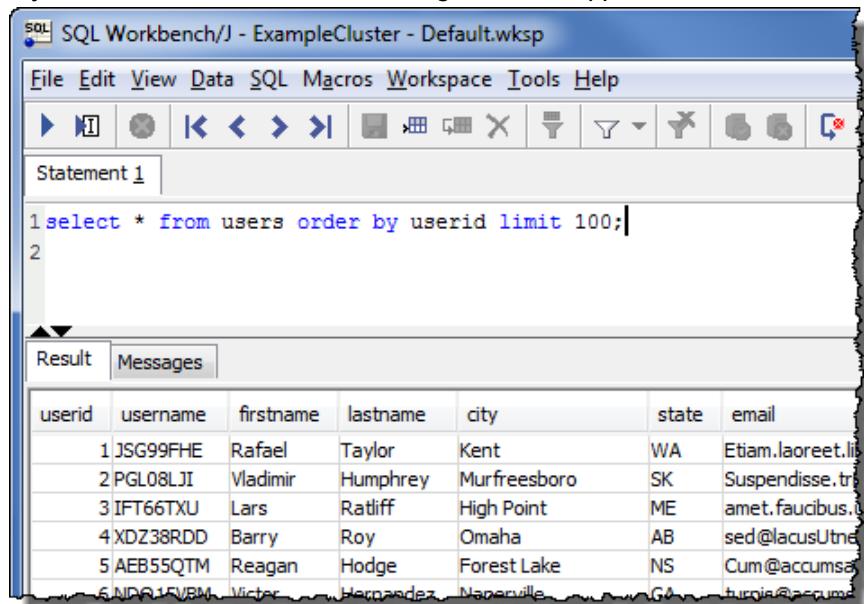
If your connection is successful, a listing of records appears in the **Results** tab.



2. Alternatively, if you loaded the sample tables and data from the [Amazon Redshift Getting Started](#), you can test your connection by typing the following query into the **Statement** window:

```
select * from users order by userid limit 100;
```

If your connection is successful, a listing of records appears in the **Results** tab.



Connect to Your Cluster by Using the psql Tool

After you create an Amazon Redshift cluster, you can use psql, a terminal-based front end from PostgreSQL, to query the data in your cluster. You can type the queries interactively or read them from a file. To connect from psql, you must specify the cluster endpoint, database, and port.

Note

Amazon Redshift does not provide the psql tool; it is installed with PostgreSQL. For information about using psql, go to <https://www.postgresql.org/docs/8.4/static/app-psql.html>. For

information about installing the PostgreSQL client tools, select your operating system from the PostgreSQL binary downloads page at <https://www.postgresql.org/download/>.

Connect by Using the psql Defaults

By default, psql does not validate the Amazon Redshift service; it makes an encrypted connection by using Secure Sockets Layer (SSL).

To connect by using psql defaults

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the left navigation pane, click **Clusters**. Click your cluster to open it. Under **Cluster Database Properties**, record the values of **Endpoint**, **Port**, and **Database Name**.

Cluster Database Properties	
Port	5439
Publicly Accessible	Yes
Elastic IP	[REDACTED]
Database Name	dev
Master Username	masteruser
Encrypted	Yes
KMS Key ID	[REDACTED]
JDBC URL	jdbc:redshift://examplecluster.[REDACTED].us-west-2.redshift.amazonaws.com:5439/dev
ODBC URL	Driver=(Amazon Redshift (x64)); Server=examplecluster.[REDACTED].us-west-2.redshift.amazonaws.com; Database=dev; UID=masteruser; PWD=insert_your_master_user_pass_word_here; Port=5439

3. At a command prompt, specify the connection information by using either command line parameters or a connection information string. To use parameters:

```
psql -h <endpoint> -U <userid> -d <dbname> -p <port>
```

Where:

- <endpoint> is the **Endpoint** you recorded in the previous step.
- <userid> is a user ID with permissions to connect to the cluster.
- <dbname> is the **Database Name** you recorded in the previous step.
- <port> is the **Port** you recorded in the previous step.

For example:

```
psql -h examplecluster.<XXXXXXXXXXXXXX>.us-west-2.redshift.amazonaws.com -U masteruser -d dev -p 5439
```

4. At the psql password prompt, enter the password for the <userid> user.

You are connected to the cluster, and you can interactively enter commands.

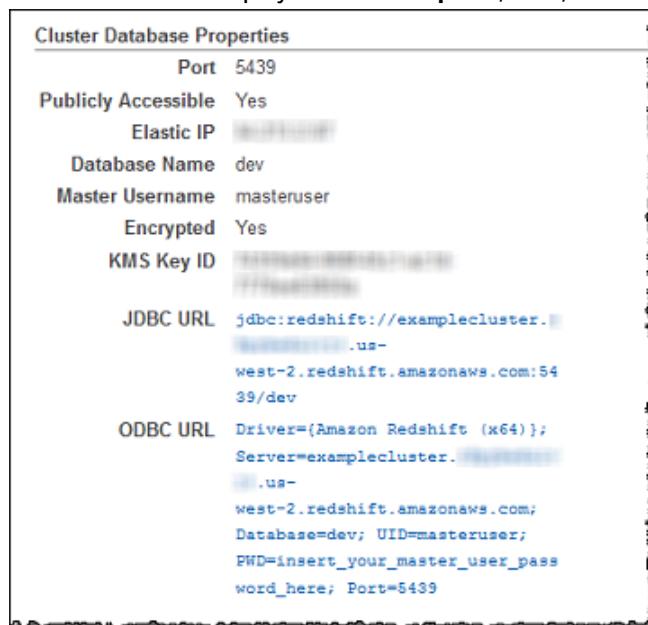
Connect by Using a Certificate

To control whether psql authenticates the service using a certificate, you must use a connection information string to specify connection information, and specify the `sslmode` keyword. By default, psql operates with `sslmode=prefer`. To specify that psql opens an encrypted connection and uses an Amazon Redshift certificate to verify the service, download an Amazon Redshift certificate to your computer. Specify `verify-full` unless you use a DNS alias. If you use a DNS alias, select `verify-ca`. Specify `sslrootcert` with the location of the certificate. For more information about `sslmode`, see [Configure Security Options for Connections \(p. 95\)](#).

For more information about connection information string parameters, see <https://www.postgresql.org/docs/8.4/static/libpq-connect.html>.

To connect by using a certificate

1. Save the download the [Redshift Certificate Authority Bundle](#) a .crt file to your computer. If you do a **File\Save as** using Internet Explorer, specify the file type as **Text file (*.txt)** and delete the .txt extension. For example, save it as the file C:\MyDownloads\redshift-ca-bundle.crt.
2. In the Amazon Redshift console, select the cluster to display the **Cluster Database Properties**. Record the values displayed in the **Endpoint**, **Port**, and **Database Name** fields.



3. At a command prompt, specify the connection information using a connection information string:

```
psql "host=<endpoint> user=<userid> dbname=<dbname> port=<port> sslmode=verify-ca  
sslrootcert=<certificate>"
```

Where:

- `<endpoint>` is the **Endpoint** you recorded in the previous step.
- `<userid>` is a user ID with permissions to connect to the cluster.
- `<dbname>` is the **Database Name** you recorded in the previous step.
- `<port>` is the **Port** you recorded in the previous step.

- `<certificate>` is the full path to the certificate file. On Windows systems, the certificate path must be specified using Linux-style / separators instead of the Windows \ separator.

On Linux and Mac OS X operating systems, the path is

```
~/.postgresql/root.crt
```

On Microsoft Windows, the path is

```
%APPDATA%/postgresql/root.crt
```

For example:

```
psql "host=examplecluster.<XXXXXXXXXXXXXX>.us-west-2.redshift.amazonaws.com
      user=masteruser dbname=dev port=5439 sslmode=verify-ca sslrootcert=C:/MyDownloads/
      redshift-ca-bundle.crt"
```

4. At the psql password prompt, enter the password for the `<userid>` user.

You are connected to the cluster, and you can interactively enter commands.

Connect to Your Cluster Programmatically

This section explains how to connect to your cluster programmatically. If you are using an application like SQL Workbench/J that manages your client connections for you, then you can skip this section.

Connecting to a Cluster by Using Java

When you use Java to programmatically connect to your cluster, you can do so with or without server authentication. If you plan to use server authentication, follow the instructions in [Configure Security Options for Connections \(p. 95\)](#) to put the Amazon Redshift server certificate into a keystore. You can refer to the keystore by specifying a property when you run your code as follows:

```
-Djavax.net.ssl.trustStore=<path to keystore>
-Djavax.net.ssl.trustStorePassword=<keystore password>
```

Example : Connect to a Cluster by Using Java

The following example connects to a cluster and runs a sample query that returns system tables. It is not necessary to have data in your database to use this example.

If you are using a server certificate to authenticate your cluster, you can restore the line that uses the keystore, which is commented out:

```
props.setProperty("ssl", "true");
```

For more information about the server certificate, see [Configure Security Options for Connections \(p. 95\)](#).

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#).

```

package connection;

import java.sql.*;
import java.util.Properties;

public class Docs {
    //Redshift driver: "jdbc:redshift://x.y.us-west-2.redshift.amazonaws.com:5439/dev";
    //or "jdbc:postgresql://x.y.us-west-2.redshift.amazonaws.com:5439/dev";
    static final String dbURL = "***jdbc cluster connection string ***";
    static final String MasterUsername = "***master user name***";
    static final String MasterUserPassword = "***master user password***";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            //Dynamically load driver at runtime.
            //Redshift JDBC 4.1 driver: com.amazon.redshift.jdbc41.Driver
            //Redshift JDBC 4 driver: com.amazon.redshift.jdbc4.Driver
            Class.forName("com.amazon.redshift.jdbc.Driver");

            //Open a connection and define properties.
            System.out.println("Connecting to database...");
            Properties props = new Properties();

            //Uncomment the following line if using a keystore.
            //props.setProperty("ssl", "true");
            props.setProperty("user", MasterUsername);
            props.setProperty("password", MasterUserPassword);
            conn = DriverManager.getConnection(dbURL, props);

            //Try a simple query.
            System.out.println("Listing system tables...");
            stmt = conn.createStatement();
            String sql;
            sql = "select * from information_schema.tables;";
            ResultSet rs = stmt.executeQuery(sql);

            //Get the data from the result set.
            while(rs.next()){
                //Retrieve two columns.
                String catalog = rs.getString("table_catalog");
                String name = rs.getString("table_name");

                //Display values.
                System.out.print("Catalog: " + catalog);
                System.out.println(", Name: " + name);
            }
            rs.close();
            stmt.close();
            conn.close();
        }catch(Exception ex){
            //For convenience, handle all errors here.
            ex.printStackTrace();
        }finally{
            //Finally block to close resources.
            try{
                if(stmt!=null)
                    stmt.close();
            }catch(Exception ex){
                // nothing we can do
            }try{
                if(conn!=null)
                    conn.close();
            }catch(Exception ex){
                ex.printStackTrace();
            }
        }
    }
}

```

```
        }
    }
    System.out.println("Finished connectivity test.");
}
```

Connecting to a Cluster by Using .NET

When you use .NET (C#) to programmatically connect to your cluster, you can do so with or without server authentication. If you plan to use server authentication, follow the instructions in [Configure Security Options for Connections \(p. 95\)](#) to download the Amazon Redshift server certificate, and then put the certificate in the correct form for your .NET code.

Example Connect to a Cluster by Using .NET

The following example connects to a cluster and runs a sample query that returns system tables. It does not show server authentication. It is not necessary to have data in your database to use this example. This example uses the [System.Data.Odbc Namespace](#), a .NET Framework Data Provider for ODBC.

```
using System;
using System.Data;
using System.Data.Odbc;

namespace redshift.amazon.com.docsamples
{
    class ConnectToClusterExample
    {
        public static void Main(string[] args)
        {

            DataSet ds = new DataSet();
            DataTable dt = new DataTable();

            // Server, e.g. "examplecluster.xyz.us-west-2.redshift.amazonaws.com"
            string server = "***provide server name part of connection string***";

            // Port, e.g. "5439"
            string port = "***provide port***";

            // MasterUserName, e.g. "masteruser".
            string masterUsername = "***provide master user name***";

            // MasterUserPassword, e.g. "mypassword".
            string masterUserPassword = "***provide master user password***";

            // DBName, e.g. "dev"
            string DBName = "***provide name of database***";

            string query = "select * from information_schema.tables;";

            try
            {
                // Create the ODBC connection string.
                //Redshift ODBC Driver - 64 bits
                /*
                string connString = "Driver={Amazon Redshift (x64)};" +
                    String.Format("Server={0};Database={1};"
                    +"UID={2};PWD={3};Port={4};SSL=true;Sslmode=Require",
                    server, DBName, masterUsername,
                    masterUserPassword, port);
                */
                //Redshift ODBC Driver - 32 bits
            }
        }
    }
}
```

```
string connString = "Driver={Amazon Redshift (x86)};" +
    String.Format("Server={0};Database={1};"
    "UID={2};PWD={3};Port={4};SSL=true;Sslmode=Require",
    server, DBName, masterUsername,
    masterUserPassword, port);

// Make a connection using the psqlODBC provider.
OdbcConnection conn = new OdbcConnection(connString);
conn.Open();

// Try a simple query.
string sql = query;
OdbcDataAdapter da = new OdbcDataAdapter(sql, conn);
da.Fill(ds);
dt = ds.Tables[0];
foreach (DataRow row in dt.Rows)
{
    Console.WriteLine(row["table_catalog"] + ", " + row["table_name"]);
}

conn.Close();
Console.ReadKey();
}
catch (Exception ex)
{
    Console.Error.WriteLine(ex.Message);
    Console.ReadKey();
}
}

}
}
```

Troubleshooting Connection Issues in Amazon Redshift

If you have issues with connecting to your cluster from a SQL client tool, there are several things that you can check to narrow down the problem. If you are using SSL or server certificates, first remove this complexity while you troubleshoot the connection issue. Then add this back when you have found a solution. For more information, see [Configure Security Options for Connections \(p. 95\)](#).

Important

Amazon Redshift has changed the way that we manage SSL certificates. If you have trouble connecting using SSL, you might need to update your current trust root CA certificates. For more information, see [Transitioning to ACM Certificates for SSL Connections \(p. 98\)](#).

The following section has some example error messages and possible solutions for connection issues. Because different SQL client tools provide different error messages, this is not a complete list, but should be a good starting point for troubleshooting issues.

Topics

- [Connecting from Outside of Amazon EC2 —Firewall Timeout Issue \(p. 110\)](#)
- [The Connection Is Refused or Fails \(p. 111\)](#)
- [The Client and Driver Are Incompatible \(p. 112\)](#)
- [Queries Appear to Hang and Sometimes Fail to Reach the Cluster \(p. 112\)](#)

Connecting from Outside of Amazon EC2 —Firewall Timeout Issue

Example Issue

Your client connection to the database appears to hang or timeout when running long queries, such as a COPY command. In this case, you might observe that the Amazon Redshift console displays that the query has completed, but the client tool itself still appears to be running the query. The results of the query might be missing or incomplete depending on when the connection stopped.

Possible Solutions

This happens when you connect to Amazon Redshift from a computer other than an Amazon EC2 instance, and idle connections are terminated by an intermediate network component, such as a firewall, after a period of inactivity. This behavior is typical when you log on from a Virtual Private Network (VPN) or your local network.

To avoid these timeouts, we recommend the following changes:

- Increase client system values that deal with TCP/IP timeouts. You should make these changes on the computer you are using to connect to your cluster. The timeout period should be adjusted for your client and network. See [Change TCP/IP Timeout Settings \(p. 110\)](#).
- Optionally, set keep-alive behavior at the DSN level. See [Change DSN Timeout Settings \(p. 111\)](#).

Change TCP/IP Timeout Settings

To change TCP/IP timeout settings, configure the timeout settings according to the operating system that you use to connect to your cluster.

- Linux — If your client is running on Linux, run the following command as the root user to change the timeout settings for the current session:

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=200 net.ipv4.tcp_keepalive_intvl=200  
net.ipv4.tcp_keepalive_probes=5
```

To persist the settings, create or modify the file `/etc/sysctl.conf` with the following values then reboot your system.

```
net.ipv4.tcp_keepalive_time=200  
net.ipv4.tcp_keepalive_intvl=200  
net.ipv4.tcp_keepalive_probes=5
```

- Windows — If your client runs on Windows, edit the values for the following registry settings under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\`:
 - KeepAliveTime: 30000
 - KeepAliveInterval: 1000
 - TcpMaxDataRetransmissions: 10

These settings use the DWORD data type. If they do not exist under the registry path, you can create the settings and specify these recommended values. For more information about editing the Windows registry, refer to Windows documentation.

After you set these values, restart your computer for the changes to take effect.

- Mac — If your client is running on a Mac, run the following commands to change the timeout settings for the current session:

```
sudo sysctl net.inet.tcp.keepintvl=200000
sudo sysctl net.inet.tcp.keepidle=200000
sudo sysctl net.inet.tcp.keepinit=200000
sudo sysctl net.inet.tcp.always_keepalive=1
```

To persist the settings, create or modify the file `/etc/sysctl.conf` with the following values:

```
net.inet.tcp.keepidle=200000
net.inet.tcp.keepintvl=200000
net.inet.tcp.keepinit=200000
net.inet.tcp.always_keepalive=1
```

Restart your computer, and then run the following commands to verify that the values are set.

```
sysctl net.inet.tcp.keepidle
sysctl net.inet.tcp.keepintvl
sysctl net.inet.tcp.keepinit
sysctl net.inet.tcp.always_keepalive
```

Change DSN Timeout Settings

You can set keep-alive behavior at the DSN level if you choose. You do this by adding or modifying the following parameters in the `odbc.ini` file:

KeepAlivesCount

The number of TCP keep-alive packets that can be lost before the connection is considered broken.

KeepAlivesIdle

The number of seconds of inactivity before the driver sends a TCP keep-alive packet.

KeepAlivesInterval

The number of seconds between each TCP keep-alive retransmission.

On Windows, you modify these parameters in the registry by adding or changing keys in `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\your_DSN`. On Linux and Mac OS, you add or modify these parameters in the target DSN entry directly in the `odbc.ini` file. For more information on modifying the `odbc.ini` file on Linux and Mac OS computers, see [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#).

If these parameters don't exist, or if they have a value of 0, the system will use the keep-alive parameters specified for TCP/IP to determine DSN keep-alive behavior. On Windows, the TCP/IP parameters can be found in the registry in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`. On Linux and Mac OS, the TCP/IP parameters can be found in the `sysctl.conf` file.

The Connection Is Refused or Fails

Example errors:

- "Failed to establish a connection to `<endpoint>`."
- "Could not connect to server: Connection timed out. Is the server running on host '`<endpoint>`' and accepting TCP/IP connections on port '`<port>`'?"

- "Connection refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections."

Possible solutions:

Generally, when you receive an error message indicating that there is a failure to establish a connection, it is an issue with permission to access the cluster.

If you attempt to connect to the cluster from a client tool outside of the network that the cluster is in, you must add an ingress rule to the cluster security group for the CIDR/IP that you are connecting from:

- If you created your Amazon Redshift cluster in an Amazon Virtual Private Cloud (Amazon VPC), you need to add your client CIDR/IP address to the VPC security group in Amazon VPC. For more information about configuring VPC security groups for your cluster, see [Managing Clusters in an Amazon Virtual Private Cloud \(VPC\) \(p. 36\)](#).
- If you created your Amazon Redshift cluster outside a VPC, you need to add your client CIDR/IP address to the cluster security group in Amazon Redshift. For more information about configuring cluster security groups, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

If you attempt to connect to the cluster from a client tool in an Amazon EC2 instance, you must add an ingress rule to the cluster security group for the Amazon EC2 security group that is associated with the Amazon EC2 instance. For more information about configuring cluster security groups, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).

Additionally, if you have a layer between your client and server, such as a firewall, make sure that the firewall accepts inbound connections over the port that you configured for your cluster.

The Client and Driver Are Incompatible

Example error:

"The specified DSN contains an architecture mismatch between the Driver and Application."

Possible solution:

When you get attempt to connect and get an error about an architecture mismatch, this means that the client tool and the driver are not compatible because their system architecture does not match. For example, this can happen if you have a 32-bit client tool but have installed the 64-bit version of the driver. Sometimes 64-bit client tools can use 32-bit drivers, but you cannot use 32-bit applications with 64-bit drivers. Make sure that the driver and client tool are using the same version of the system architecture.

Queries Appear to Hang and Sometimes Fail to Reach the Cluster

Example issue:

You experience an issue with queries completing, where the queries appear to be running but hang in the SQL client tool. Sometimes the queries fail to appear in the cluster, such as in system tables or the Amazon Redshift console.

Possible solution:

This issue can happen due to packet drop, when there is a difference in the maximum transmission unit (MTU) size in the network path between two Internet Protocol (IP) hosts. The MTU size determines the maximum size, in bytes, of a packet that can be transferred in one Ethernet frame over a network

connection. In AWS, some Amazon EC2 instance types support an MTU of 1500 (Ethernet v2 frames) and other instance types support an MTU of 9001 (TCP/IP jumbo frames).

To avoid issues that can occur with differences in MTU size, we recommend doing one of the following:

- If your cluster uses the EC2-VPC platform, configure the Amazon VPC security group with an inbound custom Internet Control Message Protocol (ICMP) rule that returns `Destination Unreachable`, thus instructing the originating host to use the lowest MTU size along the network path. For details on this approach, see [Configuring Security Groups to Allow ICMP "Destination Unreachable" \(p. 113\)](#).
- If your cluster uses the EC2-Classic platform, or you cannot allow the ICMP inbound rule, disable TCP/IP jumbo frames so that Ethernet v2 frames are used. For details on this approach, see [Configuring the MTU of an Instance \(p. 113\)](#).

Configuring Security Groups to Allow ICMP "Destination Unreachable"

When there is a difference in the MTU size in the network between two hosts, first make sure that your network settings don't block path MTU discovery (PMTUD). PMTUD enables the receiving host to respond to the originating host with the following ICMP message: `Destination Unreachable: fragmentation needed and DF set (ICMP Type 3, Code 4)`. This message instructs the originating host to use the lowest MTU size along the network path to resend the request. Without this negotiation, packet drop can occur because the request is too large for the receiving host to accept. For more information about this ICMP message, go to [RFC792](#) on the *Internet Engineering Task Force (IETF)* website.

If you don't explicitly configure this ICMP inbound rule for your Amazon VPC security group, PMTUD is blocked. In AWS, security groups are virtual firewalls that specify rules for inbound and outbound traffic to an instance. For clusters using the EC2-VPC platform, Amazon Redshift uses VPC security groups to allow or deny traffic to the cluster. By default, the security groups are locked down and deny all inbound traffic.

For more information about how to add rules to VPC security groups, see [Managing VPC Security Groups for a Cluster \(p. 39\)](#). For more information about specific PMTUD settings required in this rule, see [Path MTU Discovery](#) in the *Amazon EC2 User Guide for Linux Instances*.

Configuring the MTU of an Instance

If your cluster uses the EC2-Classic platform or you cannot allow the custom ICMP rule for inbound traffic in your case, then we recommend that you adjust the MTU to 1500 on the network interface (NIC) of the Amazon EC2 instances from which you connect to your Amazon Redshift cluster. This adjustment disables TCP/IP jumbo frames to ensure that connections consistently use the same packet size. However, note that this option reduces your maximum network throughput for the instance entirely, not just for connections to Amazon Redshift. For more information, see the following procedures.

To set MTU on a Microsoft Windows operating system

If your client runs in a Microsoft Windows operating system, you can review and set the MTU value for the Ethernet adapter by using the `netsh` command.

1. Run the following command to determine the current MTU value:

```
netsh interface ipv4 show subinterfaces
```

2. Review the MTU value for the Ethernet adapter in the output.
3. If the value is not 1500, run the following command to set it:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
```

After you set this value, restart your computer for the changes to take effect.

To set MTU on a Linux operating system

If your client runs in a Linux operating system, you can review and set the MTU value by using the `ip` command.

1. Run the following command to determine the current MTU value:

```
$ ip link show eth0
```

2. Review the value following `mtu` in the output.
3. If the value is not 1500, run the following command to set it:

```
$ sudo ip link set dev eth0 mtu 1500
```

To set MTU on a Mac operating system

- To set the MTU on a Mac operating system, follow the instructions in [Mac OS X 10.4 or later: How to change the MTU for troubleshooting purposes](#).

Amazon Redshift Enhanced VPC Routing

When you use Amazon Redshift Enhanced VPC Routing, Amazon Redshift forces all [COPY](#) and [UNLOAD](#) traffic between your cluster and your data repositories through your Amazon VPC. By using Enhanced VPC Routing, you can use standard VPC features, such as [VPC security groups](#), [network access control lists \(ACLs\)](#), [VPC endpoints](#), [VPC endpoint policies](#), [internet gateways](#), and [Domain Name System \(DNS\)](#) servers, as described in the *Amazon VPC User Guide*. You use these features to tightly manage the flow of data between your Amazon Redshift cluster and other resources. When you use Enhanced VPC Routing to route traffic through your VPC, you can also use [VPC flow logs](#) to monitor COPY and UNLOAD traffic.

If Enhanced VPC Routing is not enabled, Amazon Redshift routes traffic through the internet, including traffic to other services within the AWS network.

Important

Because Enhanced VPC Routing affects the way that Amazon Redshift accesses other resources, COPY and UNLOAD commands might fail unless you configure your VPC correctly. You must specifically create a network path between your cluster's VPC and your data resources, as described following.

When you execute a COPY or UNLOAD command on a cluster with Enhanced VPC Routing enabled, your VPC routes the traffic to the specified resource using the *strictest*, or most specific, network path available.

For example, you can configure the following pathways in your VPC:

- **VPC endpoints** – For traffic to an Amazon S3 bucket in the same AWS Region as your cluster, you can create a VPC endpoint to direct traffic directly to the bucket. When you use VPC endpoints, you can attach an endpoint policy to manage access to Amazon S3. For more information about using endpoints with Amazon Redshift, see [Working with VPC Endpoints \(p. 116\)](#).
- **NAT gateway** – You can connect to an Amazon S3 bucket in another AWS Region, and you can connect to another service within the AWS network. You can also access a host instance outside the AWS network. To do so, configure a [network address translation \(NAT\) gateway](#), as described in the *Amazon VPC User Guide*.
- **Internet gateway** – To connect to AWS services outside your VPC, you can attach an [internet gateway](#) to your VPC subnet, as described in the *Amazon VPC User Guide*. To use an internet gateway, your cluster must have a public IP to allow other services to communicate with your cluster.

For more information, see [VPC Endpoints](#) in the *Amazon VPC User Guide*.

There is no additional charge for using Enhanced VPC Routing. You might incur additional data transfer charges for certain operations. These include such operations as UNLOAD to Amazon S3 in a different AWS Region, COPY from Amazon EMR, or Secure Shell (SSH) with public IP addresses. For more information about pricing, see [Amazon EC2 Pricing](#).

Topics

- [Working with VPC Endpoints \(p. 116\)](#)
- [Enabling Enhanced VPC Routing \(p. 116\)](#)

- [Using Amazon Redshift Spectrum with Enhanced VPC Routing \(p. 118\)](#)

Working with VPC Endpoints

You can use a VPC endpoint to create a managed connection between your Amazon Redshift cluster in a VPC and Amazon Simple Storage Service (Amazon S3). When you do, COPY and UNLOAD traffic between your cluster and your data on Amazon S3 stays in your Amazon VPC. You can attach an endpoint policy to your endpoint to more closely manage access to your data. For example, you can add a policy to your VPC endpoint that permits unloading data only to a specific Amazon S3 bucket in your account.

Important

Currently, Amazon Redshift supports VPC endpoints only for connecting to Amazon S3. When Amazon VPC adds support for other AWS services to use VPC endpoints, Amazon Redshift will support those VPC endpoint connections also. To connect to an Amazon S3 bucket using a VPC endpoint, the Amazon Redshift cluster and the Amazon S3 bucket that it connects to must be in the same AWS Region.

To use VPC endpoints, create a VPC endpoint for the VPC that your cluster is in and then enable enhanced VPC routing for your cluster. You can enable enhanced VPC routing when you create your cluster in a VPC, or you can modify a cluster in a VPC to use enhanced VPC routing.

A VPC endpoint uses route tables to control the routing of traffic between a cluster in the VPC and Amazon S3. All clusters in subnets associated with the specified route tables automatically use that endpoint to access the service.

Your VPC uses the most specific, or most restrictive, route that matches your cluster's traffic to determine how to route the traffic. For example, suppose that you have a route in your route table for all internet traffic (0.0.0.0/0) that points to an internet gateway and an Amazon S3 endpoint. In this case, the endpoint route takes precedence for all traffic destined for Amazon S3. This is because the IP address range for the Amazon S3 service is more specific than 0.0.0.0/0. In this example, all other internet traffic goes to your internet gateway, including traffic that's destined for Amazon S3 buckets in other AWS Regions.

For more information about creating endpoints, see [VPC Endpoints](#) in the *Amazon VPC User Guide*.

You use endpoint policies to control access from your cluster to the Amazon S3 buckets that hold your data files. By default, the Create Endpoint wizard attaches an endpoint policy doesn't further restrict access from any user or service within the VPC. For more specific control, you can optionally attach a custom endpoint policy. For more information, see [Using Endpoint Policies](#) in the *Amazon VPC User Guide*.

There is no additional charge for using endpoints. Standard charges for data transfer and resource usage apply. For more information about pricing, see [Amazon EC2 Pricing](#).

Enabling Enhanced VPC Routing

You can enable Enhanced VPC Routing when you create a cluster, or you can modify an existing cluster to enable Enhanced VPC Routing.

To work with Enhanced VPC Routing, your cluster must meet the following requirements and constraints:

- Your cluster must be in a VPC.

If you attach an Amazon S3 VPC endpoint, your cluster uses the VPC endpoint only for access to Amazon S3 buckets in the same AWS Region. To access buckets in another AWS Region (not using the

VPC endpoint) or to access other AWS services, make your cluster publicly accessible or use a [network address translation \(NAT\) gateway](#). For more information, see [Creating a Cluster in a VPC \(p. 37\)](#).

- You must enable Domain Name Service (DNS) resolution in your VPC. Alternatively, if you're using your own DNS server, make sure that DNS requests to Amazon S3 are resolved correctly to the IP addresses that are maintained by AWS. For more information, see [Using DNS with Your VPC in the Amazon VPC User Guide](#).
- DNS hostnames must be enabled in your VPC. DNS hostnames are enabled by default.
- Your VPC endpoint policies must allow access to any Amazon S3 buckets used with COPY, UNLOAD, or CREATE LIBRARY calls in Amazon Redshift, including access to any manifest files involved. For COPY from remote hosts, your endpoint policies must allow access to each host machine. For more information, see [IAM Permissions for COPY, UNLOAD, and CREATE LIBRARY](#) in the [Amazon Redshift Database Developer Guide](#).

You can create a cluster with Enhanced VPC Routing enabled by using the AWS Management Console. To do so, choose Yes for the **Enhanced VPC Routing** option in the Launch Cluster wizard's **Configure Networking Options** section, as shown following. For more information, see [Creating a Cluster \(p. 17\)](#).

The screenshot shows the 'Configure Networking Options' section of the AWS Management Console. It includes fields for choosing a VPC, a cluster subnet group, and setting public accessibility. A prominent yellow box highlights the 'Enhanced VPC Routing' checkbox, which is checked ('Yes'). A note below it states: 'Additional VPC configuration might be required.' An 'Availability Zone' dropdown is also visible at the bottom.

Configure Networking Options:

Choose a VPC The identifier of the VPC in which you want to create your cluster

Cluster Subnet Group Selected Cluster Subnet Group may limit the choice of Availability Zones

Publicly Accessible Yes No Select Yes if you want the cluster to have a public IP address that can be accessed from the public internet; select No if you want the cluster to have a private IP address that can only be accessed from within the VPC.

Choose a Public IP Address Yes No Select Yes if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select No to have Amazon Redshift create an EIP for your instance.

Enhanced VPC Routing Yes No ⚠ Additional VPC configuration might be required. [Learn more.](#)

Availability Zone The EC2 Availability Zone that the cluster will be created in.

To modify a cluster to enable Enhanced VPC Routing using the console, choose the cluster. Then choose **Modify Cluster**, and choose Yes for the **Enhanced VPC Routing** option in the **Modify Cluster** dialog box. For more information, see [Modifying a Cluster \(p. 25\)](#).

Note

When you modify a cluster to enable Enhanced VPC Routing, the cluster automatically restarts to apply the change.

Modify Cluster

Cluster Identifier	example-cluster
Cluster Parameter Group	default.redshift-1.0
Parameter Group Description	Default parameter group for redshift-1.0
VPC Security Groups	d-906736679c_controllers ... default (sg-af8223cb) launch-wizard-4 (sg-8c43f...) launch-wizard-5 (sg-5f46a...)
Publicly Accessible	<input checked="" type="radio"/> Yes <input type="radio"/> No
Choose a Public IP Address	<input type="radio"/> Yes <input checked="" type="radio"/> No
Enhanced VPC Routing	<input checked="" type="radio"/> Yes <input type="radio"/> No
Master User Password	[REDACTED]
Automated Snapshot Retention Period	1 days
Maintenance Window Start	Start Day Saturday 03 : 30 UTC
Maintenance Window End	Start Day Saturday 04 : 00 UTC
! Additional VPC configuration might be required. Learn more.	
Cancel Modify	

You can use the following AWS Command Line Interface (AWS CLI) operations for Amazon Redshift to enable Enhanced VPC Routing:

- [create-cluster](#)
- [modify-cluster](#)

You can use the following Amazon Redshift API actions to enable Enhanced VPC Routing:

- [CreateCluster](#)
- [ModifyCluster](#)

Using Amazon Redshift Spectrum with Enhanced VPC Routing

Amazon Redshift Enhanced VPC Routing routes specific traffic through your VPC. All traffic between your cluster and your Amazon S3 buckets is forced to pass through your Amazon VPC. Redshift Spectrum runs on AWS-managed resources that are owned by Amazon Redshift. Because these resources are outside your VPC, Redshift Spectrum doesn't use Enhanced VPC Routing.

When your cluster is configured to use Enhanced VPC Routing, traffic between Redshift Spectrum and Amazon S3 is securely routed through the AWS private network, outside of your VPC. In-flight traffic is signed using Amazon Signature Version 4 protocol (SIGv4) and encrypted using HTTPS. This traffic is authorized based on the IAM role that is attached to your Amazon Redshift cluster. To further manage Redshift Spectrum traffic, you can modify your cluster's IAM role and your policy attached to the Amazon S3 bucket. You might also need to configure your VPC to allow your cluster to access AWS Glue or Athena, as detailed following.

Considerations for Using Enhanced VPC Routing for Redshift Spectrum

Following are considerations when using Redshift Spectrum Enhanced VPC Routing:

- [Bucket Access Policies \(p. 119\)](#)
- [Cluster IAM Role \(p. 119\)](#)
- [Logging and Auditing Amazon S3 Access \(p. 120\)](#)
- [Access to AWS Glue or Amazon Athena \(p. 120\)](#)

Bucket Access Policies

You can control access to data in your Amazon S3 buckets by using a bucket policy attached to the bucket and by using an IAM role attached to the cluster.

Redshift Spectrum can't access data stored in Amazon S3 buckets that use a bucket policy that restricts access to only specified VPC endpoints. Instead, use a bucket policy that restricts access to only specific principals, such as a specific AWS account or specific users.

For the IAM role that is granted access to the bucket, use a trust relationship that allows the role to be assumed only by the Amazon Redshift service principal. When attached to your cluster, the role can be used only in the context of Amazon Redshift and can't be shared outside of the cluster. For more information, see [Restricting Access to IAM Roles \(p. 236\)](#).

The following example bucket policy permits access to the specified bucket only from traffic originated by Redshift Spectrum owned by AWS account 123456789012.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "BucketPolicyForSpectrum",  
            "Effect": "Allow",  
            "Principal": {"AWS": ["arn:aws:iam::123456789012:root"]},  
            "Action": ["s3:GetObject", "s3>List*"],  
            "Resource": ["arn:aws:s3:::examplebucket/*"],  
            "Condition": {"StringEquals": {"aws:UserAgent": "AWS Redshift/Spectrum"}}  
        }  
    ]  
}
```

Cluster IAM Role

The role attached to your cluster should have a trust relationship that permits it to be assumed only by the Amazon Redshift service, as shown following.

```
{  
    "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

You can add a policy to the cluster role that prevents COPY and UNLOAD access to a specific bucket. The following policy permits traffic to the specified bucket only from Redshift Spectrum.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3>List*"],
      "Resource": "arn:aws:s3:::myBucket/*",
      "Condition": {"StringEquals": {"aws:UserAgent": "AWS Redshift/Spectrum"}}
    }
  ]
}
```

For more information, see [IAM Policies for Amazon Redshift Spectrum](#) in the *Amazon Redshift Database Developer Guide*.

Logging and Auditing Amazon S3 Access

One benefit of using Amazon Redshift Enhanced VPC Routing is that all COPY and UNLOAD traffic is logged in the VPC flow logs. Traffic originating from Redshift Spectrum to Amazon S3 doesn't pass through your VPC, so it isn't logged in the VPC flow logs. When Redshift Spectrum accesses data in Amazon S3, it performs these operations in the context of the AWS account and respective role privileges. You can log and audit Amazon S3 access using server access logging in AWS CloudTrail and Amazon S3.

AWS CloudTrail Logs

To trace all access to objects in Amazon S3, including Redshift Spectrum access, enable CloudTrail logging for Amazon S3 objects.

You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. For more information, see [Getting Started with CloudTrail](#).

By default, CloudTrail tracks only bucket-level actions. To track object-level actions (such as `GetObject`), enable data and management events for each logged bucket.

Amazon S3 Server Access Logging

Server access logging provides detailed records for the requests that are made to a bucket. Access log information can be useful in security and access audits. For more information, see [How to Enable Server Access Logging](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information, see the AWS Security blog post [How to Use Bucket Policies and Apply Defense-in-Depth to Help Secure Your Amazon S3 Data](#).

Access to AWS Glue or Amazon Athena

Redshift Spectrum accesses your data catalogs in AWS Glue or Athena through the public endpoints, not through your VPC. If access to the internet isn't an option, use a dedicated Hive Metastore for your data catalogs.

To enable access to AWS Glue or Athena, configure your VPC with an internet gateway or NAT gateway. Configure your VPC security groups to allow outbound traffic to the public endpoints for AWS Glue and Athena.

You can configure the following pathways in your VPC:

- **Internet gateway** –To connect to AWS services outside your VPC, you can attach an [internet gateway](#) to your VPC subnet, as described in the *Amazon VPC User Guide*. To use an internet gateway, your cluster must have a public IP address to allow other services to communicate with your cluster.
- **NAT gateway** –To connect to an Amazon S3 bucket in another AWS Region or to another service within the AWS network, configure a [network address translation \(NAT\) gateway](#), as described in the *Amazon VPC User Guide*. Use this configuration also to access a host instance outside the AWS network.

For more information, see [Amazon Redshift Enhanced VPC Routing \(p. 115\)](#).

Amazon Redshift Parameter Groups

Overview

In Amazon Redshift, you associate a *parameter group* with each cluster that you create. The parameter group is a group of parameters that apply to all of the databases that you create in the cluster. These parameters configure database settings such as query timeout and datestyle.

About Parameter Groups

Each parameter group has several parameters to configure settings for the database. The list of available parameters depends on the parameter group family to which the parameter group belongs. The *parameter group family* is the version of the Amazon Redshift engine to which the parameters in the parameter group apply. The format of the parameter group family name is `redshift-<version>` where `<version>` is the engine version. For example, the current version of the engine is `redshift-1.0`.

Amazon Redshift provides one default parameter group for each parameter group family. The default parameter group has preset values for each of its parameters, and it cannot be modified. The format of the default parameter group name is `default.<parameter_group_family>`, where `<parameter_group_family>` is the version of the engine to which the parameter group belongs. For example, the default parameter group for the `redshift-1.0` version is named `default.redshift-1.0`.

Note

At this time, `redshift-1.0` is the only version of the Amazon Redshift engine. Consequently, `default.redshift-1.0` is the only default parameter group.

If you want to use different parameter values than the default parameter group, you must create a custom parameter group and then associate your cluster with it. Initially, the parameter values in a custom parameter group are the same as in the default parameter group. The initial source for all of the parameters is `engine-default` because the values are preset by Amazon Redshift. After you change a parameter value, the source changes to `user` to indicate that the value has been modified from its default value.

Note

The Amazon Redshift console does not display the source of each parameter. You must use the Amazon Redshift API, the AWS CLI, or one of the AWS SDKs to view the source.

For parameter groups that you create, you can modify a parameter value at any time, or you can reset all parameter values to their defaults. You can also associate a different parameter group with a cluster. If you modify parameter values in a parameter group that is already associated with a cluster or you associate a different parameter group with the cluster, you might need to restart the cluster for the updated parameter values to take effect. If the cluster fails and is restarted by Amazon Redshift, your changes are applied at that time. For more information, see [WLM Dynamic and Static Properties \(p. 125\)](#).

Default Parameter Values

The following table shows the default parameter values at a glance with links to more in-depth information about each parameter. These are the default values for the `redshift-1.0` parameter group family.

Parameter Name	Value	More Information
analyze_threshold_percent	10	analyze_threshold_percent in the <i>Amazon Redshift Database Developer Guide</i>
datestyle	ISO, MDY	datestyle in the <i>Amazon Redshift Database Developer Guide</i>
enable_user_activity_logging	false	Database Audit Logging (p. 298) in this guide
extra_float_digits	0	extra_float_digits in the <i>Amazon Redshift Database Developer Guide</i>
query_group	default	query_group in the <i>Amazon Redshift Database Developer Guide</i>
require_ssl	false	Configure Security Options for Connections (p. 95) in this guide
search_path	\$user, public	search_path in the <i>Amazon Redshift Database Developer Guide</i>
statement_timeout	0	statement_timeout in the <i>Amazon Redshift Database Developer Guide</i>
wlm_json_configuration	[{"query_concurrency":5}]	Configuring Workload Management (p. 124) in this guide
use_fips_ssl	false	Enable FIPS-compliant SSL mode only if your system is required to be FIPS compliant.

Note

The `max_cursor_result_set_size` parameter is deprecated. For more information about cursor result set size, see [Cursor Constraints](#) in the *Amazon Redshift Database Developer Guide*.

You can temporarily override a parameter by using the `SET` command in the database. The `SET` command overrides the parameter for the duration of your current session only. In addition to the parameters listed in the preceding table, you can also temporarily adjust the slot count by setting `wlm_query_slot_count` in the database. The `wlm_query_slot_count` parameter is not available for configuration in parameter groups. For more information about adjusting the slot count, see [wlm_query_slot_count](#) in the *Amazon Redshift Database Developer Guide*. For more information about temporarily overriding the other parameters, see [Modifying the Server Configuration](#) in the *Amazon Redshift Database Developer Guide*.

Configuring Parameter Values Using the AWS CLI

To configure Amazon Redshift parameters by using the AWS CLI, you use the `modify-cluster-parameter-group` command for a specific parameter group. You specify the parameter group to modify in `parameter-group-name`. You use the `parameters` parameter (for the `modify-cluster-parameter-group` command) to specify name/value pairs for each parameter that you want to modify in the parameter group.

Note

There are special considerations when configuring the `wlm_json_configuration` parameter by using the AWS CLI. The examples in this section apply to all of the parameters except `wlm_json_configuration`. For more information about configuring

wlm_json_configuration by using the AWS CLI, see [Configuring Workload Management \(p. 124\)](#).

After you modify parameter values, you must reboot any clusters that are associated with the modified parameter group. The cluster status displays applying for ParameterApplyStatus while the values are being applied, and then pending-reboot after the values have been applied. After you reboot, the databases in your cluster begin use the new parameter values. For more information about rebooting clusters, see [Rebooting a Cluster \(p. 28\)](#).

Note

The wlm_json_configuration parameter contains some properties that are dynamic and do not require you to reboot associated clusters for the changes to be applied. For more information about dynamic and static properties, see [WLM Dynamic and Static Properties \(p. 125\)](#).

Syntax

The following syntax shows how to use the modify-cluster-parameter-group command to configure a parameter. You specify `parameter_group_name` and replace both `parameter_name` and `parameter_value` with an actual parameter to modify and a value for that parameter. If you want to modify more than one parameter at the same time, separate each parameter and value set from the next with a space.

```
aws redshift modify-cluster-parameter-group --parameter-group-name parameter_group_name --parameters ParameterName=parameter_name,ParameterValue=parameter_value
```

Example

The following example shows how to configure the statement_timeout and enable_user_activity_logging parameters for the myclusterparametergroup parameter group.

Note

For readability purposes, the example is displayed on several lines, but in the actual AWS CLI this is one line.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name myclusterparametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000
ParameterName=enable_user_activity_logging,ParameterValue=true
```

Configuring Workload Management

In Amazon Redshift, you use workload management (WLM) to define the number of query queues that are available, and how queries are routed to those queues for processing. WLM is part of parameter group configuration. A cluster uses the WLM configuration that is specified in its associated parameter group.

When you create a parameter group, the default WLM configuration contains one queue that can run up to five queries concurrently. You can add additional queues and configure WLM properties in each of them if you want more control over query processing. Each queue that you add has the same default WLM configuration until you configure its properties.

When you add additional queues, the last queue in the configuration is the *default queue*. Unless a query is routed to another queue based on criteria in the WLM configuration, it is processed by the default queue. You cannot specify user groups or query groups for the default queue.

As with other parameters, you cannot modify the WLM configuration in the default parameter group. Clusters associated with the default parameter group always use the default WLM configuration. If you want to modify the WLM configuration, you must create a parameter group and then associate that parameter group with any clusters that require your custom WLM configuration.

WLM Dynamic and Static Properties

The WLM configuration properties are either dynamic or static. Dynamic properties can be applied to the database without a cluster reboot, but static properties require a cluster reboot for changes to take effect. However, if you change dynamic and static properties at the same time, then you must reboot the cluster for all the property changes to take effect regardless of whether they are dynamic or static. While dynamic properties are being applied, your cluster status will be modifying.

The following WLM properties are static:

- User groups
- User group wildcard
- Query groups
- Query group wildcard

Adding, removing, or reordering query queues is a static change and requires a cluster reboot to take effect.

The following WLM properties are dynamic:

- Enable short query acceleration
- Maximum run time for short queries
- Concurrency
- Percent of memory to use
- Timeout

If the timeout value is changed, the new value is applied to any query that begins execution after the value is changed. If the concurrency or percent of memory to use are changed, Amazon Redshift transitions to the new configuration dynamically so that currently running queries are not affected by the change. For more information, see [WLM Dynamic Memory Allocation](#).

Properties in the `wlm_json_configuration` Parameter

You can configure WLM by using the Amazon Redshift console, the AWS CLI, Amazon Redshift API, or one of the AWS SDKs. WLM configuration uses several properties to define queue behavior, such as memory allocation across queues, the number of queries that can run concurrently in a queue, and so on. The following list describes the WLM properties that you can configure for each queue.

Note

The following properties appear with their Amazon Redshift console names, with the corresponding JSON property names in the descriptions.

Enable short query acceleration

Short query acceleration (SQA) prioritizes selected short-running queries ahead of longer-running queries. SQA executes short-running queries in a dedicated space, so that SQA queries aren't forced to wait in queues behind longer queries. With SQA, short-running queries begin executing more quickly and users see results sooner. When you enable SQA, you can also specify the maximum run time for short queries. To enable SQA, specify `true`. The default is `false`.

JSON property: short_query_queue

Maximum run time for short queries

When you enable SQA, you can specify 0 to let WLM dynamically set the maximum run time for short queries. Alternatively, you can specify a value of 1–20 seconds, in milliseconds. The default value is 0.

JSON property: max_execution_time

Concurrency

The number of queries that can run concurrently in a queue. When a queue reaches the concurrency level, any subsequent queries wait in the queue until resources are available to process them. The range is between 1 and 50.

JSON property: query_concurrency

User Groups

A comma-separated list of user group names. When members of the user group run queries in the database, their queries are routed to the queue that is associated with their user group.

JSON property: user_group

User Group Wildcard

A Boolean value that indicates whether to enable wildcards for user groups. If this is 0, wildcards are disabled; if this is 1, wildcards are enabled. When wildcards are enabled, you can use "*" or "?" to specify multiple user groups when running queries. For more information, see [Wildcards](#).

JSON property: user_group_wild_card

Query Groups

A comma-separated list of query groups. When members of the query group run queries in the database, their queries are routed to the queue that is associated with their query group.

JSON property: query_group

Query Group Wildcard

A Boolean value that indicates whether to enable wildcards for query groups. If this is 0, wildcards are disabled; if this is 1, wildcards are enabled. When wildcards are enabled, you can use "*" or "?" to specify multiple query groups when running queries. For more information, see [Wildcards](#).

JSON property: query_group_wild_card

Timeout (ms)

The maximum time, in milliseconds, queries can run before being canceled. If a read-only query, such as a SELECT statement, is canceled due to a WLM timeout, WLM attempts to route the query to the next matching queue based on the WLM Queue Assignment Rules. If the query doesn't match any other queue definition, the query is canceled; it is not assigned to the default queue. For more information, see [WLM Query Queue Hopping](#). WLM timeout doesn't apply to a query that has reached the returning state. To view the state of a query, see the [STV_WLM_QUERY_STATE](#) system table.

JSON property: max_execution_time

Memory (%)

The percentage of memory to allocate to the queue. If you specify a memory percentage for at least one of the queues, you must specify a percentage for all of the other queues up to a total of 100 percent. If your memory allocation is below 100 percent across all of the queues, the unallocated memory is managed by the service and can be temporarily given to a queue that requests additional memory for processing.

JSON property: `memory_percent_to_use`

Query Monitoring Rules

You can use WLM query monitoring rules to continuously monitor your WLM queues for queries based on criteria, or predicates, that you specify. For example, you might monitor queries that tend to consume excessive system resources, and then initiate a specified action when a query exceeds your specified performance boundaries.

Note

If you choose to create rules programmatically, we strongly recommend using the console to generate the JSON that you include in the parameter group definition.

You associate a query monitoring rule with a specific query queue. You can have up to eight rules per queue, and the total limit for all queues is eight rules.

JSON property: `rules`

JSON properties hierarchy:

```
rules
  rule_name
  predicate
    metric_name
    operator
    value
  action
```

For each rule, you specify the following properties:

- `rule_name` – Rule names must be unique within WLM configuration. Rule names can be up to 32 alphanumeric characters or underscores, and can't contain spaces or quotation marks. You can have up to eight rules per queue, and the total limit for all queues is eight rules.
 - `predicate` – You can have up to three predicates per rule. For each predicate, specify the following properties.
 - `metric_name` – For a list of metrics, see [Query Monitoring Metrics](#) in the *Amazon Redshift Database Developer Guide*.
 - `operator` – Operations are `=`, `<`, and `>`.
 - `value` – The threshold value for the specified metric that triggers an action.
 - `action` – Each rule is associated with one action. Valid actions are:
 - `log`
 - `hop`
 - `abort`

The following example shows the JSON for a WLM query monitoring rule named `rule_1`, with two predicates and the action `hop`.

```
"rules": [
  {
    "rule_name": "rule_1",
    "predicate": [
      {
        "metric_name": "query_cpu_time",
        "operator": ">",
        "value": 100000
      },
      {
        "metric_name": "query_blocks_read",
        "operator": ">",
        "value": 1000
      }
    ],
    "action": "hop"
  }
]
```

```
        },
      ],
      "action": "hop"
    }
]
```

For more information about each of these properties and strategies for configuring query queues, go to [Defining Query Queues](#) and [Implementing Workload Management](#) in the *Amazon Redshift Database Developer Guide*.

Configuring the wlm_json_configuration Parameter Using the AWS CLI

To configure WLM, you modify the `wlm_json_configuration` parameter. The value is formatted in JavaScript Object Notation (JSON). If you configure WLM by using the AWS CLI, Amazon Redshift API, or one of the AWS SDKs, use the rest of this section to learn how to construct the JSON structure for the `wlm_json_configuration` parameter.

Note

If you configure WLM by using the Amazon Redshift console, you do not need to understand JSON formatting because the console provides an easy way to add queues and configure their properties. For more information about configuring WLM by using the Amazon Redshift console, see [Modifying a Parameter Group \(p. 134\)](#).

Example

The following example is the default WLM configuration, which defines one queue with a concurrency level of five.

```
{
  "query_concurrency":5
}
```

Syntax

The default WLM configuration is very simple, with only queue and one property. You can add more queues and configure multiple properties for each queue in the JSON structure. The following syntax represents the JSON structure that you use to configure multiple queues with multiple properties:

```
[
  {
    "ParameterName": "wlm_json_configuration", "ParameterValue":
      [
        [
          {
            "q1_first_property_name": "q1_first_property_value",
            "q1_second_property_name": "q1_second_property_value",
            ...
          },
          {
            "q2_first_property_name": "q2_first_property_value",
            "q2_second_property_name": "q2_second_property_value",
            ...
          }
          ...
        ]
      ]
]
```

In the preceding example, the representative properties that begin with **q1** are objects in an array for the first queue. Each of these objects is a name/value pair; name and value together set the WLM properties for the first queue. The representative properties that begin with **q2** are objects in an array for the second queue. If you require more queues, you add another array for each additional queue and set the properties for each object.

When you modify the WLM configuration, you must include in the entire structure for your queues, even if you only want to change one property within a queue. This is because the entire JSON structure is passed in as a string as the value for the `wlm_json_configuration` parameter.

Formatting the AWS CLI Command

The `wlm_json_configuration` parameter requires a specific format when you use the AWS CLI. The format that you use depends on your client operating system. Operating systems have different ways to enclose the JSON structure so it's passed correctly from the command line. For details on how to construct the appropriate command in the Linux, Mac OS X, and Windows operating systems, see the sections following. For more information about the differences in enclosing JSON data structures in the AWS CLI in general, see [Quoting Strings](#) in the *AWS Command Line Interface User Guide*.

Examples

The following example command configures WLM for a parameter group called `example-parameter-group`. The configuration enables short-query acceleration with a maximum run time for short queries set to 0, which instructs WLM to set the value dynamically.. The `ApplyType` setting is `dynamic`. This setting means that any changes made to dynamic properties in the parameter are applied immediately unless other static changes have been made to the configuration. The configuration defines three queues with the following:

- The first queue enables users to specify `report` as a label (as specified in the `query_group` property) in their queries to help in routing queries to that queue. Wildcard searches are enabled for the `report*` label, so the label doesn't need to be exact for queries to be routed to the queue. For example, `reports` and `reporting` both match this query group. The queue is allocated 25 percent of the total memory across all queues, and can run up to four queries at the same time. Queries are limited to a maximum time of 20000 milliseconds (ms).
- The second queue enables users who are members of `admin` or `dba` groups in the database to have their queries routed to the queue for processing. Wildcard searches are disabled for user groups, so users must be matched exactly to groups in the database in order for their queries to be routed to the queue. The queue is allocated 40 percent of the total memory across all queues, and it can run up to five queries at the same time.
- The last queue in the configuration is the default queue. This queue is allocated 35 percent of the total memory across all queues, and it can process up to five queries at a time.

Note

The example is shown on several lines for demonstration purposes. Actual commands should not have line breaks.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-parameter-group
--parameters
'[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "[
      {
        "query_group": ["report*"],
        "query_group_wild_card": 1,
        "query_concurrency": 4,
        "max_execution_time": 20000,
```

```
        "memory_percent_to_use":25
    },
    {
        "user_group":["admin","dba"],
        "user_group_wild_card":0,
        "query_concurrency":5,
        "memory_percent_to_use":40
    },
    {
        "query_concurrency":5,
        "memory_percent_to_use":35
    },
    {
        "short_query_queue": true,
        "max_execution_time": 0
    }
],
"ApplyType":"dynamic"
}
]
```

The following is an example of configuring WLM query monitoring rules. The example creates a parameter group named `example-monitoring-rules`. The configuration defines the same three queues as the previous example, and adds the following rules:

- The first queue defines a rule named `rule_1`. The rule has two predicates: `query_cpu_time > 10000000` and `query_blocks_read > 1000`. The rule action is `log`.
- The second queue defines a rule named `rule_2`. The rule has two predicates: `query_execution_time > 600000000` and `scan_row_count > 1000000000`. The rule action is `hop`.
- The last queue in the configuration is the default queue.

Note

The example is shown on several lines for demonstration purposes. Actual commands should not have line breaks.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-monitoring-rules
--parameters
'[
    {
        "ParameterName":"wlm_json_configuration",
        "ParameterValue":>[
            {
                "query_group":["report*"],
                "query_group_wild_card":1,
                "query_concurrency":4,
                "max_execution_time":20000,
                "memory_percent_to_use":25,
                "rules": [{"rule_name": "rule_1",
                           "predicate": [
                               {"metric_name": "query_cpu_time",
                                "operator": ">",
                                "value": 1000000},
                               {"metric_name": "query_blocks_read",
                                "operator": ">",
                                "value": 1000}],
                           "action": "log"}]
            },
            {
                "user_group":["admin","dba"],
```

```
"user_group_wild_card":0,  
"query_concurrency":5,  
"memory_percent_to_use":40,  
"rules": [{"rule_name": "rule_2",  
    "predicate": [  
        {"metric_name": "query_execution_time",  
            "operator": ">",  
            "value": 600000000},  
        {"metric_name": "scan_row_count",  
            "operator": ">",  
            "value": 1000000000}],  
    "action": "hop"}]  
  
,  
{  
    "query_concurrency":5,  
    "memory_percent_to_use":35  
,  
{  
    "short_query_queue": true,  
    "max_execution_time": 5000  
}  
]  
,  
"ApplyType":"dynamic"  
}  
]
```

Rules for Configuring WLM by Using the AWS CLI in the Command Line on the Linux and Mac OS X Operating Systems

- The entire JSON structure must be enclosed in single quotation marks (') and brackets ([]).
- All parameter names and parameter values must be enclosed in double quotation marks (").
- Within the ParameterValue value, you must enclose the entire nested structure in double-quotation marks (") and brackets ([]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({ }).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark (").
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace ({ }) and the beginning of the next queue's curly brace ({ }).

Example

The following example shows how to configure the queues described in this section by using the AWS CLI on the Linux and Mac OS X operating systems.

Note

This example must be submitted on one line in the AWS CLI.

```
aws redshift modify-cluster-parameter-group --parameter-group-name example-parameter-group --parameters '[{"ParameterName": "wlm_json_configuration", "ParameterValue": "\[{\\"query_group\\":\[\"report*\\\"], \"query_group_wild_card\\\":1, \"query_concurrency\\\":4, \"max_execution_time\\\":20000, \"memory_percent_to_use\\\":25}, {\\"user_group\\":\[\"admin\", \"dba\"], \"user_group_wild_card\\\":1, \"query_concurrency\\\":5, \"memory_percent_to_use\\\":40}, {\"query_concurrency\\\":5, \"memory_percent_to_use\\\":35}, {\"short_query_queue\\\": true, \"max_execution_time\\\": 5000 }\\\"]\\\", \"ApplyType\\\":\\\"dynamic\\\"}]"
```

Rules for Configuring WLM by Using the AWS CLI in Windows PowerShell on Microsoft Windows Operating Systems

- The entire JSON structure must be enclosed in single quotation marks ('') and brackets ([]).
- All parameter names and parameter values must be enclosed in double quotation marks ("").
- Within the `ParameterValue` value, you must enclose the entire nested structure in double-quotation marks ("") and brackets ([]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({}).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark ("") and its backslash (\) escape character. This requirement means that you will use three backslashes and a double quotation mark to make sure that the properties are passed in correctly (\\\:).
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace (}) and the beginning of the next queue's curly brace ({}).

Example

The following example shows how to configure the queues described in this section by using the AWS CLI in Windows PowerShell on Windows operating systems.

Note

This example must be submitted on one line in the AWS CLI.

```
aws redshift modify-cluster-parameter-group --parameter-group-name example-parameter-group  
--parameters '[{"ParameterName": "wlm_json_configuration", "ParameterValue": "[{\\"query_group\\\": [\"report\"], \\"query_group_wild_card\\\": 1, \\"query_concurrency\\\": 4, \\"max_execution_time\\\": 20000, \\"memory_percent_to_use\\\": 25}, {\\"user_group\\\": [\"admin\", \"dba\"], \\"user_group_wild_card\\\": 1, \\"query_concurrency\\\": 5, \\"memory_percent_to_use\\\": 40}, {\\"query_concurrency\\\": 5, \\"memory_percent_to_use\\\": 35}, {\\"short_query_queue\\\": true, \\"max_execution_time\\\": 5000 }]}", "ApplyType": "dynamic"}]
```

Rules for Configuring WLM by Using the Command Prompt on Windows Operating Systems

- The entire JSON structure must be enclosed in double-quotation marks ("") and brackets ([]).
- All parameter names and parameter values must be enclosed in double quotation marks ("").
- Within the `ParameterValue` value, you must enclose the entire nested structure in double-quotation marks ("") and brackets ([]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({}).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark ("") and its backslash (\) escape character. This requirement means that you will use three backslashes and a double quotation mark to make sure that the properties are passed in correctly (\\\:).
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace (}) and the beginning of the next queue's curly brace ({}).

Example

The following example shows how to configure the queues described in this section by using the AWS CLI in the command prompt on Windows operating systems.

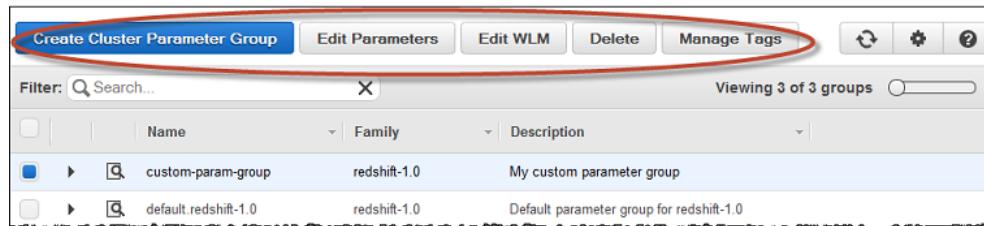
Note

This example must be submitted on one line in the AWS CLI.

```
aws redshift modify-cluster-parameter-group --parameter-group-name example-parameter-group
--parameters "[{\\"ParameterName\\":\\"wlm_json_configuration\\",\\"ParameterValue\\":\"[{\\"query_group\\\"::[\\\\\"report*\\\\\"],\\\\\"query_group_wild_card\\\":1,\\\\\"query_concurrency\\\":4,\\\\\"max_execution_time\\\":2000,\\\\\"memory_percent_to_use\\\":25},{\\\\\"user_group\\\":{\\\\\"admin\\\",\\\\\"dba\\\"},\\\\\"user_group_wild_card\\\":1,\\\\\"query_concurrency\\\":5,\\\\\"memory_percent_to_use\\\":40},{\\\\\"query_concurrency\\\":5,\\\\\"memory_percent_to_use\\\":35},{ \\\\\"short_query_queue\\\": true, \\\\\"max_execution_time\\\": 5000 }]\",\\"ApplyType\\\":\\\"dynamic\\\"}]"
```

Managing Parameter Groups Using the Console

You can view, create, modify, and delete parameter groups by using the Amazon Redshift console. To initiate these tasks, use the buttons on the **Parameter Groups** page, as shown in the following screenshot.



You can expand any of the parameter groups in the list to see a summary of the values for parameters and workload management (WLM) configuration. In the following screenshot, the parameter group called `custom-parameter-group` is expanded to show the summary of parameter values.

A screenshot of the Amazon Redshift Parameter Groups page showing the expanded details for the 'custom-param-group'. At the top, it shows the same buttons and search/filter functionality as the previous screenshot. Below the table, there is a section titled 'Parameters' which lists the following configuration options:

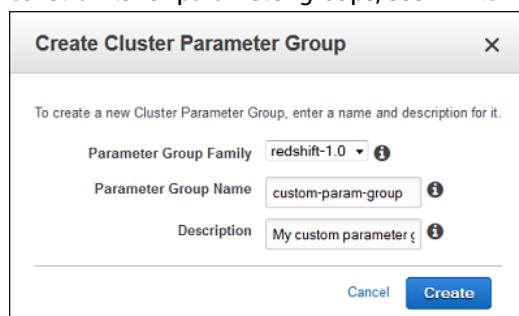
Parameter	Value
datestyle	ISO, MDY
enable_user_activity_logging	false
extra_float_digits	0
force_acm	false
query_group	default
require_ssl	false
search_path	\$user, public
statement_timeout	0
use_fips_ssl	false
wlm_json_configuration	[{"query_group":[], "user_group":[]}]

Creating a Parameter Group

You can create a parameter group if you want to set parameter values that are different from the default parameter group.

To create a parameter group

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 **Parameter Groups**.
3. On the **Parameter Groups** page, choose **Create Cluster Parameter Group**.
4. In the **Create Cluster Parameter Group** dialog box, choose a parameter group family, and then type a parameter group name and a parameter group description. For more information about naming constraints for parameter groups, see [Limits in Amazon Redshift \(p. 323\)](#).



5. Choose **Create**.

Modifying a Parameter Group

You can modify parameters to change the parameter settings and WLM configuration properties.

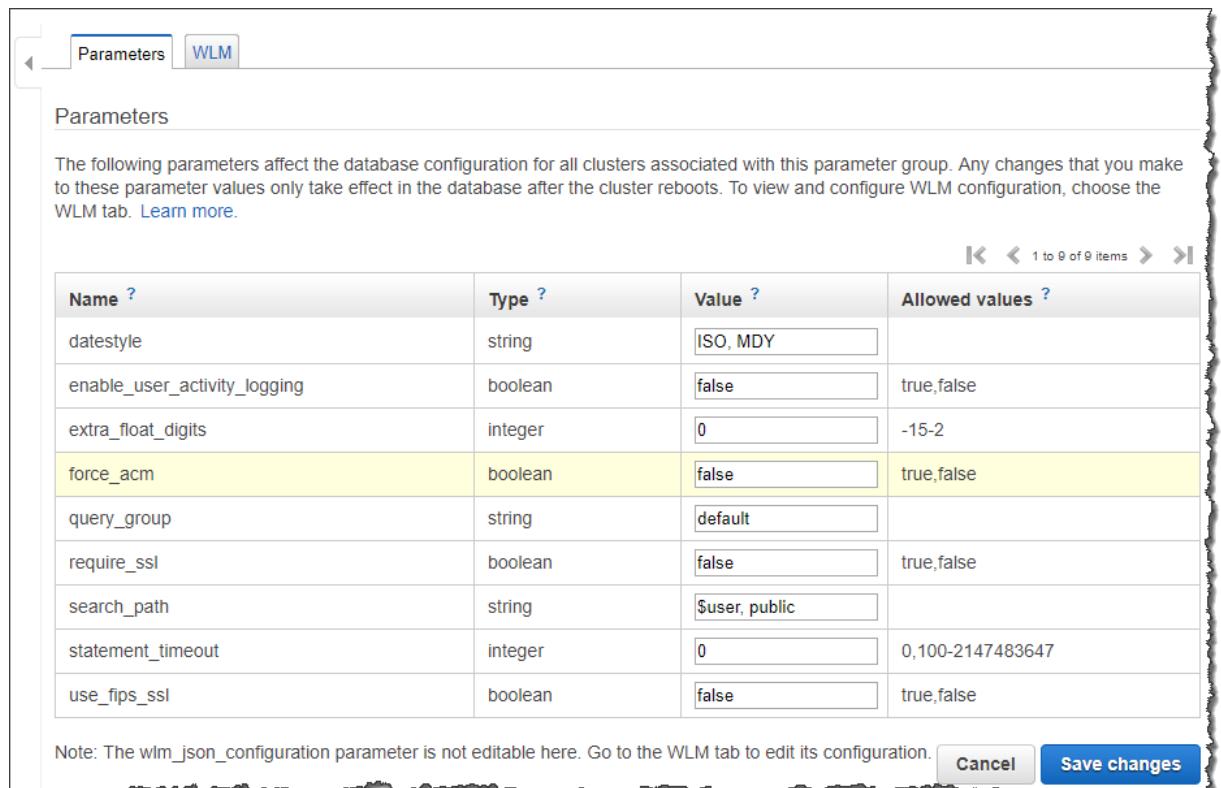
Note

You cannot modify the default parameter group.

To modify parameters in a parameter group

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 **Parameter Groups**.
3. On the **Parameter Groups** page, in the parameter group list, select the row of the parameter group that you want to modify.
4. To edit the parameters other than the WLM configuration parameter, choose **Edit Parameters**.

The **Parameters** tab opens as shown in the following screenshot.



The screenshot shows the 'Parameters' tab of a parameter group in the Amazon Redshift console. The table lists various configuration parameters:

Name ?	Type ?	Value ?	Allowed values ?
datestyle	string	ISO, MDY	
enable_user_activity_logging	boolean	false	true,false
extra_float_digits	integer	0	-15-2
force_acm	boolean	false	true,false
query_group	string	default	
require_ssl	boolean	false	true,false
search_path	string	\$user, public	
statement_timeout	integer	0	0,100-2147483647
use_fips_ssl	boolean	false	true,false

Note: The wlm_json_configuration parameter is not editable here. Go to the WLM tab to edit its configuration.

[Cancel](#) [Save changes](#)

5. In the **Value** box that corresponds to the parameter you want to modify, type a new value. For more information about these parameters, see [Amazon Redshift Parameter Groups \(p. 122\)](#).
6. Choose **Save Changes**.

Note

If you modify these parameters in a parameter group that is already associated with a cluster, reboot the cluster for the changes to be applied. For more information, see [Rebooting a Cluster \(p. 28\)](#).

To modify the WLM configuration in a parameter group

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 **Workload management**.
3. For **Parameter groups**, choose the parameter group that you want to modify.

Note

You cannot modify the default parameter group.

4. To edit the WLM configuration, choose **Edit**.

A screenshot of the AWS Management Console for Amazon Redshift. The top navigation bar shows 'Parameters' and 'WLM'. The 'WLM' tab is selected. The main content area is titled 'Workload Management Configuration'. Under 'Parameter groups', 'sqa' is selected. A large blue button labeled 'Add queue' is visible. The 'Short Query Acceleration' section is expanded, showing the 'Enable short query acceleration' checkbox checked and the 'Maximum run time for short queries (1 to 20 seconds)' dropdown set to 'dynamic'. Below this, the 'Default queue' configuration is shown with a concurrency limit of 5. A note at the bottom of this section says 'No rules have been defined.'

5. To enable short query acceleration (SQA), choose **Enable short query acceleration**.
 6. When you enable SQA, **Maximum run time for short queries (1 to 20 seconds)** is set to **Dynamic** by default. To set the maximum run time to a fixed value, choose a value of 1–20.
 7. Do one or more of the following to modify the queue configuration:
 - To create a queue, choose **Add Queue**.
 - To modify a queue, change property values in the table.
 - To change the order of queues, choose the **Up** and **Down** arrow buttons in the table.
 - To delete a queue, choose **Delete** in the queue's row in the table.
 8. To have changes applied to associated clusters after their next reboot, choose **Apply dynamic changes after cluster reboot**.
- Note**
Some changes require a cluster reboot regardless of this setting. For more information, see [WLM Dynamic and Static Properties \(p. 125\)](#).
9. Choose **Save**.

Creating or Modifying a Query Monitoring Rule Using the Console

You can use the AWS Management Console to create and modify WLM query management rules. Query monitoring rules are part of the WLM configuration parameter for a parameter group. For more information, see [WLM Query Monitoring Rules](#).

When you create a rule, you define the rule name, one or more predicates, and an action.

When you save WLM configuration that includes a rule, you can view the JSON code for the rule definition as part of the JSON for the WLM configuration parameter.

To create a query monitoring rule

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 **Workload management**.
3. For **Parameter groups**, choose the parameter group that you want to modify.

Note

You can't modify the default parameter group.

4. To edit the WLM configuration, choose **Edit**.

- 5.

The screenshot shows the 'Workload Management Configuration' page in the AWS Management Console. At the top, there are tabs for 'Parameters' and 'WLM', with 'WLM' selected. Below the tabs, the title 'Workload Management Configuration' is displayed. A note states: 'Workload Management (WLM) is an ordered set of query queues that define how resources are allocated and how queries are routed for processing. Learn more.' A dropdown menu under 'Parameter groups' shows 'sqa'. A large blue button labeled 'Add queue' is visible. The 'Short Query Acceleration' section contains a checkbox for 'Enable short query acceleration' which is checked, and a dropdown for 'Maximum run time for short queries (1 to 20 seconds)' set to 'dynamic'. The 'Default queue' section has a 'Concurrency' input field set to '5'. Below it, a table titled 'Rules for Default queue' shows a message: 'No rules have been defined.' There are buttons for 'Add rule from templates' and 'Add custom'.

6. Choose **Add queue**. A new queue appears, as shown in the following screenshot.

Workload Management Configuration

Workload Management (WLM) is an ordered set of query queues that define how resources are allocated and how queries are routed for processing. You can add query monitoring rules to your WLM queues. Rules define metrics-based performance boundaries and specify what action to take when a query exceeds those boundaries. [Learn more](#).

Parameter groups: custom-param-group

Add queue

Short Query Acceleration

Short query acceleration (SQA) routes queries that run quickly to a dedicated short-query queue. When SQA is enabled, you can reduce concurrency in other queues for longer-running queries. Overall system performance is improved when your concurrency is lower. For optimal performance with SQA, configure the concurrency across all WLM queues at 15 or less. [Learn more](#).

Enable short query acceleration: Maximum run time for short queries: 5 (1 to 20 seconds)

Queue 1

Concurrency	User groups	Match wildcards	Query groups	Match wildcards	Timeout (ms)	Memory limit (MB)
5		+ <input type="checkbox"/>		+ <input type="checkbox"/>	0	

Rules for Queue 1

No rules have been defined.

- To create a new rule using a predefined template, in the **Rules for Queue 1** group, choose **Add Rule from Templates**. The **Rule Templates** dialog appears, as shown in the following screenshot.

Rule Templates

Template Name	Predicate
<input type="checkbox"/> Nested loop join	Nested loop join row count (rows) > 100
<input type="checkbox"/> Query returns a high number of rows	Return row count (rows) > 1000000
<input type="checkbox"/> Join with a high number of rows	Rows joined (rows) > 1000000000
<input type="checkbox"/> High disk usage when writing intermediate results	Memory to disk (1 MB blocks) > 100000
<input checked="" type="checkbox"/> Long running query with high I/O skew	Segment execution time (seconds) > 120 And I/O skew (ratio) > 2

- Choose one or more rule templates. WLM creates one rule for each template you choose. For this example, choose **Long running query with high I/O skew** and then choose **Select**.

A new rule appears with two predicates, as shown in the following screenshot.

Queue 1						Done Editing	Delete
Concurrency	User groups	Match wildcards	Query groups	Match wildcards	Timeout (ms)	Memory	
5		<input checked="" type="checkbox"/>		<input type="checkbox"/>	0		
Rules for Queue 1							Add Rule from Templates Add Custom Rule
Rule names	Predicates			Actions			Delete
<input type="text"/>	Segment execution time (seconds)	>	120	<input checked="" type="radio"/>	log	<input type="radio"/>	Delete
	I/O skew (ratio)	>	2	<input checked="" type="radio"/>	hop	<input type="radio"/>	

9. Type a **Rule name**. The name can be up to 32 alphanumeric characters and must not contain spaces or quotation mark characters. For this example, type **HighIOskew**.
10. Optionally, modify the predicates.
11. Choose an **Action**. Each rule has one action. For this example, choose **hop**. Hop terminates the query and WLM routes the query to the next matching queue, if one is available.
12. Choose **Save**.

Queue 1						Edit	Delete
Concurrency	User groups	Match wildcards	Query groups	Match wildcards	Timeout (ms)	Memory	
5		<input checked="" type="checkbox"/>		<input type="checkbox"/>	0		
Rules for Queue 1							
Rule names	Predicates			Actions			
HighIOskew	segment_execution_time > 120 and io_skew > 2			hop			

13. To modify the rules for a queue, choose **Edit**.
14. To add a new queue from scratch, choose **Add Custom Rule**. You can a maximum of five rules per queue, and a total of eight rules for all queues.
15. Type a **Rule name**; for example, **NestedLoop**.
16. Define a **Predicate**. Choose a predicate name, an operator, and a value. For this example, choose **Nested loop join count (rows)**. Leave the operator at greater than (>), and for the value type **1000**. The following screen shot shows the new rule with one predicate.

The screenshot shows the 'Rules for Queue 1' section of the WLM configuration. It contains two rules:

- HighOskew:** Predicates: Segment execution time (seconds) > 120, I/O skew (ratio) > 2. Action: log.
- NestedLoop:** Predicate: Nested loop join row count (rows) > 1000. Action: log.

17. To add additional predicates, choose the add icon to the right of the predicates. You can have up to three predicates per rule. If all of the predicates are met, WLM triggers the associated action.
18. Choose an **Action**. Each rule has one action. For this example, accept the default action, **Log**. The Log action writes a record to the **STL_WLM_RULE_ACTION** system table and leaves the query running in the queue.
19. Choose **Done Editing**. The queue details collapse.
20. Choose **Save**.
21. Amazon Redshift generates your WLM configuration parameter in JSON format and displays the JSON in a window at the bottom of the screen, as shown in the following screenshot.

JSON format of your saved WLM configuration parameter. [Learn more.](#)

```
1 [ {  
2   "query_group" : [ ],  
3   "user_group" : [ ],  
4   "query_group_wild_card" : 0,  
5   "rules" : [ {  
6     "predicate" : [ {  
7       "metric_name" : "segment_execution_time",  
8       "value" : 120,  
9       "operator" : ">"  
10    }, {  
11      "metric_name" : "io_skew",  
12      "value" : 2,  
13      "operator" : ">"  
14    } ],  
15  } ]
```

Deleting a Parameter Group

You can delete a parameter group if you no longer need it and it is not associated with any clusters. You can only delete custom parameter groups.

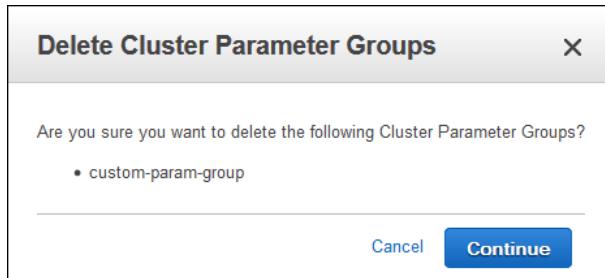
To delete a parameter group

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 **Parameter Groups**.
3. Select the row of the parameter group that you want to delete, and then choose **Delete**.

Note

You cannot delete the default parameter group.

4. In the **Delete Cluster Parameter Groups** dialog box, choose **Continue**.



Associating a Parameter Group with a Cluster

When you launch a cluster, you must associate it with a parameter group. If you want to change the parameter group later, you can modify the cluster and choose a different parameter group. For more information, see [Creating a Cluster by Using Launch Cluster \(p. 18\)](#) and [To modify a cluster \(p. 26\)](#).

Managing Parameter Groups Using the AWS SDK for Java

This example demonstrates the following tasks related to parameter groups:

- Creating a parameter group
- Modifying a parameter group
- Associating a parameter group with a cluster
- Getting information about parameter groups

This example creates a new parameter group, `parametergroup1`, and makes the following updates:

- Changes the parameter `extra_float_digits` to 2 from the default value of 0.
- Replaces the existing workload management configuration (`wlm_json_configuration` parameter) with the following JSON which defines a queue in addition to the default queue.

```
[  
  {  
    "user_group": [  
      "example_user_group1"  
    ],  
    "query_group": [  
      "example_query_group1"  
    ],  
    "query_concurrency": 7  
  },  
  {  
    "query_concurrency": 5  
  }  
]
```

The preceding JSON is an array of two objects, one for each queue. The first object defines a queue with specific user group and query group. It also sets the concurrency level to 7.

```
{
```

```

"user_group": [
    "example_user_group1"
],
"query_group": [
    "example_query_group1"
],
"query_concurrency": 7
}

```

Because this example replaces the WLM configuration, this JSON configuration also defines the default queue with no specific user group or query group. It sets the concurrency to the default value, 5.

```
{
    "query_concurrency": 5
}
```

For more information about Workload Management (WML) configuration, go to [Implementing workload management](#).

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and provide a cluster identifier.

Example

```

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterParameterGroup {

    public static AmazonRedshiftClient client;
    public static String clusterParameterGroupName = "parametergroup1";
    public static String clusterIdentifier = "***provide cluster identifier***";
    public static String parameterGroupFamily = "redshift-1.0";

    public static void main(String[] args) throws IOException {
        AWS Credentials credentials = new PropertiesCredentials(
            CreateAndModifyClusterParameterGroup.class
                .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
            createClusterParameterGroup();
            modifyClusterParameterGroup();
            associateParameterGroupWithCluster();
            describeClusterParameterGroups();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createClusterParameterGroup() {
        CreateClusterParameterGroupRequest request = new
        CreateClusterParameterGroupRequest()
            .withDescription("my cluster parameter group")
            .withParameterGroupName(clusterParameterGroupName)
    }
}

```

```

        .withParameterGroupFamily(parameterGroupFamily);
    client.createClusterParameterGroup(request);
    System.out.println("Created cluster parameter group.");
}

private static void describeClusterParameterGroups() {
    DescribeClusterParameterGroupsResult result =
client.describeClusterParameterGroups();
    printResultClusterParameterGroups(result);
}

private static void modifyClusterParameterGroup() {
    List<Parameter> parameters = new ArrayList<Parameter>();
    parameters.add(new Parameter()
        .withParameterName("extra_float_digits")
        .withParameterValue("2"));
    // Replace WLM configuration. The new configuration defines a queue (in addition to
    the default).
    parameters.add(new Parameter()
        .withParameterName("wlm_json_configuration")
        .withParameterValue("[{\\"user_group\": [\"example_user_group1\"], \"query_group\":
[\"example_query_group1\"], \"query_concurrency\": 7}, {\"query_concurrency\": 5}]}"));
    ModifyClusterParameterGroupRequest request = new
ModifyClusterParameterGroupRequest()
    .withParameterGroupName(clusterParameterGroupName)
    .withParameters(parameters);
    client.modifyClusterParameterGroup(request);
}

private static void associateParameterGroupWithCluster() {

    ModifyClusterRequest request = new ModifyClusterRequest()
    .withClusterIdentifier(clusterIdentifier)
    .withClusterParameterGroupName(clusterParameterGroupName);

    Cluster result = client.modifyCluster(request);

    System.out.format("Parameter Group %s is used for Cluster %s\n",
                      clusterParameterGroupName,
                      result.getClusterParameterGroups().get(0).getParameterGroupName());
}
private static void
printResultClusterParameterGroups(DescribeClusterParameterGroupsResult result)
{
    if (result == null)
    {
        System.out.println("\nDescribe cluster parameter groups result is null.");
        return;
    }

    System.out.println("\nPrinting parameter group results:\n");
    for (ClusterParameterGroup group : result.getParameterGroups()) {
        System.out.format("\nDescription: %s\n", group.getDescription());
        System.out.format("Group Family Name: %s\n", group.getParameterGroupFamily());
        System.out.format("Group Name: %s\n", group.getParameterGroupName());

        describeClusterParameters(group.getParameterGroupName());
    }
}

private static void describeClusterParameters(String parameterGroupName) {
    DescribeClusterParametersRequest request = new DescribeClusterParametersRequest()
        .withParameterGroupName(parameterGroupName);
}

```

```
        DescribeClusterParametersResult result = client.describeClusterParameters(request);

        printResultClusterParameters(result, parameterGroupName);
    }

    private static void printResultClusterParameters(DescribeClusterParametersResult
result, String parameterGroupName)
{
    if (result == null)
    {
        System.out.println("\nCluster parameters is null.");
        return;
    }

    System.out.format("\nPrinting cluster parameters for \"%s\"\n",
parameterGroupName);
    for (Parameter parameter : result.getParameters()) {
        System.out.println("  Name: " + parameter.getParameterName() + ", Value: " +
parameter.getParameterValue());
        System.out.println("  DataType: " + parameter.getDataType() + ", "
MinEngineVersion: " + parameter.getMinimumEngineVersion());
        System.out.println("  AllowedValues: " + parameter.getAllowedValues() + ", "
Source: " + parameter.getSource());
        System.out.println("  IsModifiable: " + parameter.getIsModifiable() + ", "
Description: " + parameter.getDescription());
    }
}
}
```

Managing Parameter Groups Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage parameter groups.

- [create-cluster-parameter-group](#)
- [delete-cluster-parameter-group](#)
- [describe-cluster-parameters](#)
- [describe-cluster-parameter-groups](#)
- [describe-default-cluster-parameters](#)
- [modify-cluster-parameter-group](#)
- [reset-cluster-parameter-group](#)

You can use the following Amazon Redshift APIs to manage parameter groups.

- [CreateClusterParameterGroup](#)
- [DeleteClusterParameterGroup](#)
- [DescribeClusterParameters](#)
- [DescribeClusterParameterGroups](#)
- [DescribeDefaultClusterParameters](#)
- [ModifyClusterParameterGroup](#)
- [ResetClusterParameterGroup](#)

Amazon Redshift Snapshots

Topics

- [Overview \(p. 145\)](#)
- [Managing Snapshots Using the Console \(p. 151\)](#)
- [Managing Snapshots Using the AWS SDK for Java \(p. 159\)](#)
- [Managing Snapshots Using the Amazon Redshift CLI and API \(p. 161\)](#)

Overview

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: *automated* and *manual*. Amazon Redshift stores these snapshots internally in Amazon S3 by using an encrypted Secure Sockets Layer (SSL) connection. If you need to restore from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot that you specify.

When you restore from a snapshot, Amazon Redshift creates a new cluster and makes the new cluster available before all of the data is loaded, so you can begin querying the new cluster immediately. The cluster streams data on demand from the snapshot in response to active queries, then loads the remaining data in the background.

Amazon Redshift periodically takes incremental snapshots that track changes to the cluster since the previous snapshot. Amazon Redshift retains all of the data required to restore a cluster from a snapshot.

You can monitor the progress of snapshots by viewing the snapshot details in the AWS Management Console, or by calling [describe-cluster-snapshots](#) in the CLI or the [DescribeClusterSnapshots](#) API action. For an in-progress snapshot, these display information such as the size of the incremental snapshot, the transfer rate, the elapsed time, and the estimated time remaining.

Amazon Redshift provides free storage for snapshots that is equal to the storage capacity of your cluster until you delete the cluster. After you reach the free snapshot storage limit, you are charged for any additional storage at the normal rate. Because of this, you should evaluate how many days you need to keep automated snapshots and configure their retention period accordingly, and delete any manual snapshots that you no longer need. For pricing information, go to the Amazon Redshift [product detail page](#).

Automated Snapshots

When automated snapshots are enabled for a cluster, Amazon Redshift periodically takes snapshots of that cluster, usually every eight hours or following every 5 GB per node of data changes, or whichever comes first. Automated snapshots are enabled by default when you create a cluster. These snapshots are deleted at the end of a retention period. The default retention period is one day, but you can modify it by using the Amazon Redshift console or programmatically by using the Amazon Redshift API.

To disable automated snapshots, set the retention period to zero. If you disable automated snapshots, Amazon Redshift stops taking snapshots and deletes any existing automated snapshots for the cluster.

Only Amazon Redshift can delete an automated snapshot; you cannot delete them manually. Amazon Redshift deletes automated snapshots at the end of a snapshot's retention period, when you disable automated snapshots, or when you delete the cluster. [Amazon Redshift retains the latest automated snapshot until you disable automated snapshots or delete the cluster.](#)

If you want to keep an automated snapshot for a longer period, you can create a copy of it as a manual snapshot. The automated snapshot is retained until the end of retention period, but the corresponding manual snapshot is retained until you manually delete it.

Manual Snapshots

Regardless of whether you enable automated snapshots, you can take a manual snapshot whenever you want. Amazon Redshift will never automatically delete a manual snapshot. Manual snapshots are retained even after you delete your cluster.

Because manual snapshots accrue storage charges, it's important that you manually delete them if you no longer need them. If you delete a manual snapshot, you cannot start any new operations that reference that snapshot. However, if a restore operation is in progress, that restore operation will run to completion.

Amazon Redshift has a quota that limits the total number of manual snapshots that you can create; this quota is per AWS account per region. The default quota is listed at [AWS Service Limits](#).

Excluding Tables From Snapshots

By default, all user-defined permanent tables are included in snapshots. If a table, such as a staging table, doesn't need to be backed up, you can significantly reduce the time needed to create snapshots and restore from snapshots. You also reduce storage space on Amazon S3 by using a no-backup table. To create a no-backup table, include the BACKUP NO parameter when you create the table. For more information, see [CREATE TABLE](#) and [CREATE TABLE AS](#) in the *Amazon Redshift Database Developer Guide*.

Copying Snapshots to Another Region

You can configure Amazon Redshift to automatically copy snapshots (automated or manual) for a cluster to another region. When a snapshot is created in the cluster's primary region, it will be copied to a secondary region; these are known respectively as the *source region* and *destination region*. By storing a copy of your snapshots in another region, you have the ability to restore your cluster from recent data if anything affects the primary region. You can configure your cluster to copy snapshots to only one destination region at a time. For a list of Amazon Redshift regions, go to [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

When you enable Amazon Redshift to automatically copy snapshots to another region, you specify the destination region where you want snapshots to be copied. In the case of automated snapshots, you can also specify the retention period that they should be kept in the destination region. After an automated snapshot is copied to the destination region and it reaches the retention time period there, it is deleted from the destination region, keeping your snapshot usage low. You can change this retention period if you need to keep the automated snapshots for a shorter or longer period of time in the destination region.

The retention period that you set for automated snapshots that are copied to the destination region is separate from the retention period for automated snapshots in the source region. The default retention period for copied snapshots is seven days. That seven-day period only applies to automated snapshots. Manual snapshots are not affected by the retention period in either the source or destination regions, and they remain until you manually delete them.

You can disable automatic snapshot copy for a cluster at any time. When you disable this feature, snapshots are no longer copied from the source region to the destination region. Any automated snapshots copied to the destination region are deleted as they reach the retention period limit, unless you create manual snapshot copies of them. These manual snapshots, and any manual snapshots that were copied from the destination region, are retained in the destination region until you manually delete them.

If you want to change the destination region that you copy snapshots to, you have to first disable the automatic copy feature and then re-enable it, specifying the new destination region.

Copying snapshots across regions incurs data transfer charges. Once a snapshot is copied to the destination region, it becomes active and available for restoration purposes.

If you want to copy snapshots for AWS KMS-encrypted clusters to another region, you must create a grant for Amazon Redshift to use a AWS KMS customer master key (CMK) in the destination region. Then you must select that grant when you enable copying of snapshots in the source region. For more information about configuring snapshot copy grants, see [Copying AWS KMS-Encrypted Snapshots to Another Region \(p. 164\)](#).

Restoring a Cluster from a Snapshot

A snapshot contains data from any databases that are running on your cluster, and also information about your cluster, including the number of nodes, node type, and master user name. If you need to restore your cluster from a snapshot, Amazon Redshift uses the cluster information to create a new cluster and then restores all the databases from the snapshot data. The new cluster that Amazon Redshift creates from the snapshot will have same configuration, including the number and type of nodes, as the original cluster from which the snapshot was taken. The cluster is restored in the same region and a random, system-chosen Availability Zone, unless you specify another Availability Zone in your request.

You can monitor the progress of a restore by either calling the [DescribeClusters](#) API action, or viewing the cluster details in the AWS Management Console. For an in-progress restore, these display information such as the size of the snapshot data, the transfer rate, the elapsed time, and the estimated time remaining. For a description of these metrics, go to [RestoreStatus](#).

You cannot use a snapshot to revert an active cluster to a previous state.

Note

When you restore a snapshot into a new cluster, the default security group and parameter group are used unless you specify different values.

Restoring a Table from a Snapshot

You can restore a single table from a snapshot instead of restoring an entire cluster. When you restore a single table from a snapshot, you specify the source snapshot, database, schema, and table name, and the target cluster, schema, and a new table name for the restored table.

The new table name cannot be the name of an existing table. To replace an existing table with a restored table from a snapshot, rename or drop the existing table before you restore the table from the snapshot.

The target table is created using the source table's column definitions, table attributes, and column attributes except for foreign keys. To prevent conflicts due to dependencies, the target table doesn't inherit foreign keys from the source table. Any dependencies, such as views or permissions granted on the source table, are not applied to the target table.

If the owner of the source table exists, then that user is the owner of the restored table, provided that the user has sufficient permissions to become the owner of a relation in the specified database and schema. Otherwise, the restored table is owned by the master user that was created when the cluster was launched.

The restored table returns to the state it was in at the time the backup was taken. This includes transaction visibility rules defined by Redshift's adherence to [serializable isolation](#), meaning that data will be immediately visible to in flight transactions started after the backup.

Restoring a table from a snapshot has the following limitations:

- You can restore a table only to the current, active running cluster and from a snapshot that was taken of that cluster.
- You can restore only one table at a time.
- You cannot restore a table from a cluster snapshot that was taken prior to a cluster being resized.

To restore a table from a snapshot using the Amazon Redshift console

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. Choose **Clusters**.
3. Choose the **Table restore** tab.

Table	Database	Schema	Status
new_table	dev	public	PENDING
test_no_proxy	dev	public	SUCCEEDED

4. Choose **Restore table**.
5. In the **Table restore** panel, select a date range that contains the cluster snapshot that you want to restore from. For example, you might select `Last 1 Week` for cluster snapshots taken in the previous week.
6. Add the following information:
 - **From snapshot** – The identifier of the cluster snapshot that contains the table to restore from.
 - **Source table to restore from**
 - **Database** – The name of the database from the cluster snapshot that contains the table to restore from.
 - **Schema** – The name of the database schema from the cluster snapshot that contains the table to restore from.
 - **Table** – The name of the table from the cluster snapshot to restore from.
 - **Target table to restore to**
 - **Database** – The name of the database in the target cluster to restore the table to.
 - **Schema** – The name of the database schema in the target cluster to restore the table to.
 - **New table name** – The new name of the restored table. This name cannot be the name of an existing table in the target database.

Table Level Restore

Restore a table to cluster: restore-from-cluster

Select from snapshots in time range: Last 1 Week

From snapshot:*

Source table to restore from

Database:*

Schema:*

Table:*

Target table to restore to

Database:*

Schema:*

New table name: *

Cancel **Restore**

- Choose **Restore** to restore the table.

If you have restored at least one table from a cluster snapshot, you can copy the values from a previous table restore request into a new table restore request. This approach means you don't have to retype values that will be the same for several table restore operations.

To copy from a previous table restore request to a new table restore operation:

- In the **Table restore** tab, choose an existing table restore status.

Table	Database	Schema	Status
new_table	dev	public	SUCCEEDED

- Choose **Copy restore request**.

Example Example: Restoring a Table from a Snapshot Using the AWS CLI

The following example uses the `restore-table-from-cluster-snapshot` AWS CLI command to restore the `my-source-table` table from the `sample-database` schema in the `mysnapshot-id`. The example restores the snapshot to the `mycluster-example` cluster with a new table name of `my-new-table`.

```
aws redshift restore-table-from-cluster-snapshot --cluster-identifier mycluster-example
--new-table-name my-new-table
```

```
--snapshot-identifier my-snapshot-id
--source-database-name sample-database
--source-table-name my-source-table
```

Sharing Snapshots

You can share an existing manual snapshot with other AWS customer accounts by authorizing access to the snapshot. You can authorize up to 20 for each snapshot and 100 for each AWS Key Management Service (AWS KMS) key. That is, if you have 10 snapshots that are encrypted with a single KMS key, then you can authorize 10 AWS accounts to restore each snapshot, or other combinations that add up to 100 accounts and do not exceed 20 accounts for each snapshot. A person logged in as a user in one of the authorized accounts can then describe the snapshot or restore it to create a new Amazon Redshift cluster under their account. For example, if you use separate AWS customer accounts for production and test, a user can log on using the production account and share a snapshot with users in the test account. Someone logged on as a test account user can then restore the snapshot to create a new cluster that is owned by the test account for testing or diagnostic work.

A manual snapshot is permanently owned by the AWS customer account under which it was created. Only users in the account owning the snapshot can authorize other accounts to access the snapshot, or to revoke authorizations. Users in the authorized accounts can only describe or restore any snapshot that has been shared with them; they cannot copy or delete snapshots that have been shared with them. An authorization remains in effect until the snapshot owner revokes it. If an authorization is revoked, the previously authorized user loses visibility of the snapshot and cannot launch any new actions referencing the snapshot. If the account is in the process of restoring the snapshot when access is revoked, the restore runs to completion. You cannot delete a snapshot while it has active authorizations; you must first revoke all of the authorizations.

AWS customer accounts are always authorized to access snapshots owned by the account. Attempts to authorize or revoke access to the owner account will receive an error. You cannot restore or describe a snapshot that is owned by an inactive AWS customer account.

After you have authorized access to an AWS customer account, no IAM users in that account can perform any actions on the snapshot unless they have IAM policies that allow them to do so.

- IAM users in the snapshot owner account can authorize and revoke access to a snapshot only if they have an IAM policy that allows them to perform those actions with a resource specification that includes the snapshot. For example, the following policy allows a user in AWS account 012345678912 to authorize other accounts to access a snapshot named `my-snapshot20130829`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift:AuthorizeSnapshotAccess",
                "redshift:RevokeSnapshotAccess"
            ],
            "Resource": [
                "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snapshot20130829"
            ]
        }
    ]
}
```

- IAM users in an AWS account with which a snapshot has been shared cannot perform actions on that snapshot unless they have IAM policies allowing those actions:

- To list or describe a snapshot, they must have an IAM policy that allows the `DescribeClusterSnapshots` action. The following code shows an example:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "redshift:DescribeClusterSnapshots"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

- To restore a snapshot, users must have an IAM policy that allows the `RestoreFromClusterSnapshot` action and has a resource element that covers both the cluster they are attempting to create and the snapshot. For example, if a user in account 012345678912 has shared snapshot `my-snapshot20130829` with account 219876543210, in order to create a cluster by restoring the snapshot, a user in account 219876543210 must have a policy such as the following:

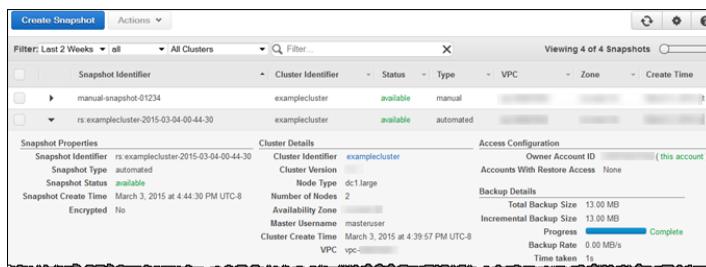
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "redshift:RestoreFromClusterSnapshot"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snapshot20130829",  
                "arn:aws:redshift:us-east-1:219876543210:cluster:from-another-account"  
            ]  
        }  
    ]  
}
```

- Once access to a snapshot has been revoked from an AWS account, no users in that account can access the snapshot, even if they have IAM policies that allow actions on the previously shared snapshot resource.

Managing Snapshots Using the Console

Amazon Redshift takes automatic, incremental snapshots of your data periodically and saves them to Amazon S3. Additionally, you can take manual snapshots of your data whenever you want. This section explains how to manage your snapshots from the Amazon Redshift console. For more information about snapshots, see [Amazon Redshift Snapshots \(p. 145\)](#).

All snapshot tasks in the Amazon Redshift console start from the snapshot list. You can filter the list by using the snapshot type, a time range, and the cluster associated with the snapshot. When you select an existing snapshot, the snapshot details are shown inline in the list, as shown in the example following. Depending on the snapshot type that you select, you will have different options available for working with the snapshot.

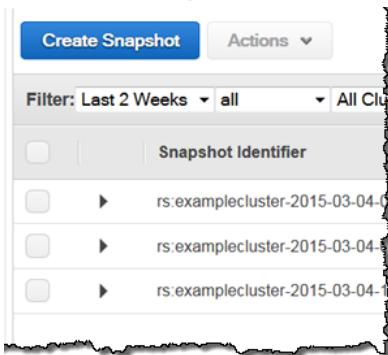


Creating a Manual Snapshot

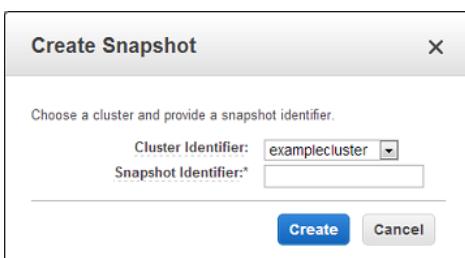
You can create a manual snapshot of a cluster from the snapshots list as follows. Or, you can take a snapshot of a cluster in the cluster configuration pane. For more information, see [Taking a Snapshot of a Cluster \(p. 31\)](#).

To create a manual snapshot

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, click **Snapshots**.
3. Click **Create Snapshot**.



4. In the **Create Snapshot** dialog box, do the following:
 - a. In the **Cluster Identifier** box, click the cluster that you want to take a snapshot of.
 - b. In the **Snapshot Identifier** box, type a name for the snapshot.



5. Click **Create**.

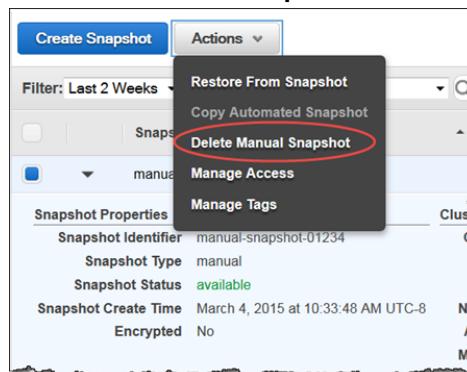
The snapshot might take some time to complete. The new snapshot is displayed in the list of snapshots with its current status. The example following shows that `examplecluster-manual-02-13-13` is in the process of being created.

Snapshot Identifier	Cluster	Status	Type
examplecluster-manual-02-13-13	examplecluster	creating	manual
cm:examplecluster-2013-01-31-20-24-31	examplecluster	available	automated

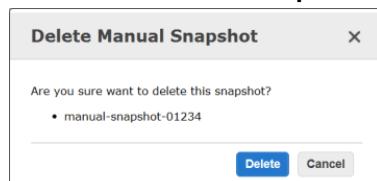
Deleting a Manual Snapshot

To delete a manual snapshot

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, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to delete, do any or all of the following:
 - In the **Time Range** box, click a time range that will narrow your search appropriately.
 - In the **Type** box, click **manual**.
 - In the **Cluster** box, click the cluster whose snapshot you want to delete.
4. In the snapshot list, click the row that contains the snapshot that you want to delete.
5. Click **Delete Manual Snapshot**.



6. In the **Delete Manual Snapshot** dialog box, click **Delete**.

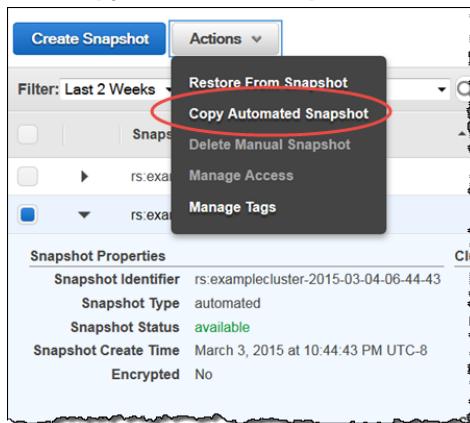


Copying an Automated Snapshot

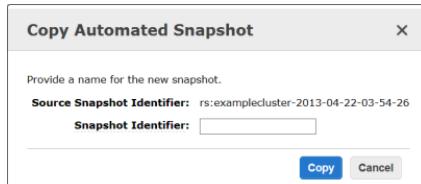
Automated snapshots are automatically deleted when their retention period expires, when you disable automated snapshots, or when you delete a cluster. If you want to keep an automated snapshot, you can copy it to a manual snapshot. Because Amazon Redshift never automatically deletes manual snapshots, you can keep this copy as long as you want.

To copy an automated snapshot

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, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to copy, do any or all of the following:
 - In the **Time Range** box, click a time range that will narrow your search appropriately.
 - In the **Type** box, click **automated**.
 - In the **Cluster** box, click the cluster whose snapshot you want to copy.
4. In the snapshot list, click the row of the snapshot that you want to copy.
5. Click **Copy Automated Snapshot**.



6. In the **Snapshot Identifier** box of the **Copy Automated Snapshot** dialog box, type a name for the snapshot copy.



7. Click **Copy**.

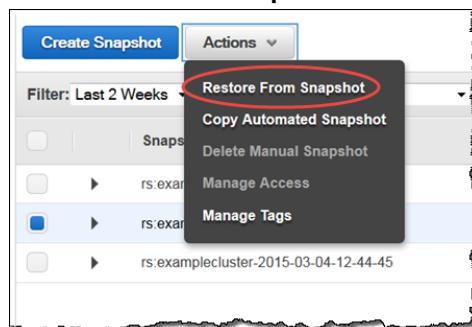
Restoring a Cluster from a Snapshot

When you restore a cluster from a snapshot, Amazon Redshift creates a new cluster with all the snapshot data on the new cluster.

To restore a cluster from a snapshot

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, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to use, do any or all of the following:
 - In the **Time Range** box, click a time range that will narrow your search appropriately.
 - In the **Type** box, click **manual** or **automated**.

- In the **Cluster** box, click the cluster whose snapshot you want to restore.
4. In the snapshot list, click the row that contains the snapshot that you want to use.
 5. Click **Restore From Snapshot**.



6. In the **Restore Cluster from Snapshot** dialog box, do the following:
 - a. In the **Cluster Identifier** box, type a cluster identifier for the restored cluster.

Cluster identifiers must meet the following conditions:

 - They must contain from 1 to 255 alphanumeric characters or hyphens.
 - Alphanumeric characters must be lowercase.
 - The first character must be a letter.
 - They cannot end with a hyphen or contain two consecutive hyphens.
 - They must be unique for all clusters within an AWS account.
 - b. In the **Port** box, accept the port from the snapshot or change the value as appropriate.
 - c. Select **Allow Version Upgrade** as appropriate.
 - d. In **Cluster Subnet Group**, select the subnet group into which you want to restore the cluster.

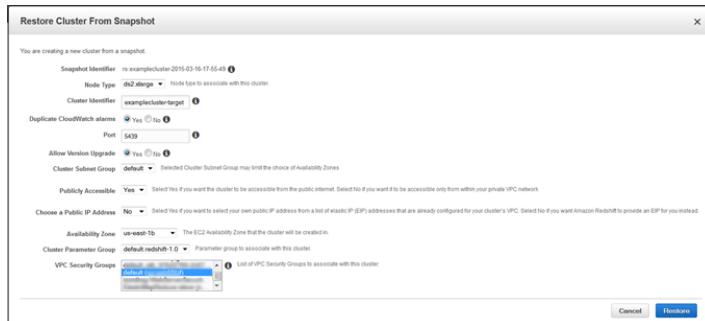
This option only appears if you restore the cluster into the EC2-VPC platform.
 - e. In **Publicly Accessible**, select **Yes** if you want the cluster to have a public IP address that can be accessed over a public connection to the Internet, and select **No** if you want the cluster to have a private IP address that can only be accessed from within the VPC. If your AWS account allows you to create EC2-Classic clusters, the default is **No**. Otherwise, the default is **Yes**.
 - f. In **Choose a Public IP Address**, select **Yes** if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select **No** to have Amazon Redshift create an EIP for your instance.

This option only appears if you restore the cluster into the EC2-VPC platform.
 - g. In **Elastic IP**, select an EIP to use to connect to the cluster from outside of the VPC.

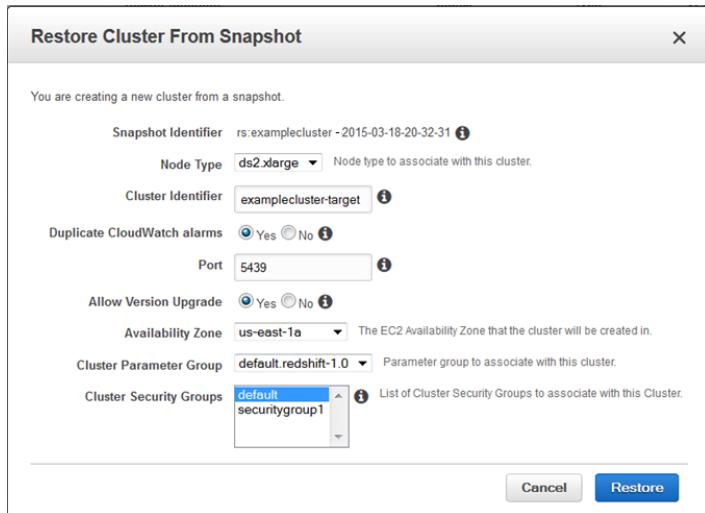
This option only appears if you restore the cluster into the EC2-VPC platform and you select **Yes** in **Choose a Public IP Address**.
 - h. In the **Availability Zone** box, accept the Availability Zone from the snapshot or change the value as appropriate.
 - i. In **Cluster Parameter Group**, select a parameter group to associate with the cluster.
 - j. In **Cluster Security Groups** or **VPC Security Groups**, select a security group to associate with the cluster. The types of security group that appear here depend on whether you're restoring the cluster into the EC2-Classic or EC2-VPC platform.

The option to select a cluster security group or a VPC security group depends on whether you restore the cluster into the EC2-Classic platform or the EC2-VPC platform.

The following is an example of restoring a snapshot into a cluster that uses the EC2-VPC platform.



The following is an example of restoring a snapshot into a cluster that uses the EC2-Classic platform.



7. Click **Restore**.

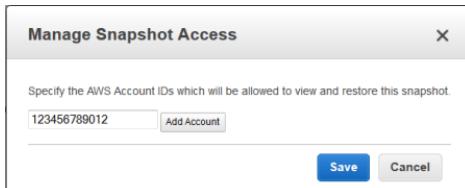
Sharing a Cluster Snapshot

You can authorize other users to access a manual snapshot you own, and you can later revoke that access when it is no longer required.

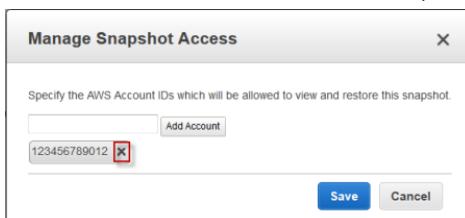
To share a cluster snapshot

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, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to share, do any or all of the following:
 - In the **Time Range** box, click a time range that will narrow your search appropriately.
 - In the **Cluster** box, click the cluster whose snapshot you want to share.
4. In the snapshot list, click the row that contains the snapshot that you want to use.
5. Click **Manage Access**.
6. In the **Manage Snapshot Access** dialog box, you can either authorize a user to access the snapshot or revoke a previously authorized access.

- To authorize a user to access the snapshot, type that user's 12-digit AWS account ID in the box (omit the dashes), and then click **Add Account**.



- To revoke the authorization for a user, click X beside that user's AWS account ID.



7. Click **Save** to save your changes, or **Cancel** to roll back the changes.

Configuring Cross-Region Snapshot Copy for a Non-Encrypted Cluster

You can configure Amazon Redshift to copy snapshots for a cluster to another region. To configure cross-region snapshot copy, you need to enable this copy feature for each cluster and configure where to copy snapshots and how long to keep copied automated snapshots in the destination region. When cross-region copy is enabled for a cluster, all new manual and automatic snapshots are copied to the specified region. Copied snapshot names are prefixed with **copy:**.

To configure cross-region snapshot copy for a non-encrypted cluster

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, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross-Region Snapshots**.
4. In the **Configure Cross-Region Snapshots** dialog box, for **Copy Snapshots** choose **Yes**.
5. In **Destination Region**, choose the region to which to copy snapshots.
6. In **Retention Period (days)**, choose the number of days for which you want automated snapshots to be retained in the destination region before they are deleted.
7. Click **Save**.

Configure Cross-Region Snapshot Copy for an AWS KMS-Encrypted Cluster

When you launch an Amazon Redshift cluster, you can choose to encrypt it with a master key from the AWS Key Management Service (AWS KMS). AWS KMS keys are specific to a region. If you want to enable cross-region snapshot copy for an AWS KMS-encrypted cluster, you must configure a *snapshot copy grant* for a master key in the destination region so that Amazon Redshift can perform encryption operations in the destination region. The following procedure describes the process of enabling cross-region snapshot

copy for an AWS KMS-encrypted cluster. For more information about encryption in Amazon Redshift and snapshot copy grants, see [Copying AWS KMS-Encrypted Snapshots to Another Region \(p. 164\)](#).

To configure cross-region snapshot copy for an AWS KMS-encrypted cluster

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation pane, click **Clusters**.
3. In the cluster list, choose a cluster name to open the **Configuration** view for the cluster.
4. Click **Backup**, and then click **Configure Cross-Region Snapshots**.
5. In the **Configure Cross-Region Snapshots** dialog box, for **Copy Snapshots** choose **Yes**.
6. In **Destination Region**, choose the region to which to copy snapshots.
7. In **Retention Period (days)**, choose the number of days for which you want automated snapshots to be retained in the destination region before they are deleted.
8. For **Existing Snapshot Copy Grant**, do one of the following:
 - a. Choose **No** to create a new snapshot copy grant. For **KMS Key**, choose the AWS KMS key for which to create the grant, and then type a name in **Snapshot Copy Grant Name**.
 - b. Choose **Yes** to choose an existing snapshot copy grant from the destination region. Then choose a grant from **Snapshot Copy Grant**.
9. Click **Save**.

Modifying the Retention Period for Cross-Region Snapshot Copy

After you configure cross-region snapshot copy, you might want to change the settings. You can easily change the retention period by selecting a new number of days and saving the changes.

Warning

You cannot modify the destination region after cross-region snapshot copy is configured. If you want to copy snapshots to a different region, you must first disable cross-region snapshot copy, and then re-enable it with a new destination region and retention period. Because any copied automated snapshots are deleted after you disable cross-region snapshot copy, you should determine if there are any that you want to keep and copy them to manual snapshots before disabling cross-region snapshot copy.

To modify the retention period for snapshots copied to a destination cluster

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, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross Region Snapshots**.
4. In the **Retention Period** box, select the new number of days that you want automated snapshots to be retained in the destination region.

If you select a smaller number of days to retain snapshots in the destination region, any automated snapshots that were taken before the new retention period will be deleted. If you select a larger number of days to retain snapshots in the destination region, the retention period for existing automated snapshots will be extended by the difference between the old value and the new value.

5. Click **Save Configuration**.

Disabling Cross-Region Snapshot Copy

You can disable cross-region snapshot copy for a cluster when you no longer want Amazon Redshift to copy snapshots to a destination region.

To disable cross-region snapshot copy 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. In the navigation pane, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross Region Snapshots** to open the **Configure Cross Region Snapshots** dialog box.
4. In the **Enable Cross Region Snapshots** box, click **No**.
5. Click **Save Configuration**.

Managing Snapshots Using the AWS SDK for Java

The following example demonstrates these common operations involving a snapshot:

- Creating a manual cluster snapshot of a cluster.
- Displaying information about all the snapshots of a cluster.
- Deleting manual snapshots of a cluster.

In this example, a snapshot of the cluster is initiated. When the snapshot is successfully created, all manual snapshots for the cluster that were created before the new snapshot are deleted. When creation of the manual snapshot is initiated, the snapshot is not immediately available. Therefore, this example uses a loop to poll for the status of the snapshot by calling the `describeClusterSnapshot` method. It normally takes a few moments for a snapshot to become available after initiation. For more information about snapshots, see [Amazon Redshift Snapshots \(p. 145\)](#).

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and provide a cluster identifier.

Example

```
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.CreateClusterSnapshotRequest;
import com.amazonaws.services.redshift.model.DeleteClusterSnapshotRequest;
import com.amazonaws.services.redshift.model.DescribeClusterSnapshotsRequest;
import com.amazonaws.services.redshift.model.DescribeClusterSnapshotsResult;
import com.amazonaws.services.redshift.model.Snapshot;

public class CreateAndDescribeSnapshot {

    public static AmazonRedshiftClient client;
    public static String clusterIdentifier = "***provide cluster identifier***";
    public static long sleepTime = 10;
```

```

public static void main(String[] args) throws IOException {
    AWSCredentials credentials = new PropertiesCredentials(
        CreateAndDescribeSnapshot.class
            .getResourceAsStream("AwsCredentials.properties"));

    client = new AmazonRedshiftClient(credentials);

    try {
        // Unique snapshot identifier
        String snapshotId = "my-snapshot-" + (new SimpleDateFormat("yyyy-MM-dd-HH-mm-
ss")).format(new Date());

        Date createDate = createManualSnapshot(snapshotId);
        waitForSnapshotAvailable(snapshotId);
        describeSnapshots();
        deleteManualSnapshotsBefore(createDate);
        describeSnapshots();

    } catch (Exception e) {
        System.err.println("Operation failed: " + e.getMessage());
    }
}

private static Date createManualSnapshot(String snapshotId) {

    CreateClusterSnapshotRequest request = new CreateClusterSnapshotRequest()
        .withClusterIdentifier(clusterIdentifier)
        .withSnapshotIdentifier(snapshotId);
    Snapshot snapshot = client.createClusterSnapshot(request);
    System.out.format("Created cluster snapshot: %s\n", snapshotId);
    return snapshot.getSnapshotCreateTime();
}

private static void describeSnapshots() {

    DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshotsRequest()
        .withClusterIdentifier(clusterIdentifier);
    DescribeClusterSnapshotsResult result = client.describeClusterSnapshots(request);

    printResultSnapshots(result);
}

private static void deleteManualSnapshotsBefore(Date creationDate) {

    DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshotsRequest()
        .withEndTime(creationDate)
        .withClusterIdentifier(clusterIdentifier)
        .withSnapshotType("manual");

    DescribeClusterSnapshotsResult result = client.describeClusterSnapshots(request);

    for (Snapshot s : result.getSnapshots()) {
        DeleteClusterSnapshotRequest deleteRequest = new DeleteClusterSnapshotRequest()
            .withSnapshotIdentifier(s.getSnapshotIdentifier());
        Snapshot deleteResult = client.deleteClusterSnapshot(deleteRequest);
        System.out.format("Deleted snapshot %s\n",
            deleteResult.getSnapshotIdentifier());
    }
}

private static void printResultSnapshots(DescribeClusterSnapshotsResult result) {
    System.out.println("\nSnapshot listing:");
    for (Snapshot snapshot : result.getSnapshots()) {
        System.out.format("Identifier: %s\n", snapshot.getSnapshotIdentifier());
        System.out.format("Snapshot type: %s\n", snapshot.getSnapshotType());
    }
}

```

```
        System.out.format("Snapshot create time: %s\n",
snapshot.getSnapshotCreateTime());
        System.out.format("Snapshot status: %s\n\n", snapshot.getStatus());
    }

    private static Boolean waitForSnapshotAvailable(String snapshotId) throws
InterruptedException {
    Boolean snapshotAvailable = false;
    System.out.println("Waiting for snapshot to become available.");
    while (!snapshotAvailable) {
        DescribeClusterSnapshotsResult result = client.describeClusterSnapshots(new
DescribeClusterSnapshotsRequest()
        .withSnapshotIdentifier(snapshotId));
        String status = (result.getSnapshots()).get(0).getStatus();
        if (status.equalsIgnoreCase("available")) {
            snapshotAvailable = true;
        }
        else {
            System.out.print(".");
            Thread.sleep(sleepTime*1000);
        }
    }
    return snapshotAvailable;
}

}
```

Managing Snapshots Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage snapshots.

- [authorize-snapshot-access](#)
- [copy-cluster-snapshot](#)
- [create-cluster-snapshot](#)
- [delete-cluster-snapshot](#)
- [describe-cluster-snapshots](#)
- [disable-snapshot-copy](#)
- [enable-snapshot-copy](#)
- [modify-snapshot-copy-retention-period](#)
- [restore-from-cluster-snapshot](#)
- [revoke-snapshot-access](#)

You can use the following Amazon Redshift API actions to manage snapshots.

- [AuthorizeSnapshotAccess](#)
- [CopyClusterSnapshot](#)
- [CreateClusterSnapshot](#)
- [DeleteClusterSnapshot](#)
- [DescribeClusterSnapshots](#)
- [DisableSnapshotCopy](#)
- [EnableSnapshotCopy](#)

- [ModifySnapshotCopyRetentionPeriod](#)
- [RestoreFromClusterSnapshot](#)
- [RevokeSnapshotAccess](#)

For more information about Amazon Redshift snapshots, see [Amazon Redshift Snapshots \(p. 145\)](#).

Amazon Redshift Database Encryption

In Amazon Redshift, you can enable database encryption for your clusters to help protect data at rest. When you enable encryption for a cluster, the data blocks and system metadata are encrypted for the cluster and its snapshots.

Encryption is an optional, immutable setting of a cluster. If you want encryption, you enable it during the cluster launch process. To go from an unencrypted cluster to an encrypted cluster or the other way around, unload your data from the existing cluster and reload it in a new cluster with the chosen encryption setting. For more information, see [Migrating an Unencrypted Cluster to an Encrypted Cluster \(p. 167\)](#)

Though encryption is an optional setting in Amazon Redshift, we recommend enabling it for clusters that contain sensitive data. Additionally, you might be required to use encryption depending on the guidelines or regulations that govern your data. For example, the Payment Card Industry Data Security Standard (PCI DSS), the Sarbanes-Oxley Act (SOX), the Health Insurance Portability and Accountability Act (HIPAA), and other such regulations provide guidelines for handling specific types of data.

Amazon Redshift uses a hierarchy of encryption keys to encrypt the database. You can use either AWS Key Management Service (AWS KMS) or a hardware security module (HSM) to manage the top-level encryption keys in this hierarchy. The process that Amazon Redshift uses for encryption differs depending on how you manage keys.

Additionally, Amazon Redshift automatically integrates with AWS KMS but not with an HSM. When you use an HSM, you must use client and server certificates to configure a trusted connection between Amazon Redshift and your HSM.

Database Encryption for Amazon Redshift Using AWS KMS

When you choose AWS KMS for key management with Amazon Redshift, there is a four-tier hierarchy of encryption keys. These keys, in hierarchical order, are the master key, a cluster encryption key (CEK), a database encryption key (DEK), and data encryption keys.

When you launch your cluster, Amazon Redshift returns a list of the customer master keys (CMKs) that your AWS account has created or has permission to use in AWS KMS. You select a CMK to use as your master key in the encryption hierarchy.

By default, Amazon Redshift selects your default key as the master key. Your default key is an AWS-managed key that is created for your AWS account to use in Amazon Redshift. AWS KMS creates this key the first time you launch an encrypted cluster in a region and choose the default key.

If you don't want to use the default key, you must have (or create) a customer-managed CMK separately in AWS KMS before you launch your cluster in Amazon Redshift. Customer-managed CMKs give you more flexibility, including the ability to create, rotate, disable, define access control for, and audit the encryption keys used to help protect your data. For more information about creating CMKs, go to [Creating Keys in the AWS Key Management Service Developer Guide](#).

If you want to use a AWS KMS key from another AWS account, you must have permission to use the key and specify its ARN in Amazon Redshift. For more information about access to keys in AWS KMS, go to [Controlling Access to Your Keys in the AWS Key Management Service Developer Guide](#).

After you choose a master key, Amazon Redshift requests that AWS KMS generate a data key and encrypt it using the selected master key. This data key is used as the CEK in Amazon Redshift. AWS KMS exports the encrypted CEK to Amazon Redshift, where it is stored internally on disk in a separate network from the cluster along with the grant to the CMK and the encryption context for the CEK. Only the encrypted CEK is exported to Amazon Redshift; the CMK remains in AWS KMS. Amazon Redshift also passes the encrypted CEK over a secure channel to the cluster and loads it into memory. Then, Amazon Redshift calls AWS KMS to decrypt the CEK and loads the decrypted CEK into memory. For more information about grants, encryption context, and other AWS KMS-related concepts, go to [Concepts](#) in the *AWS Key Management Service Developer Guide*.

Next, Amazon Redshift randomly generates a key to use as the DEK and loads it into memory in the cluster. The decrypted CEK is used to encrypt the DEK, which is then passed over a secure channel from the cluster to be stored internally by Amazon Redshift on disk in a separate network from the cluster. Like the CEK, both the encrypted and decrypted versions of the DEK are loaded into memory in the cluster. The decrypted version of the DEK is then used to encrypt the individual encryption keys that are randomly generated for each data block in the database.

When the cluster reboots, Amazon Redshift starts with the internally stored, encrypted versions of the CEK and DEK, reloads them into memory, and then calls AWS KMS to decrypt the CEK with the CMK again so it can be loaded into memory. The decrypted CEK is then used to decrypt the DEK again, and the decrypted DEK is loaded into memory and used to encrypt and decrypt the data block keys as needed.

For more information about creating Amazon Redshift clusters that are encrypted with AWS KMS keys, see [Creating a Cluster \(p. 17\)](#) and [Manage Clusters Using the Amazon Redshift CLI and API \(p. 35\)](#).

Copying AWS KMS-Encrypted Snapshots to Another Region

AWS KMS keys are specific to a region. If you enable copying of Amazon Redshift snapshots to another region, and the source cluster and its snapshots are encrypted using a master key from AWS KMS, you need to configure a grant for Amazon Redshift to use a master key in the destination region. This grant enables Amazon Redshift to encrypt snapshots in the destination region. For more information cross-region snapshot copy, see [Copying Snapshots to Another Region \(p. 146\)](#).

Note

If you enable copying of snapshots from an encrypted cluster and use AWS KMS for your master key, you cannot rename your cluster because the cluster name is part of the encryption context.

If you must rename your cluster, you can disable copying of snapshots in the source region, rename the cluster, and then configure and enable copying of snapshots again.

The process to configure the grant for copying snapshots is as follows.

1. In the destination region, create a snapshot copy grant by doing the following:
 - If you do not already have an AWS KMS key to use, create one. For more information about creating AWS KMS keys, go to [Creating Keys](#) in the *AWS Key Management Service Developer Guide*.
 - Specify a name for the snapshot copy grant. This name must be unique in that region for your AWS account.
 - Specify the AWS KMS key ID for which you are creating the grant. If you do not specify a key ID, the grant applies to your default key.
2. In the source region, enable copying of snapshots and specify the name of the snapshot copy grant that you created in the destination region.

This preceding process is only necessary if you enable copying of snapshots using the AWS CLI, the Amazon Redshift API, or SDKs. If you use the console, Amazon Redshift provides the proper workflow

to configure the grant when you enable cross-region snapshot copy. For more information about configuring cross-region snapshot copy for AWS KMS-encrypted clusters by using the console, see [Configure Cross-Region Snapshot Copy for an AWS KMS-Encrypted Cluster \(p. 157\)](#).

Before the snapshot is copied to the destination region, Amazon Redshift decrypts the snapshot using the master key in the source region and re-encrypts it temporarily using a randomly generated RSA key that Amazon Redshift manages internally. Amazon Redshift then copies the snapshot over a secure channel to the destination region, decrypts the snapshot using the internally managed RSA key, and then re-encrypts the snapshot using the master key in the destination region.

For more information about configuring snapshot copy grants for AWS KMS-encrypted clusters, see [Configuring Amazon Redshift to Use AWS KMS Encryption Keys Using the Amazon Redshift API and AWS CLI \(p. 174\)](#).

Encryption for Amazon Redshift Using Hardware Security Modules

If you don't use AWS KMS for key management, you can use a hardware security module (HSM) for key management with Amazon Redshift.

Important

HSM encryption is not supported for DC2 node types.

HSMs are devices that provide direct control of key generation and management. They provide greater security by separating key management from the application and database layers. Amazon Redshift supports AWS CloudHSM Classic for key management. The encryption process is different when you use HSM to manage your encryption keys instead of AWS KMS.

Important

Amazon Redshift supports only AWS CloudHSM Classic. We don't support the newer AWS CloudHSM service. AWS CloudHSM Classic isn't available in all regions. For more information about available regions, see [Region Table](#).

When you configure your cluster to use an HSM, Amazon Redshift sends a request to the HSM to generate and store a key to be used as the CEK. However, unlike AWS KMS, the HSM doesn't export the CEK to Amazon Redshift. Instead, Amazon Redshift randomly generates the DEK in the cluster and passes it to the HSM to be encrypted by the CEK. The HSM returns the encrypted DEK to Amazon Redshift, where it is further encrypted using a randomly-generated, internal master key and stored internally on disk in a separate network from the cluster. Amazon Redshift also loads the decrypted version of the DEK in memory in the cluster so that the DEK can be used to encrypt and decrypt the individual keys for the data blocks.

If the cluster is rebooted, Amazon Redshift decrypts the internally-stored, double-encrypted DEK using the internal master key to return the internally stored DEK to the CEK-encrypted state. The CEK-encrypted DEK is then passed to the HSM to be decrypted and passed back to Amazon Redshift, where it can be loaded in memory again for use with the individual data block keys.

Configuring a Trusted Connection Between Amazon Redshift and an HSM

When you opt to use an HSM for management of your cluster key, you need to configure a trusted network link between Amazon Redshift and your HSM. Doing this requires configuration of client and server certificates. The trusted connection is used to pass the encryption keys between the HSM and Amazon Redshift during encryption and decryption operations.

Amazon Redshift creates a public client certificate from a randomly generated private and public key pair. These are encrypted and stored internally. You download and register the public client certificate in your HSM, and assign it to the applicable HSM partition.

You provide Amazon Redshift with the HSM IP address, HSM partition name, HSM partition password, and a public HSM server certificate, which is encrypted by using an internal master key. Amazon Redshift completes the configuration process and verifies that it can connect to the HSM. If it cannot, the cluster is put into the INCOMPATIBLE_HSM state and the cluster is not created. In this case, you must delete the incomplete cluster and try again.

Important

When you modify your cluster to use a different HSM partition, Amazon Redshift verifies that it can connect to the new partition, but it does not verify that a valid encryption key exists. Before you use the new partition, you must replicate your keys to the new partition. If the cluster is restarted and Amazon Redshift cannot find a valid key, the restart fails. For more information, see [Replicating Keys Across HSMs](#).

For more information about configuring Amazon Redshift to use an HSM, see [Configuring Amazon Redshift to Use an HSM Using the Amazon Redshift console \(p. 169\)](#) and [Configuring Amazon Redshift to use an HSM Using the Amazon Redshift API and AWS CLI \(p. 174\)](#).

After initial configuration, if Amazon Redshift fails to connect to the HSM, an event is logged. For more information about these events, see [Amazon Redshift Event Notifications](#).

About Rotating Encryption Keys in Amazon Redshift

In Amazon Redshift, you can rotate encryption keys for encrypted clusters. When you start the key rotation process, Amazon Redshift rotates the CEK for the specified cluster and for any automated or manual snapshots of the cluster. Amazon Redshift also rotates the DEK for the specified cluster, but cannot rotate the DEK for the snapshots while they are stored internally in Amazon Simple Storage Service (Amazon S3) and encrypted using the existing DEK.

While the rotation is in progress, the cluster is put into a ROTATING_KEYS state until completion, at which time the cluster returns to the AVAILABLE state. Amazon Redshift handles decryption and re-encryption during the key rotation process.

Note

You cannot rotate keys for snapshots without a source cluster. Before you delete a cluster, consider whether its snapshots rely on key rotation.

Because the cluster is momentarily unavailable during the key rotation process, you should rotate keys only as often as your data needs require or when you suspect the keys might have been compromised. As a best practice, you should review the type of data that you store and plan how often to rotate the keys that encrypt that data. The frequency for rotating keys varies depending on your corporate policies for data security, and any industry standards regarding sensitive data and regulatory compliance. Ensure that your plan balances security needs with availability considerations for your cluster.

For more information about rotating keys, see [Rotating Encryption Keys Using the Amazon Redshift console \(p. 173\)](#) and [Rotating Encryption Keys Using the Amazon Redshift API and AWS CLI \(p. 175\)](#).

Migrating an Unencrypted Cluster to an Encrypted Cluster

You enable encryption when you launch a cluster. To migrate from an unencrypted cluster to an encrypted cluster, you first unload your data from the existing, source cluster. Then you reload the data in a new, target cluster with the chosen encryption setting. For more information about launching an encrypted cluster, see [Amazon Redshift Database Encryption \(p. 163\)](#). During the migration process, your source cluster is available for read-only queries until the last step. The last step is to rename the target and source clusters, which switches endpoints so all traffic is routed to the new, target cluster. The target cluster is unavailable until you reboot following the rename. Suspend all data loads and other write operations on the source cluster while data is being transferred.

To prepare for migration

1. Identify all the dependent systems that interact with Amazon Redshift, for example business intelligence (BI) tools and extract, transform, and load (ETL) systems.
2. Identify validation queries to test the migration.

For example, you can use the following query to find the number of user-defined tables.

```
select count(*)
from pg_table_def
where schemaname != 'pg_catalog';
```

The following query returns a list of all user-defined tables and the number of rows in each table.

```
select "table", tbl_rows
from svv_table_info;
```

3. Choose a good time for your migration. To find a time when cluster usage is lowest, monitor cluster metrics such as CPU utilization and number of database connections. For more information, see [Viewing Cluster Performance Data \(p. 261\)](#).
4. Drop unused tables.

To create a list of tables and the number of the times each table has been queried, run the following query.

```
select database,
schema,
table_id,
"table",
round(size::float/(1024*1024)::float,2) as size,
sortkey1,
nvl(s.num_qs,0) num_qs
from svv_table_info t
left join (select tbl,
perm_table_name,
count(distinct query) num_qs
from stl_scan s
where s.userid > 1
and s.perm_table_name not in ('internal worktable','s3')
group by tbl,
perm_table_name) s on s.tbl = t.table_id
where t."schema" not in ('pg_internal');
```

5. Launch a new, encrypted cluster.

Use the same port number for the target cluster as for the source cluster. For more information about launching an encrypted cluster, see [Amazon Redshift Database Encryption \(p. 163\)](#).

6. Set up the unload and load process.

You can use the [Amazon Redshift Unload/Copy Utility](#) to help you to migrate data between clusters. The utility exports data from the source cluster to a location on Amazon S3. The data is encrypted with AWS KMS. The utility then automatically imports the data into the target. Optionally, you can use the utility to clean up Amazon S3 after migration is complete.

7. Run a test to verify your process and estimate how long write operations must be suspended.

During the unload and load operations, maintain data consistency by suspending data loads and other write operations. Using one of your largest tables, run through the unload and load process to help you estimate timing.

8. Create database objects, such as schemas, views, and tables. To help you generate the necessary data definition language (DDL) statements, you can use the scripts in [AdminViews](#) in the AWS GitHub repository.

To migrate your cluster:

1. Stop all ETL processes on the source cluster.

To confirm that there are no write operations in process, use the Amazon Redshift Management Console to monitor write IOPS. For more information, see [Viewing Cluster Performance Data \(p. 261\)](#).

2. Run the validation queries you identified earlier to collect information about the unencrypted source cluster before migration.
3. (Optional) Create one workload management (WLM) queue to use the maximum available resources in both the source and target cluster. For example, create a queue named `data_migrate` and configure the queue with memory of 95 percent and concurrency of 4. For more information, see [Routing Queries to Queues Based on User Groups and Query Groups](#) in the *Amazon Redshift Database Developer Guide*.
4. Using the `data_migrate` queue, run the `UnloadCopyUtility`.
Monitor the UNLOAD and COPY process using the Amazon Redshift Console.
5. Run the validation queries again and verify that the results match the results from the source cluster.
6. Rename your source and target clusters to swap the endpoints. For more information, see [Step 6: Rename the Source and Target Clusters \(p. 320\)](#). To avoid disruption, perform this operation outside of business hours.
7. Verify that you can connect to the target cluster using all of your SQL clients, such as ETL and reporting tools.
8. Shut down the unencrypted source cluster.

Configuring Database Encryption Using the Console

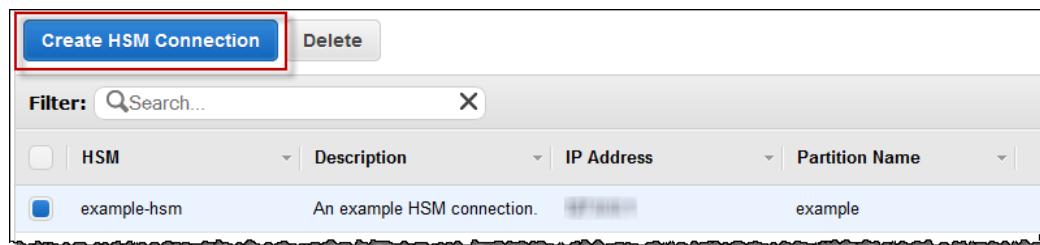
You can use the Amazon Redshift console to configure Amazon Redshift to use an HSM and to rotate encryption keys. For information about how to create clusters using AWS KMS encryption keys, see [Creating a Cluster \(p. 17\)](#) and [Manage Clusters Using the Amazon Redshift CLI and API \(p. 35\)](#).

Configuring Amazon Redshift to Use an HSM Using the Amazon Redshift console

You can use the following procedures to specify HSM connection and configuration information for Amazon Redshift by using the Amazon Redshift console.

To create an HSM Connection

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the left navigation pane, click **Security**, and then click the **HSM Connections** tab.
3. Click **Create HSM Connection**.



4. On the **Create HSM Connection** page, type the following information:
 - a. In the **HSM Connection Name** box, type a name to identify this connection.
 - b. In the **Description** box, type a description about the connection.
 - c. In the **HSM IP Address** box, type the IP address for your HSM.
 - d. In the **HSM Partition Name** box, type the name of the partition that Amazon Redshift should connect to.
 - e. In the **HSM Partition Password** box, type the password that is required to connect to the HSM partition.
 - f. Copy the public server certificate from your HSM and paste it in the **Paste the HSM's public server certificate here** box.
 - g. Click **Create**.

Create HSM Connection

Provide an identifier and description to designate this connection:

HSM Connection Name* i

Description* i

Enter the details required for connecting to the HSM below. Please note: only SafeNet HSMs are supported.

HSM IP Address* i

HSM Partition Name* i

HSM Partition Password* i

Confirm HSM Partition Password*

Paste the HSM's public server certificate here* i

*Required Cancel Create

5. After the connection is created, you can create an HSM client certificate. If you want to create an HSM client certificate immediately after creating the connection, click **Yes** and complete the steps in the next procedure. Otherwise, click **Not now** to return to the list of HSM connections and complete the remainder of the process at another time.

Create HSM Connection

✓ Your HSM Connection **example-hsm** was successfully created.

In order to configure a cluster to use this HSM configuration, you must also create an HSM client certificate. You can do this now or you may do it later (if you have not done so already).

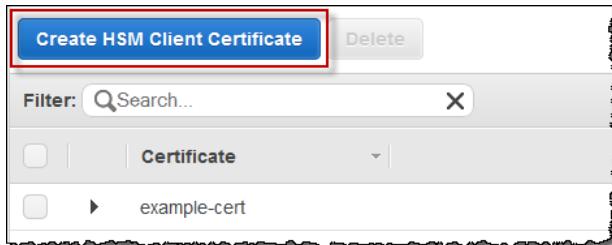
Would you like to create an HSM client certificate now?

Yes Not now

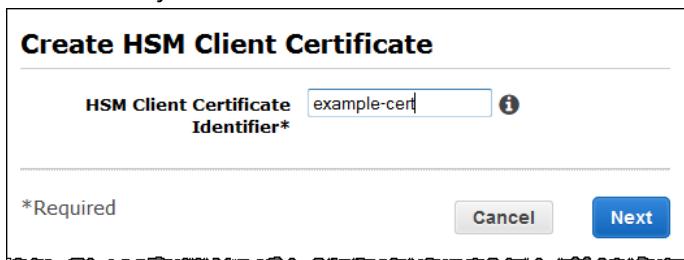
*Required Cancel Close

To create an HSM client certificate

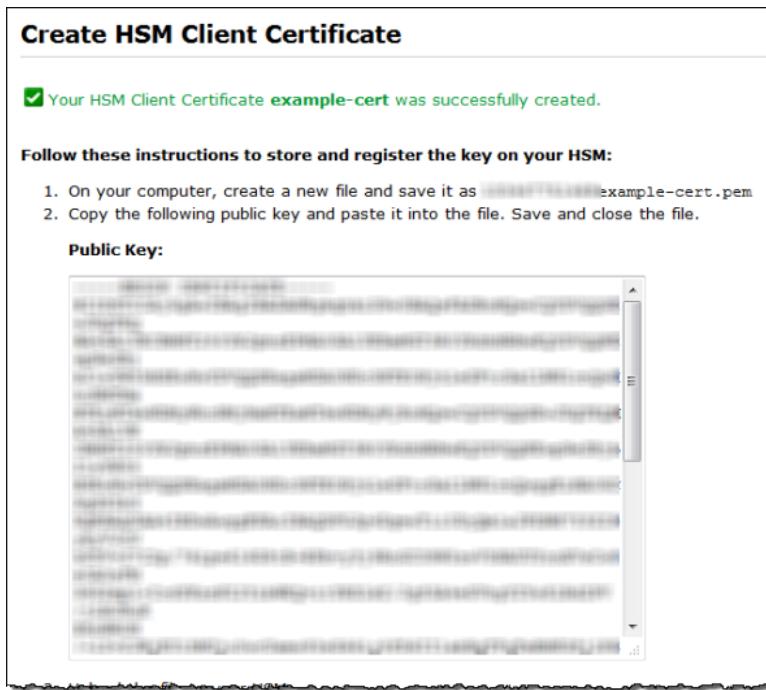
1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the left navigation pane, click **Security**, and then click the **HSM Certificates** tab.
3. Click **Create HSM Client Certificate**.



4. On the **Create HSM Client Certificate** page, type a name in the **HSM Client Certificate Identifier** box to identify this client certificate.



5. Click **Next**.
6. After the certificate is created, a confirmation page appears with information to register the key on your HSM. If you do not have permission to configure the HSM, coordinate the following steps with an HSM administrator.



- a. On your computer, open a new text file.
- b. In the Amazon Redshift console, on the **Create HSM Client Certificate** confirmation page, copy the public key.
- c. Paste the public key into the open file and save it with the file name displayed in step 1 from the confirmation page. Make sure that you save the file with the .pem file extension, for example: 123456789mykey.pem.

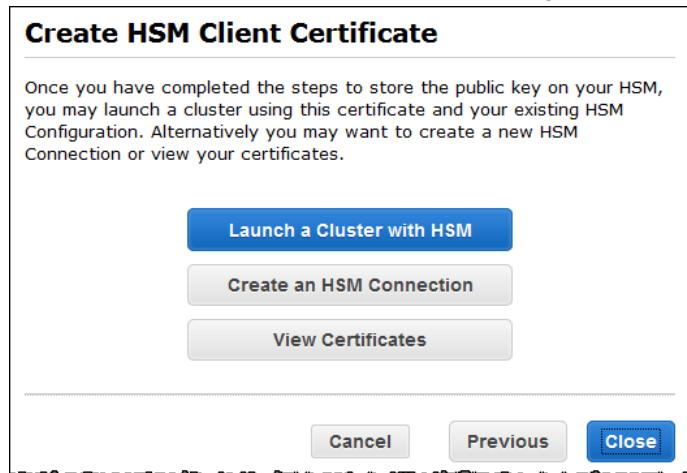
- d. Upload the .pem file to your HSM.
- e. On the HSM, open a command-prompt window and run the commands listed in step 4 on the confirmation page to register the key. The command uses the following format, with *ClientName*, *KeyFilename*, and *PartitionName* being values you need to replace with your own:

```
client register -client ClientName -hostname KeyFilename  
client assignPartition -client ClientName -partition PartitionName
```

For example:

```
client register -client MyClient -hostname 123456789mykey  
client assignPartition -client MyClient -partition MyPartition
```

- f. After you register the key on the HSM, click **Next**.
7. After the HSM client certificate is created and registered, click one of the following buttons:



- **Launch a Cluster with HSM.** This option starts the process of launching a new cluster. During the process, you can select an HSM to store encryption keys. For more information about the launch cluster process, see [Managing Clusters Using the Console \(p. 15\)](#).

Create an HSM Connection. This option starts the [Create HSM Connection](#) process.

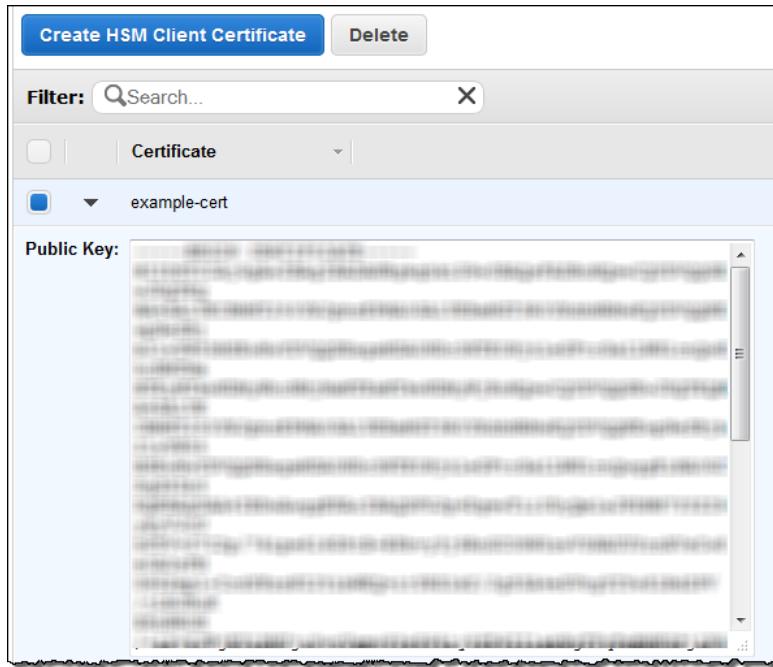
View Certificates. This option returns you to **HSM** in the navigation pane and displays a list of client certificates on the **Certificates** tab.

Previous. This option returns you to the [Create HSM Client Certificates](#) confirmation page.

Close. This option returns you to **HSM** in the navigation pane and displays a list of HSM connections on the **Connections** tab.

To display the public key for an HSM client certificate

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, click **Security**, and then click the **HSM Certificates** tab.
3. Click the HSM client certificate to display the public key. This key is the same one that you added to the HSM in the procedure preceding procedure, [To create an HSM client certificate \(p. 170\)](#)



To delete an HSM connection

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the left navigation pane, click **Security**, and then click the **HSM Connections** tab.
3. Click the HSM connection that you want to delete.
4. In the **Delete HSM Connection** dialog box, click **Delete** to delete the connection from Amazon Redshift, or click **Cancel** to return to the **HSM Connections** tab without deleting the connection.

To delete an HSM client certificate

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, click **Security** and select the **HSM Certificates** tab.
3. In the list, click the HSM client certificate that you want to delete.
4. In the **Delete HSM Client Certificate** dialog box, click **Delete** to delete the certificate from Amazon Redshift, or click **Cancel** to return to the **Certificates** tab without deleting the certificate.

Rotating Encryption Keys Using the Amazon Redshift console

You can use the following procedure to rotate encryption keys by using the Amazon Redshift console.

To rotate an encryption key

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, click **Clusters**.

3. In the list, click the cluster for which you want to rotate keys.
4. Click **Database**, and then click **Rotate Encryption Keys**.
5. Click **Yes, Rotate Keys** if you want to rotate the keys or **Cancel** if you do not.

Note

Your cluster will be momentarily unavailable until the key rotation process completes.

Configuring Database Encryption Using the Amazon Redshift API and AWS CLI

Use the Amazon Redshift API and AWS Command Line Interface (AWS CLI) to configure encryption key options for Amazon Redshift databases. For more information about database encryption, see [Amazon Redshift Database Encryption \(p. 163\)](#).

Configuring Amazon Redshift to Use AWS KMS Encryption Keys Using the Amazon Redshift API and AWS CLI

You can use the following Amazon Redshift API actions to configure Amazon Redshift to use AWS KMS encryption keys.

- [CreateCluster](#)
- [CreateSnapshotCopyGrant](#)
- [DescribeSnapshotCopyGrants](#)
- [DeleteSnapshotCopyGrant](#)
- [DisableSnapshotCopy](#)
- [EnableSnapshotCopy](#)

You can use the following Amazon Redshift CLI operations to configure Amazon Redshift to use AWS KMS encryption keys.

- `create-cluster`
- `create-snapshot-copy-grant`
- `describe-snapshot-copy-grants`
- `delete-snapshot-copy-grant`
- `disable-snapshot-copy`
- `enable-snapshot-copy`

Configuring Amazon Redshift to use an HSM Using the Amazon Redshift API and AWS CLI

You can use the following Amazon Redshift API actions to manage hardware security modules.

- [CreateHsmClientCertificate](#)
- [CreateHsmConfiguration](#)
- [DeleteHsmClientCertificate](#)

- [DeleteHsmConfiguration](#)
- [DescribeHsmClientCertificates](#)
- [DescribeHsmConfigurations](#)

You can use the following AWS CLI operations to manage hardware security modules.

- [create-hsm-client-certificate](#)
- [create-hsm-configuration](#)
- [delete-hsm-client-certificate](#)
- [delete-hsm-configuration](#)
- [describe-hsm-client-certificates](#)
- [describe-hsm-configurations](#)

Rotating Encryption Keys Using the Amazon Redshift API and AWS CLI

You can use the following Amazon Redshift API actions to rotate encryption keys.

- [RotateEncryptionKey](#)

You can use the following AWS CLI operations to rotate encryption keys.

- [rotate-encryption-key](#)

Purchasing Amazon Redshift Reserved Nodes

Overview

In AWS, the charges that you accrue for using Amazon Redshift are based on compute nodes. Each compute node is billed at an hourly rate. The hourly rate varies depending on factors such as region, node type, and whether the node receives on-demand node pricing or reserved node pricing.

On-demand node pricing is the most expensive, but most flexible option in Amazon Redshift. With on-demand rates, you are charged only for compute nodes that you have in a running cluster. If you shut down or delete a cluster, you are no longer charged for compute nodes that were in that cluster. You are billed only for the compute nodes that you use, and no more. The hourly rate that you are charged for each compute node varies depending on factors such as region and node type.

Reserved node pricing is less expensive than on-demand pricing because compute nodes are billed at discounted hourly rates. However, to receive these discounted rates, you must purchase reserved node offerings. When you purchase an offering, you make a reservation. The reservation sets a discounted rate for each node that you reserve for the duration of the reservation. The discounted rate in an offering varies depending on factors such as the region, node type, duration, and payment option.

This topic discusses what reserved node offerings are and how you can purchase them to reduce the cost of running your Amazon Redshift clusters. This topic discusses rates in general terms as on-demand or discounted so you can understand pricing concepts and how pricing affects billing. For more information about specific rates, go to [Amazon Redshift Pricing](#).

About Reserved Node Offerings

If you intend to keep your Amazon Redshift cluster running continuously for a prolonged period, you should consider purchasing reserved node offerings. These offerings provide significant savings over on-demand pricing, but they require you to reserve compute nodes and commit to paying for those nodes for either a one-year or three-year duration.

Reserved nodes are a billing concept that is used strictly to determine the rate at which you are charged for nodes. Reserving a node does not actually create any nodes for you. You are charged for reserved nodes regardless of usage, which means that you must pay for each node that you reserve for the duration of the reservation, whether or not you have any nodes in a running cluster to which the discounted rate applies.

In the evaluation phase of your project or when you're developing a proof of concept, on-demand pricing gives you the flexibility to pay as you go, to pay only for what you use, and to stop paying at any time by shutting down or deleting clusters. After you have established the needs of your production environment and begin the implementation phase, you should consider reserving compute nodes by purchasing one or more offerings.

An offering can apply to one or more compute nodes. You specify the number of compute nodes to reserve when you purchase the offering. You might choose to purchase one offering for multiple compute nodes, or you might choose to purchase multiple offerings and specify a certain number of compute nodes in each offering.

For example, any of the following are valid ways to purchase an offering for three compute nodes:

- Purchase one offering and specify three compute nodes.
- Purchase two offerings, and specify one compute node for the first offering and two compute nodes for the second offering.
- Purchase three offerings, and specify one compute node for each of the offerings.

Comparing Pricing Among Reserved Node Offerings

Amazon Redshift provides several payment options for offerings. The payment option that you choose affects the payment schedule and the discounted rate that you are charged for the reservation. The more that you pay upfront for the reservation, the better the overall savings are.

The following payment options are available for offerings. The offerings are listed in order from least to most savings over on-demand rates.

Note

You are charged the applicable hourly rate for every hour in the specified duration of the reservation, regardless of whether you use the reserved node or not. The payment option just determines the frequency of payments and the discount to be applied. For more information, see [About Reserved Node Offerings \(p. 176\)](#).

Comparing Reserved Node Offerings

Payment Option	Payment Schedule	Comparative Savings	Duration	Upfront Charges	Recurring Monthly Charges
No Upfront	Monthly installments for the duration of the reservation. No upfront payment.	About a 20 percent discount over on-demand rates.	One-year term	None	Yes
Partial Upfront	Partial upfront payment, and monthly installments for the duration of the reservation.	Up to 41 percent to 73 percent discount depending on duration.	One-year or three-year term	Yes	Yes
All Upfront	Full upfront payment for the reservation. No monthly charges.	Up to 42 percent to 76 percent discount depending on duration.	One-year or three-year term	Yes	None

Note

If you previously purchased **Heavy Utilization** offerings for Amazon Redshift, the comparable offering is the **Partial Upfront** offering.

How Reserved Nodes Work

With reserved node offerings, you pay according to the payment terms as described in the preceding section. You pay this way whether you already have a running cluster or you launch a cluster after you have a reservation.

When you purchase an offering, your reservation has a status of **payment-pending** until the reservation is processed. If the reservation fails to be processed, the status displays as **payment-failed** and you can try the process again. Once your reservation is successfully processed, its status changes to **active**.

The applicable discounted rate in your reservation is not applied to your bill until the status changes to **active**. After the reservation duration elapses, the status changes to **retired** but you can continue to access information about the reservation for historical purposes. When a reservation is **retired**, your clusters continue to run but you might be billed at the on-demand rate unless you have another reservation that applies discounted pricing to the nodes.

Reserved nodes are specific to the region in which you purchase the offering. If you purchase an offering by using the Amazon Redshift console, select the AWS region in which you want to purchase an offering, and then complete the reservation process. If you purchase an offering programmatically, the region is determined by the Amazon Redshift endpoint that you connect to. For more information about Amazon Redshift regions, go to [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

To ensure that the discounted rate is applied to all of the nodes when you launch a cluster, make sure that the region, the node type, and the number of nodes that you select match one or more active reservations. Otherwise, you'll be charged at the on-demand rate for nodes that don't match an active reservation.

In a running cluster, if you exceed the number of nodes that you have reserved, you begin to accrue charges for those additional nodes at the on-demand rate. This accrual means that it is possible for you to be charged varying rates for nodes in the same cluster depending on how many nodes you've reserved. You can purchase another offering to cover those additional nodes, and then the discounted rate is applied to those nodes for the remainder of the duration once the reservation status becomes **active**.

If you resize your cluster into a different node type and you haven't reserved nodes of that type, you'll be charged at the on-demand rate. You can purchase another offering with the new node type if you want to receive discounted rates for your resized cluster. However, you also continue to pay for the original reservation until its duration elapses. If you need to alter your reservations before the term expires, please create a support case using the [AWS Console](#).

Reserved Nodes and Consolidated Billing

The pricing benefits of Reserved Nodes are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The hourly usage across all sub-accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Nodes logic is applied to calculate the bill. For more information, see [Consolidated Billing](#) in the AWS Billing and Cost Management User Guide.

Reserved Node Examples

The scenarios in this section demonstrate how nodes accrue charges based on on-demand and discounted rates using the following reservation details:

- Region: US West (Oregon)
- Node Type: ds1.xlarge
- Payment Option: No Upfront
- Duration: one year
- Number of Reserved Nodes: 16

Example 1

You have one ds1.xlarge cluster in the US West (Oregon) region with 20 nodes.

In this scenario, 16 of the nodes receive the discounted rate from the reservation, but the additional 4 nodes in the cluster are billed at the on-demand rate.

Example 2

You have one ds1.xlarge cluster in the US West (Oregon) region with 12 nodes.

In this scenario, all 12 nodes in the cluster receive the discounted rate from the reservation. However, you also pay for the remaining reserved nodes in the reservation even though you don't currently have a running cluster to which they apply.

Example 3

You have one ds1.xlarge cluster in the US West (Oregon) region with 12 nodes. You run the cluster for several months with this configuration, and then you need to add nodes to the cluster. You resize the cluster, choosing the same node type and specifying a total of 16 nodes.

In this scenario, you are billed the discounted rate for 16 nodes. Your charges remain the same for the full year duration because the number of nodes that you have in the cluster is equal to the number of nodes that you have reserved.

Example 4

You have one ds1.xlarge cluster in the US West (Oregon) region with 16 nodes. You run the cluster for several months with this configuration, and then you need to add nodes. You resize the cluster, choosing the same node type and specifying a total of 20 nodes.

In this scenario, you are billed the discounted rate for all the nodes prior to the resize. After the resize, you are billed the discounted rate for 16 of the nodes for the rest of the year, and you are billed at the on-demand rate for the additional 4 nodes that you added to the cluster.

Example 5

You have two ds1.xlarge clusters in the US West (Oregon) region. One of the clusters has 6 nodes, and the other has 10 nodes.

In this scenario, you are billed at the discounted rate for all of the nodes because the total number of nodes in both clusters is equal to the number of nodes that you have reserved.

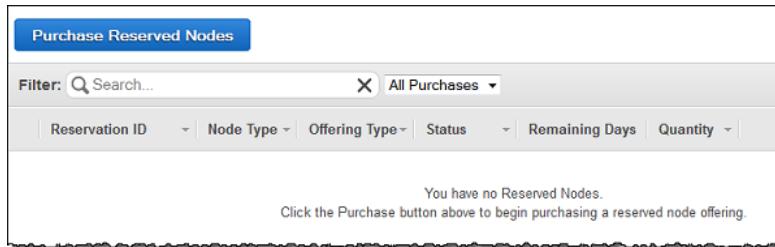
Example 6

You have two ds1.xlarge clusters in the US West (Oregon) region. One of the clusters has 4 nodes, and the other has 6 nodes.

In this scenario, you are billed the discounted rate for the 10 nodes that you have in running clusters, and you also pay the discounted rate for the additional 6 nodes that you have reserved even though you don't currently have any running clusters to which they apply.

Purchasing a Reserved Node Offering with the Amazon Redshift Console

You use the **Reserved Nodes** page in the Amazon Redshift console to purchase reserved node offerings, and to view current and past reservations. If you don't have any reservations, the **Reserved Node** page appears as follows.



After you purchase an offering, the **Reserved Node** list displays your reservations and the details of each one, such as the node type, number of nodes, and status of the reservation. For more information about the reservation details, see [How Reserved Nodes Work \(p. 177\)](#).

Purchase Reserved Nodes						
Filter: <input type="text"/> Search... X All Purchases						
Reservation ID	Node Type	Offering Type	Status	Remaining Days	Quantity	
506f6589-90e9-4ed4-9945-8d59cf13e9d3	ds2.8xlarge	Partial Upfront	active	1094 days	1	Status: active Node Type: ds2.8xlarge Fixed Price: \$24,000.00 Usage Price: \$0.912 (Hourly) Start Date: June 4, 2015 at 4:17:14 PM UTC-7
5b4131d9-e98e-42bc-8caf-44ba2ddb3804	ds2.xlarge	Partial Upfront	payment-pending	N/A	1	Status: payment-pending Node Type: ds2.xlarge Fixed Price: \$3,000.00 Usage Price: \$0.114 (Hourly) Start Date: June 5, 2015 at 2:03:08 PM UTC-7

To view reserved node reservations

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 **Reserved Nodes**.
3. (Optional) In **Filter**, specify search criteria to reduce the results in the reservations list.

To purchase a reserved node offering

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 **Reserved Nodes**.
3. Choose **Purchase Reserved Nodes**.
4. In the **Purchase Reserved Nodes** wizard, select **Node Type**, **Term**, and **Offering Type**.

Purchase Reserved Nodes

Select from the options below, then enter the Number of Nodes you wish to reserve with this order. When you are done, click the Continue button to review your order.

Node Type	ds2.xlarge
Term	3 years
Offering Type	Partial Upfront
One-time Payment: \$3,000.00 (per node)	
Number of Nodes	1
Total One-time Payment*: \$3,000.00 (Due Now)	
*Additional taxes may apply	
Usage Charges*: \$0.11 (Hourly) (per node)	
Charges for your usage will appear on your monthly bill.	
*Additional taxes may apply	

Cancel **Continue**

5. For **Number of Nodes**, type the number of nodes to reserve.
6. Choose **Continue**.
7. Review the offering details, and then choose **Purchase**.

Purchase Reserved Nodes

You are about to purchase Reserved Node(s) with the following information.

Region	US East (N. Virginia)
Node Type	ds2.xlarge
Offering Type	Partial Upfront
Term	3 years
Quantity	1
Price Per Node	\$3,000.00
Total Payment Due Now	\$3,000.00

Purchasing these Reserved Nodes will charge \$3,000.00 the payment method associated with this Amazon Web Services account.

Are you sure you would like to proceed?

Purchase

Back **Cancel**

8. On the **Reserved Nodes** page, the reservation displays in the reservations list with a status of **payment-pending**.

Upgrade a Reserved Node from DC1 to DC2

You can upgrade your DC1 reserved nodes to DC2 nodes for the remainder of your current term at no cost. DC2 is designed for demanding data warehousing workloads that require low latency and high throughput.

Prerequisites

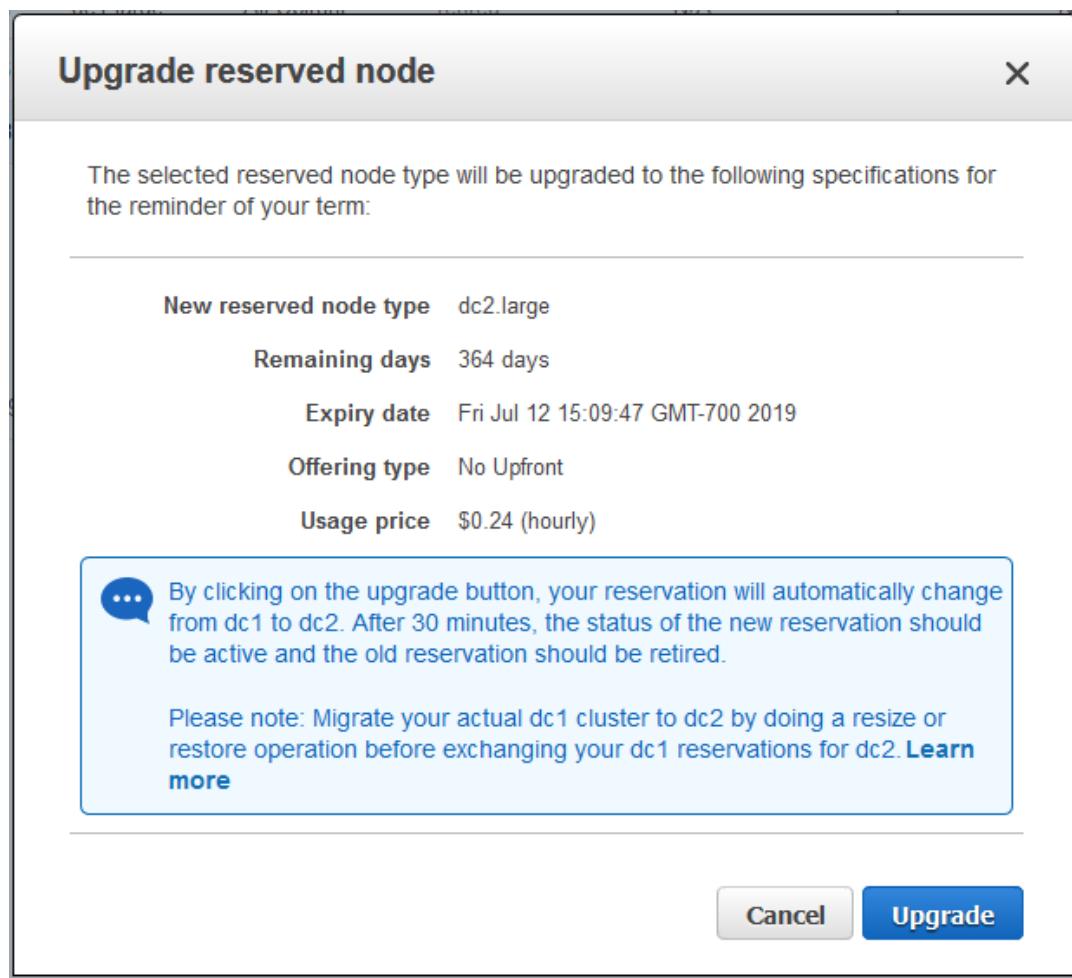
Migrate the cluster that includes the node you plan to upgrade before upgrading the reserved nodes. To migrate your DC1 cluster to DC2, use the resize or restore operation. If your cluster is a **DC1.large cluster**, you can restore to a new DC2.large cluster using an existing snapshot. If your cluster is a **DC1.8xlarge cluster**, you can resize it to be a DC2.8xlarge cluster. Make sure the DC1 cluster is shut

down before you upgrade the reserved nodes. The DC2 cluster will accrue on-demand pricing until you upgrade the DC1 reserved nodes.

For more information about restoring from a snapshot, see [Amazon Redshift Snapshots \(p. 145\)](#). For more information about resizing a cluster, see [Resizing Clusters in Amazon Redshift \(p. 311\)](#).

To upgrade a reserved node reservation

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. Choose **Reserved Nodes** from navigation pane.
3. Choose the DC1 reserved node you want to upgrade.
4. Review the details on the **Upgrade reserved node** dialog box and choose **Upgrade**.



To upgrade a reserved node reservation with the Amazon CLI

1. Obtain a list of ReservedNodeOfferingID's for offerings that meet your requirements for payment type, term, and charges. The following example illustrates this step:

```
aws redshift get-reserved-node-exchange-offerings --reserved-node-id xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

```
{  
    "ReservedNodeOfferings": [  
        {  
            "Duration": 31536000,  
            "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",  
            "UsagePrice": 0.0,  
            "NodeType": "dc2.large",  
            "RecurringCharges": [  
                {  
                    "RecurringChargeFrequency": "Hourly",  
                    "RecurringChargeAmount": 0.2  
                }  
            ],  
            "CurrencyCode": "USD",  
            "OfferingType": "No Upfront",  
            "ReservedNodeOfferingType": "Regular",  
            "FixedPrice": 0.0  
        }  
    ]  
}
```

2. Call `accept-reserved-node-exchange` and provide the ID for the DC1 reserved node you want to exchange along with the `ReservedNodeOfferingID` you obtained in the previous step.

The following example illustrates this step:

```
aws redshift accept-reserved-node-exchange --reserved-node-id xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx --target-reserved-node-offering-id yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy  
{  
    "ExchangedReservedNode": {  
        "UsagePrice": 0.0,  
        "OfferingType": "No Upfront",  
        "State": "exchanging",  
        "FixedPrice": 0.0,  
        "CurrencyCode": "USD",  
        "ReservedNodeId": "zzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzz",  
        "NodeType": "dc2.large",  
        "NodeCount": 1,  
        "RecurringCharges": [  
            {  
                "RecurringChargeFrequency": "Hourly",  
                "RecurringChargeAmount": 0.2  
            }  
        ],  
        "ReservedNodeOfferingType": "Regular",  
        "StartTime": "2018-06-27T18:02:58Z",  
        "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",  
        "Duration": 31536000  
    }  
}
```

You can confirm that the exchange is complete by calling `describe-reserved-nodes` and checking the value for `Node_type`.

Purchasing a Reserved Node Offering Using the AWS SDK for Java

The following example demonstrates how to use the AWS SDK for Java to do the following:

- List existing reserved nodes.
- Search for a new reserved node offering based on specified node criteria.
- Purchase a reserved node.

This example, first selects all the reserved node offerings that match a specified node type and fixed price value. Then, this example goes through each offering found and lets you purchase the offering.

Important

If you run this example and accept the offer to purchase a reserved node offering, you will be charged for the offering.

For step-by-step instructions to run this example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). To get information about a node type and fixed price other than those listed, update the code and provide that node type and fixed price.

Example

```
import java.io.DataInput;
import java.io.DataInputStream;
import java.io.IOException;
import java.util.ArrayList;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.DescribeReservedNodeOfferingsRequest;
import com.amazonaws.services.redshift.model.DescribeReservedNodeOfferingsResult;
import com.amazonaws.services.redshift.model.DescribeReservedNodesResult;
import com.amazonaws.services.redshift.model.PurchaseReservedNodeOfferingRequest;
import com.amazonaws.services.redshift.model.ReservedNode;
import com.amazonaws.services.redshift.model.ReservedNodeAlreadyExistsException;
import com.amazonaws.services.redshift.model.ReservedNodeOffering;
import com.amazonaws.services.redshift.model.ReservedNodeOfferingNotFoundException;
import com.amazonaws.services.redshift.model.ReservedNodeQuotaExceededException;

public class ListAndPurchaseReservedNodeOffering {

    public static AmazonRedshiftClient client;
    public static String nodeTypeToPurchase = "ds1.xlarge";
    public static Double fixedPriceLimit = 10000.00;
    public static ArrayList<ReservedNodeOffering> matchingNodes = new
    ArrayList<ReservedNodeOffering>();

    public static void main(String[] args) throws IOException {

        AWS Credentials credentials = new PropertiesCredentials(
            ListAndPurchaseReservedNodeOffering.class
            .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
```

```

        listReservedNodes();
        findReservedNodeOffer();
        purchaseReservedNodeOffer();

    } catch (Exception e) {
        System.err.println("Operation failed: " + e.getMessage());
    }
}

private static void listReservedNodes() {
    DescribeReservedNodesResult result = client.describeReservedNodes();
    System.out.println("Listing nodes already purchased.");
    for (ReservedNode node : result.getReservedNodes()) {
        printReservedNodeDetails(node);
    }
}

private static void findReservedNodeOffer()
{
    DescribeReservedNodeOfferingsRequest request = new
DescribeReservedNodeOfferingsRequest();
    DescribeReservedNodeOfferingsResult result =
client.describeReservedNodeOfferings(request);
    Integer count = 0;

    System.out.println("\nFinding nodes to purchase.");
    for (ReservedNodeOffering offering : result.getReservedNodeOfferings())
    {
        if (offering.getNodeType().equals(nodeTypeToPurchase)){

            if (offering.getFixedPrice() < fixedPriceLimit) {
                matchingNodes.add(offering);
                printOfferingDetails(offering);
                count +=1;
            }
        }
    }
    if (count == 0) {
        System.out.println("\nNo reserved node offering matches found.");
    } else {
        System.out.println("\nFound " + count + " matches.");
    }
}

private static void purchaseReservedNodeOffer() throws IOException {
    if (matchingNodes.size() == 0) {
        return;
    } else {
        System.out.println("\nPurchasing nodes.");

        for (ReservedNodeOffering offering : matchingNodes) {
            printOfferingDetails(offering);
            System.out.println("WARNING: purchasing this offering will incur costs.");
            System.out.print("Purchase this offering [Y or N]?");
            DataInputStream in = new DataInputStream(System.in);
            String purchaseOpt = in.readLine();
            if (purchaseOpt.equalsIgnoreCase("y")){

                try {
                    PurchaseReservedNodeOfferingRequest request = new
PurchaseReservedNodeOfferingRequest()

                    .withReservedNodeOfferingId(offering.getReservedNodeOfferingId());
                    ReservedNode reservedNode =
client.purchaseReservedNodeOffering(request);
                    printReservedNodeDetails(reservedNode);
                }
            }
        }
    }
}

```

```
        }
        catch (ReservedNodeAlreadyExistsException ex1){
        }
        catch (ReservedNodeOfferingNotFoundException ex2){
        }
        catch (ReservedNodeQuotaExceededException ex3){
        }
        catch (Exception ex4){
        }
    }
}
System.out.println("Finished.");

}

private static void printOfferingDetails(
    ReservedNodeOffering offering) {
    System.out.println("\nOffering Match:");
    System.out.format("Id: %s\n", offering.getReservedNodeOfferingId());
    System.out.format("Node Type: %s\n", offering.getNodeType());
    System.out.format("Fixed Price: %s\n", offering.getFixedPrice());
    System.out.format("Offering Type: %s\n", offering.getOfferingType());
    System.out.format("Duration: %s\n", offering.getDuration());
}

private static void printReservedNodeDetails(ReservedNode node) {
    System.out.println("\nPurchased Node Details:");
    System.out.format("Id: %s\n", node.getReservedNodeOfferingId());
    System.out.format("State: %s\n", node.getState());
    System.out.format("Node Type: %s\n", node.getNodeType());
    System.out.format("Start Time: %s\n", node.getStartTime());
    System.out.format("Fixed Price: %s\n", node.getFixedPrice());
    System.out.format("Offering Type: %s\n", node.getOfferingType());
    System.out.format("Duration: %s\n", node.getDuration());
}
}
```

Purchasing a Reserved Node Offering Using the AWS CLI and Amazon Redshift API

You can use the following AWS CLI operations to purchase reserved node offerings.

- [purchase-reserved-node-offering](#)
- [describe-reserved-node-offerings](#)
- [describe-orderable-cluster-options](#)

You can use the following Amazon Redshift APIs to purchase reserved node offerings.

- [PurchaseReservedNodeOffering](#)
- [DescribeReservedNodeOfferings](#)
- [DescribeOrderableClusterOptions](#)

Security

Access to Amazon Redshift resources is controlled at four levels:

- **Cluster management** – The ability to create, configure, and delete clusters is controlled by the permissions given to the user or account associated with your AWS security credentials. AWS users with the proper permissions can use the AWS Management Console, AWS Command Line Interface (CLI), or Amazon Redshift Application Programming Interface (API) to manage their clusters. This access is managed by using IAM policies. For details, see [Authentication and Access Control for Amazon Redshift \(p. 187\)](#).
- **Cluster connectivity** – Amazon Redshift security groups specify the AWS instances that are authorized to connect to an Amazon Redshift cluster in Classless Inter-Domain Routing (CIDR) format. For information about creating Amazon Redshift, Amazon EC2, and Amazon VPC security groups and associating them with clusters, see [Amazon Redshift Cluster Security Groups \(p. 203\)](#).
- **Database access** – The ability to access database objects, such as tables and views, is controlled by user accounts in the Amazon Redshift database. Users can only access resources in the database that their user accounts have been granted permission to access. You create these Amazon Redshift user accounts and manage permissions by using the [CREATE USER](#), [CREATE GROUP](#), [GRANT](#), and [REVOKE](#) SQL statements. For more information, see [Managing Database Security](#).
- **Temporary database credentials and single sign-on** – In addition to creating and managing database users using SQL commands, such as [CREATE USER](#) and [ALTER USER](#), you can configure your SQL client with custom Amazon Redshift JDBC or ODBC drivers. These drivers manage the process of creating database users and temporary passwords as part of the database logon process.

The drivers authenticate database users based on AWS Identity and Access Management (IAM) authentication. If you already manage user identities outside of AWS, you can use a SAML 2.0-compliant identity provider (IdP) to manage access to Amazon Redshift resources. You use an IAM role to configure your IdP and AWS to permit your federated users to generate temporary database credentials and log on to Amazon Redshift databases. For more information, see [Using IAM Authentication to Generate Database User Credentials \(p. 217\)](#).

Authentication and Access Control for Amazon Redshift

Access to Amazon Redshift requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an Amazon Redshift cluster. The following sections provide details on how you can use [AWS Identity and Access Management \(IAM\)](#) and Amazon Redshift to help secure your resources by controlling who can access them:

- [Authentication \(p. 187\)](#)
- [Access Control \(p. 188\)](#)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the](#)

root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An [IAM user](#) is an identity within your AWS account that has specific custom permissions (for example, permissions to create a cluster in Amazon Redshift). You can use an IAM user name and password to sign in to secure AWS webpages like the [AWS Management Console](#), [AWS Discussion Forums](#), or the [AWS Support Center](#).

In addition to a user name and password, you can also generate [access keys](#) for each user. You can use these keys when you access AWS services programmatically, either through [one of the several SDKs](#) or by using the [AWS Command Line Interface \(CLI\)](#). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. Amazon Redshift supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 Signing Process](#) in the [AWS General Reference](#).

- **IAM role** – An [IAM role](#) is an IAM identity that you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:
 - **Federated user access** – Instead of creating an IAM user, you can use existing user identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an [identity provider](#). For more information about federated users, see [Federated Users and Roles](#) in the *IAM User Guide*.
 - **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account's resources. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see [Creating a Role to Delegate Permissions to an AWS Service](#) in the *IAM User Guide*.
 - **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances](#) in the *IAM User Guide*.

Access Control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access Amazon Redshift resources. For example, you must have permissions to create an Amazon Redshift cluster, create a snapshot, add an event subscription, and so on.

The following sections describe how to manage permissions for Amazon Redshift. We recommend that you read the overview first.

- [Overview of Managing Access Permissions to Your Amazon Redshift Resources \(p. 188\)](#)
- [Using Identity-Based Policies \(IAM Policies\) for Amazon Redshift \(p. 193\)](#)

Overview of Managing Access Permissions to Your Amazon Redshift Resources

Every AWS resource is owned by an AWS account, and permissions to create or access the resources are governed by permissions policies. An account administrator can attach permissions policies to IAM

identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

Note

An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see [IAM Best Practices](#) in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, which resources they get permissions for, and the specific actions that you want to allow on those resources.

Amazon Redshift Resources and Operations

In Amazon Redshift, the primary resource is a *cluster*. Amazon Redshift supports other resources that can be used with the primary resource such as snapshots, parameter groups, and event subscriptions. These are referred to as *subresources*.

These resources and subresources have unique Amazon Resource Names (ARNs) associated with them as shown in the following table.

Resource Type	ARN Format
Cluster	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:cluster:<i>cluster-name</i></code>
Cluster database	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:dbname:<i>cluster-name</i>/<i>database-name</i></code>
Cluster database user	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:dbuser:<i>cluster-name</i>/<i>database-user-name</i></code>
Cluster database user group	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:dbgroup:<i>cluster-name</i>/<i>database-group-name</i></code>
Cluster parameter group	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:parametergroup:<i>parameter-group-name</i></code>
Cluster security group	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:securitygroup:<i>security-group-name</i></code>
CIDR/IP address	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:securitygroupingress:<i>security-group-name</i>/cidrip/<i>IP-range</i></code>
EC2 security group	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:securitygroupingress:<i>security-group-name</i>/ec2securitygroup/<i>owner</i>/<i>EC2-security-group-id</i></code>
Event subscription	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:eventsSubscription:<i>event-subscription-name</i></code>
Hardware security module (HSM) client certificate	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:hsmclientcertificate:<i>HSM-client-certificate-id</i></code>
HSM configuration	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:hsmconfiguration:<i>HSM-configuration-id</i></code>
Parameter group	<code>arn:aws:redshift:<i>region</i>:<i>account-id</i>:parametergroup:<i>parameter-group-name</i></code>

Resource Type	ARN Format
Snapshot	arn:aws:redshift: <i>region</i> : <i>account-id</i> :snapshot: <i>cluster-name</i> / <i>snapshot-name</i>
Snapshot copy grant	arn:aws:redshift: <i>region</i> : <i>account-id</i> :snapshotcopygrant: <i>snapshot-copy-grant-name</i>
Subnet group	arn:aws:redshift: <i>region</i> : <i>account-id</i> :subnetgroup: <i>subnet-group-name</i>

Amazon Redshift provides a set of operations to work with the Amazon Redshift resources. For a list of available operations, see [Amazon Redshift API Permissions Reference \(p. 201\)](#).

Understanding Resource Ownership

A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a DB cluster, your AWS account is the owner of the Amazon Redshift resource.
- If you create an IAM user in your AWS account and grant permissions to create Amazon Redshift resources to that user, the user can create Amazon Redshift resources. However, your AWS account, to which the user belongs, owns the Amazon Redshift resources.
- If you create an IAM role in your AWS account with permissions to create Amazon Redshift resources, anyone who can assume the role can create Amazon Redshift resources. Your AWS account, to which the role belongs, owns the Amazon Redshift resources.

Managing Access to Resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

Note

This section discusses using IAM in the context of Amazon Redshift. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see [What Is IAM?](#) in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see [AWS IAM Policy Reference](#) in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM policies) and policies attached to a resource are referred to as *resource-based* policies. Amazon Redshift supports only identity-based policies (IAM policies).

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user. Such a policy grants permissions for that user to create an Amazon Redshift resource, such as a cluster.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see [Access Management](#) in the *IAM User Guide*.

The following is an example policy that allows a user to create, delete, modify, and reboot Amazon Redshift clusters for your AWS account.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowManageClusters",  
            "Effect": "Allow",  
            "Action": [  
                "redshift:CreateCluster",  
                "redshift>DeleteCluster",  
                "redshift:ModifyCluster",  
                "redshift:RebootCluster"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

For more information about using identity-based policies with Amazon Redshift, see [Using Identity-Based Policies \(IAM Policies\) for Amazon Redshift \(p. 193\)](#). For more information about users, groups, roles, and permissions, see [Identities \(Users, Groups, and Roles\)](#) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. Amazon Redshift doesn't support resource-based policies.

Specifying Policy Elements: Actions, Effects, Resources, and Principals

For each Amazon Redshift resource (see [Amazon Redshift Resources and Operations \(p. 189\)](#)), the service defines a set of API operations (see [Actions](#)). To grant permissions for these API operations, Amazon Redshift defines a set of actions that you can specify in a policy. Performing an API operation can require permissions for more than one action.

The following are the basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see [Amazon Redshift Resources and Operations \(p. 189\)](#).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `redshift:DescribeClusters` permission allows the user permissions to perform the Amazon Redshift `DescribeClusters` operation.

- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). Amazon Redshift doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see [AWS IAM Policy Reference](#) in the *IAM User Guide*.

For a table showing all of the Amazon Redshift API actions and the resources that they apply to, see [Amazon Redshift API Permissions Reference \(p. 201\)](#).

Specifying Conditions in a Policy

When you grant permissions, you can use the access policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in an access policy language, see [IAM JSON Policy Elements: Condition](#) in the *IAM User Guide*.

To identify conditions where a permissions policy applies, include a `Condition` element in your IAM permissions policy. For example, you can create a policy that permits a user to create a cluster using the `redshift:CreateCluster` action, and you can add a `Condition` element to restrict that user to only create the cluster in a specific region. For details, see [Using IAM Policy Conditions for Fine-Grained Access Control \(p. 192\)](#). For a list showing all of condition key values and the Amazon Redshift actions and resources that they apply to, see [Amazon Redshift API Permissions Reference \(p. 201\)](#).

Using IAM Policy Conditions for Fine-Grained Access Control

In Amazon Redshift, you can use two condition keys to restrict access to resources based on the tags for those resources. The following are common Amazon Redshift condition keys.

Condition Key	Description
<code>redshift:RequestTag</code>	Requires users to include a tag key (name) and value whenever they create a resource. The <code>redshift:RequestTag</code> condition key only applies to Amazon Redshift API actions that create a resource.
<code>redshift:ResourceTag</code>	Restricts user access to resources based on specific tag keys and values.

For information on tags, see [Tagging Overview \(p. 326\)](#).

For a list of the API actions that support the `redshift:RequestTag` and `redshift:ResourceTag` condition keys, see [Amazon Redshift API Permissions Reference \(p. 201\)](#).

The following condition keys can be used with the Amazon Redshift `GetClusterCredentials` action.

Condition Key	Description
<code>redshift:DurationSeconds</code>	limits the number of seconds that can be specified for duration.
<code>redshift:DbName</code>	Restricts database names that can be specified.

Condition Key	Description
redshift:DbUser	Restricts database user names that can be specified.

Example 1: Restricting Access by Using the redshift:ResourceTag Condition Key

Use the following IAM policy to let a user modify an Amazon Redshift cluster only for a specific AWS account in the us-west-2 region with a tag named environment with a tag value of test.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowModifyTestCluster",
            "Effect": "Allow",
            "Action": "redshift:ModifyCluster",
            "Resource": "arn:aws:redshift:us-west-2:123456789012:cluster:/*"
            "Condition": {"StringEquals": {"redshift:ResourceTag/environment": "test"}}
        }
    ]
}
```

Example 2: Restricting Access by Using the redshift:RequestTag Condition Key

Use the following IAM policy to let a user create an Amazon Redshift cluster only if the command to create the cluster includes a tag named usage and a tag value of production.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateProductionCluster",
            "Effect": "Allow",
            "Action": "redshift>CreateCluster",
            "Resource": "*"
            "Condition": {"StringEquals": {"redshift:RequestTag/usage": "production"}}
        }
    ]
}
```

Using Identity-Based Policies (IAM Policies) for Amazon Redshift

This topic provides examples of identity-based policies in which an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles).

Important

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your Amazon Redshift resources. For more information, see [Overview of Managing Access Permissions to Your Amazon Redshift Resources \(p. 188\)](#).

The following shows an example of a permissions policy. The policy allows a user to create, delete, modify, and reboot all clusters, and then denies permission to delete or modify any clusters where the cluster identifier starts with production.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {

```

```
    "Sid": "AllowClusterManagement",
    "Action": [
        "redshift>CreateCluster",
        "redshift>DeleteCluster",
        "redshift>ModifyCluster",
        "redshift>RebootCluster"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow"
},
{
    "Sid": "DenyDeleteModifyProtected",
    "Action": [
        "redshift>DeleteCluster",
        "redshift>ModifyCluster"
    ],
    "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:production*"
    ],
    "Effect": "Deny"
}
]
```

The policy has two statements:

- The first statement grants permissions for a user to create, delete, modify, and reboot clusters. The statement specifies a wildcard character (*) as the Resource value so that the policy applies to all Amazon Redshift resources owned by the root AWS account.
- The second statement denies permission to delete or modify a cluster. The statement specifies a cluster Amazon Resource Name (ARN) for the Resource value that includes a wildcard character (*). As a result, this statement applies to all Amazon Redshift clusters owned by the root AWS account where the cluster identifier begins with production.

Permissions Required to Use the Amazon Redshift Console

For a user to work with the Amazon Redshift console, that user must have a minimum set of permissions that allows the user to describe the Amazon Redshift resources for their AWS account. These permissions must also allow the user to describe other related information, including Amazon EC2 security and network information.

If you create an IAM policy that is more restrictive than the minimum required permissions, the console doesn't function as intended for users with that IAM policy. To ensure that those users can still use the Amazon Redshift console, also attach the [AmazonRedshiftReadOnlyAccess](#) managed policy to the user, as described in [AWS-Managed \(Predefined\) Policies for Amazon Redshift \(p. 195\)](#).

To give a user access to the Query Editor on the Amazon Redshift console, attach the [AmazonRedshiftQueryEditor](#) managed policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the Amazon Redshift API.

Resource Policies for GetClusterCredentials

To connect to a cluster database using a JDBC or ODBC connection with IAM database credentials, or to programmatically call the `GetClusterCredentials` action, you need a minimum set of permissions. At

a minimum, you need permission to call the `redshift:GetClusterCredentials` action with access to a `dbuser` resource.

If you use a JDBC or ODBC connection, instead of `server` and `port` you can specify `cluster_id` and `region`, but to do so your policy must permit the `redshift:DescribeClusters` action with access to the `cluster` resource.

If you call `GetClusterCredentials` with the optional parameters `Autocreate`, `DbGroups`, and `DbName`, you need to also allow the actions and permit access to the resources listed in the following table.

GetClusterCred Parameter	Action	Resource
Autocreate	<code>redshift:CreateClusterUser</code>	<code>clusteruser</code>
DbGroups	<code>redshift:JoinGroup</code>	<code>group</code>
DbName	NA	<code>dbname</code>

For more information about resources, see [Amazon Redshift Resources and Operations \(p. 189\)](#).

You can also include the following conditions in your policy:

- `redshift:DurationSeconds`
- `redshift:DbName`
- `redshift:DbUser`

For more information about conditions, see [Specifying Conditions in a Policy \(p. 192\)](#)

AWS-Managed (Predefined) Policies for Amazon Redshift

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see [AWS Managed Policies](#) in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to Amazon Redshift:

- **AmazonRedshiftReadOnlyAccess** – Grants read-only access to all Amazon Redshift resources for the AWS account.
- **AmazonRedshiftFullAccess** – Grants full access to all Amazon Redshift resources for the AWS account.
- **AmazonRedshiftQueryEditor** – Grants full access to the Query Editor on the Amazon Redshift console.

You can also create your own custom IAM policies to allow permissions for Amazon Redshift API actions and resources. You can attach these custom policies to the IAM users or groups that require those permissions.

Customer Managed Policy Examples

In this section, you can find example user policies that grant permissions for various Amazon Redshift actions. These policies work when you are using the Amazon Redshift API, AWS SDKs, or the AWS CLI.

Note

All examples use the US West (Oregon) Region (`us-west-2`) and contain fictitious account IDs.

Example 1: Allow User Full Access to All Amazon Redshift Actions and Resources

The following policy allows access to all Amazon Redshift actions on all resources.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRedshift",  
            "Action": [  
                "redshift:*"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

The value `redshift:*` in the `Action` element indicates all of the actions in Amazon Redshift.

Example 2: Deny a User Access to a Set of Amazon Redshift Actions

By default, all permissions are denied. However, sometimes you need to explicitly deny access to a specific action or set of actions. The following policy allows access to all the Amazon Redshift actions and explicitly denies access to any Amazon Redshift action where the name starts with Delete. This policy applies to all Amazon Redshift resources in us-west-2.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowUSWest2Region",  
            "Action": [  
                "redshift:*"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:redshift:us-west-2:/*"  
        },  
        {  
            "Sid": "DenyDeleteUSWest2Region",  
            "Action": [  
                "redshift:Delete*"  
            ],  
            "Effect": "Deny",  
            "Resource": "arn:aws:redshift:us-west-2:/*"  
        }  
    ]  
}
```

Example 3: Allow a User to Manage Clusters

The following policy allows a user to create, delete, modify, and reboot all clusters, and then denies permission to delete any clusters where the cluster name starts with `protected`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {
```

```

    "Sid": "AllowClusterManagement",
    "Action": [
        "redshift>CreateCluster",
        "redshift>DeleteCluster",
        "redshift>ModifyCluster",
        "redshift>RebootCluster"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow"
},
{
    "Sid": "DenyDeleteProtected",
    "Action": [
        "redshift>DeleteCluster"
    ],
    "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:protected*"
    ],
    "Effect": "Deny"
}
]
}

```

Example 4: Allow a User to Authorize and Revoke Snapshot Access

The following policy allows a user, for example User A, to do the following:

- Authorize access to any snapshot created from a cluster named `shared`.
- Revoke snapshot access for any snapshot created from the `shared` cluster where the snapshot name starts with `revokable`.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSharedSnapshots",
            "Action": [
                "redshift>AuthorizeSnapshotAccess"
            ],
            "Resource": [
                "arn:aws:redshift:us-west-2:123456789012:shared/*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "AllowRevokableSnapshot",
            "Action": [
                "redshift>RevokeSnapshotAccess"
            ],
            "Resource": [
                "arn:aws:redshift:us-west-2:123456789012:snapshot:*/revokable*"
            ],
            "Effect": "Allow"
        }
    ]
}

```

If User A has allowed User B to access a snapshot, User B must have a policy such as the following to allow User B to restore a cluster from the snapshot. The following policy allows User B to describe and

restore from snapshots, and to create clusters. The name of these clusters must start with `from-other-account`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowDescribeSnapshots",  
            "Action": [  
                "redshift:DescribeClusterSnapshots"  
            ],  
            "Resource": [  
                "*"  
            ],  
            "Effect": "Allow"  
        },  
        {  
            "Sid": "AllowUserRestoreFromSnapshot",  
            "Action": [  
                "redshift:RestoreFromClusterSnapshot"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:snapshot:/*/*",  
                "arn:aws:redshift:us-west-2:444455556666:cluster:from-other-account*"  
            ],  
            "Effect": "Allow"  
        }  
    ]  
}
```

Example 5: Allow a User to Copy a Cluster Snapshot and Restore a Cluster from a Snapshot

The following policy allows a user to copy any snapshot created from the cluster named `big-cluster-1`, and restore any snapshot where the snapshot name starts with `snapshot-for-restore`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowCopyClusterSnapshot",  
            "Action": [  
                "redshift:CopyClusterSnapshot"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:snapshot:big-cluster-1/*"  
            ],  
            "Effect": "Allow"  
        },  
        {  
            "Sid": "AllowRestoreFromClusterSnapshot",  
            "Action": [  
                "redshift:RestoreFromClusterSnapshot"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:snapshot:*/snapshot-for-restore*",  
                "arn:aws:redshift:us-west-2:123456789012:cluster:/*"  
            ],  
            "Effect": "Allow"  
        }  
    ]  
}
```

Example 6: Allow a User Access to Amazon Redshift, and Common Actions and Resources for Related AWS Services

The following example policy allows access to all actions and resources for Amazon Redshift, Amazon Simple Notification Service (Amazon SNS), and Amazon CloudWatch. It also allows specified actions on all related Amazon EC2 resources under the account.

Note

Resource-level permissions are not supported for the Amazon EC2 actions that are specified in this example policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRedshift",  
            "Effect": "Allow",  
            "Action": [  
                "redshift:*"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Sid": "AllowSNS",  
            "Effect": "Allow",  
            "Action": [  
                "sns:*"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Sid": "AllowCloudWatch",  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Sid": "AllowEC2Actions",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AllocateAddress",  
                "ec2:AssociateAddress",  
                "ec2:AttachNetworkInterface",  
                "ec2:DescribeAccountAttributes",  
                "ec2:DescribeAddresses",  
                "ec2:DescribeAvailabilityZones",  
                "ec2:DescribeInternetGateways",  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeVpcs"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]
```

}

Example Policy for Using GetClusterCredentials

The following policy uses these sample parameter values:

- Region: us-west-2
- AWS Account: 123456789012
- Cluster name: examplecluster

The following policy enables the `GetCredentials`, `CreateCluserUser`, and `JoinGroup` actions. The policy uses condition keys to allow the `GetClusterCredentials` and `CreateClusterUser` actions only when the AWS user ID matches "AIDIODR4TAW7CSEXAMPLE: `${redshift:DbUser}`@yourdomain.com". IAM access is requested for the "testdb" database only. The policy also allows users to join a group named "common_group".

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "GetClusterCredsStatement",  
            "Effect": "Allow",  
            "Action": [  
                "redshift:GetClusterCredentials"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/${redshift:DbUser}",  
                "arn:aws:redshift:us-west-2:123456789012:dbname:examplecluster/testdb",  
                "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/common_group"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:userid": "AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"  
                }  
            }  
        },  
        {  
            "Sid": "CreateClusterUserStatement",  
            "Effect": "Allow",  
            "Action": [  
                "redshift:CreateClusterUser"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/${redshift:DbUser}"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:userid": "AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"  
                }  
            }  
        },  
        {  
            "Sid": "RedshiftJoinGroupStatement",  
            "Effect": "Allow",  
            "Action": [  
                "redshift:JoinGroup"  
            ],  
            "Resource": [  
                "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/common_group"  
            ]  
        }  
    ]  
}
```

```
    ]  
}  
  
}
```

Amazon Redshift API Permissions Reference

When you set up [Access Control \(p. 188\)](#) and write permissions policies that you can attach to an IAM identity (identity-based policies), you can use the following table as a reference. The each Amazon Redshift API operation and the corresponding actions for which you can grant permissions to perform the action. It also includes the AWS resource for which you can grant the permissions, and condition keys that you can include for fine-grained access control. (For more information about conditions, see [Using IAM Policy Conditions for Fine-Grained Access Control \(p. 192\)](#).) You specify the actions in the policy's Action field, the resource value in the policy's Resource field, and conditions in the policy's Condition field.

Note

To specify an action, use the `redshift:` prefix followed by the API operation name (for example, `redshift:CreateCluster`).

Amazon Redshift also supports the following actions that are not based on the Amazon Redshift API:

- The `redshift:ViewQueriesInConsole` action controls whether a user can see queries in the Amazon Redshift console in the **Queries** tab of the **Cluster** section.
- The `redshift:CancelQuerySession` action controls whether a user can terminate running queries and loads from the **Cluster** section in the Amazon Redshift console.
- The following actions control access to the Query Editor on the Amazon Redshift console:
 - `redshift>ListSchemas`
 - `redshift>ListTables`
 - `redshift>ListDatabases`
 - `redshift:ExecuteQuery`
 - `redshift:FetchResults`
 - `redshift:CancelQuery`
 - `redshift:DescribeQuery`
 - `redshift:DescribeTable`
 - `redshift:ViewQueriesFromConsole`
 - `redshift>ListSavedQueries`
 - `redshift>CreateSavedQuery`
 - `redshift>DeleteSavedQueries`
 - `redshift:ModifySavedQuery`

Using Service-Linked Roles for Amazon Redshift

Amazon Redshift uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Amazon Redshift. Service-linked roles are predefined by Amazon Redshift and include all the permissions that the service requires to call AWS services on behalf of your Amazon Redshift cluster.

A service-linked role makes setting up Amazon Redshift easier because you don't have to manually add the necessary permissions. The role is linked to Amazon Redshift use cases and has predefined

permissions. Only Amazon Redshift can assume the role, and only the service-linked role can use the predefined permissions policy. Amazon Redshift creates a service-linked role in your account the first time you create a cluster. You can delete the service-linked role only after you delete all of the Amazon Redshift clusters in your account. This protects your Amazon Redshift resources because you can't inadvertently remove permissions needed for access to the resources.

Amazon Redshift supports using service-linked roles in all of the regions where the service is available. For more information, see [AWS Regions and Endpoints](#).

For information about other services that support service-linked roles, see [AWS Services That Work with IAM](#) and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Amazon Redshift

Amazon Redshift uses the service-linked role named **AWSServiceRoleForRedshift** – Allows Amazon Redshift to call AWS services on your behalf.

The **AWSServiceRoleForRedshift** service-linked role trusts only redshift.amazonaws.com to assume the role.

The **AWSServiceRoleForRedshift** service-linked role permissions policy allows Amazon Redshift to complete the following on all related resources:

- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeAddress`
- `ec2:AssociateAddress`
- `ec2:DisassociateAddress`
- `ec2>CreateNetworkInterface`
- `ec2>DeleteNetworkInterface`
- `ec2:ModifyNetworkInterfaceAttribute`

To allow an IAM entity to create **AWSServiceRoleForRedshift** service-linked roles

Add the following policy statement to the permissions for that IAM entity:

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam:CreateServiceLinkedRole"  
    ],  
    "Resource": "arn:aws:iam:<AWS-account-ID>:role/aws-service-role/  
redshift.amazonaws.com/AWSServiceRoleForRedshift",  
    "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}  
}
```

To allow an IAM entity to delete **AWSServiceRoleForRedshift** service-linked roles

Add the following policy statement to the permissions for that IAM entity:

```
{
```

```
"Effect": "Allow",
"Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
],
"Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift",
"Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}
}
```

Alternatively, you can use an AWS managed policy to [provide full access to Amazon Redshift](#).

Creating a Service-Linked Role for Amazon Redshift

You don't need to manually create an AWSServiceRoleForRedshift service-linked role. Amazon Redshift creates the service-linked role for you. If the AWSServiceRoleForRedshift service-linked role has been deleted from your account, Amazon Redshift creates the role when you launch a new Amazon Redshift cluster.

Important

If you were using the Amazon Redshift service before September 18, 2017, when it began supporting service-linked roles, then Amazon Redshift created the AWSServiceRoleForRedshift role in your account. To learn more, see [A New Role Appeared in My IAM Account](#).

Editing a Service-Linked Role for Amazon Redshift

Amazon Redshift does not allow you to edit the AWSServiceRoleForRedshift service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using the IAM console, the AWS Command Line Interface (AWS CLI), or IAM API. For more information, see [Modifying a Role](#) in the *IAM User Guide*.

Deleting a Service-Linked Role for Amazon Redshift

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained.

Before you can delete a service-linked role for an account, you need to shut down and delete any clusters in the account. For more information, see [Shutting Down and Deleting Clusters \(p. 14\)](#).

You can use the IAM console, the AWS CLI, or the IAM API to delete a service-linked role. For more information, see [Deleting a Service-Linked Role](#) in the *IAM User Guide*.

Amazon Redshift Cluster Security Groups

When you provision an Amazon Redshift cluster, it is locked down by default so nobody has access to it. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group. If you are on the EC2-Classic platform, you define a cluster security group and associate it with a cluster as described following. If you are on the EC2-VPC platform, you can either use an existing Amazon VPC security group or define a new one and then associate it with a cluster. For more information on managing a cluster on the EC2-VPC platform, see [Managing Clusters in an Amazon Virtual Private Cloud \(VPC\) \(p. 36\)](#).

Topics

- [Overview \(p. 204\)](#)
- [Managing Cluster Security Groups Using the Console \(p. 204\)](#)
- [Managing Cluster Security Groups Using the AWS SDK for Java \(p. 213\)](#)
- [Manage Cluster Security Groups Using the Amazon Redshift CLI and API \(p. 216\)](#)

Overview

A cluster security group consists of a set of rules that control access to your cluster. Individual rules identify either a range of IP addresses or an Amazon EC2 security group that is allowed access to your cluster. When you associate a cluster security group with a cluster, the rules that are defined in the cluster security group control access to the cluster.

You can create cluster security groups independent of provisioning any cluster. You can associate a cluster security group with an Amazon Redshift cluster either at the time you provision the cluster or later. Also, you can associate a cluster security group with multiple clusters.

Amazon Redshift provides a cluster security group called default, which is created automatically when you launch your first cluster. Initially, this cluster security group is empty. You can add inbound access rules to the default cluster security group and then associate it with your Amazon Redshift cluster.

If the default cluster security group is enough for you, you don't need to create your own. However, you can optionally create your own cluster security groups to better manage inbound access to your cluster. For example, suppose that you are running a service on an Amazon Redshift cluster, and you have a few companies as your customers. If you don't want to provide the same access to all your customers, you might want to create separate cluster security groups, one for each company. You can add rules in each cluster security group to identify the Amazon EC2 security groups and the IP address ranges specific to a company. You can then associate all these cluster security groups with your cluster.

You can associate a cluster security group with many clusters, and you can associate many cluster security groups with a cluster, subject to AWS service limits. For more information, see [Amazon Redshift Limits](#).

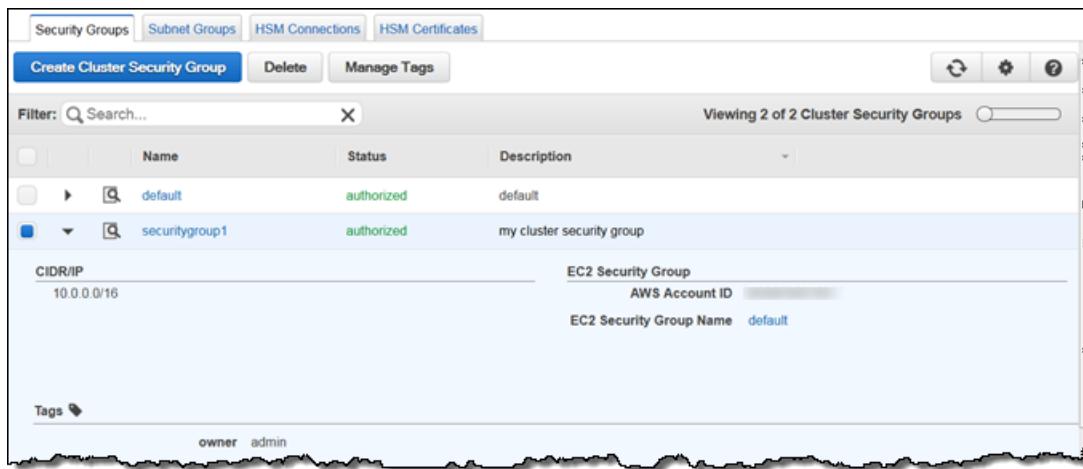
You can manage cluster security groups using the Amazon Redshift console, and you can manage cluster security groups programmatically by using the Amazon Redshift API or the AWS SDKs.

Amazon Redshift applies changes to a cluster security group immediately. So if you have associated the cluster security group with a cluster, inbound cluster access rules in the updated cluster security group apply immediately.

Managing Cluster Security Groups Using the Console

You can create, modify, and delete cluster security groups by using the Amazon Redshift console. You can also manage the default cluster security group in the Amazon Redshift console. All of the tasks start from the cluster security group list. You must choose a cluster security group to manage it.

In the example cluster security group list below, there are two cluster security groups, the default cluster security group and a custom cluster security group called `securitygroup1`. Because `securitygroup1` is selected (highlighted), you can delete it or manage tags for it, and also see the rules and tags associated with it.



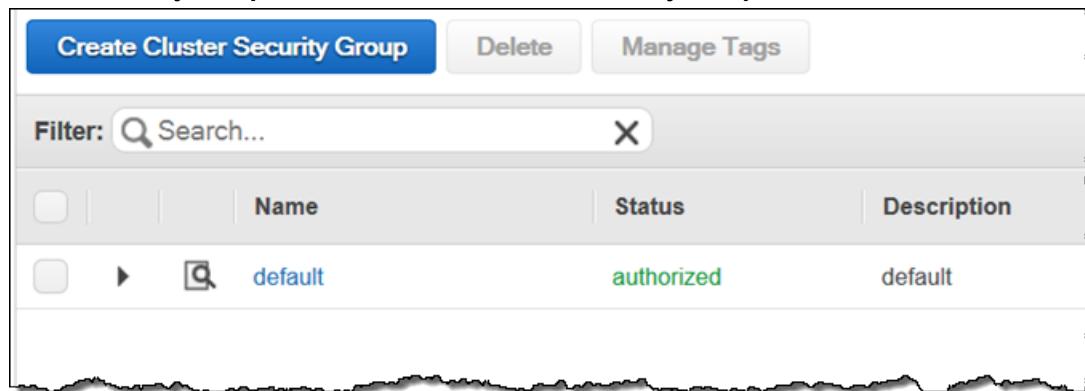
You cannot delete the default cluster security group, but you can modify it by authorizing or revoking ingress access.

To add or modify the rules associated with a security group, choose the security group to go to the **Security Group Connections** page.

Creating a Cluster Security Group

To create a cluster security group

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 **Security**.
3. On the **Security Groups** tab, choose **Create Cluster Security Group**.



4. In the **Create Cluster Security Group** dialog box, specify a cluster security group name and description.



The dialog box is titled "Create Cluster Security Group". It contains fields for "Cluster Security Group Name" (set to "securitygroup1") and "Description" (set to "my cluster security gro"). There are "Cancel" and "Create" buttons at the bottom.

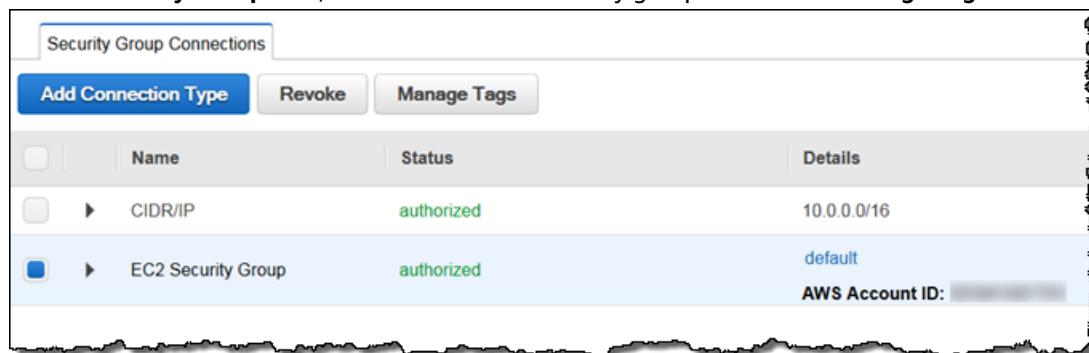
5. Choose **Create**.

The new group is displayed in the list of cluster security groups.

Tagging a Cluster Security Group

To tag a cluster security group

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 **Security**.
3. On the **Security Groups** tab, choose the cluster security group and choose **Manage Tags**.



The dialog box shows a table of security group connections. The table has columns: Name, Status, and Details. It lists two entries: "CIDR/IP" (status: authorized, details: 10.0.0.0/16) and "EC2 Security Group" (status: authorized, details: default, AWS Account ID: [redacted]).

	Name	Status	Details
<input type="checkbox"/>	CIDR/IP	authorized	10.0.0.0/16
<input checked="" type="checkbox"/>	EC2 Security Group	authorized	default AWS Account ID: [redacted]

4. In the **Manage Tags** dialog box, do one of the following:

- a. Remove a tag.
 - In the **Applied Tags** section, choose **Delete** next to the tag you want to remove.
 - Choose **Apply Changes**.

Manage Tags

Apply tags to your resources to help organize and identify them.
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn More](#) about tagging your AWS resources.

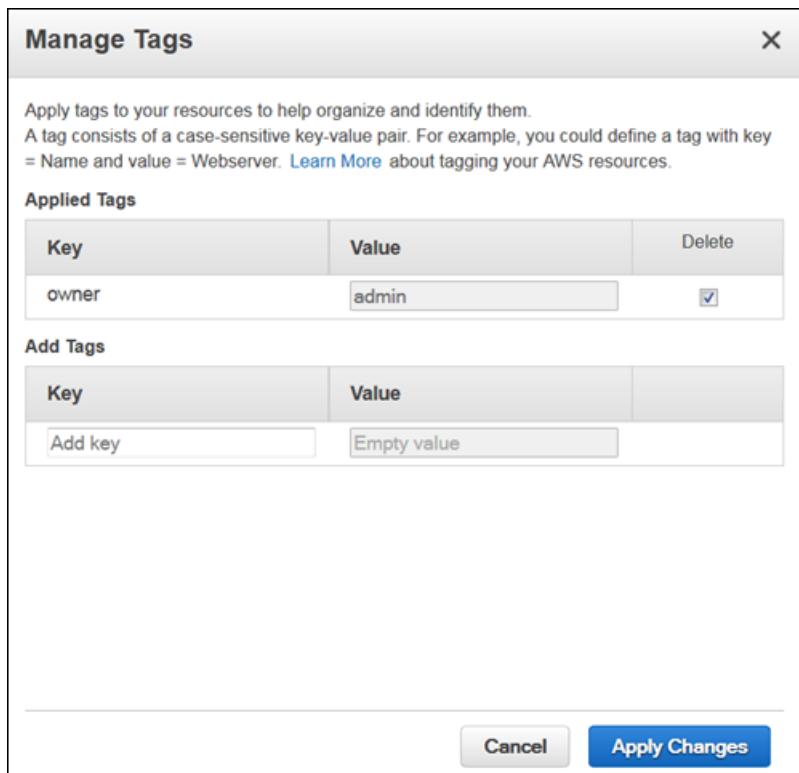
Applied Tags

Key	Value	Delete
owner	admin	<input checked="" type="checkbox"/>

Add Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Empty value"/>

Buttons



The screenshot shows the 'Manage Tags' dialog box. It has two main sections: 'Applied Tags' and 'Add Tags'. In the 'Applied Tags' section, there is one tag listed: 'owner' with a value of 'admin'. A 'Delete' button is next to it, which has a checked checkbox. In the 'Add Tags' section, there are input fields for 'Key' and 'Value'. Below these fields are buttons for 'Add key' and 'Empty value'. At the bottom of the dialog are 'Cancel' and 'Apply Changes' buttons.

- b. Add a tag.
- In the **Add Tags** section, type a key-value pair for the tag.
 - Choose **Apply Changes**.

Manage Tags

Apply tags to your resources to help organize and identify them.
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn More](#) about tagging your AWS resources.

Applied Tags

Key	Value	Delete
owner	admin	<input type="checkbox"/>

Add Tags

Key	Value	
sg_type	finance	<input type="checkbox"/>
Add key	Empty value	

Cancel **Apply Changes**

For more information about tagging an Amazon Redshift resource, see [How to Manage Tags in the Amazon Redshift Console \(p. 329\)](#).

Managing Ingress Rules for a Cluster Security Group

To manage ingress rules for a cluster security group

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 **Security**.
3. On the **Security Groups** tab, in the cluster security group list, choose the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, choose **Add Connection Type**.

Security Group Connections

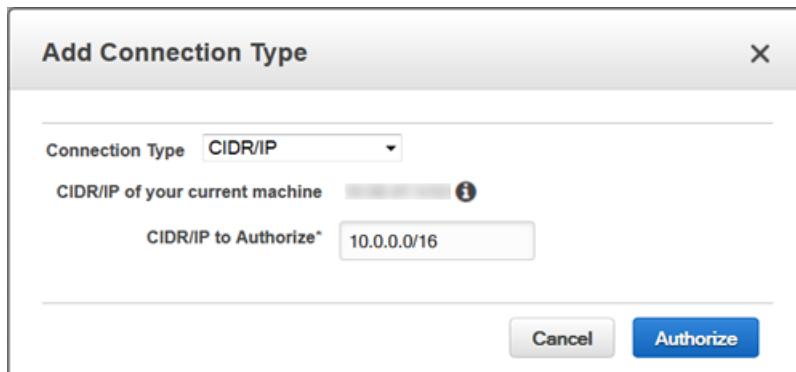
Add Connection Type **Revoke** **Manage Tags**

	Name	Status	Details
<input type="checkbox"/>	EC2 Security Group	authorized	default AWS Account ID: [REDACTED]

5. In the **Add Connection Type** dialog, do one of the following:

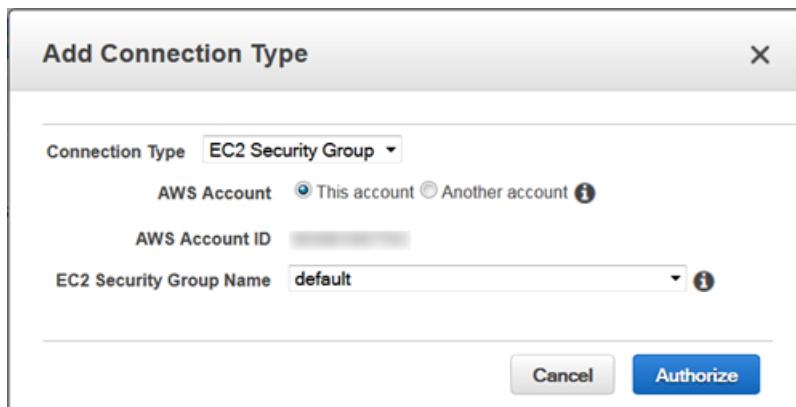
- a. Add an ingress rule based on CIDR/IP:

- In the **Connection Type** box, choose **CIDR/IP**.
- In the **CIDR/IP to Authorize** box, specify the range.
- Choose **Authorize**.



- b. Add an ingress rule based on an EC2 security group:

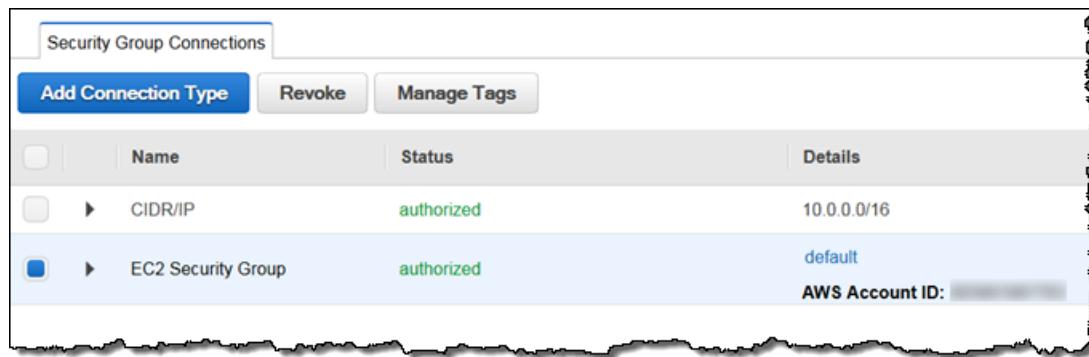
- Under **Connection Type**, choose **EC2 Security Group**.
- Choose the AWS account to use. By default, the account currently logged into the console is used. If you choose **Another account**, you must specify the AWS account ID.
- Choose the name of the EC2 security group you want in the **EC2 Security Group Name** box.
- Choose **Authorize**.



Revoking Ingress Rules for a Cluster Security Group

To revoke ingress rules for a cluster security group

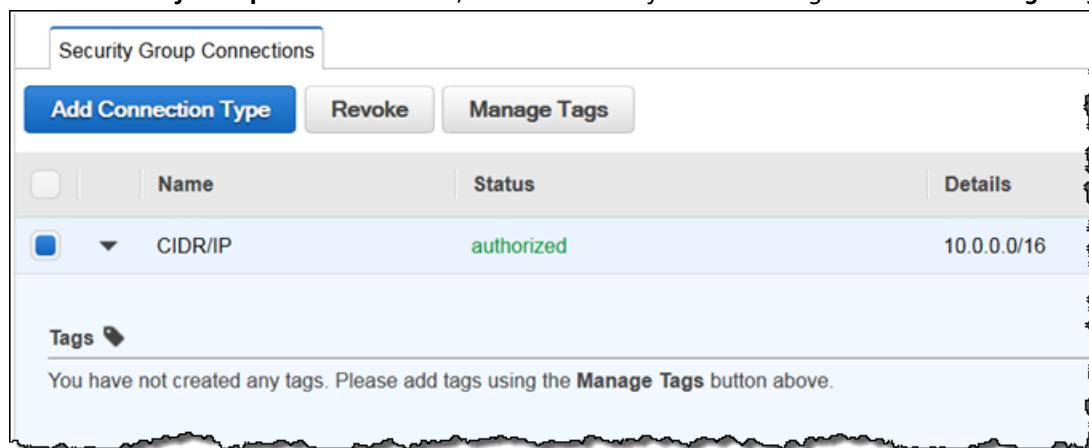
1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation pane, choose **Security**.
3. On the **Security Groups** tab, in the cluster security group list, choose the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, choose the rule you want to remove and choose **Revoke**.



Tagging Ingress Rules for a Cluster Security Group

To tag ingress rules for a cluster security group

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 **Security**.
3. On the **Security Groups** tab, choose the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, choose the rule you want to tag and choose **Manage Tags**.



5. In the **Manage Tags** dialog box, do one of the following:

- a. Remove a tag:
 - In the **Applied Tags** section, choose **Delete** next to the tag you want to remove.
 - Choose **Apply Changes**.

Manage Tags

Apply tags to your resources to help organize and identify them.
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn More](#) about tagging your AWS resources.

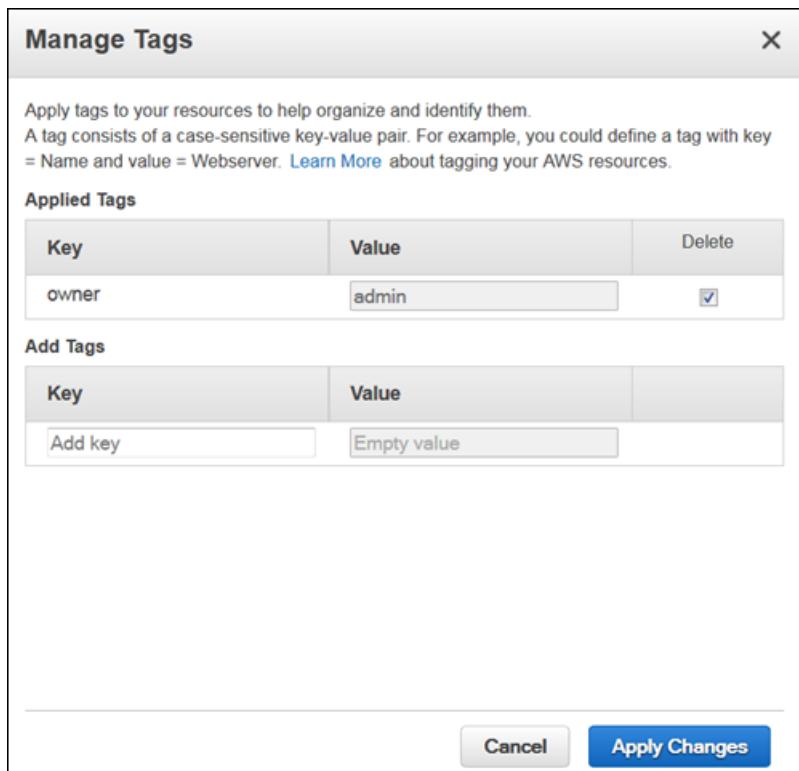
Applied Tags

Key	Value	Delete
owner	admin	<input checked="" type="checkbox"/>

Add Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Empty value"/>

Cancel **Apply Changes**



The screenshot shows the 'Manage Tags' dialog box. It has two main sections: 'Applied Tags' and 'Add Tags'. In the 'Applied Tags' section, there is one tag listed: 'owner' with 'Value' 'admin' and a checked 'Delete' checkbox. In the 'Add Tags' section, there are input fields for 'Key' and 'Value', both currently empty. At the bottom are 'Cancel' and 'Apply Changes' buttons.

- b. Add a tag.

Note

Tagging an EC2 Security Group rule only tags that rule, not the EC2 Security Group itself. If you want the EC2 Security Group tagged as well, you must do that separately.

- In the **Add Tags** section, type a key-value pair for the tag.
- Choose **Apply Changes**.

Manage Tags

Apply tags to your resources to help organize and identify them.
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn More](#) about tagging your AWS resources.

Applied Tags

Key	Value	Delete
owner	admin	<input type="checkbox"/>

Add Tags

Key	Value	
sg_type	finance	<input type="button" value="X"/>
Add key	Empty value	

Cancel **Apply Changes**

For more information about tagging an Amazon Redshift resource, see [How to Manage Tags in the Amazon Redshift Console \(p. 329\)](#).

Deleting a Cluster Security Group

If a cluster security group is associated with one or more clusters, you cannot delete it.

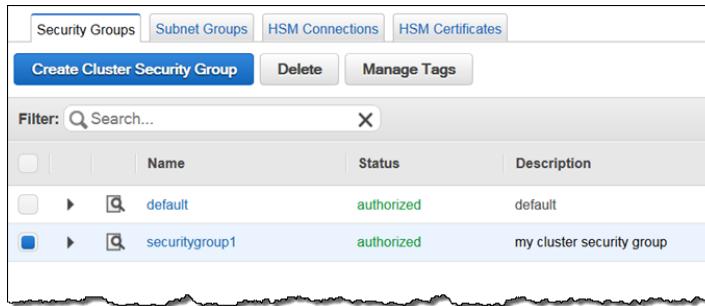
To delete a cluster security group

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 **Security**.
3. On the **Security Groups** tab, choose the cluster security group that you want to delete, and then choose **Delete**.

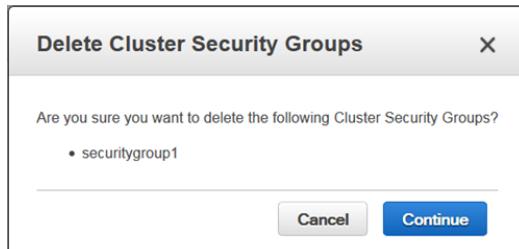
One row must be selected for the **Delete** button to be enabled.

Note

You cannot delete the default cluster security group.



4. In the Delete Cluster Security Groups dialog box, choose Continue.



If the cluster security group is used by a cluster, you can't delete it. The following example shows that `securitygroup1` is used by `examplecluster2`.



Associating a Cluster Security Group with a Cluster

Each cluster you provision on the EC2-Classic platform has one or more cluster security groups associated with it. You can associate a cluster security group with a cluster when you create the cluster, or you can associate a cluster security group later by modifying the cluster. For more information, see [Creating a Cluster by Using Launch Cluster \(p. 18\)](#) and [To modify a cluster \(p. 26\)](#). If you are on the EC2-VPC platform, see [Managing VPC Security Groups for a Cluster \(p. 39\)](#) for more information about associating VPC security groups with your cluster.

Managing Cluster Security Groups Using the AWS SDK for Java

The following example demonstrates common operations on cluster security groups, including:

- Creating a new cluster security group.
- Adding ingress rules to a cluster security group.
- Associating a cluster security group with a cluster by modifying the cluster configuration.

By default, when a new cluster security group is created, it has no ingress rules. This example modifies a new cluster security group by adding two ingress rules. One ingress rule is added by specifying a CIDR/IP range; the other is added by specifying an owner ID and Amazon EC2 security group combination.

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and provide a cluster identifier and AWS account number.

Example

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSecurityGroup {

    public static AmazonRedshiftClient client;
    public static String clusterSecurityGroupName = "securitygroup1";
    public static String clusterIdentifier = "***provide cluster identifier***";
    public static String ownerID = "***provide account id***";

    public static void main(String[] args) throws IOException {

        AWS Credentials credentials = new PropertiesCredentials(
            CreateAndModifyClusterSecurityGroup.class
                .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
            createClusterSecurityGroup();
            describeClusterSecurityGroups();
            addIngressRules();
            associateSecurityGroupWithCluster();
        } catch (Exception e) {
            System.out.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createClusterSecurityGroup() {
        CreateClusterSecurityGroupRequest request = new CreateClusterSecurityGroupRequest()
            .withDescription("my cluster security group")
            .withClusterSecurityGroupName(clusterSecurityGroupName);

        client.createClusterSecurityGroup(request);
        System.out.format("Created cluster security group: '%s'\n",
            clusterSecurityGroupName);
    }

    private static void addIngressRules() {

        AuthorizeClusterSecurityGroupIngressRequest request = new
        AuthorizeClusterSecurityGroupIngressRequest()
            .withClusterSecurityGroupName(clusterSecurityGroupName)
            .withCIDRIP("192.168.40.5/32");

        ClusterSecurityGroup result = client.authorizeClusterSecurityGroupIngress(request);

        request = new AuthorizeClusterSecurityGroupIngressRequest()
            .withClusterSecurityGroupName(clusterSecurityGroupName)
            .withEC2SecurityGroupName("default")
            .withEC2SecurityGroupOwnerId(ownerID);
        result = client.authorizeClusterSecurityGroupIngress(request);
    }
}
```

```
System.out.format("\nAdded ingress rules to security group '%s'\n",
clusterSecurityGroupName);
    printResultSecurityGroup(result);
}

private static void associateSecurityGroupWithCluster() {

    // Get existing security groups used by the cluster.
    DescribeClustersRequest request = new DescribeClustersRequest()
        .withClusterIdentifier(clusterIdentifier);

    DescribeClustersResult result = client.describeClusters(request);
    List<ClusterSecurityGroupMembership> membershipList =
        result.getClusters().get(0).getClusterSecurityGroups();

    List<String> secGroupNames = new ArrayList<String>();
    for (ClusterSecurityGroupMembership mem : membershipList) {
        secGroupNames.add(mem.getClusterSecurityGroupName());
    }
    // Add new security group to the list.
    secGroupNames.add(clusterSecurityGroupName);

    // Apply the change to the cluster.
    ModifyClusterRequest request2 = new ModifyClusterRequest()
        .withClusterIdentifier(clusterIdentifier)
        .withClusterSecurityGroups(secGroupNames);

    Cluster result2 = client.modifyCluster(request2);
    System.out.format("\nAssociated security group '%s' to cluster '%s'.",
clusterSecurityGroupName, clusterIdentifier);
}

private static void describeClusterSecurityGroups() {
    DescribeClusterSecurityGroupsRequest request = new
DescribeClusterSecurityGroupsRequest();

    DescribeClusterSecurityGroupsResult result =
client.describeClusterSecurityGroups(request);
    printResultSecurityGroups(result.getClusterSecurityGroups());
}

private static void printResultSecurityGroups(List<ClusterSecurityGroup> groups)
{
    if (groups == null)
    {
        System.out.println("\nDescribe cluster security groups result is null.");
        return;
    }

    System.out.println("\nPrinting security group results:");
    for (ClusterSecurityGroup group : groups)
    {
        printResultSecurityGroup(group);
    }
}
private static void printResultSecurityGroup(ClusterSecurityGroup group) {
    System.out.format("\nName: '%s', Description: '%s'\n",
group.getClusterSecurityGroupName(), group.getDescription());
    for (EC2SecurityGroup g : group.getEC2SecurityGroups())
    {
        System.out.format("EC2group: '%s', '%s', '%s'\n",
g.getEC2SecurityGroupName(),
g.getEC2SecurityGroupOwnerId(), g.getStatus());
    }
    for (IPRange range : group.getIPRanges())
    {
        System.out.format("IPRanges: '%s', '%s'\n",
range.getCIDRIP(),
range.getStatus());
    }
}
```

```
}
```

Manage Cluster Security Groups Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage cluster security groups.

- [authorize-cluster-security-group-ingress](#)
- [create-cluster-security-group](#)
- [delete-cluster-security-group](#)
- [describe-cluster-security-groups](#)
- [revoke-cluster-security-group-ingress](#)

You can use the following Amazon Redshift APIs to manage cluster security groups.

- [AuthorizeClusterSecurityGroupIngress](#)
- [CreateClusterSecurityGroup](#)
- [DeleteClusterSecurityGroup](#)
- [DescribeClusterSecurityGroups](#)
- [RevokeClusterSecurityGroupIngress](#)

Using IAM Authentication to Generate Database User Credentials

To better manage the access your users have to your Amazon Redshift database, you can generate temporary database credentials based on permissions granted through an AWS Identity and Access Management (IAM) permissions policy.

Commonly, Amazon Redshift database users log on to the database by providing a database user name and password. As an alternative to maintaining user names and passwords in your Amazon Redshift database, you can configure your system to permit users to create user credentials and log on to the database based on their IAM credentials. You can also configure your system to let users sign on using federated single sign-on (SSO) through a SAML 2.0-compliant identity provider.

Topics

- [Overview \(p. 217\)](#)
- [Creating Temporary IAM User Credentials \(p. 218\)](#)
- [Options for Providing IAM Credentials \(p. 228\)](#)
- [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#)
- [Generating IAM Database Credentials Using the Amazon Redshift CLI or API \(p. 232\)](#)

Overview

Amazon Redshift provides the `GetClusterCredentials` API action to generate temporary database user credentials. You can configure your SQL client with Amazon Redshift JDBC or ODBC drivers that manage the process of calling the `GetClusterCredentials` action. They do so by retrieving the database user credentials, and establishing a connection between your SQL client and your Amazon Redshift database. You can also use your database application to programmatically call the `GetClusterCredentials` action, retrieve database user credentials, and connect to the database.

If you already manage user identities outside of AWS, you can use a SAML 2.0-compliant identity provider (IdP) to manage access to Amazon Redshift resources. You configure your IdP to permit your federated users access to an IAM Role. With that IAM Role, you can generate temporary database credentials and log on to Amazon Redshift databases.

Your SQL client needs permission to call the `GetClusterCredentials` action on your behalf. You manage those permissions by creating an IAM role and attaching an IAM permissions policy that grants or restricts access to the `GetClusterCredentials` action and related actions.

The policy also grants or restricts access to specific resources, such as Amazon Redshift clusters, databases, database user names, and user group names.

Note

We recommend using the Amazon Redshift JDBC or ODBC drivers to manage the process of calling the `GetClusterCredentials` action and logging on to the database. For simplicity, we assume that you are using a SQL client with the JDBC or ODBC drivers throughout this topic.

For specific details and examples of using the `GetClusterCredentials` action or the parallel `get-cluster-credentials` CLI action, see [GetClusterCredentials](#) and [get-cluster-credentials](#).

Creating Temporary IAM User Credentials

In this section, you can find the steps to configure your system to generate temporary IAM-based database user credentials and log on to your database using the new credentials.

At a high level, the process flows as follows:

1. [Step 1: Create an IAM Role for IAM Single Sign-On \(SSO\) Access \(p. 218\)](#)

(Optional) You can authenticate users for access to an Amazon Redshift database by integrating IAM authentication and a third-party identity provider (IdP), such as PingFederate, Okta, or ADFS.

2. [Step 2: Configure SAML Assertions for Your IdP \(p. 218\)](#)

(Optional) To use IAM authentication using an IdP, you need to define a claim rule in your IdP application that maps users or groups in your organization to the IAM role. Optionally, you can include attribute elements to set `GetClusterCredentials` parameters.

3. [Step 3: Create an IAM Role or User With Permissions to Call GetClusterCredentials \(p. 220\)](#)

Your SQL client application assumes the IAM role when it calls the `GetClusterCredentials` action. If you created an IAM role for identity provider access, you can add the necessary permission to that role.

4. [Step 4: Create a Database User and Database Groups \(p. 221\)](#)

(Optional) By default, `GetClusterCredentials` returns credentials for existing users. You can choose to have `GetClusterCredentials` create a new user if the user name doesn't exist. You can also choose to specify user groups that users join at logon. By default, database users join the PUBLIC group.

5. [Step 5: Configure a JDBC or ODBC Connection to Use IAM Credentials \(p. 222\)](#)

To connect to your Amazon Redshift database, you configure your SQL client to use an Amazon Redshift JDBC or ODBC driver.

Step 1: Create an IAM Role for IAM Single Sign-On (SSO) Access

If you don't use an identity provider for single-sign on access, you can skip this step.

If you already manage user identities outside of AWS, you can authenticate users for access to an Amazon Redshift database by integrating IAM authentication and a third-party SAML-2.0 identity provider (IdP), such as ADFS, PingFederate, or Okta.

For more information, see [Identity Providers and Federation](#) in the *AWS IAM User Guide*.

Before you can use Amazon Redshift IdP authentication, you need to create an AWS SAML identity provider. You create an identity provider in the IAM console to inform AWS about the IdP and its configuration. Doing this establishes trust between your AWS account and the IdP. For steps to create a role, see [Creating a Role for SAML 2.0 Federation \(Console\)](#).

Step 2: Configure SAML Assertions for Your IdP

After you create the IAM role, you need to define a claim rule in your IdP application that maps users or groups in your organization to the IAM role. For more information, see [Configuring SAML Assertions for the Authentication Response](#).

If you choose to use the optional GetClusterCredentials parameters DbUser, AutoCreate, and DbGroups, you can set the values for the parameters with your JDBC or ODBC connection or you can set the values by adding SAML Attribute elements to your IdP. For more information about the DbUser, AutoCreate, and DbGroups parameters, see [Step 5: Configure a JDBC or ODBC Connection to Use IAM Credentials \(p. 222\)](#).

Note

If you use an IAM policy variable \${redshift:DbUser}, as described in [Resource Policies for GetClusterCredentials \(p. 194\)](#) the value for DbUser is replaced with the value retrieved by the API operation's request context. The Amazon Redshift drivers use the value for the DbUser variable provided by the connection URL, rather than the value supplied as a SAML attribute.

To help secure this configuration, we recommend that you use a condition in an IAM policy to validate the DbUser value with the RoleSessionName. You can find examples of how to set a condition using an IAM policy in [Example Policy for Using GetClusterCredentials \(p. 200\)](#).

To configure your IdP to set the DbUser, AutoCreate, and DbGroups parameters, include the following Attribute elements:

- An Attribute element with the Name attribute set to "https://redshift.amazonaws.com/SAML/Attributes/DbUser"

Set the AttributeValue to the name of a user that will connect to the Amazon Redshift database.

The value in the AttributeValue element must be lowercase, begin with a letter, contain only alphanumeric characters, underscore ('_'), plus sign ('+'), dot ('.'), at ('@'), or hyphen ('-'), and be less than 128 characters. Typically, the user name is a user ID (for example, bobsmith) or an email address (for example bobsmith@example.com). The value can't include a space (for example, a user's display name such as Bob Smith).

```
<Attribute Name="https://redshift.amazonaws.com/SAML/Attributes/DbUser">
  <AttributeValue>user-name</AttributeValue>
</Attribute>
```

- An Attribute element with the Name attribute set to "https://redshift.amazonaws.com/SAML/Attributes/AutoCreate"

Set the AttributeValue element to true to create a new database user if one doesn't exist. Set the AttributeValue to false to specify that the database user must exist in the Amazon Redshift database.

```
<Attribute Name="https://redshift.amazonaws.com/SAML/Attributes/AutoCreate">
  <AttributeValue>true</AttributeValue>
</Attribute>
```

- An Attribute element with the Name attribute set to "https://redshift.amazonaws.com/SAML/Attributes/DbGroups"

This element contains one or more AttributeValue elements. Set each AttributeValue element to a database group name that the DbUser joins for the duration of the session when connecting to the Amazon Redshift database.

```
<Attribute Name="https://redshift.amazonaws.com/SAML/Attributes/DbGroups">
  <AttributeValue>group1</AttributeValue>
  <AttributeValue>group2</AttributeValue>
  <AttributeValue>group3</AttributeValue>
</Attribute>
```

Step 3: Create an IAM Role or User With Permissions to Call GetClusterCredentials

Your SQL client needs authorization to call the `GetClusterCredentials` action on your behalf. To provide that authorization, you create an IAM user or role and attach a policy that grants the necessary permissions.

To create an IAM role with permissions to call `GetClusterCredentials`

1. Using the IAM service, create an IAM user or role. You can also use an existing user or role. For example, if you created an IAM role for identity provider access, you can attach the necessary IAM policies to that role.
2. Attach a permission policy with permission to call the `redshift:GetClusterCredentials` action. Depending on which optional parameters you specify, you can also allow or restrict additional actions and resources in your policy:
 - To permit your SQL client to retrieve cluster ID, region, and port, include permission to call the `redshift:DescribeClusters` action with the Redshift cluster resource.
 - If you use the AutoCreate option, include permission to call `redshift>CreateClusterUser` with the `dbuser` resource. The following Amazon Resource Name (ARN) specifies the Amazon Redshift `dbuser`. Replace `region`, `account-id`, and `cluster-name` with the values for your region, account, and cluster, respectively. For `dbuser-name`, specify the user name that will be used to log on to the cluster database.

```
arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name
```

- Optionally, add an ARN that specifies the Amazon Redshift `dbname` resource in the following format. Replace `region`, `account-id`, and `cluster-name` with the values for your region, account, and cluster, respectively. For `database-name`, specify the name of a database that the user will log on to.

```
arn:aws:redshift:region:account-id:dbname:cluster-name/database-name
```

- If you use the `DbGroups` option, include permission to call the `redshift:JoinGroup` action with the Amazon Redshift `dbgroup` resource in the following format. Replace `region`, `account-id`, and `cluster-name` with the values for your region, account, and cluster, respectively. For `dbgroup-name`, specify the name of a user group that the user joins at logon.

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

For more information and examples, see [Resource Policies for `GetClusterCredentials` \(p. 194\)](#).

The following example shows a policy that allows the IAM role to call the `GetClusterCredentials` action. Specifying the Amazon Redshift `dbuser` resource grants the role access to the database user name `temp_creds_user` on the cluster named `examplecluster`.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "redshift:GetClusterCredentials",  
        "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/  
temp_creds_user"
```

```
}
```

You can use a wildcard (*) to replace all, or a portion of, the cluster name, user name, and database group names. The following example allows any user name beginning with `temp_` with any cluster in the specified account.

Important

The statement in the following example specifies a wildcard character (*) as part of the value for the resource so that the policy permits any resource that begins with the specified characters.

Using a wildcard character in your IAM policies might be overly permissive. As a best practice, we recommend using the most restrictive policy feasible for your business application.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "redshift:GetClusterCredentials",
        "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:*/temp_*"
    }
}
```

The following example shows a policy that allows the IAM role to call the `GetClusterCredentials` action with the option to automatically create a new user and specify groups the user joins at logon. The `"Resource": "*"` clause grants the role access to any resource, including clusters, database users, or user groups.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "redshift:GetClusterCredentials",
            "redshift>CreateClusterUser",
            "redshift:JoinGroup"
        ],
        "Resource": "*"
    }
}
```

For more information, see [Amazon Redshift ARN syntax](#).

Step 4: Create a Database User and Database Groups

Optionally, you can create a database user that you use to log on to the cluster database. If you create temporary user credentials for an existing user, you can disable the user's password to force the user to log on with the temporary password. Alternatively, you can use the `GetClusterCredentials` `Autocreate` option to automatically create a new database user.

You can create database user groups with the permissions you want the IAM database user to join at logon. When you call the `GetClusterCredentials` action, you can specify a list of user group names that the new user joins at logon. These group memberships are valid only for sessions created using credentials generated with the given request.

To create a database user and database groups

1. Log on to your Amazon Redshift database and create a database user using `CREATE USER` or alter an existing user using `ALTER USER`.

2. Optionally, specify the PASSWORD DISABLE option to prevent the user from using a password. When a user's password is disabled, the user can log on only using temporary IAM user credentials. If the password is not disabled, the user can log on either with the password or using temporary IAM user credentials. You can't disable the password for a superuser.

The following example creates a user with password disabled.

```
create user temp_creds_user password disable;
```

The following example disables the password for an existing user.

```
alter user temp_creds_user password disable;
```

3. Create database user groups using [CREATE GROUP](#).
4. Use the [GRANT](#) command to define access privileges for the groups.

Step 5: Configure a JDBC or ODBC Connection to Use IAM Credentials

You can configure your SQL client with an Amazon Redshift JDBC or ODBC driver that manages the process of creating database user credentials and establishing a connection between your SQL client and your Amazon Redshift database.

To configure a JDBC connection to use IAM credentials

1. Download the latest Amazon Redshift JDBC driver from the [Configure a JDBC Connection \(p. 62\)](#) page.

Important

The Amazon Redshift JDBC driver must be version 1.2.7.1003 or later.

2. Create a JDBC URL with the IAM credentials options in one of the following formats. To use IAM authentication, add iam: to the Amazon Redshift JDBC URL following jdbc:redshift: as shown in the following example.

```
jdbc:redshift:iam://
```

Replace *cluster-name*, *region*, and *dbname* with values for your cluster name, region, and database name. The JDBC driver uses your IAM account information and cluster name to retrieve the cluster ID, region, and port number. To do so, your IAM user or role must have permission to call the redshift:DescribeClusters action with the specified cluster.

```
jdbc:redshift:iam://cluster-name:region/dbname
```

If your IAM user or role doesn't have permission to call the redshift:DescribeClusters action, include the cluster ID, region, and port as shown in the following example. The port number is optional. The default port is 5439.

```
jdbc:redshift:iam://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/  
dev
```

3. Add JDBC options to provide IAM credentials. You use different combinations of JDBC options to provide IAM credentials. For details, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

The following URL specifies AccessKeyID and SecretAccessKey for an IAM user.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?  
AccessKeyId=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtnFEMI/K7MDENG/  
bPxRfiCYEXAMPLEKEY
```

The following example specifies a named profile that contains the IAM credentials.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?Profile=user2
```

4. Add JDBC options that the JDBC driver uses to call the GetClusterCredentials API action. Don't include these options if you call the GetClusterCredentials API action programmatically. For more details, see [Configure a JDBC or ODBC Connection to Use IAM Credentials \(p. 222\)](#).

The following example includes the JDBC GetClusterCredentials options.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?  
Profile=user2&DbUser=newuser&AutoCreate=true&DbGroups=group1,group2
```

To configure an ODBC connection to use IAM credentials

In this topic, you can find steps only to configure IAM authentication. For steps to use standard authentication, using a database user name and password, see [Configure an ODBC Connection \(p. 77\)](#).

1. 1. Install and configure the latest Amazon Redshift ODBC driver for your operating system. For more information, see [Configure an ODBC Connection \(p. 77\)](#) page.

Important

The Amazon Redshift ODBC driver must be version 1.3.6.1000 or later.

2. Follow the steps for your operating system to configure connection settings.

For more information, see one of the following topics:

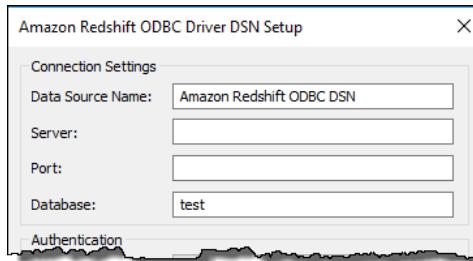
- [Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems \(p. 79\)](#)
 - [Configure the ODBC Driver on Linux and Mac OS X Operating Systems \(p. 84\)](#)
3. On Microsoft Windows Operation Systems, access the Amazon Redshift ODBC Driver DSN Setup window.

- a. Under **Connection Settings**, type the following information:

- **Data Source Name**
- **Server** (optional)
- **Port** (optional)
- **Database**

If your IAM user or role has permission to call the `redshift:DescribeClusters` action, only **Data Source Name** and **Database** are required. Amazon Redshift uses **ClusterId** and **Region** to get the server and port by calling the `DescribeCluster` action.

If your IAM user or role doesn't have permission to call the `redshift:DescribeClusters` action, specify **Server** and **Port**. The default port is 5439.



- b. Under **Authentication**, choose an **Auth Type**.

For each authentication type, specific fields appear as shown following.

AWS Profile

Enter the following information:

- **ClusterID**
- **Region**

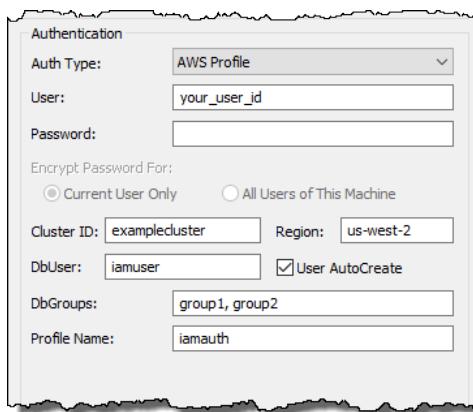
Optionally, provide details for options that the ODBC driver uses to call the GetClusterCredentials API action.

- **DbUser**
- **User AutoCreate**
- **DbGroups**

For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

- **Profile name**

Type the name of a profile in an AWS config file that contains values for the ODBC connection options. For more information, see [Using a Configuration Profile \(p. 230\)](#).



AWS IAM Credentials

Enter the following information:

- **ClusterID**
- **Region**

Provide details for options that the ODBC driver uses to call the GetClusterCredentials API action.

- **DbUser** (required)
- **User AutoCreate** (optional)
- **DbGroups** (optional)

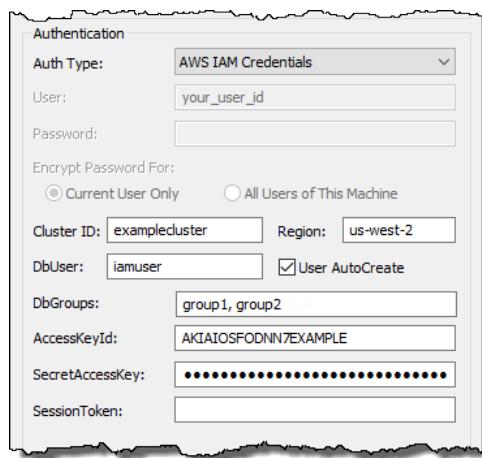
For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

- **AccessKeyId** and **SecretAccessKey**

The access key ID and secret access key for the IAM role or IAM user configured for IAM database authentication.

- **SessionToken**

SessionToken is required for an IAM role with temporary credentials. For more information, see [Temporary Security Credentials](#).



Identity Provider: AD FS

For Windows Integrated Authentication with AD FS, leave **User** and **Password** empty.

Optionally, provide details for options that the ODBC driver uses to call the GetClusterCredentials API action.

- **DbUser**
- **User AutoCreate**
- **DbGroups**

For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

Provide IdP details.

- **IdP Host**

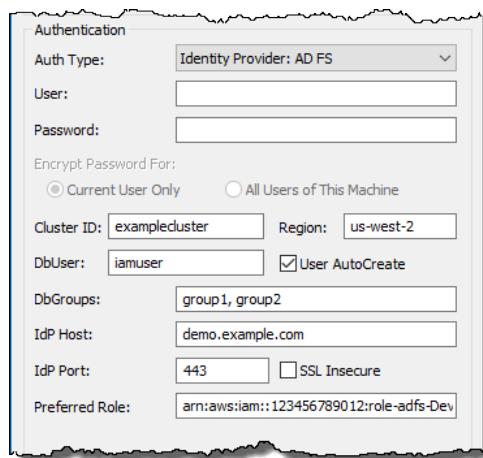
The name of the corporate identity provider host. This name should not include any slashes (/).

- **IdP Port (optional)**

The port used by identity provider. The default is 443.

- **Preferred Role**

A role Amazon Resource Name (ARN) from the AttributeValue elements for the Role attribute in the SAML assertion. Work with your IdP administrator to find the appropriate value for the preferred role. For more information, see [Configure SAML Assertions for Your IdP \(p. 218\)](#).



Identity Provider: PingFederate

For **User** and **Password**, type your IdP user name and password.

Optionally, provide details for options that the ODBC driver uses to call the GetClusterCredentials API action.

- **DbUser**
- **User AutoCreate**
- **DbGroups**

For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

Provide IdP details.

- **IdP Host**

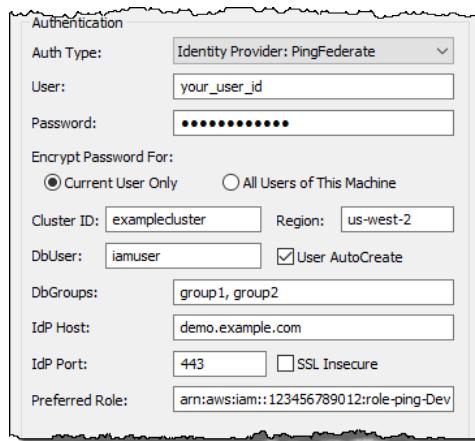
The name of the corporate identity provider host. This name should not include any slashes (/).

- **IdP Port (optional)**

The port used by identity provider. The default is 443.

- **Preferred Role**

A role Amazon Resource Name (ARN) from the AttributeValue elements for the Role attribute in the SAML assertion. Work with your IdP administrator to find the appropriate value for the preferred role. For more information, see [Configure SAML Assertions for Your IdP \(p. 218\)](#).



Identity Provider: Okta

For **User** and **Password**, type your IdP user name and password.

Optionally, provide details for options that the ODBC driver uses to call the GetClusterCredentials API action.

- **DbUser**
- **User AutoCreate**
- **DbGroups**

For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

Provide IdP details.

- **IdP Host**

The name of the corporate identity provider host. This name should not include any slashes (/).

- **IdP Port**

IdP Port is not used by Okta.

- **Preferred Role**

A role Amazon Resource Name (ARN) from the AttributeValue elements for the Role attribute in the SAML assertion. Work with your IdP administrator to find the appropriate value for the preferred role. For more information, see [Configure SAML Assertions for Your IdP \(p. 218\)](#).

- **Okta App ID**

An ID for an Okta application. The value for App ID follows "amazon_aws" in the Okta Application Embed Link. Work with your IdP administrator to get this value. The following is an example of an application embed link.

https://example.okta.com/home/amazon_aws/0oa2hylwrpM8UGehd1t7/272

The screenshot shows a configuration interface for an Okta Identity Provider. The 'Auth Type' is set to 'Identity Provider: Okta'. The 'User' field contains 'your_user_id' and the 'Password' field contains a redacted password. Under 'Encrypt Password For', the radio button for 'Current User Only' is selected. The 'Cluster ID' is 'examplecluster' and the 'Region' is 'us-west-2'. The 'DbUser' is 'iamuser' and the 'DbGroups' are 'group1, group2'. The 'IdP Host' is 'demo.example.com', 'IdP Port' is '443', and the 'Preferred Role' is 'arn:aws:iam::123456789012:role/okta-Dev'. The 'Okta App ID' is '0oa2hylwrpM8UGehd1t7/272'. A checkbox for 'User AutoCreate' is checked.

Options for Providing IAM Credentials

To provide IAM credentials for a JDBC or ODBC connection, choose one of the following authentication types.

- **AWS Profile**

As an alternative to providing credentials values in the form of JDBC or ODBC settings, you can put the values in a named profile.

- **AWS IAM Credentials**

Provide values for AccessKeyID, SecretAccessKey, and, optionally, SessionToken in the form of JDBC or ODBC settings. SessionToken is required only for an IAM role with temporary credentials. For more information, see [Temporary Security Credentials](#).

- **Identity Provider**

If you use an identity provider for authentication, specify the name of an identity provider plugin. The Amazon Redshift JDBC and ODBC drivers include plugins for the following SAML-based credential providers:

- AD FS
- PingFederate
- Okta

You can provide the plugin name and related values in the form of JDBC or ODBC settings or by using a profile.

For more information, see [Configure a JDBC or ODBC Connection to Use IAM Credentials \(p. 222\)](#).

JDBC and ODBC Options for Providing IAM Credentials

The following table lists the JDBC and ODBC options for providing IAM credentials.

Option	Description
Iam	For use only in an ODBC connection string. Set to 1 to use IAM authentication..
AccessKeyId SecretAccessKey SessionToken	The access key ID and secret access key for the IAM role or IAM user configured for IAM database authentication. SessionToken is required only for an IAM role with temporary credentials. SessionToken is not used for an IAM user. For more information, see Temporary Security Credentials .
Plugin_Name	The fully qualified class name that implements a credentials provider. The Amazon Redshift JDBC driver includes SAML-based credential provider plug-ins. If plugin_name is provided, other related parameters are available. For more information, see Using a Credentials Provider Plugin (p. 229) .
Profile	The name of a profile in an AWS credentials or config file that contains values for the JDBC connection options. For more information, see Using a Configuration Profile (p. 230) .

Using a Credentials Provider Plugin

The following credential provider plugins are included with the Amazon Redshift JDBC driver.

- Active directory federation service (AD FS)
- Ping Federate (Ping)

Ping is supported only with the predetermined PingFederate IdP Adapter using Forms authentication.
• Okta

Okta is supported only for the Okta-supplied AWS Console default application.

To use a SAML-based credential provider plugin, specify the following options using JDBC or ODBC options or in a named profile:

Option	Description
plugin_name	For JDBC, the class name that implements a credentials provider. Specify one of the following: <ul style="list-style-type: none"> • For ADFS <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"><code>com.amazon.redshift.plugin.AdfsCredentialsProvider</code></div> • For Okta <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"><code>com.amazon.redshift.plugin.OktaCredentialsProvider</code></div> • For PingFederate

Option	Description
	<pre style="border: 1px solid black; padding: 5px;">com.amazon.redshift.plugin.PingCredentialsProvider</pre> <p>For ODBC, specify one of the following:</p> <ul style="list-style-type: none"> • For AD FS: adfs • For Okta: okta • For PingFederate: ping
idp_host	The name of the corporate identity provider host. This name should not include any slashes ('/'). For an Okta identity provider, the value for idp_host should end with .okta.com.
idp_port	The port used by identity provider. The default is 443. Port is ignored for Okta.
preferred_role	A role Amazon Resource Name (ARN) from the AttributeValue elements for the Role attribute in the SAML assertion. Work with your IdP administrator to find the appropriate value for the preferred role. For more information, see Configure SAML Assertions for Your IdP (p. 218) Configure SAML Assertions for Your IdP.
user	A corporate user name, including the domain when applicable. For example, for Active Directory, the domain name is required in the format <i>domain\username</i> .
password	The corporate user's password. We recommend not using this option. Instead, use your SQL client to supply the password.
ssl_insecure	Set to true (JDBC) or 1 (ODBC) to use insecure SSL with IdP (not recommended).
app_id	An ID for a Okta application. Used only with Okta. The value for app_id follows amazon_aws in the Okta Application Embed Link. Work with your IdP administrator to get this value. The following is an example of an application embed link: https://example.okta.com/home/amazon_aws/0oa2hy1wprM8UGehd1t7/272

The following example shows credentials provider plugin parameters in a named profile.

```
[plug-in-creds]
plugin_name=com.amazon.redshift.plugin.AdfsCredentialsProvider
idp_host=demo.example.com
idp_port=443
preferred_role=arn:aws:iam::123456789012:role/ADFS-Dev
user=example\user
password=Password1234
```

Using a Configuration Profile

You can supply the IAM credentials options and GetClusterCredentials options as settings in named profiles in your AWS configuration file. Provide the profile name by using the Profile JDBC option.

The configuration is stored in a file named `config` in a folder named `.aws` in your home directory. Home directory location varies but can be referred to using the environment variables `%UserProfile%` in Windows and `$HOME` or `~` (tilde) in Unix-like systems.

When using the Amazon Redshift JDBC driver or ODBC driver with a bundled SAML-based credential provider plugin, the following settings are supported. If `plugin_name` is not used, the listed options are ignored.

- `plugin_name`
- `idp_host`
- `idp_port`
- `preferred_role`
- `user`
- `password`
- `ssl_insecure`
- `app_id` (for Okta only)

The following example shows a configuration file with three profiles. The `plug-in-creds` example includes the optional `DbUser`, `AutoCreate`, and `DbGroups` options.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user2]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
session_token=AQoDYXdzEPT///////////
wEXAMPLEtc764bNrC9SAPBSM22wD0k4x4HZ8j4FZTwdQWLWsKWHGBuFqwaMicRXmxfpSPfleoIYRqTflfKD8YUuwthAx7mSEI/
qkPpKPi/kMcGd
QrmGdeehM4IC1NtBmUp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
9HFv1Rd8Tx6q6fE8YQcHNVXAkiY9q6d+xoOrKwT38xVqr7ZD0u0iPPkUL641IZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSvKTr4rvx3iSiLTJabIQwj2ICCR/oLxBA==

[plug-in-creds]
plugin_name=com.amazon.redshift.plugin.AdfsCredentialsProvider
idp_host=demo.example.com
idp_port=443
preferred_role=arn:aws:iam::1234567:role/ADFS-Dev
user=example\user
password=Password1234
```

To use the credentials for the `user2` example, specify `Profile=user2` in the JDBC URL. To use the credentials for the `plug-in creds` example, specify `Profile=plug-in-creds` in the JDBC URL.

For more information, see [Named Profiles](#) in the AWS Command Line Interface User Guide.

JDBC and ODBC Options for Creating Database User Credentials

To use the Amazon Redshift JDBC or ODBC driver to create database user credentials, provide the database user name as a JDBC or ODBC option. Optionally, you can have the driver create a new database user if one doesn't exist, and you can specify a list of database user groups the user joins at logon.

If you use an identity provider (IdP), work with your IdP administrator to determine the correct values for these options. Your IdP administrator can also configure your IdP to provide these options, in which case you don't need to provide them as JDBC or ODBC options. For more information, see [Configure SAML Assertions for Your IdP \(p. 218\)](#).

Note

If you use an IAM policy variable `#{redshift:DbUser}`, as described in [Resource Policies for GetClusterCredentials \(p. 194\)](#) the value for `DbUser` is replaced with the value retrieved by the API operation's request context. The Amazon Redshift drivers use the value for the `DbUser` variable provided by the connection URL, rather than the value supplied as a SAML attribute. To help secure this configuration, we recommend that you use a condition in an IAM policy to validate the `DbUser` value with the `RoleSessionName`. You can find examples of how to set a condition using an IAM policy in [Example Policy for Using GetClusterCredentials \(p. 200\)](#).

The following table lists the options for creating database user credentials.

Option	Description
DbUser	The name of a database user. If a user named <code>DbUser</code> exists in the database, the temporary user credentials have the same permissions as the existing user. If <code>DbUser</code> doesn't exist in the database and <code>AutoCreate</code> is true, a new user named <code>DbUser</code> is created. Optionally, disable the password for an existing user. For more information, see ALTER_USER
AutoCreate	Specify <code>true</code> to create a database user with the name specified for <code>DbUser</code> if one does not exist. The default is <code>false</code> .
DbGroups	A comma-delimited list of the names of one or more existing database groups the database user joins for the current session. By default, the new user is added only to <code>PUBLIC</code> .

Generating IAM Database Credentials Using the Amazon Redshift CLI or API

To programmatically generate temporary database user credentials, Amazon Redshift provides the [get-cluster-credentials](#) command for the AWS Command Line Interface (AWS CLI) and the [GetClusterCredentials](#) API action. Alternatively, you can configure your SQL client with Amazon Redshift JDBC or ODBC drivers that manage the process of calling the `GetClusterCredentials` action, retrieving the database user credentials, and establishing a connection between your SQL client and your Amazon Redshift database. For more information, see [JDBC and ODBC Options for Creating Database User Credentials \(p. 231\)](#).

Note

We recommend using the Amazon Redshift JDBC or ODBC drivers to generate database user credentials.

In this section, you can find steps to programmatically call the `GetClusterCredentials` action or `get-cluster-credentials` command, retrieve database user credentials, and connect to the database.

To generate and use temporary database credentials

1. Create or modify an IAM user or role with the required permissions. For more information about IAM permissions, see [Create an IAM Role or User With Permissions to Call GetClusterCredentials \(p. 220\)](#).
2. As an IAM user or role you authorized in the previous step, execute the `get-cluster-credentials` CLI command or call the `GetClusterCredentials` API action and provide the following values:
 - **Cluster identifier** – The name of the cluster that contains the database.
 - **Database user name** – The name of an existing or new database user.

- If the user doesn't exist in the database and AutoCreate is true, a new user is created with PASSWORD disabled.
 - If the user doesn't exist, and AutoCreate is false, the request fails.
 - For this example, the database user name is `temp_creds_user`.
- **Autocreate** – (Optional) Create a new user if the database user name doesn't exist.
 - **Database name** – (Optional) The name of the database that the user is authorized to log on to. If database name is not specified, the user can log on to any cluster database.
 - **Database groups** – (Optional) A list of existing database user groups. Upon successful log on, the database user is added to the specified user groups. If no group is specified, the user has only PUBLIC permissions. This user group names must match the dbgroup resources ARNs specified in the IAM policy attached to the IAM user or role.
 - **Expiration time** – (Optional) The time, in seconds, until the temporary credentials expire. You can specify a value between 900 seconds (15 minutes) and 3600 seconds (60 minutes). The default is 900 seconds.
3. Amazon Redshift verifies that the IAM user has permission to call the GetClusterCredentials operation with the specified resources.
 4. Amazon Redshift returns a temporary password and the database user name.

The following example uses the Amazon Redshift CLI to generate temporary database credentials for an existing user named `temp_creds_user`.

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp_creds_user --db-name exampledb --duration-seconds 3600
```

The result is as follows.

```
{  
  "DbUser": "IAM:temp_creds_user",  
  "Expiration": "2016-12-08T21:12:53Z",  
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/  
gqX2Eeaq6P3DgTzgPg=="  
}
```

The following example uses the Amazon Redshift CLI with autocreate to generate temporary database credentials for a new user and add the user to the group `example_group`.

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp_creds_user --autocreate true --db-name exampledb --db-groups example_group --duration-seconds 3600
```

The result is as follows.

```
{  
  "DbUser": "IAMA:temp_creds_user:example_group",  
  "Expiration": "2016-12-08T21:12:53Z",  
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/  
gqX2Eeaq6P3DgTzgPg=="  
}
```

5. Establish a Secure Socket Layer (SSL) authentication connection with the Amazon Redshift cluster and send a logon request with the user name and password from the GetClusterCredentials response. Include the IAM: or IAMA: prefix with the user name, for example, `IAM:temp_creds_user` or `IAMA:temp_creds_user`.

Important

Configure your SQL client to require SSL. Otherwise, if your SQL client automatically tries to connect with SSL, it can fall back to non-SSL if there is any kind of failure. In that case, the first connection attempt might fail because the credentials are expired or invalid, then a second connection attempt fails because the connection is not SSL. If that occurs, the first error message might be missed. For more information about connecting to your cluster using SSL, see [Configure Security Options for Connections \(p. 95\)](#).

6. If the connection doesn't use SSL, the connection attempt fails.
7. The cluster sends an authentication request to the SQL client.
8. The SQL client then sends the temporary password to the cluster.
9. If the password is valid and has not expired, the cluster completes the connection.

Authorizing Amazon Redshift to Access Other AWS Services on Your Behalf

Some Amazon Redshift features require Amazon Redshift to access other AWS services on your behalf. For example, the [COPY](#) and [UNLOAD](#) commands can load or unload data into your Amazon Redshift cluster using an Amazon Simple Storage Service (Amazon S3) bucket. Amazon Redshift Spectrum can use a data catalog in Amazon Athena or AWS Glue. For your Amazon Redshift clusters to act on your behalf, you supply security credentials to your clusters. The preferred method to supply security credentials is to specify an AWS Identity and Access Management (IAM) role. For COPY and UNLOAD, you can provide AWS access keys.

Following, find out how to create an IAM role with the appropriate permissions to access other AWS services. You also need to associate the role with your cluster and specify the Amazon Resource Name (ARN) of the role when you execute the Amazon Redshift command. For more information, see [Authorizing COPY, UNLOAD, and CREATE EXTERNAL SCHEMA Operations Using IAM Roles \(p. 240\)](#).

Creating an IAM Role to Allow Your Amazon Redshift Cluster to Access AWS Services

To create an IAM role to permit your Amazon Redshift cluster to communicate with other AWS services on your behalf, take the following steps.

To create an IAM role to allow Amazon Redshift to access AWS services

1. Open the [IAM Console](#).
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. Choose **AWS service**, and then choose **Redshift**.
5. Under **Select your use case**, choose **Redshift - Customizable** and then choose **Next: Permissions**.
6. The **Attach permissions policy** page appears. For access to Amazon S3 using COPY and UNLOAD, choose **AmazonS3ReadOnlyAccess**. For Redshift Spectrum, in addition to Amazon S3 access, add **AWSGlueConsoleFullAccess** or **AmazonAthenaFullAccess**. Choose **Next: Review**.
7. For **Role name**, type a name for your role, for example **RedshiftCopyUnload**. Choose **Create role**.
8. The new role is available to all users on clusters that use the role. To restrict access to only specific users on specific clusters, or to clusters in specific regions, edit the trust relationship for the role. For more information, see [Restricting Access to IAM Roles \(p. 236\)](#).
9. Associate the role with your cluster. You can associate an IAM role with a cluster when you create the cluster, or you add the role to an existing cluster. For more information, see [Associating IAM Roles with Clusters \(p. 240\)](#).

Restricting Access to IAM Roles

By default, IAM roles that are available to an Amazon Redshift cluster are available to all users on that cluster. You can choose to restrict IAM roles to specific Amazon Redshift database users on specific clusters or to specific regions.

To permit only specific database users to use an IAM role, take the following steps.

To identify specific database users with access to an IAM role

1. Identify the Amazon Resource Name (ARN) for the database users in your Amazon Redshift cluster. The ARN for a database user is in the format: `arn:aws:redshift:region:account-id:dbuser:cluster-name/user-name`.
2. Open the [IAM Console](https://console.aws.amazon.com/) at [url="https://console.aws.amazon.com/](https://console.aws.amazon.com/).
3. In the navigation pane, choose **Roles**.
4. Choose the IAM role that you want to restrict to specific Amazon Redshift database users.
5. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**. A new IAM role that allows Amazon Redshift to access other AWS services on your behalf has a trust relationship as follows:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "redshift.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

6. Add a condition to the `sts:AssumeRole` action section of the trust relationship that limits the `sts:ExternalId` field to values that you specify. Include an ARN for each database user that you want to grant access to the role.

For example, the following trust relationship specifies that only database users `user1` and `user2` on cluster `my-cluster` in region `us-west-2` have permission to use this IAM role.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "redshift.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole",  
            "Condition": {  
                "StringEquals": {  
                    "sts:ExternalId": [  
                        "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user1",  
                        "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user2"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

}

7. Choose **Update Trust Policy**.

Restricting an IAM Role to an AWS Region

You can restrict an IAM role to only be accessible in a certain AWS Region. By default, IAM roles for Amazon Redshift are not restricted to any single region.

To restrict use of an IAM role by region, take the following steps.

To identify permitted regions for an IAM role

1. Open the [IAM Console](https://console.aws.amazon.com) at <https://console.aws.amazon.com>.
2. In the navigation pane, choose **Roles**.
3. Choose the role that you want to modify with specific regions.
4. Choose the **Trust Relationships** tab and then choose **Edit Trust Relationship**. A new IAM role that allows Amazon Redshift to access other AWS services on your behalf has a trust relationship as follows:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "redshift.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

5. Modify the **Service** list for the **Principal** with the list of the specific regions that you want to permit use of the role for. Each region in the **Service** list must be in the following format: `redshift.region.amazonaws.com`.

For example, the following edited trust relationship permits the use of the IAM role in the `us-east-1` and `us-west-2` regions only.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": [  
                    "redshift.us-east-1.amazonaws.com",  
                    "redshift.us-west-2.amazonaws.com"  
                ]  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

6. Choose **Update Trust Policy**

Chaining IAM Roles in Amazon Redshift

When you attach a role to your cluster, your cluster can assume that role to access Amazon S3, Athena, and AWS Glue on your behalf. If a role attached to your cluster doesn't have access to the necessary resources, you can chain another role, possibly belonging to another account. Your cluster then temporarily assumes the chained role to access the data. You can also grant cross-account access by chaining roles. Each role in the chain assumes the next role in the chain, until the cluster assumes the role at the end of chain. You can chain a maximum of 10 roles.

For example, suppose Company A wants to access data in an Amazon S3 bucket that belongs to Company B. Company A creates an AWS service role for Amazon Redshift named `RoleA` and attaches it to their cluster. Company B creates a role named `RoleB` that's authorized to access the data in the Company B bucket. To access the data in the Company B bucket, Company A runs a `COPY` command using an `iam_role` parameter that chains `RoleA` and `RoleB`. For the duration of the `COPY` operation, `RoleA` temporarily assumes `RoleB` to access the Amazon S3 bucket.

To chain roles, you establish a trust relationship between the roles. A role that assumes another role (for example, `RoleA`) must have a permissions policy that allows it to assume the next chained role (for example, `RoleB`). In turn, the role that passes permissions (`RoleB`) must have a trust policy that allows it to pass its permissions to the previous chained role (`RoleA`). For more information, see [Using IAM Roles](#) in the IAM User Guide.

The first role in the chain must be a role attached to the cluster. The first role, and each subsequent role that assumes the next role in the chain, must have a policy that includes a specific statement. This statement has the `Allow` effect on the `sts:AssumeRole` action and the Amazon Resource Name (ARN) of the next role in a `Resource` element. In our example, `RoleA` has the following permission policy that allows it to assume `RoleB`, owned by AWS account 210987654321.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1487639602000",  
            "Effect": "Allow",  
            "Action": [  
                "sts:AssumeRole"  
            ],  
            "Resource": "arn:aws:iam::210987654321:role/RoleB"  
        }  
    ]  
}
```

A role that passes to another role must establish a trust relationship with the role that assumes the role or with the AWS account that owns the role. In our example, `RoleB` has the following trust policy to establish a trust relationship with `RoleA`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::role/RoleA"  
            }  
        }  
    ]  
}
```

The following trust policy establishes a trust relationship with the owner of `RoleA`, AWS account 123456789012.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:root"  
            }  
        }  
    ]  
}
```

When you run an UNLOAD, COPY, or CREATE EXTERNAL SCHEMA command, you chain roles by including a comma-separated list of role ARNs in the `iam_role` parameter. The following shows the syntax for chaining roles in the `iam_role` parameter.

```
unload ('select * from venue limit 10')  
to 's3://acmedata/redshift/venue_pipe_'  
IAM_ROLE 'arn:aws:iam::<aws-account-id-1>:role/<role-name-1>[,arn:aws:iam::<aws-account-id-2>:role/<role-name-2>][,...]';
```

Note

The entire role chain is enclosed in single quotes and must not contain spaces.

In the following examples, RoleA is attached to the cluster belonging to AWS account 123456789012. RoleB, which belongs to account 210987654321, has permission to access the bucket named `s3://companyb/redshift/`. The following example chains RoleA and RoleB to UNLOAD data to the `s3://companyb/redshift/` bucket.

```
unload ('select * from venue limit 10')  
to 's3://companyb/redshift/venue_pipe_'  
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

The following example uses a COPY command to load the data that was unloaded in the previous example.

```
copy venue  
from 's3://companyb/redshift/venue_pipe_'  
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

In the following example, CREATE EXTERNAL SCHEMA uses chained roles to assume the role RoleB.

```
create external schema spectrumexample from data catalog  
database 'exampleredb' region 'us-west-2'  
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

Related Topics

- [Authorizing COPY, UNLOAD, and CREATE EXTERNAL SCHEMA Operations Using IAM Roles \(p. 240\)](#)

Authorizing COPY, UNLOAD, and CREATE EXTERNAL SCHEMA Operations Using IAM Roles

You can use the [COPY](#) command to load (or import) data into Amazon Redshift and the [UNLOAD](#) command to unload (or export) data from Amazon Redshift. When you use Amazon Redshift Spectrum, you use the [CREATE EXTERNAL SCHEMA](#) command to specify the location of an Amazon S3 bucket that contains your data. When you run the COPY, UNLOAD, or CREATE EXTERNAL SCHEMA commands, you must provide security credentials that authorize your Amazon Redshift cluster to read or write data to and from your target destination, such as an Amazon S3 bucket. The preferred method to supply security credentials is to specify an AWS Identity and Access Management (IAM) role. For COPY and UNLOAD, you can provide AWS access keys. For information about creating an IAM role, see [Authorizing Amazon Redshift to Access Other AWS Services on Your Behalf \(p. 235\)](#).

The steps for using an IAM role are as follows:

- Create an IAM role for use with your Amazon Redshift cluster.
- Associate the IAM role with the cluster.
- Include the IAM role's ARN when you call the COPY, UNLOAD, or CREATE EXTERNAL SCHEMA command.

In this topic you learn how to associate an IAM role with an Amazon Redshift cluster.

Associating IAM Roles with Clusters

After you have created an IAM role that authorizes Amazon Redshift to access other AWS services on your behalf, you must associate that role with an Amazon Redshift cluster before you can use the role to load or unload data.

Permissions Required to Associate an IAM Role with a Cluster

To associate an IAM role with a cluster, an IAM user must have `iam:PassRole` permission for that IAM role. This permission allows an administrator to restrict which IAM roles a user can associate with Amazon Redshift clusters.

The following example shows an IAM policy that can be attached to an IAM user that allows the user to take these actions:

- Get the details for all Amazon Redshift clusters owned by that user's account.
- Associate any of three IAM roles with either of two Amazon Redshift clusters.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "redshift:DescribeClusters",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "redshift:ModifyClusterIamRoles",  
                "redshift>CreateCluster"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": [
            "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-cluster",
            "arn:aws:redshift:us-east-1:123456789012:cluster:cluster:my-second-
redshift-cluster"
        ]
    },
    {
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": [
            "arn:aws:iam::123456789012:role/MyRedshiftRole",
            "arn:aws:iam::123456789012:role/SecondRedshiftRole",
            "arn:aws:iam::123456789012:role/ThirdRedshiftRole"
        ]
    }
]
```

Once an IAM user has the appropriate permissions, that user can associate an IAM role with an Amazon Redshift cluster for use with the COPY or UNLOAD command or other Amazon Redshift commands.

For more information on IAM policies, see [Overview of IAM Policies](#) in the *IAM User Guide*.

Managing IAM Role Association With a Cluster

You can associate an IAM role with an Amazon Redshift cluster when you create the cluster, or you can modify an existing cluster and add or remove one or more IAM role associations. Note the following:

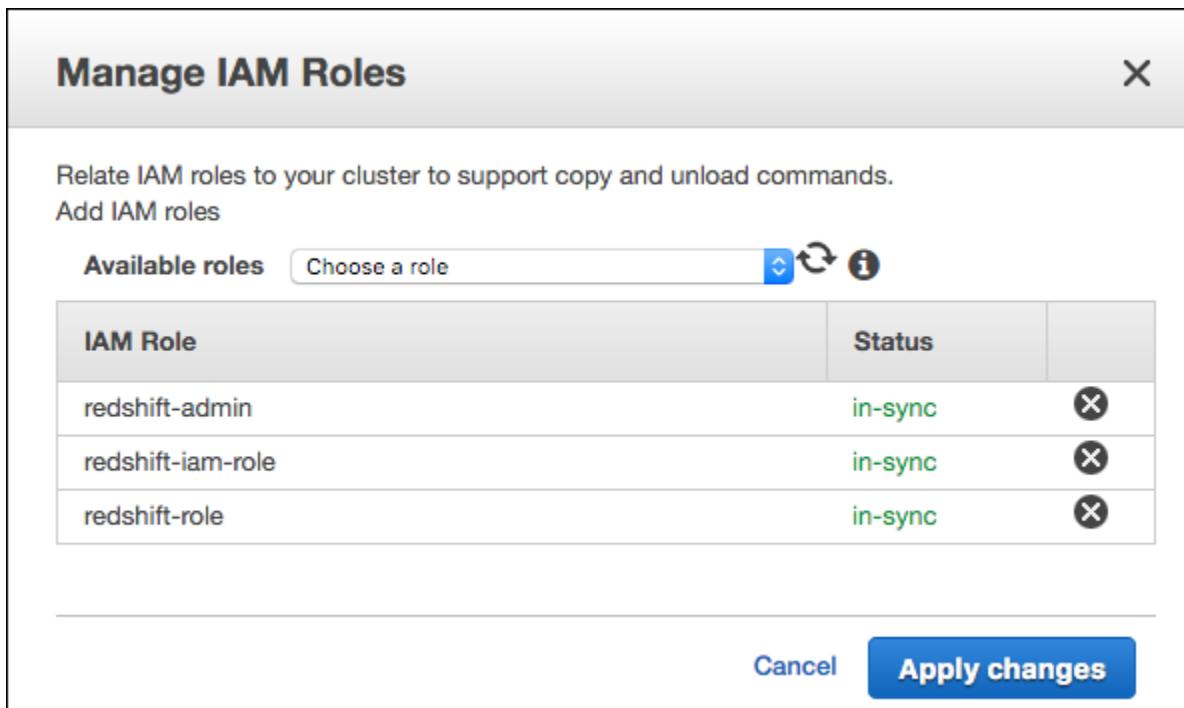
- You can associate a maximum of 10 IAM roles with an Amazon Redshift cluster.
- An IAM role can be associated with multiple Amazon Redshift clusters.
- An IAM role can be associated with an Amazon Redshift cluster only if both the IAM role and the cluster are owned by the same AWS account.

Using the Console to Manage IAM Role Associations

You can manage IAM role associations for a cluster with the console by using the following procedure.

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 **Clusters**.
3. In the list, choose the cluster that you want to manage IAM role associations for.
4. Choose **Manage IAM Roles**.
5. To associate an IAM role with the cluster, select your IAM role from the **Available roles** list. You can also manually enter an IAM role if you don't see it included the list (for example, if the IAM role hasn't been created yet).
6. To disassociate an IAM role from the cluster, choose X for the IAM role that you want to disassociate.
7. After you have finished modifying the IAM role associations for the cluster, choose **Apply Changes** to update the IAM roles that are associated with the cluster.

The **Manage IAM Roles** panel shows you the status of your cluster IAM role associations. Roles that have been associated with the cluster show a status of **in-sync**. Roles that are in the process of being associated with the cluster show a status of **adding**. Roles that are being disassociated from the cluster show a status of **removing**.



Using the AWS CLI to Manage IAM Role Associations

You can manage IAM role associations for a cluster with the AWS CLI by using the following approaches.

Associating an IAM Role with a Cluster Using the AWS CLI

To associate an IAM role with a cluster when the cluster is created, specify the Amazon Resource Name (ARN) of the IAM role for the `--iam-role-arns` parameter of the `create-cluster` command. You can specify up to 10 IAM roles to add when calling the `create-cluster` command.

Associating and disassociating IAM roles with Amazon Redshift clusters is an asynchronous process. You can get the status of all IAM role cluster associations by calling the `describe-clusters` command.

The following example associates two IAM roles with the newly created cluster named `my-redshift-cluster`.

```
aws redshift create-cluster \
--cluster-identifier "my-redshift-cluster" \
--node-type "dc1.large" \
--number-of-nodes 16 \
--iam-role-arns "arn:aws:iam::123456789012:role/RedshiftCopyUnload" \
"arn:aws:iam::123456789012:role/SecondRedshiftRole"
```

To associate an IAM role with an existing Amazon Redshift cluster, specify the Amazon Resource Name (ARN) of the IAM role for the `--add-iam-roles` parameter of the `modify-cluster-iam-roles` command. You can specify up to 10 IAM roles to add when calling the `modify-cluster-iam-roles` command.

The following example associates an IAM role with an existing cluster named `my-redshift-cluster`.

```
aws redshift modify-cluster-iam-roles \
--cluster-identifier "my-redshift-cluster" \
```

```
--add-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

Disassociating an IAM Role from a Cluster Using the AWS CLI

To disassociate an IAM role from a cluster, specify the ARN of the IAM role for the `--remove-iam-roles` parameter of the `modify-cluster-iam-roles` command. You can specify up to 10 IAM roles to remove when calling the `modify-cluster-iam-roles` command.

The following example removes the association for an IAM role for the 123456789012 AWS account from a cluster named `my-redshift-cluster`.

```
aws redshift modify-cluster-iam-roles \
--cluster-identifier "my-redshift-cluster" \
--remove-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

Listing IAM Role Associations for a Cluster Using the AWS CLI

To list all of the IAM roles that are associated with an Amazon Redshift cluster, and the status of the IAM role association, call the `describe-clusters` command. The ARN for each IAM role associated with the cluster is returned in the `IamRoles` list as shown in the following example output.

Roles that have been associated with the cluster show a status of `in-sync`. Roles that are in the process of being associated with the cluster show a status of `adding`. Roles that are being disassociated from the cluster show a status of `removing`.

```
{
    "Clusters": [
        {
            "ClusterIdentifier": "my-redshift-cluster",
            "NodeType": "dc1.large",
            "NumberOfNodes": 16,
            "IamRoles": [
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                }
            ],
            ...
        },
        {
            "ClusterIdentifier": "my-second-redshift-cluster",
            "NodeType": "dc1.large",
            "NumberOfNodes": 10,
            "IamRoles": [
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/ThirdRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                }
            ],
        }
    ]
}
```

```
        }      ...
    }
```

For more information on using the AWS CLI, see [AWS Command Line Interface User Guide](#).

Using the Amazon Redshift Management Interfaces

Amazon Redshift supports several management interfaces that you can use to create, manage, and delete Amazon Redshift clusters: the AWS SDKs, the AWS Command Line Interface (AWS CLI), and the Amazon Redshift management API.

The Amazon Redshift Query API – You can call this Amazon Redshift management API by submitting a query request. Query requests are HTTP or HTTPS requests that use the HTTP verbs `GET` or `POST` with a query parameter named `Action`. Calling the Query API is the most direct way to access the Amazon Redshift service. However, it requires that your application handle low-level details such as error handling and generating a hash to sign the request.

- For information about building and signing a Query API request, see [Signing an HTTP Request \(p. 248\)](#).
- For information about the Query API actions and data types for Amazon Redshift, see the [Amazon Redshift API Reference](#).

AWS SDKs – You can use the AWS SDKs to perform Amazon Redshift cluster-related operations. Several of the SDK libraries wrap the underlying Amazon Redshift Query API. They integrate the API functionality into the specific programming language and handle many of the low-level details, such as calculating signatures, handling request retries, and error handling. Calling the wrapper functions in the SDK libraries can greatly simplify the process of writing an application to manage an Amazon Redshift cluster.

- Amazon Redshift is supported by the AWS SDKs for Java, .NET, PHP, Python, Ruby, and Node.js. The wrapper functions for Amazon Redshift are documented in the reference manual for each SDK. For a list of the AWS SDKs and links to their documentation, see [Tools for Amazon Web Services](#).
- This guide provides examples of working with Amazon Redshift using the Java SDK. For more general AWS SDK code examples, see [Sample Code & Libraries](#).

AWS CLI – The CLI provides a set of command line tools that you can use to manage AWS services from Windows, Mac, and Linux computers. The AWS CLI includes commands based on the Amazon Redshift Query API actions.

- For information about installing and setting up the Amazon Redshift CLI, see [Setting Up the Amazon Redshift CLI \(p. 250\)](#).
- For reference material on the Amazon Redshift CLI commands, see [Amazon Redshift in the AWS CLI Reference](#).

Topics

- [Using the AWS SDK for Java with Amazon Redshift \(p. 246\)](#)
- [Signing an HTTP Request \(p. 248\)](#)
- [Setting Up the Amazon Redshift CLI \(p. 250\)](#)

Using the AWS SDK for Java with Amazon Redshift

The AWS SDK for Java provides a class named `AmazonRedshiftClientBuilder`, which you can use to interact with Amazon Redshift. For information about downloading the AWS SDK for Java, go to [AWS SDK for Java](#).

Note

The AWS SDK for Java provides thread-safe clients for accessing Amazon Redshift. As a best practice, your applications should create one client and reuse the client between threads.

You can use the `AmazonRedshiftClientBuilder` and `AwsClientBuilder` classes to configure an endpoint and create an `AmazonRedshift` client. You can then use the client object to create an instance of a `Cluster` object. The `Cluster` object includes methods that map to underlying Amazon Redshift Query API actions. (These actions are described in the Amazon Redshift [API Reference](#)). When you call a method, you must create a corresponding request object. The request object includes information that you must pass with the actual request. The `Cluster` object provides information returned from Amazon Redshift in response to the request.

The following example illustrates using the `AmazonRedshiftClientBuilder` class to configure an endpoint and then create a 2-node `ds1.xlarge` cluster.

```
String endpoint = "https://redshift.us-east-1.amazonaws.com/";
String region = "us-east-1";
AwsClientBuilder.EndpointConfiguration config = new
    AwsClientBuilder.EndpointConfiguration(endpoint, region);
AmazonRedshiftClientBuilder clientBuilder = AmazonRedshiftClientBuilder.standard();
clientBuilder.setEndpointConfiguration(config);
AmazonRedshift client = clientBuilder.build();

CreateClusterRequest request = new CreateClusterRequest()
    .withClusterIdentifier("exampleclusterusingjava")
    .withMasterUsername("masteruser")
    .withMasterUserPassword("12345678Aa")
    .withNodeType("ds1.xlarge")
    .withNumberOfNodes(2);

Cluster createResponse = client.createCluster(request);
System.out.println("Created cluster " + createResponse.getClusterIdentifier());
```

Running Java Examples for Amazon Redshift Using Eclipse

General Process of Running Java Code Examples Using Eclipse

1. Create a new **AWS Java Project** in Eclipse.

Follow the steps in [Setting Up the AWS Toolkit for Eclipse](#) in the [AWS Toolkit for Eclipse Getting Started Guide](#).

2. Copy the sample code from the section of this document that you are reading and paste it into your project as a new Java class file.
3. Run the code.

Running Java Examples for Amazon Redshift from the Command Line

General Process of Running Java Code Examples from the Command Line

1. Set up and test your environment as follows:
 - a. Create a directory to work in and in it create `src`, `bin`, and `sdk` subfolders.
 - b. Download the AWS SDK for Java and unzip it to the `sdk` subfolder you created. After you unzip the SDK, you should have four subdirectories in the `sdk` folder, including a `lib` and `third-party` folder.
 - c. Supply your AWS credentials to the SDK for Java. For more information, go to [Providing AWS Credentials in the AWS SDK for Java](#) in the [AWS SDK for Java Developer Guide](#).
 - d. Ensure that you can run the Java program compiler (`javac`) and the Java application launcher (`java`) from your working directory. You can test by running the following commands:

```
javac -help  
java -help
```

2. Put the code that you want to run in a `.java` file, and save the file in the `src` folder. To illustrate the process, we use the code from [Managing Cluster Security Groups Using the AWS SDK for Java \(p. 213\)](#) so that the file in the `src` directory is `CreateAndModifyClusterSecurityGroup.java`.
3. Compile the code.

```
javac -cp sdk/lib/aws-java-sdk-1.3.18.jar -d bin src  
\CreateAndModifyClusterSecurityGroup.java
```

If you are using a different version of the AWS SDK for Java, adjust the classpath (`-cp`) for your version.

4. Run the code. In the following command, line breaks are added for readability.

```
java -cp "bin;  
        sdk/lib/*;  
        sdk/third-party/commons-logging-1.1.1/*;  
        sdk/third-party/httpcomponents-client-4.1.1/*;  
        sdk/third-party/jackson-core-1.8/*"  
        CreateAndModifyClusterSecurityGroup
```

Change the class path separator as needed for your operating system. For example, for Windows, the separator is ";" (as shown), and for Unix, it is ":". Other code examples may require more libraries than are shown in this example, or the version of the AWS SDK you are working with may have different third-party folder names. For these cases, adjust the classpath (`-cp`) as appropriate.

To run samples in this document, use a version of the AWS SDK that supports Amazon Redshift. To get the latest version of the AWS SDK for Java, go to [AWS SDK for Java](#).

Setting the Endpoint

By default, the AWS SDK for Java uses the endpoint `https://redshift.us-east-1.amazonaws.com/`. You can set the endpoint explicitly with the `client.setEndpoint` method as shown in the following Java code snippet.

Example

```
client = new AmazonRedshiftClient(credentials);
client.setEndpoint("https://redshift.us-east-1.amazonaws.com/");
```

For a list of supported AWS regions where you can provision a cluster, go to the [Regions and Endpoints](#) section in the *Amazon Web Services Glossary*.

Signing an HTTP Request

Amazon Redshift requires that every request you send to the management API be authenticated with a signature. This topic explains how to sign your requests.

If you are using one of the AWS Software Development Kits (SDKs) or the AWS Command Line Interface, request signing is handled automatically, and you can skip this section. For more information about using AWS SDKs, see [Using the Amazon Redshift Management Interfaces \(p. 245\)](#). For more information about using the Amazon Redshift Command Line Interface, go to [Amazon Redshift Command Line Reference](#).

To sign a request, you calculate a digital signature by using a cryptographic hash function. A cryptographic hash is a function that returns a unique hash value that is based on the input. The input to the hash function includes the text of your request and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the `Authorization` header of your request.

Note

For API access, you need an access key ID and secret access key. Use IAM user access keys instead of AWS account root user access keys. IAM lets you securely control access to AWS services and resources in your AWS account. For more information about creating access keys, see [How Do I Get Security Credentials?](#) in the *AWS General Reference*.

After Amazon Redshift receives your request, it recalculates the signature by using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, Amazon Redshift processes the request; otherwise, the request is rejected.

Amazon Redshift supports authentication using [AWS Signature Version 4](#). The process for calculating a signature is composed of three tasks. These tasks are illustrated in the example that follows.

- [Task 1: Create a Canonical Request](#)

Rearrange your HTTP request into a canonical form. Using a canonical form is necessary because Amazon Redshift uses the same canonical form to calculate the signature it compares with the one you sent.

- [Task 2: Create a String to Sign](#)

Create a string that you will use as one of the input values to your cryptographic hash function. The string, called the *string to sign*, is a concatenation of the name of the hash algorithm, the request date, a *credential scope* string, and the canonicalized request from the previous task. The *credential scope* string itself is a concatenation of date, region, and service information.

- [Task 3: Create a Signature](#)

Create a signature for your request by using a cryptographic hash function that accepts two input strings, your string to sign and a *derived key*. The derived key is calculated by starting with your secret access key and using the credential scope string to create a series of hash-based message authentication codes (HMAC-SHA256).

Example Signature Calculation

The following example walks you through the details of creating a signature for [CreateCluster](#) request. You can use this example as a reference to check your own signature calculation method. Other reference calculations are included in the [Signature Version 4 Test Suite](#) of the Amazon Web Services Glossary.

You can use a GET or POST request to send requests to Amazon Redshift. The difference between the two is that for the GET request your parameters are sent as query string parameters. For the POST request they are included in the body of the request. The example below shows a POST request.

The example assumes the following:

- The time stamp of the request is `Fri, 07 Dec 2012 00:00:00 GMT`.
- The endpoint is US East (Northern Virginia) Region, `us-east-1`.

The general request syntax is:

```
https://redshift.us-east-1.amazonaws.com/
?Action=CreateCluster
&ClusterIdentifier=examplecluster
&MasterUsername=masteruser
&MasterUserPassword=12345678Aa
&NumberOfNodes=2
&NodeType=ds1.xlarge
&Version=2012-12-01
&x-amz-algorithm=AWS4-HMAC-SHA256
&x-amz-credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_request
&x-amz-date=20121207T00000Z
&x-amz-signedheaders=content-type;host;x-amz-date
```

The canonical form of the request calculated for [Task 1: Create a Canonical Request \(p. 248\)](#) is:

```
POST
/
content-type:application/x-www-form-urlencoded; charset=utf-8
host:redshift.us-east-1.amazonaws.com
x-amz-date:20121207T00000Z

content-type;host;x-amz-date
55141b5d2aff6042cc9d2af808fdf95ac78255e25b823d2dbd720226de1625d
```

The last line of the canonical request is the hash of the request body. The third line in the canonical request is empty because there are no query parameters for this API.

The string to sign for [Task 2: Create a String to Sign \(p. 248\)](#) is:

```
AWS4-HMAC-SHA256
20121207T00000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b60c627cc6c5b5b5a959b9902b5ac2187be80cbac0714
```

The first line of the *string to sign* is the algorithm, the second line is the time stamp, the third line is the *credential scope*, and the last line is a hash of the canonical request from [Task 1: Create a Canonical Request \(p. 248\)](#). The service name to use in the credential scope is `redshift`.

For [Task 3: Create a Signature \(p. 248\)](#), the derived key can be represented as:

```
derived key = HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey,"20121207"), "us-east-1"), "redshift"), "aws4_request")
```

The derived key is calculated as series of hash functions. Starting from the inner HMAC statement in the formula above, you concatenate the phrase "AWS4" with your secret access key and use this as the key to hash the data "us-east-1". The result of this hash becomes the key for the next hash function.

After you calculate the derived key, you use it in a hash function that accepts two input strings, your string to sign and the derived key. For example, if you use the secret access key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY and the string to sign given earlier, then the calculated signature is as follows:

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

The final step is to construct the Authorization header. For the demonstration access key AKIAIOSFODNN7EXAMPLE, the header (with line breaks added for readability) is:

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_request,  
SignedHeaders=content-type;host;x-amz-date,  
Signature=9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

Setting Up the Amazon Redshift CLI

This section explains how to set up and run the AWS CLI command line tools for use in managing Amazon Redshift. The Amazon Redshift command line tools run on the AWS Command Line Interface (AWS CLI), which in turn uses Python (<https://www.python.org/>). The AWS CLI can be run on any operating system that supports Python.

Installation Instructions

To begin using the Amazon Redshift command line tools, you first set up the AWS CLI, and then you add configuration files that define the Amazon Redshift CLI options.

If you have already installed and configured the AWS CLI for another AWS service, you can skip this procedure.

To install the AWS Command Line Interface

1. Go to [Getting Set Up with the AWS Command Line Interface](#), and then follow the instructions for installing the AWS CLI.

For CLI access, you need an access key ID and secret access key. Use IAM user access keys instead of AWS account root user access keys. IAM lets you securely control access to AWS services and resources in your AWS account. For more information about creating access keys, see [Understanding and Getting Your Security Credentials](#) in the [AWS General Reference](#).

2. Create a file containing configuration information such as your access keys, default region, and command output format. Then set the `AWS_CONFIG_FILE` environment variable to reference that file. For detailed instructions, go to [Configuring the AWS Command Line Interface](#) in the AWS Command Line Interface User Guide.
3. Run a test command to confirm that the AWS CLI interface is working. For example, the following command should display help information for the AWS CLI:

```
aws help
```

The following command should display help information for Amazon Redshift:

```
aws redshift help
```

For reference material on the Amazon Redshift CLI commands, go to [Amazon Redshift](#) in the AWS CLI Reference.

Getting Started with the AWS Command Line Interface

To help you get started using the command line interface, this section shows how to perform basic administrative tasks for an Amazon Redshift cluster. These tasks are very similar to those in the [Amazon Redshift Getting Started](#), but they are focused on the command line interface rather than the Amazon Redshift console.

This section walks you through the process of creating a cluster, creating database tables, uploading data, and testing queries. You will use the Amazon Redshift CLI to provision a cluster and to authorize necessary access permissions. You will then use the SQL Workbench client to connect to the cluster and create sample tables, upload sample data, and execute test queries.

Step 1: Before You Begin

If you don't already have an AWS account, you must sign up for one. Then you'll need to set up the Amazon Redshift command line tools. Finally, you'll need to download client tools and drivers in order to connect to your cluster.

Step 1.1: Sign Up for an AWS account

For information about signing up for an AWS user account, go to the [Amazon Redshift Getting Started](#).

Step 1.2: Download and Install the AWS Command Line Interface (CLI)

If you have not installed the AWS Command Line Interface, see [Setting Up the Amazon Redshift CLI \(p. 250\)](#).

Step 1.3: Download the Client Tools and Drivers

You can use any SQL client tools to connect to an Amazon Redshift cluster with PostgreSQL JDBC or ODBC drivers. If you do not currently have such software installed, you can use SQL Workbench, a free cross-platform tool that you can use to query tables in an Amazon Redshift cluster. The examples in this section will use the SQL Workbench client.

To download SQL Workbench and the PostgreSQL drivers, go to the [Amazon Redshift Getting Started Guide](#).

Step 2: Launch a Cluster

Now you're ready to launch a cluster by using the AWS Command Line Interface (CLI).

Important

The cluster that you're about to launch will be live (and not running in a sandbox). You will incur the standard usage fees for the cluster until you terminate it. For pricing information, go to [the Amazon Redshift pricing page](#).

If you complete the exercise described here in one sitting and terminate your cluster when you are finished, the total charges will be minimal.

The `create-cluster` command has a large number of parameters. For this exercise, you will use the parameter values that are described in the following table. Before you create a cluster in a production environment, we recommend that you review all the required and optional parameters so that your cluster configuration matches your requirements. For more information, see [create-cluster](#)

Parameter Name	Parameter Value for This Exercise
Cluster Identifier	<code>examplecluster</code>
Master Username	<code>masteruser</code>
Master Password	<code>TopSecret1</code>
Node Type	<code>ds1.xlarge</code> or the node size that you want to use. For more information, see Clusters and Nodes in Amazon Redshift (p. 6)
Cluster Type	<code>single-node</code>

To create your cluster, type the following command:

```
aws redshift create-cluster --cluster-identifier examplecluster --master-username masteruser --master-user-password TopSecret1 --node-type ds1.xlarge --cluster-type single-node
```

The cluster creation process will take several minutes to complete. To check the status, type the following command:

```
aws redshift describe-clusters --cluster-identifier examplecluster
```

The output will look similar to this:

```
{  
  "Clusters": [  
    {  
      ...output omitted...  
  
      "ClusterStatus": "creating",  
      "ClusterIdentifier": "examplecluster",  
  
      ...output omitted...  
    }  
  ]  
}
```

When the **ClusterStatus** field changes from `creating` to `available`, your cluster is ready for use.

In the next step, you will authorize access so that you can connect to the cluster.

Step 3: Authorize Inbound Traffic for Cluster Access

You must explicitly grant inbound access to your client in order to connect to the cluster. Your client can be an Amazon EC2 instance or an external computer.

When you created a cluster in the previous step, because you did not specify a security group, you associated the default cluster security group with the cluster. The default cluster security group contains no rules to authorize any inbound traffic to the cluster. To access the new cluster, you must add rules for inbound traffic, which are called ingress rules, to the cluster security group.

Ingress Rules for Applications Running on the Internet

If you are accessing your cluster from the Internet, you will need to authorize a Classless Inter-Domain Routing IP (CIDR/IP) address range. For this example, we will use a CIDR/IP rule of 192.0.2.0/24; you will need to modify this range to reflect your actual IP address and netmask.

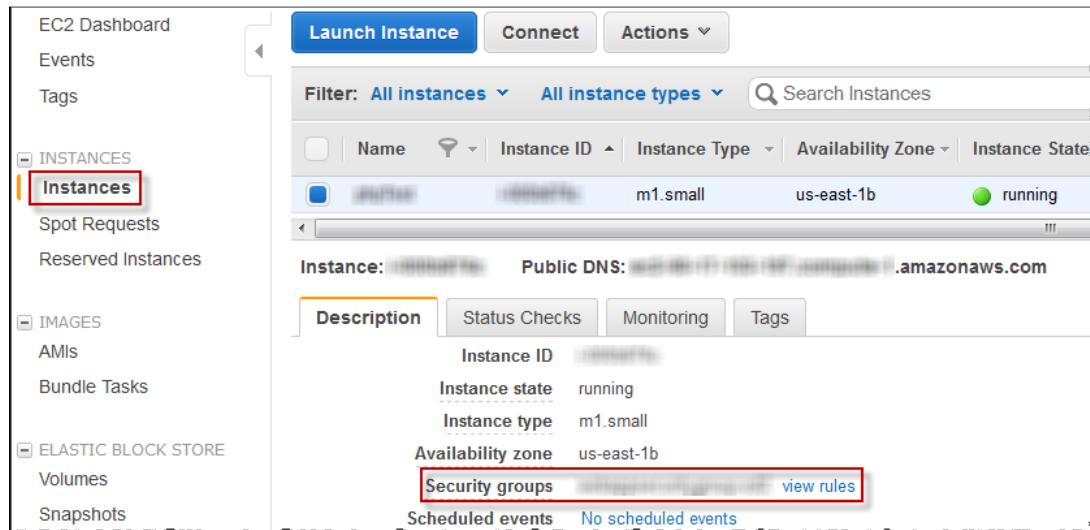
To allow network ingress to your cluster, type the following command:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name default  
--cidr 192.0.2.0/24
```

Ingress Rules for EC2 Instances

If you are accessing your cluster from an Amazon EC2 instance, you will need to authorize an Amazon EC2 security group. To do so, you specify the security group name, along with the 12-digit account number of the EC2 security group owner.

You can use the Amazon EC2 console to determine the EC2 security group associated with your instance:



To find your AWS account number, go to <https://aws.amazon.com/> and sign in to the My Account page. Your AWS account number is shown in the upper right-hand corner of that page.

For this example, we will use `myec2securitygroup` for the Amazon EC2 security group name, and `123456789012` for the account number. You will need to modify these to suit your needs.

To allow network ingress to your cluster, type the following command:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name  
default --ec2-security-group-name myec2securitygroup --ec2-security-group-owner  
123456789012
```

Step 4: Connect to Your Cluster

Now that you have added an ingress rule to the default cluster security group, incoming connections from a specific CIDR/IP or EC2 Security Group to `examplecluster` are authorized.

You are now ready to connect to the cluster.

For information about connecting to your cluster, go to the [Amazon Redshift Getting Started Guide](#).

Step 5: Create Tables, Upload Data, and Try Example Queries

For information about creating tables, uploading data, and issuing queries, go to the [Amazon Redshift Getting Started](#).

Step 6: Delete Your Sample Cluster

After you have launched a cluster and it is available for use, you are billed for the time the cluster is running, even if you are not actively using it. When you no longer need the cluster, you can delete it.

When you delete a cluster, you must decide whether to create a final snapshot. Because this is an exercise and your test cluster should not have any important data in it, you can skip the final snapshot.

To delete your cluster, type the following command:

```
aws redshift delete-cluster --cluster-identifier examplecluster --skip-final-cluster-snapshot
```

Congratulations! You successfully launched, authorized access to, connected to, and terminated a cluster.

Monitoring Amazon Redshift Cluster Performance

Amazon Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. In this section, we discuss the types of data you can work with in Amazon Redshift and specifically, in the Amazon Redshift console.

Topics

- [Overview \(p. 255\)](#)
- [Amazon Redshift Performance Data \(p. 256\)](#)
- [Working with Performance Data in the Amazon Redshift Console \(p. 261\)](#)

Overview

The performance data that you can use in the Amazon Redshift console falls into two categories:

- **Amazon CloudWatch Metrics** – Amazon CloudWatch metrics help you monitor physical aspects of your cluster, such as CPU utilization, latency, and throughput. Metric data is displayed directly in the Amazon Redshift console. You can also view it in the Amazon CloudWatch console. Alternatively, you can consume it in any other way you work with metrics, such as with the AWS CLI or one of the AWS SDKs.
- **Query/Load Performance Data** – Performance data helps you monitor database activity and performance. This data is aggregated in the Amazon Redshift console to help you easily correlate what you see in Amazon CloudWatch metrics with specific database query and load events. You can also create your own custom performance queries and run them directly on the database. Query and load performance data is displayed only in the Amazon Redshift console. It is not published as Amazon CloudWatch metrics.

Performance data is integrated into the Amazon Redshift console, yielding a richer experience in the following ways:

- Performance data associated with a cluster is displayed contextually when you view a cluster, where you might need it to make decisions about the cluster such as resizing.
- Some performance metrics are displayed in more appropriately scaled units in the Amazon Redshift console as compared to Amazon CloudWatch. For example, `WriteThroughput`, is displayed in GB/s (as compared to bytes/s in Amazon CloudWatch), which is a more relevant unit for the typical storage space of a node.
- You can easily display performance data for the nodes of a cluster together on the same graph. This way, you can easily monitor the performance of all nodes of a cluster. You can also view performance data for each node.

Amazon Redshift provides performance data (both Amazon CloudWatch metrics and query and load data) at no additional charge. Performance data is recorded every minute. You can access historical values of performance data in the Amazon Redshift console. For detailed information about using Amazon CloudWatch to access the Amazon Redshift performance data that is exposed as Amazon CloudWatch metrics, see [What is Amazon CloudWatch?](#) in the *Amazon CloudWatch User Guide*.

Amazon Redshift Performance Data

Using Amazon CloudWatch metrics for Amazon Redshift, you can get information about your cluster's health and performance and see information at the node level. When working with these metrics, keep in mind that each metric has one or more dimensions associated with it. These dimensions tell you what the metric is applicable to, that is the scope of the metric. Amazon Redshift has the following two dimensions:

- Metrics that have a `NodeID` dimension are metrics that provide performance data for nodes of a cluster. This set of metrics includes leader and compute nodes. Examples of these metrics include `CPUUtilization`, `ReadIOPS`, `WriteIOPS`.
- Metrics that have only a `ClusterIdentifier` dimension are metrics that provide performance data for clusters. Examples of these metrics include `HealthStatus` and `MaintenanceMode`.

Note

In some metric cases, a cluster-specific metric represents an aggregation of node behavior. In these cases, take care in the interpretation of the metric value because the leader node's behavior is aggregated with the compute node.

For general information about CloudWatch metrics and dimensions, see [Amazon CloudWatch Concepts](#) in the [Amazon CloudWatch User Guide](#).

For a further description of CloudWatch metrics for Amazon Redshift, see the following sections.

Amazon Redshift Metrics

The AWS/Redshift namespace includes the following metrics.

Metric	Description
<code>CPUUtilization</code>	<p>The percentage of CPU utilization. For clusters, this metric represents an aggregation of all nodes (leader and compute) CPU utilization values.</p> <p>Units: Percent</p> <p>Dimensions: <code>NodeID</code>, <code>ClusterIdentifier</code></p>
<code>DatabaseConnections</code>	<p>The number of database connections to a cluster.</p> <p>Units: Count</p> <p>Dimensions: <code>ClusterIdentifier</code></p>
<code>HealthStatus</code>	<p>Indicates the health of the cluster. Every minute the cluster connects to its database and performs a simple query. If it is able to perform this operation successfully, the cluster is considered healthy. Otherwise, the cluster is unhealthy. An unhealthy status can occur when the cluster database is under extremely heavy load or if there is a configuration problem with a database on the cluster.</p> <p>Note</p> <p>In Amazon CloudWatch this metric is reported as 1 or 0 whereas in the Amazon Redshift console, this metric is displayed with the words <code>HEALTHY</code> or <code>UNHEALTHY</code> for convenience. When this metric is displayed in the Amazon Redshift console, sampling averages are ignored</p>

Metric	Description
	<p>and only HEALTHY or UNHEALTHY are displayed. In Amazon CloudWatch, values different than 1 and 0 may occur because of sampling issue. Any value below 1 for <code>HealthStatus</code> is reported as 0 (UNHEALTHY).</p> <p>Units: 1/0 (HEALTHY/UNHEALTHY in the Amazon Redshift console)</p> <p>Dimensions: <code>ClusterIdentifier</code></p>
<code>MaintenanceMode</code>	<p>Indicates whether the cluster is in maintenance mode.</p> <p>Note In Amazon CloudWatch this metric is reported as 1 or 0 whereas in the Amazon Redshift console, this metric is displayed with the words ON or OFF for convenience. When this metric is displayed in the Amazon Redshift console, sampling averages are ignored and only ON or OFF are displayed. In Amazon CloudWatch, values different than 1 and 0 may occur because of sampling issues. Any value greater than 0 for <code>MaintenanceMode</code> is reported as 1 (ON).</p> <p>Units: 1/0 (ON/OFF in the Amazon Redshift console).</p> <p>Dimensions: <code>ClusterIdentifier</code></p>
<code>NetworkReceiveThroughput</code>	<p>The rate at which the node or cluster receives data.</p> <p>Units: Bytes/seconds (MB/s in the Amazon Redshift console)</p> <p>Dimensions: <code>NodeID, ClusterIdentifier</code></p>
<code>NetworkTransmitThroughput</code>	<p>The rate at which the node or cluster writes data.</p> <p>Units: Bytes/second (MB/s in the Amazon Redshift console)</p> <p>Dimensions: <code>NodeID, ClusterIdentifier</code></p>
<code>PercentageDiskSpaceUsed</code>	<p>The percent of disk space used.</p> <p>Units: Percent</p> <p>Dimensions: <code>NodeID, ClusterIdentifier</code></p>
<code>QueriesCompletedPerSecond</code>	<p>This metric is used to determine Query Throughput. The metric is the average number of queries completed per second, reported in five-minute intervals.</p> <p>Units: Count/second</p> <p>Dimensions: latency</p>
<code>QueryDuration</code>	<p>The average amount of time to complete a query. Reported in five-minute intervals.</p> <p>Units: Microseconds</p> <p>Dimensions: latency</p>

Metric	Description
QueryRuntimeBreakdown	<p>The amount of time all active queries have spent in various stages of execution during the previous five minutes.</p> <p>Units: Milliseconds</p> <p>Dimensions: Stage</p>
ReadIOPS	<p>The average number of disk read operations per second.</p> <p>Units: Count/second</p> <p>Dimensions: NodeID</p>
ReadLatency	<p>The average amount of time taken for disk read I/O operations.</p> <p>Units: Seconds</p> <p>Dimensions: NodeID</p>
ReadThroughput	<p>The average number of bytes read from disk per second.</p> <p>Units: Bytes (GB/s in the Amazon Redshift console)</p> <p>Dimensions: NodeID</p>
TotalTableCount	<p>The number of user tables created at a particular point in time. This total doesn't include Amazon Redshift Spectrum tables.</p> <p>Units: Count</p> <p>Dimensions: ClusterIdentifier</p>
WLMQueriesCompletedPerSecond	<p>This metric is used to determine Query Throughput for a Workload Management queue. The metric is the average number of queries completed per second for a Workload Management (WLM) queue, reported in five-minute intervals.</p> <p>Units: Count/second</p> <p>Dimensions: wlmid</p>
WLMQueryDuration	<p>The average length of time to complete a query for a Workload Management (WLM) queue. Reported in five-minute intervals.</p> <p>Units: Microseconds</p> <p>Dimensions: wlmid</p>
WLMQueueLength	<p>The number of queries in the queue for a Workload Management (WLM) queue.</p> <p>Units: Count</p> <p>Dimensions: service class</p>

Metric	Description
WriteIOPS	The average number of disk write operations per second. Units: Count/seconds Dimensions: NodeID
WriteLatency	The average amount of time taken for disk write I/O operations. Units: Seconds Dimensions: NodeID
WriteThroughput	The average number of bytes written to disk per second. Units: Bytes (GB/s in the Amazon Redshift console) Dimensions: NodeID

Amazon Redshift data can be filtered along any of the dimensions in the table following.

Dimension	Description
latency	Values are short, medium, and long. Short is less than 10 seconds, medium is between 10 seconds and 10 minutes, and long is over 10 minutes.
NodeID	Filters requested data that is specific to the nodes of a cluster. NodeID will be either "Leader", "Shared", or "Compute-N" where N is 0, 1, ... for the number of nodes in the cluster. "Shared" means that the cluster has only one node, i.e. the leader node and compute node are combined. Metrics are reported for the leader node and compute nodes only for CPUUtilization, NetworkTransmitThroughput, and ReadIOPS. Other metrics that use the NodeId dimension are reported only for compute nodes.
ClusterIdentifier	Filters requested data that is specific to the cluster. Metrics that are specific to clusters include HealthStatus, MaintenanceMode, and DatabaseConnections. In general metrics in for this dimension (e.g. ReadIOPS) that are also metrics of nodes represent an aggregate of the node metric data. You should take care in interpreting these metrics because they aggregate behavior of leader and compute nodes.
service class	The identifier for a WLM service class.
Stage	The execution stages for a query. The possible values are: <ul style="list-style-type: none"> • QueryPlanning: Time spent parsing and optimizing SQL statements . • QueryWaiting: Time spent waiting in the wlm queue. • QueryExecutingRead: Time spent executing read queries. • QueryExecutingInsert: Time spent executing insert queries.

Dimension	Description
	<ul style="list-style-type: none"> • <code>QueryExecutingDelete</code>: Time spent executing delete queries. • <code>QueryExecutingUpdate</code>: Time spent executing update queries. • <code>QueryExecutingCtas</code>: Time spent executing create table as ... queries. • <code>QueryExecutingUnload</code>: Time spent executing unload queries. • <code>QueryExecutingCopy</code>: Time spent executing copy queries. • <code>QueryCommit</code>: Time spent committing.
<code>wmlid</code>	The identifier for a Workload Management Queue.

Amazon Redshift Query/Load Performance Data

In addition to the Amazon CloudWatch metrics, Amazon Redshift provides query and load performance data. Query and load performance data can be used to help you understand the relation between database performance and cluster metrics. For example, if you notice that a cluster's CPU spiked, you can find the spike on the cluster CPU graph and see the queries that were running at that time. Conversely, if you are reviewing a specific query, metric data (like CPU) is displayed in context so that you can understand the query's impact on cluster metrics.

Query and load performance data are not published as Amazon CloudWatch metrics and can only be viewed in the Amazon Redshift console. Query and load performance data are generated from querying with your database's system tables (see [System Tables Reference](#) in the *Amazon Redshift Developer Guide*). You can also generate your own custom database performance queries, but we recommend starting with the query and load performance data presented in the console. For more information about measuring and monitoring your database performance yourself, see [Managing Performance](#) in the *Amazon Redshift Developer Guide*.

The following table describes different aspects of query and load data you can access in the Amazon Redshift console.

Query/Load Data	Description
Query summary	A list of queries in a specified time period. The list can be sorted on values such as query ID, query run time, and status. Access this data in the Queries tab of the cluster detail page.
Query Detail	<p>Provides details on a particular query including:</p> <ul style="list-style-type: none"> • Query properties such as the query ID, type, cluster the query was run on, and run time. • Details such as the status of the query and the number of errors. • The SQL statement that was run. • An explain plan if available. • Cluster performance data during the query execution (see Amazon Redshift Performance Data (p. 256)).
Load Summary	<p>Lists all the loads in a specified time period. The list can be sorted on values such as query ID, query run time, and status. Access this data in the Loads tab of the cluster detail page. Access this data in the Queries tab of the cluster detail page.</p>
Load Detail	Provides details on a particular load operation including:

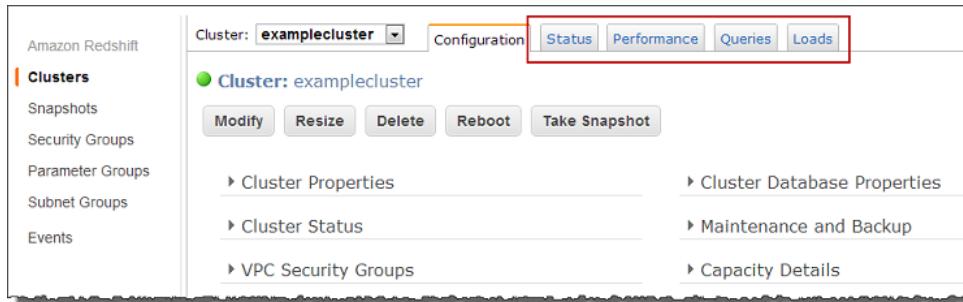
Query/Load Data	Description
	<ul style="list-style-type: none"> • Load properties such as the query ID, type, cluster the query was run on, and run time. • Details such as the status of the load and the number of errors. • The SQL statement that was run. • A list of loaded files. • Cluster performance data during the load operation (see Amazon Redshift Performance Data (p. 256)).

Working with Performance Data in the Amazon Redshift Console

This section explains how to view performance data in the Amazon Redshift console, which includes information about cluster and query performance. Additionally, you can create alarms on cluster metrics directly from the Amazon Redshift console.

When you view performance data in the Amazon Redshift console, you view it by cluster. The performance data graphs for a cluster are designed to give you access to data to answer your most common performance questions. For some performance data (see [Amazon Redshift Performance Data \(p. 256\)](#)), you can also use Amazon CloudWatch to further customize your metrics graphs, for example, choose longer times or combine metrics across clusters. For more information about working with the Amazon CloudWatch console, see [Working with Performance Metrics in the Amazon CloudWatch Console \(p. 278\)](#).

To start working with performance data, find your cluster in the *cluster performance dashboard*. The dashboard is a list of clusters that shows at a glance the status of the cluster (for example, **available**), the **DB Health** of the cluster (for example, **healthy**), whether the cluster is undergoing maintenance, and count of recent events. From the dashboard, select a cluster to work with and go to the details of the cluster. From this page, you can access the **Events+Alarms**, **Performance**, **Queries**, and **Loads** tabs that contain the performance data. These tabs are shown in the following example.



Viewing Cluster Performance Data

Cluster metrics in Amazon Redshift enable the following common performance use cases:

- Determine if cluster metrics are abnormal over a specified time range and, if so, identify the queries responsible for the performance hit.
- Check if historical or current queries are impacting cluster performance. If you identify a problematic query, you can view details about it including the cluster performance during the query's execution. You can use this information in diagnosing why the query was slow and what can be done to improve its performance.

The default cluster view shows all nodes graphed together, an Average statistic, and data for the last hour. You can change this view as needed. Some metrics, such as `HealthStatus`, are only applicable for the leader node while others, such as `WriteOps`, are only applicable for compute nodes. Switching the node display mode resets all filters.

To view cluster performance data

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 **Clusters**.
3. For **Cluster**, choose the magnifying glass icon beside the cluster for which you want to view performance data.

4. Choose the **Performance** tab.

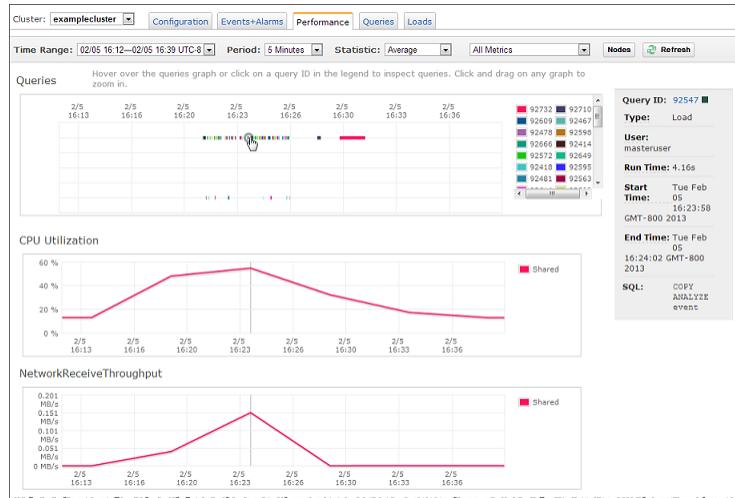
By default, the performance view displays cluster performance over the past hour. If you need to fine-tune the view, you have *filters* that you can use as described in the following table.

To Do This	Use This Filter
Change the time range for which data is displayed	Select a time range from the Time Range list. By default, the last hour is shown.
Change the period for which data is displayed	Select a period from the Period list. By default, a five-minute period is shown. Use a period smaller than five minutes if you need more detail when investigating a metric (drilling in) and displaying metrics over a small time period, for example ten minutes. Similarly, use a period greater than five minutes when viewing metrics over a large period of time, for example days.
Change the statistic that is displayed for metrics	Select a statistic from the Statistic list. By default, the Average statistic is used.

To Do This	Use This Filter
Change what metrics are shown, all or a specific metric	Choose a metric from the Metrics list. By default, all metrics are shown.
Change whether node metrics are displayed separately or together on the same graph	Choose Nodes . By default, node data for a given metric is shown on a combined graph. If you choose to display node data on separate graphs, you can additionally show or hide individual nodes.

Cluster Metrics: Examples

The following example shows `CPUUtilization` and `NetworkReceiveThroughput` metrics for a single node cluster. In this case, the graphs for cluster metrics show one line marked as **Shared** because the leader and compute node are combined. The example shows that multiple queries were run in the time period shown. On the **Queries** graph, the cursor is positioned over the query running at the peak values of the two metrics and the **Query ID** is displayed on the right. You can then choose the Query ID to find out more about the query running.

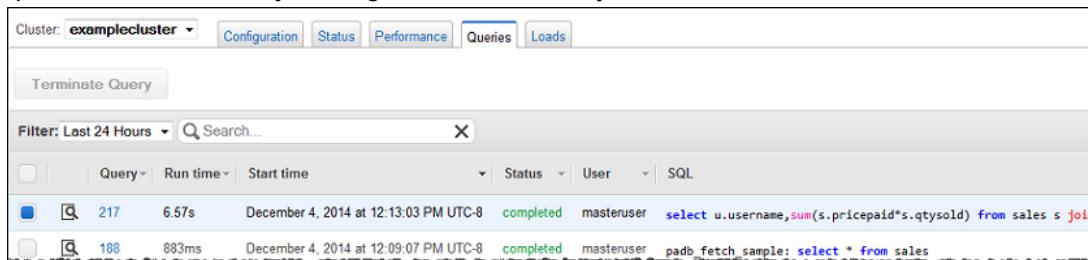


The following example shows the `NetworkReceiveThroughput` for a cluster with two nodes. It shows a line for the leader and two compute nodes. The leader node metrics appear as flat and aren't of interest because data is only loaded on the compute nodes. The example shows that one long query ran in the time period shown. On the **Queries** graph, the cursor is positioned over the long running query and the **Query ID** is displayed on the right. You can then choose the **Query ID** to find out more about the query running. The `NetworkReceiveThroughput` value is displayed during the query execution.



Viewing Query Performance Data

The Amazon Redshift console provides information about performance of queries that run in the database. You can use this information to identify and troubleshoot queries that take a long time to process and that create bottlenecks preventing other queries from processing efficiently. You can use the **Queries** tab on the cluster details page to view this information. The **Queries** tab shows a table that lists queries that are currently running or have run recently in the cluster.

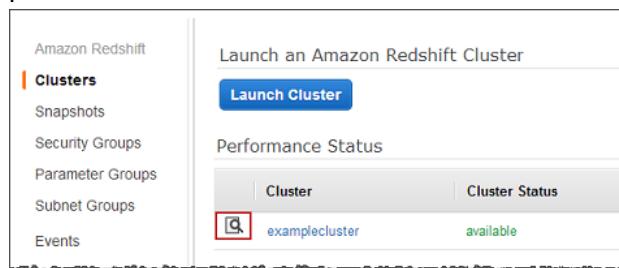


Use the button bar, shown following, to refresh the data in the table, to configure the columns that appear in the table, or to open the Amazon Redshift documentation.



To view query performance data

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 **Clusters**.
3. For **Cluster**, choose the magnifying glass icon beside the cluster for which you want to view performance data.



4. Choose the **Queries** tab.

By default, the **Queries** tab displays query performance over the past 24 hours. To change the data displayed, use the **Filter** list to select the time period for which you want to view queries. You can also type a keyword in the **Search** box to search for queries that match your search criteria.

Terminating a Running Query

You can also use the **Queries** page to terminate a query that is currently in progress.

Note

The ability to terminate queries and loads in the Amazon Redshift console requires specific permission. If you want users to be able to terminate queries and loads, make sure to add the `redshift:CancelQuerySession` action to your AWS Identity and Access Management (IAM) policy. This requirement applies whether you select the **Amazon Redshift Read Only** AWS-managed policy or create a custom policy in IAM. Users who have the **Amazon Redshift Full Access** policy already have the necessary permission to terminate queries and loads. For more information about actions in IAM policies for Amazon Redshift, see [Managing Access to Resources \(p. 190\)](#).

To terminate a running query

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 **Clusters**.
3. For **Cluster**, choose the cluster you want to open.
4. Choose the **Queries** tab.
5. Do one of the following:
 - In the list, select the query or queries that you want to terminate, and choose **Terminate Query**.
 - In the list, open a query if you want to review the query information first, and then choose **Terminate Query**.
6. In the **Terminate Queries** dialog box, choose **Confirm**.

Viewing Query Details

You can view details for a particular query by clicking an individual query in the table on the **Queries** page to open the **Query ID** view. The following list describes the information available for individual queries:

- **Query Properties.** Displays a summary of information about the query such as the query ID, the database user who ran the query, and the duration.
- **Details.** Displays the status of the query.
- **SQL.** Displays the query text in a friendly, human-readable format.
- **Query Execution Details.** Displays information about how the query was processed. This section includes both planned and actual execution data for the query. For information on using the **Query Execution Details** section, see [Analyzing Query Execution \(p. 266\)](#).
- **Cluster Performance During Query Execution.** Displays performance metrics from Amazon CloudWatch. For information on using the **Cluster Performance During Query Execution** section, see [Viewing Cluster Performance During Query Execution \(p. 270\)](#).

The **Query** view looks similar to the following when you open it.

The screenshot shows the 'Queries' tab selected in the navigation bar. Below it, a detailed view of a specific query is shown.

Query properties:

Cluster	examplecluster	Run time	8.06s
Query ID	189829	Start time	Fri May 04 17:21:09 GMT-700 2018
Type	Query	End time	Fri May 04 17:21:17 GMT-700 2018
User	gutenkauf	Status	Completed

SQL:

```

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
  
```

Query execution details:

Plan tab selected.

```

XN Merge (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
  -> XN Network (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
    -> XN Sort (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
      -> XN Hash Join DS_DIST_INNER (cost=1000000004530.79..1000001005655.67 rows=10 width=27)
        -> XN Seq Scan on users (cost=0.00..499.90 rows=49990 width=23)
        -> XN Hash (cost=1000000004530.77..1000000004530.77 rows=10 width=12)
          -> XN Subquery Scan q (cost=1000000004530.64..1000000004530.77 rows=10 width=12)
            -> XN Limit (cost=1000000004530.64..1000000004530.67 rows=10 width=6)
      -> XN HashAggregate (cost=2586.84..2651.00 rows=25665 width=6)
        -> XN Seq Scan on sales (cost=0.00..1724.56 rows=172456 width=6)
  
```

Analyzing Query Execution

The **Query Execution Details** section of the **Query** view provides information about the way the query was processed. This section combines data from [SVL_QUERY_REPORT](#), [STL_EXPLAIN](#), and other system views and tables.

The **Query Execution Details** section has three tabs:

- **Plan.** This tab shows the explain plan for the query that is displayed.

▼ Query execution details

Plan **Actual** **Metrics**

```
XN Merge (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
-> XN Network (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
-> XN Sort (cost=2000001005655.84..2000001005655.86 rows=10 width=27)
-> XN Hash Join DS_DIST_INNER (cost=1000000004530.79..1000001005655.67 rows=10 width=27)
-> XN Seq Scan on users (cost=0.00..499.90 rows=49990 width=23)
-> XN Hash (cost=1000000004530.77..1000000004530.77 rows=10 width=12)
-> XN Subquery Scan q (cost=1000000004530.64..1000000004530.77 rows=10 width=12)
-> XN Limit (cost=1000000004530.64..1000000004530.67 rows=10 width=6)
-> XN HashAggregate (cost=2586.84..2651.00 rows=25665 width=6)
-> XN Seq Scan on sales (cost=0.00..1724.56 rows=172456 width=6)
```

The information on the **Plan** tab is analogous to running the EXPLAIN command in the database. The EXPLAIN command examines your query text, and returns the query plan. You use this information to evaluate queries, and revise them for efficiency and performance if necessary. The EXPLAIN command doesn't actually run the query.

The following example shows a query that returns the top five sellers in San Diego, based on the number of tickets sold in 2008, and the query plan for that query.

```
explain
select sellerid, username, (firstname || ' ' || lastname) as name,
city, sum(qtysold)
from sales, date, users
where sales.sellerid = users.userid
and sales.dateid = date.dateid
and year = 2008
and city = 'San Diego'
group by sellerid, username, name, city
order by 5 desc
limit 5;
```

QUERY PLAN
XN HashAggregate (cost=10347367091.97..10347367466.90 rows=49990 width=24) -> XN Hash Join DS_DIST_OUTER (cost=624.88..10347366229.69 rows=172456 width=24) Outer Dist Key: s.buyerid Hash Cond: ("outer".buyerid = "inner".userid) -> XN Seq Scan on sales s (cost=0.00..1724.56 rows=172456 width=16) -> XN Hash (cost=499.90..499.90 rows=49990 width=16) -> XN Seq Scan on users u (cost=0.00..499.90 rows=49990 width=16)

For more information about understanding the explain plan, see [Analyzing the Explain Plan](#) in the *Amazon Redshift Database Developer Guide*.

When you actually run the query (omitting the EXPLAIN command), the engine might find ways to optimize the query performance and change the way it processes the query. The actual performance data for the query is stored in the system views, such as [SVL_QUERY_REPORT](#) and [SVL_QUERY_SUMMARY](#).

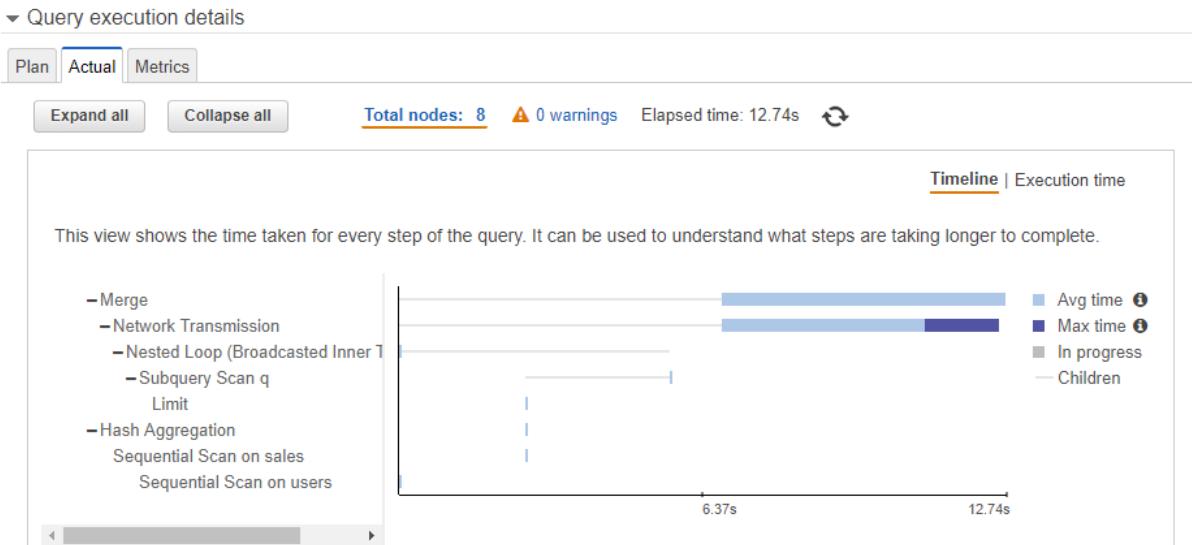
The Amazon Redshift console uses a combination of [STL_EXPLAIN](#), [SVL_QUERY_REPORT](#), and other system views and tables to present the actual query performance and compare it to the explain plan for the query. This information appears on the **Actual** tab. In some cases, you might see that the explain plan and the actual query execution steps differ. In these cases, you might need to perform some operations in the database, such as [ANALYZE](#), to update statistics and make the explain plan more effective.

Additionally, sometimes the query optimizer breaks complex SQL queries into parts and creates temporary tables with the naming convention `volt_tt_guid` to process the query more efficiently. In

this case, both the explain plan and the actual query execution summary apply to the last statement that was run. You can review previous query IDs to see the explain plan and actual query execution summary for each of the corresponding parts of the query.

For more information about the difference between the explain plan and system views and logs, see [Analyzing the Query Summary](#) in the *Amazon Redshift Database Developer Guide*.

- **Actual.** This tab shows the actual steps and statistics for the query that was executed. This information displays in a textual hierarchy and visual charts for **Timeline** and **Execution time**.



The **Timeline** view shows the sequence in which the actual steps of the query are executed.

The **Execution time** view shows the time taken for every step of the query. It can be used to understand what steps are taking longer to complete.

Each view shows **Avg time** and **Max time**.

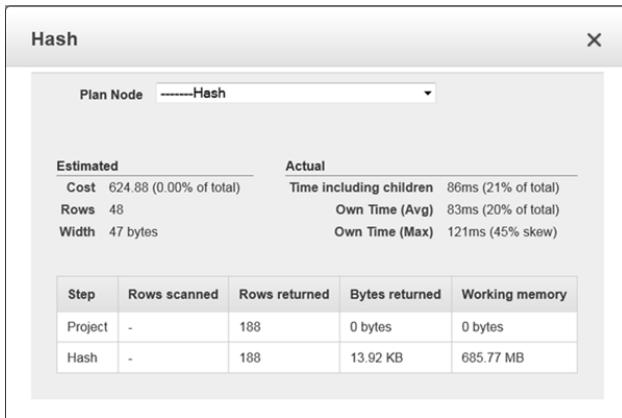
The **Avg** statistic shows the average execution time for the step across data slices, and the percentage of the total query runtime that represents. The **Max** statistic shows the longest execution time for the step on any of the data slices, and the skew. The *skew* is the difference between the average and maximum execution times for the step.

You might want to investigate a step if two conditions are both true. One condition is that the maximum execution time is consistently more than twice the average execution time over multiple runs of the query. The other condition is that the step also takes a significant amount of time (for example, being one of the top three steps in execution time in a large query).

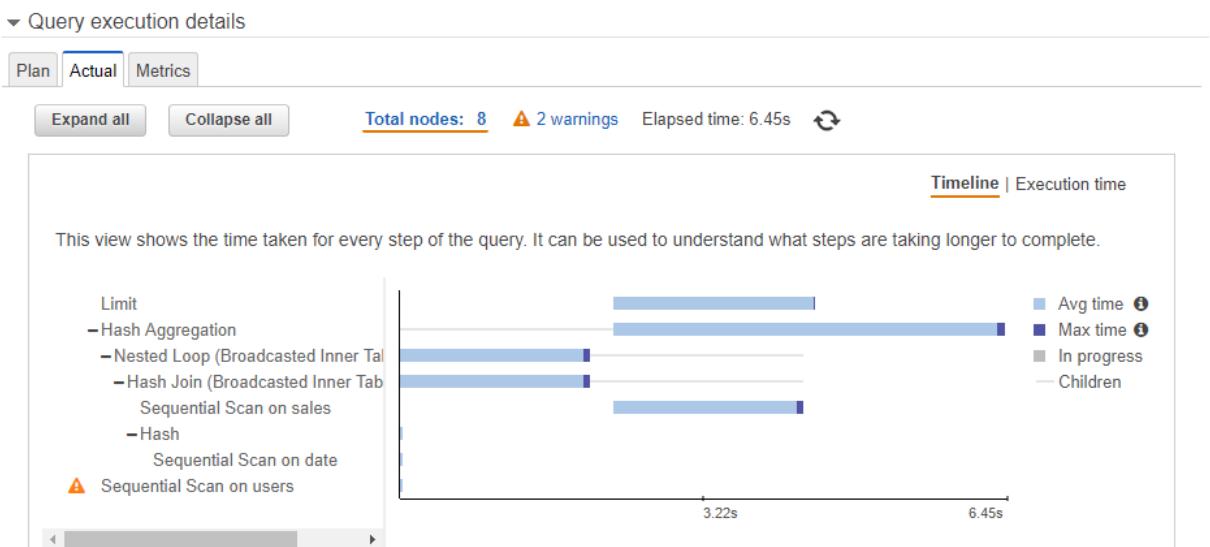
Note

When possible, you should run a query twice to see what its execution details typically are. Compilation adds overhead to the first run of the query that is not present in subsequent runs.

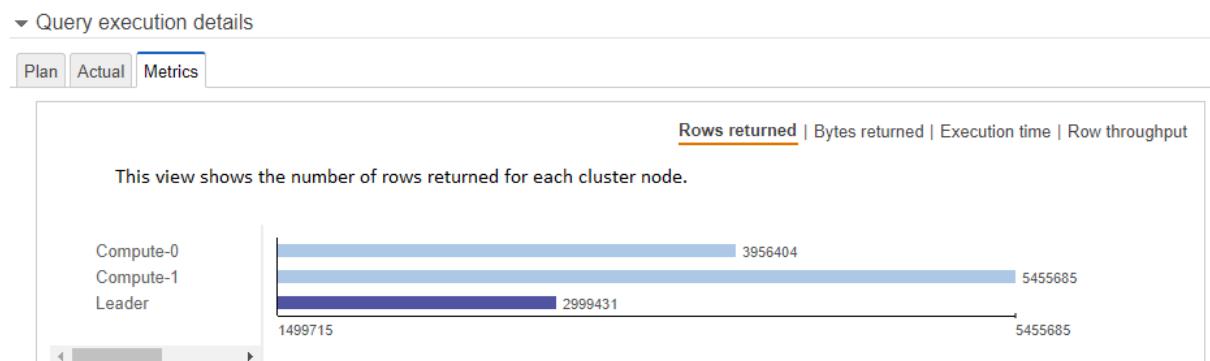
You can choose any bar in the chart to compare the data estimated from the explain plan with the actual performance of the query, as shown following.



If the query optimizer posted alerts for the query in the `STL_ALERT_EVENT_LOG` system table, then the plan nodes associated with the alerts are flagged with an alert icon.



- **Metrics.** This tab shows the metrics for the query that was executed.



The **Rows returned** metric is the sum of the number of rows produced during each step of the query.

The **Bytes returned** metric shows the number of bytes returned for each cluster node.

The **Execution time** metric shows the query execution time for each cluster node.

The **Row throughput** metric shows the number of rows returned divided by query execution time for each cluster node.

If a query runs slower than expected, you can use the **Metrics** tab to troubleshoot the cause. Look at the **Row throughput** metric. If one of the cluster nodes appears to have a much higher row throughput than the other nodes, the workload is unevenly distributed among the cluster nodes. One possible cause is that your data is unevenly distributed, or skewed, across node slices. For more information, see [Identifying Tables with Data Skew or Unsorted Rows](#).

If your data is evenly distributed, your query might be filtering for rows that are located mainly on that node. To fix this issue, look at the distribution styles for the tables in the query and see if any improvements can be made. Remember to weigh the performance of this query against the performance of other important queries and the system overall before making any changes. For more information, see [Choosing a Data Distribution Style](#).

Note

The metrics tab is not available for a single-node cluster.

[Viewing Query Execution Details Using the Console](#)

Use the following procedure to look at the details of query execution.

To view query execution details

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 **Clusters**.
3. For **Cluster**, choose the cluster for which you want to view query execution details.
4. Choose the **Queries** tab, and open the query for which you want to view performance data.
5. Expand the **Query Execution Details** section and do the following:
 - a. On the **Plan** tab, review the explain plan for the query. In some cases, you might find that your explain plan differs from the actual query execution on the **Actual** tab. In these cases, you might need to run ANALYZE to update statistics or perform other maintenance on the database to optimize the queries that you run. For more information about query optimization, see [Tuning Query Performance](#) in the *Amazon Redshift Database Developer Guide*.
 - b. On the **Actual** tab, review the performance data associated with each of the plan nodes in the query execution. You can choose an individual plan node in the hierarchy to view performance data associated with that specific plan node. This data includes both the estimated and actual performance data.
 - c. On the **Metrics** tab, review the metrics for each of the cluster nodes.

[Viewing Cluster Performance During Query Execution](#)

You can use the **Cluster Performance During Query Execution** section of the **Query** view to see cluster metrics during query execution. Doing so helps identify poorly performing queries, look for bottleneck queries, and determine if you need to resize your cluster for your workload.

To view cluster metrics during query execution

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 **Clusters**.
3. For **Cluster**, choose the cluster for which you want to view cluster performance during query execution.

The screenshot shows the 'Clusters' section of the Amazon Redshift Cluster Management Guide. On the left, there's a sidebar with links: 'Amazon Redshift', 'Clusters' (which is selected and highlighted in orange), 'Snapshots', 'Security Groups', 'Parameter Groups', 'Subnet Groups', and 'Events'. The main area has a title 'Launch an Amazon Redshift Cluster' with a 'Launch Cluster' button. Below that is a 'Performance Status' section with a table:

Cluster	Cluster Status
examplecluster	available

4. Choose the **Queries** tab.

The screenshot shows the 'Queries' tab for the 'examplecluster'. The top navigation bar includes tabs for Configuration, Status, Performance, Queries (which is selected and highlighted in red), and Loads. Below that is a 'Time Range' dropdown set to '02/12 17:46—02/13 17:46 UTC-8 (Last 24 Hours)'. The main area is a table with columns: Query, Run time, Start time, Status, User, and SQL. The 'Run time' column is highlighted with a red box. The table contains three rows:

Query	Run time	Start time	Status	User	SQL
253246	32.51s	Wed Feb 13 17:46:10 GMT-800	running	masterus	-- using def
253109	1m 58.55s	Wed Feb 13 17:39:10 GMT-800	completed	masterus	-- using def
253076	39.18s	Wed Feb 13 17:37:41 GMT-800	completed	masterus	-- using def

5. In the query list, find the query you want to work with, and choose the query ID in the **Query** column.

In the following example, the queries are sorted by **Run time** to find the query with the maximum run time.

The screenshot shows the 'Queries' tab for the 'examplecluster'. The top navigation bar includes tabs for Configuration, Status, Performance, Queries (selected and highlighted in red), and Loads. Below that is a 'Time Range' dropdown set to '02/05 14:45—02/06 14:45 UTC-8 (Last 24 Hours)'. The main area is a table with columns: Query, Run time, Start time, Status, User, and SQL. The 'Run time' column is highlighted with a red box. The table contains eight rows, with the first row (97610) having a run time of 19m 25.72s:

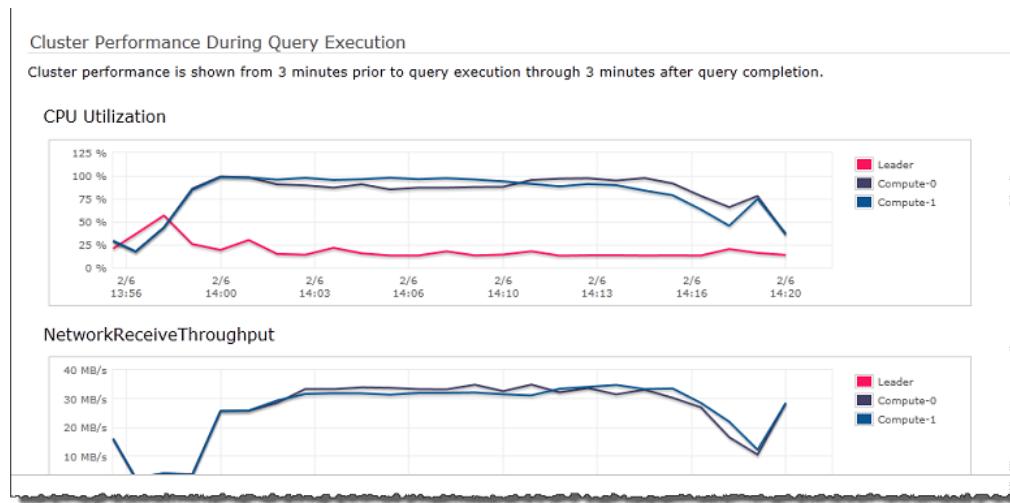
Query	Run time	Start time	Status	User	SQL
97610	19m 25.72s	Wed Feb 06 13:59:12 GMT-800	completed	masterus	copy part from 's3://.../part'
97615	19m 10.75s	Wed Feb 06 13:59:25 GMT-800	completed	masterus	copy customer from 's3://.../customer'
97504	33.71s	Wed Feb 06 13:55:33 GMT-800	completed	masterus	copy supplier from 's3://.../supplier'
97475	16.9s	Wed Feb 06 13:54:43 GMT-800	completed	masterus	COPY ANALYZE supplier
97563	16.11s	Wed Feb 06 13:58:09 GMT-800	completed	masterus	COPY ANALYZE part
79810	9.08s	Tue Feb 05 16:28:14 GMT-800	completed	masterus	COPY ANALYZE event
79728	9.03s	Tue Feb 05 16:26:19 GMT-800	completed	masterus	copy users from 's3://awssampledbs'
79679	8.61s	Tue Feb 05 16:25:17 GMT-800	completed	masterus	COPY ANALYZE users

6. In the **Query** page that opens, scroll to the **Cluster Performance During Query Execution** section to view cluster metrics.

In the following example, the **CPUUtilization** and **NetworkReceiveThroughput** metrics are displayed for the time that this query was running.

Tip

You can close the details of the **Query Execution Details** or **SQL** sections to manage how much information is displayed in the pane.



Viewing Cluster Metrics During Load Operations

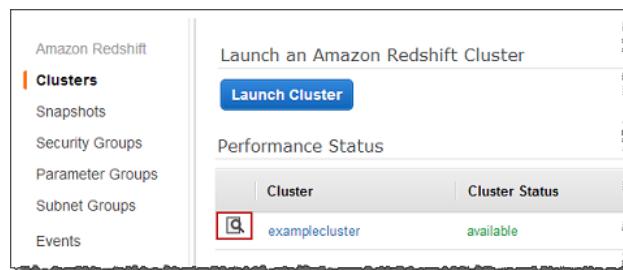
When you view cluster performance during load operations, you can identify queries that are consuming resources and take action to mitigate their effect. You can terminate a load if you don't want it to run to completion.

Note

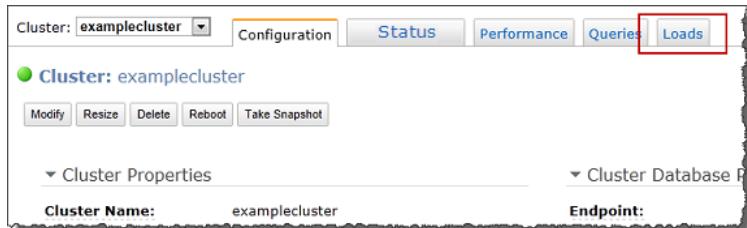
The ability to terminate queries and loads in the Amazon Redshift console requires specific permission. If you want users to be able to terminate queries and loads, make sure to add the `redshift:CancelQuerySession` action to your AWS Identity and Access Management (IAM) policy. This requirement applies whether you select the **Amazon Redshift Read Only** AWS-managed policy or create a custom policy in IAM. Users who have the **Amazon Redshift Full Access** policy already have the necessary permission to terminate queries and loads. For more information about actions in IAM policies for Amazon Redshift, see [Managing Access to Resources \(p. 190\)](#).

To view cluster metrics during load operations

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 **Clusters**.
3. For **Cluster**, choose the cluster for which you want to view cluster performance during query execution.



4. Choose the **Loads** tab.



5. In the load list, find the load operation you want to work with, and choose the load ID in the **Load** column.

The screenshot shows the 'Loads' tab with a table of completed load operations. The table has columns: Load, Run time, Start time, Status, Completion, User, and SQL. The data is as follows:

Load	Run time	Start time	Status	Completion	User	SQL
79679	8.61s	Tue Feb 05 16:25:17 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE users
79738	4.77s	Tue Feb 05 16:26:40 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE venue
79759	6.84s	Tue Feb 05 16:27:08 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE category
79779	7.31s	Tue Feb 05 16:27:35 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE date
79810	9.08s	Tue Feb 05 16:28:14 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE event
79839	8.3s	Tue Feb 05 16:28:46 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE listing
79874	4.64s	Tue Feb 05 16:29:32 GMT-800	COMPLETED	100%	masterus	COPY ANALYZE sales

6. In the new **Query** tab that is opened, you can view the details of the load operation.

At this point, you can work with the **Query** tab as shown in [Viewing Query Performance Data \(p. 264\)](#). You can review the details of the query and see the values of cluster metrics during the load operation.

To terminate a running load

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 **Clusters**.
3. For **Cluster**, choose the cluster that you want to open.
4. Choose the **Loads** tab.
5. Do one of the following:
 - In the list, select the load or loads that you want to terminate, and choose **Terminate Load**.
 - In the list, open a load if you want to review the load information first, and then choose **Terminate Load**.
6. In the **Terminate Loads** dialog box, choose **Confirm**.

Analyzing Workload Performance

You can get a detailed view of your workload's performance by looking at the Workload Execution Breakdown chart in the console. We build the chart with data provided by the CloudWatch QueryRuntimeBreakdown metric. With this chart, you can see how much time your queries spend in the various processing stages, such as waiting and planning.

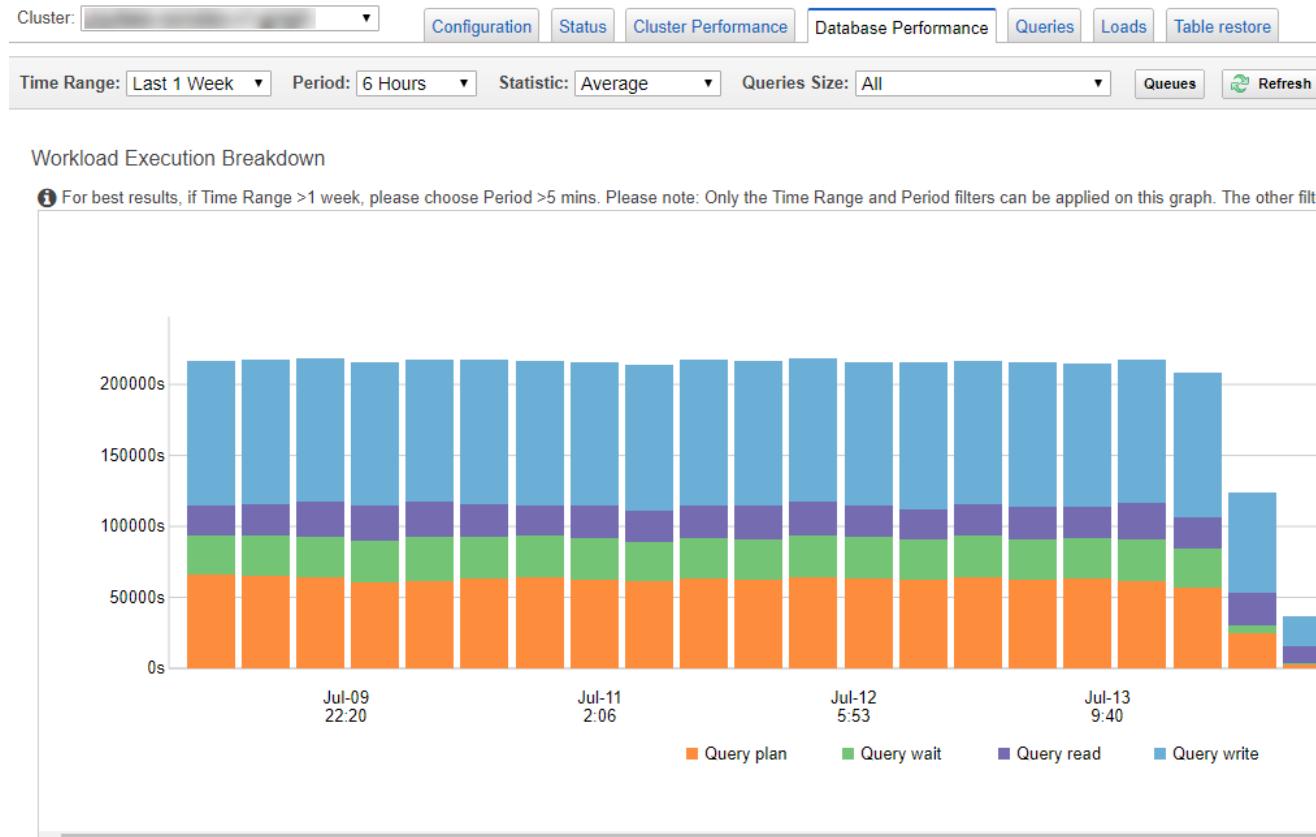
Note

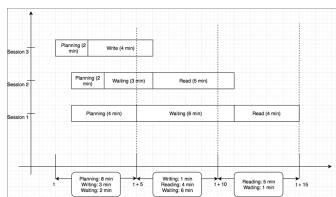
The Workload Execution Breakdown chart isn't shown for single-node clusters.

The following list of metrics describes the various processing stages:

- **QueryPlanning:** Time spent parsing and optimizing SQL statements.
- **QueryWaiting:** Time spent waiting in the Workload Management (WLM) queue.
- **QueryExecutingRead:** Time spent running read queries.
- **QueryExecutingInsert:** Time spent running insert queries.
- **QueryExecutingDelete:** Time spent running delete queries.
- **QueryExecutingUpdate:** Time spent running update queries.
- **QueryExecutingCtas:** Time spent running CREATE TABLE AS queries.
- **QueryExecutingUnload:** Time spent running unload queries.
- **QueryExecutingCopy:** Time spent running copy queries.

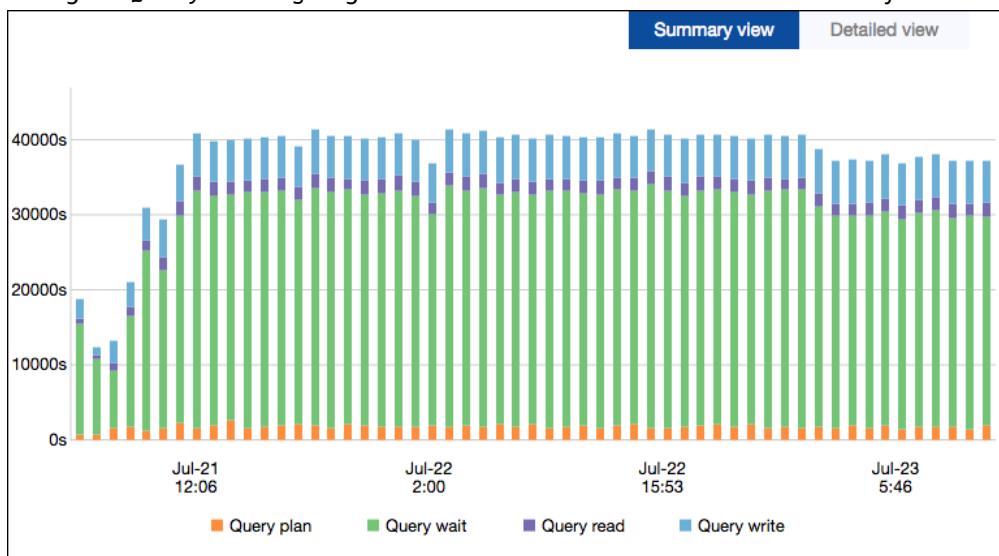
For example, the following graph in the Amazon Redshift console shows the amount of time that queries have spent in the plan, wait, read, and write stages. You can combine the findings from this graph with other metrics for further analysis. In some cases, your graph might show that queries with a short duration (as measured by the `QueryDuration` metric) are spending a long time in the wait stage. In these cases, you can increase the WLM concurrency rate for a particular queue to increase throughput.



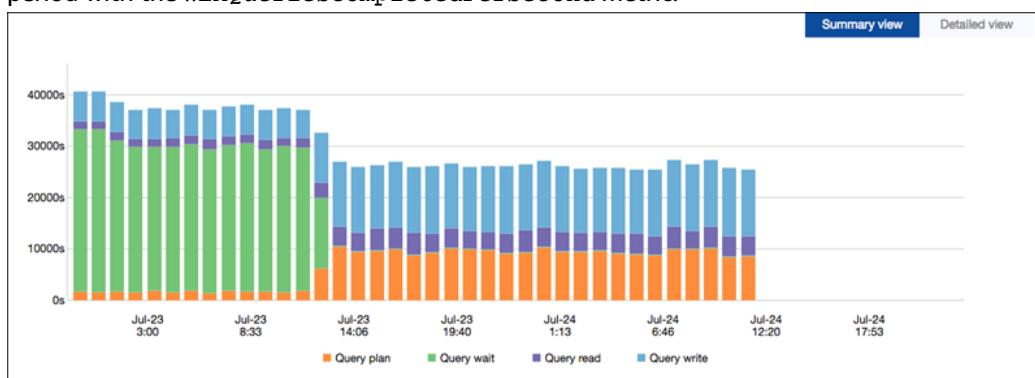


Example Analysis with the Workload Execution Breakdown Chart

The following diagrams illustrate how you can use the Workload Execution Breakdown chart to optimize your cluster's performance. In the first example chart, you can see that a majority of the query time was during the `QueryWaiting` stage. This effect was due to a low WLM concurrency value.



The following chart illustrates the query runtime breakdown after adjusting the concurrency to a higher value. In the updated chart, you can see that the majority of time usage has switched from the `QueryWaiting` stage to the `QueryExecutingRead` and `QueryPlanning` stages. In this case, more overall time is spent in the planning phase because more queries are now running during the time window after adjusting concurrency. You can check the number of queries running during a specific time period with the `WLMQueriesCompletedPerSecond` metric.



These charts demonstrate how changing your cluster's settings affect the amount of time that queries spend in the various stages. In the case preceding, queries initially spent a relatively long time waiting because the concurrency setting was low. After increasing concurrency, more queries are processed in parallel, thus decreasing the wait time and increasing query throughput.

Viewing the Workload Breakdown Chart

You can view the workload breakdown chart in the console.

To view the Workload Execution Breakdown Chart

1. Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. Choose **Clusters** in the navigation pane.
3. Choose the cluster that you want to analyze.
4. Choose the **Database Performance** tab.

Creating an Alarm

Alarms you create in the Amazon Redshift console are Amazon CloudWatch alarms. They are useful because they help you make proactive decisions about your cluster and its databases. You can set one or more alarms on any of the metrics listed in [Amazon Redshift Performance Data \(p. 256\)](#). For example, setting an alarm for high `CPUUtilization` on a cluster node helps indicate when the node is overutilized. Likewise, setting an alarm for low `CPUUtilization` on a cluster node helps indicate when the node is underutilized.

In this section, you can find how to create an alarm using the Amazon Redshift console. You can create an alarm using the Amazon CloudWatch console or any other way you work with metrics, such as with the AWS CLI or an AWS SDK. To delete an alarm, you must use the Amazon CloudWatch console.

To create an alarm on a cluster metric in the Amazon Redshift console

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 **Clusters**.
3. For **Cluster**, choose the cluster for which you want to view cluster performance during query execution.

The screenshot shows the 'Clusters' page of the Amazon Redshift console. On the left, there's a sidebar with links: 'Clusters' (which is highlighted in orange), 'Snapshots', 'Security Groups', 'Parameter Groups', 'Subnet Groups', and 'Events'. The main area has a title 'Launch an Amazon Redshift Cluster' with a 'Launch Cluster' button. Below that is a 'Performance Status' section with a table:

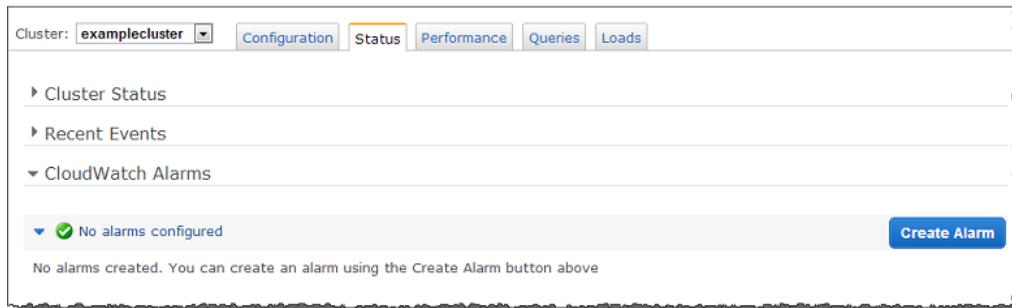
Cluster	Cluster Status
examplecluster	available

4. Choose the **Events+Alarms** tab.

The screenshot shows the 'Events+Alarms' tab for the 'examplecluster'. At the top, there's a navigation bar with tabs: 'Cluster: examplecluster', 'Configuration', 'Status' (which is highlighted with a red box), 'Performance', 'Queries', and 'Loads'. Below the tabs is a section titled 'Cluster Status' with the following details:

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

5. Choose **Create Alarm**.

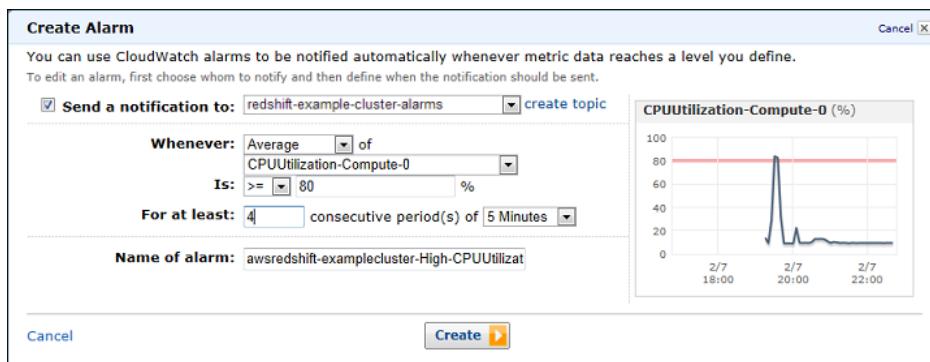


- In the **Create Alarm** dialog box, configure an alarm, and choose **Create**.

Note

The notifications that are displayed the **Send a notification to** box are your Amazon Simple Notification Service (Amazon SNS) topics. To learn more about Amazon SNS and creating topics, see [Create a Topic](#) in the *Amazon Simple Notification Service Getting Started Guide*. If you don't have any topics in Amazon SNS, you can create a topic in the Create Alarm dialog by clicking the [create topic](#) link.

The details of your alarm vary with your circumstance. In the following example, the average CPU utilization of a node (Compute-0) has an alarm set. If the CPU goes above 80 percent for four consecutive five-minute periods, this alarm sends a notification to the topic **redshift-example-cluster-alarms**.



- In the list of alarms, find your new alarm.

You might need to wait a few moments as sufficient data is collected to determine the state of the alarm as shown in the following example.



After a few moments, the state should turn to **OK**.



8. (Optional) Choose **Name** for the alarm to change the configuration of the alarm or choose the view link under **More Options** to go to this alarm in the Amazon CloudWatch console.

Working with Performance Metrics in the Amazon CloudWatch Console

When working with Amazon Redshift metrics in the Amazon CloudWatch console, keep a couple of things in mind:

- Query and load performance data is only available in the Amazon Redshift console.
- Some Metrics in the Amazon CloudWatch have different units than those used in the Amazon Redshift console. For example, `WriteThroughput`, is displayed in GB/s (as compared to Bytes/s in Amazon CloudWatch) which is a more relevant unit for the typical storage space of a node.

When working with Amazon Redshift metrics in the Amazon CloudWatch console, command line tools, or an Amazon SDK, there are two concepts to keep in mind.

- First, you specify the metric dimension to work with. A dimension is a name-value pair that helps you to uniquely identify a metric. The dimensions for Amazon Redshift are `ClusterIdentifier` and `NodeID`. In the Amazon CloudWatch console, the `Redshift Cluster` and `Redshift Node` views are provided to easily select cluster and node-specific dimensions. For more information about dimensions, see [Dimensions in the Amazon CloudWatch Developer Guide](#).
- Second, you specify the metric name, such as `ReadIOPS`.

The following table summarizes the types of Amazon Redshift metric dimensions that are available to you. All data is available in 1-minute periods at no charge.

Amazon CloudWatch Namespace	Dimension	Description
AWS/Redshift	NodeID	Filters requested data that is specific to the nodes of a cluster. <code>NodeID</code> is either "Leader", "Shared", or "Compute-N" where N is 0, 1, ... for the number of nodes in the cluster. "Shared" means that the cluster has only one node, that is the leader node and compute node are combined.
	ClusterIdentifier	Filters requested data that is specific to the cluster. Metrics that are specific to clusters include <code>HealthStatus</code> , <code>MaintenanceMode</code> , and <code>DatabaseConnections</code> . General metrics for this dimension (for example, <code>ReadIOPS</code>) that are also metrics of nodes represent an aggregate of the node metric data. Take care in interpreting these metrics because they aggregate behavior of leader and compute nodes.

Working with gateway and volume metrics is similar to working with other service metrics. Many of the common tasks are outlined in the Amazon CloudWatch documentation and are listed following for your convenience:

- [View Available Metrics](#)
- [Get Statistics for a Metric](#)
- [Creating Amazon CloudWatch Alarms](#)

Amazon Redshift Events

Topics

- [Overview \(p. 280\)](#)
- [Viewing Events Using the Console \(p. 280\)](#)
- [Viewing Events Using the AWS SDK for Java \(p. 281\)](#)
- [View Events Using the Amazon Redshift CLI and API \(p. 282\)](#)
- [Amazon Redshift Event Notifications \(p. 283\)](#)

Overview

Amazon Redshift tracks events and retains information about them for a period of several weeks in your AWS account. For each event, Amazon Redshift reports information such as the date the event occurred, a description, the event source (for example, a cluster, a parameter group, or a snapshot), and the source ID.

You can use the Amazon Redshift console, the Amazon Redshift API, or the AWS SDKs to obtain event information. You can obtain a list of all events, or you can apply filters—such as event duration or start and end date—to obtain events information for a specific period. You can also obtain events that were generated by a specific source type, such as cluster events or parameter group events.

You can create Amazon Redshift event notification subscriptions that specify a set of event filters. When an event occurs that matches the filter criteria, Amazon Redshift uses Amazon Simple Notification Service to actively inform you that the event has occurred.

For a list of Amazon Redshift events by source type and category, see [the section called “Amazon Redshift Event Categories and Event Messages” \(p. 284\)](#)

Viewing Events Using the Console

You can view events in the Amazon Redshift console by click on **Events** on the left navigation. In the list of events you can filter the results using the **Source Type** filter or a custom **Filter** that filters for text in all fields of the list. For example, if you search for "12 Dec 2012" you will match **Date** fields that contain this value.

Filter: Cluster	Search...	X	Viewing 42 of 42 events
Date	Event	Source	Source
2013 February 12 11:22:00 UTC-8	Cluster rebooted	examplecluster	cluster
2013 February 12 11:11:52 UTC-8	Cluster created	examplecluster2	cluster
2013 February 12 10:24:48 UTC-8	Cluster rebooted	examplecluster	cluster
2013 February 12 10:23:22 UTC-8	Your parameter group has been updated but changes won't get applied until you reboot the associated Clusters.	examplecluster	cluster
2013 February 11 16:22:33 UTC-8	Cluster deleted	examplecluster2	cluster
2013 February 11 16:20:21 UTC-8	Cluster deleted	examplecluster7	cluster
2013 February 11 15:22:57 UTC-8	Finished maintenance on the Cluster	examplecluster2	cluster
2013 February 11 15:14:11 UTC-8	Beginning maintenance on the Cluster. Your Cluster may not be available during maintenance	examplecluster2	cluster
2013 February 11 15:07:35 UTC-8	Finished maintenance on the Cluster	examplecluster	cluster
2013 February 11 15:00:45 UTC-8	Beginning maintenance on the Cluster. Your Cluster may not be available during maintenance	examplecluster	cluster

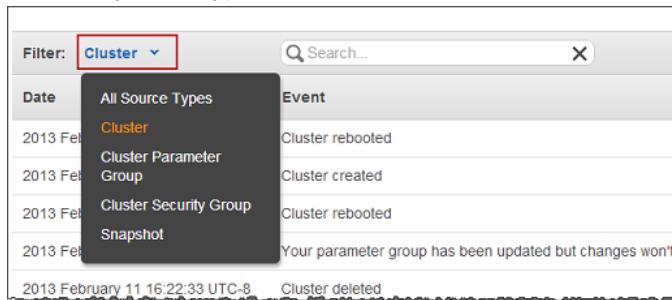
An event source type indicates what the event was about. The following source types are possible: **Cluster**, **Cluster Parameter Group**, **Cluster Security Group**, and **Snapshot**.

Filtering Events

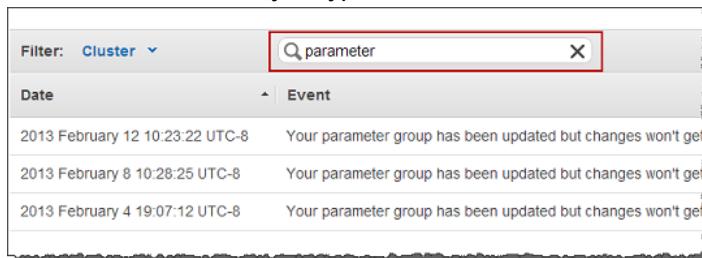
Sometimes you want to find a specific category of events or events for a specific cluster. In these cases, you can filter the events displayed.

To filter events

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, click **Events**.
3. To filter events do one of the following:
 - a. To filter by event type, click **Filter Cluster** and select the source type.



- b. To filter by text that appears in the event description, type in the in the search box and the list narrows based on what you type.



Viewing Events Using the AWS SDK for Java

The following example lists the events for a specified cluster and specified event source type. The example shows how to use pagination.

For step-by-step instructions to run the following example, see [Running Java Examples for Amazon Redshift Using Eclipse \(p. 246\)](#). You need to update the code and specify a cluster identifier and event source type.

Example

```
import java.io.IOException;
import java.util.Date;

import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class ListEvents {

    public static AmazonRedshiftClient client;
    public static String clusterIdentifier = "***provide cluster identifier***";
    public static String eventSourceType = "***provide source type***"; // e.g. cluster-
snapshot
```

```
public static void main(String[] args) throws IOException {
    AWSCredentials credentials = new PropertiesCredentials(
        ListEvents.class
        .getResourceAsStream("AwsCredentials.properties"));

    client = new AmazonRedshiftClient(credentials);

    try {
        listEvents();
    } catch (Exception e) {
        System.err.println("Operation failed: " + e.getMessage());
    }
}

private static void listEvents() {
    long oneWeeksAgoMilli = (new Date()).getTime() - (7L*24L*60L*60L*1000L);
    Date oneWeekAgo = new Date();
    oneWeekAgo.setTime(oneWeeksAgoMilli);
    String marker = null;

    do {
        DescribeEventsRequest request = new DescribeEventsRequest()
            .withSourceIdentifier(clusterIdentifier)
            .withSourceType(eventSourceType)
            .withStartTime(oneWeekAgo)
            .withMaxRecords(20);
        DescribeEventsResult result = client.describeEvents(request);
        marker = result.getMarker();
        for (Event event : result.getEvents()) {
            printEvent(event);
        }
    } while (marker != null);

}
static void printEvent(Event event)
{
    if (event == null)
    {
        System.out.println("\nEvent object is null.");
        return;
    }

    System.out.println("\nEvent metadata:\n");
    System.out.format("SourceID: %s\n", event.getSourceIdentifier());
    System.out.format("Type: %s\n", event.getSourceType());
    System.out.format("Message: %s\n", event.getMessage());
    System.out.format("Date: %s\n", event.getDate());
}
}
```

View Events Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operation to manage events.

- [describe-events](#)

Amazon Redshift provides the following API to view events.

- [DescribeEvents](#)

Amazon Redshift Event Notifications

Overview

Amazon Redshift uses the Amazon Simple Notification Service (Amazon SNS) to communicate notifications of Amazon Redshift events. You enable notifications by creating an Amazon Redshift event subscription. In the Amazon Redshift subscription, you specify a set of filters for Amazon Redshift events and an Amazon SNS topic. Whenever an event occurs that matches the filter criteria, Amazon Redshift publishes a notification message to the Amazon SNS topic. Amazon SNS then transmits the message to any Amazon SNS consumers that have an Amazon SNS subscription to the topic. The messages sent to the Amazon SNS consumers can be in any form supported by Amazon SNS for an AWS region, such as an email, a text message, or a call to an HTTP endpoint. For example, all regions support email notifications, but SMS notifications can only be created in the US East (N. Virginia) Region.

When you create an event notification subscription, you specify one or more event filters. Amazon Redshift sends notifications through the subscription any time an event occurs that matches all of the filter criteria. The filter criteria include source type (such as cluster or snapshot), source ID (such as the name of a cluster or snapshot), event category (such as Monitoring or Security), and event severity (such as INFO or ERROR).

You can easily turn off notification without deleting a subscription by setting the **Enabled** radio button to **No** in the AWS Management Console or by setting the `Enabled` parameter to `false` using the Amazon Redshift CLI or API.

Billing for Amazon Redshift event notification is through the Amazon Simple Notification Service (Amazon SNS). Amazon SNS fees apply when using event notification; for more information on Amazon SNS billing, go to [Amazon Simple Notification Service Pricing](#).

You can also view Amazon Redshift events that have occurred by using the management console. For more information, see [Amazon Redshift Events \(p. 280\)](#).

Topics

- [Subscribing to Amazon Redshift Event Notifications \(p. 283\)](#)

Subscribing to Amazon Redshift Event Notifications

You can create an Amazon Redshift event notification subscription so you can be notified when an event occurs for a given cluster, snapshot, security group, or parameter group. The simplest way to create a subscription is with the Amazon SNS console. For information on creating an Amazon SNS topic and subscribing to it, see [Getting Started with Amazon SNS](#).

You can create an Amazon Redshift event notification subscription so you can be notified when an event occurs for a given cluster, snapshot, security group, or parameter group. The simplest way to create a subscription is with the AWS Management Console. If you choose to create event notification subscriptions using the CLI or API, you must create an Amazon Simple Notification Service topic and subscribe to that topic with the Amazon SNS console or Amazon SNS API. You will also need to retain the Amazon Resource Name (ARN) of the topic because it is used when submitting CLI commands or API actions. For information on creating an Amazon SNS topic and subscribing to it, see [Getting Started with Amazon SNS](#).

An Amazon Redshift event subscription can specify these event criteria:

- Source type, the values are cluster, snapshot, parameter-groups, and security-groups.

- Source ID of a resource, such as `my-cluster-1` or `my-snapshot-20130823`. The ID must be for a resource in the same region as the event subscription.
- Event category, the values are Configuration, Management, Monitoring, and Security.
- Event severity, the values are INFO or ERROR.

The event criteria can be specified independently, except that you must specify a source type before you can specify source IDs in the console. For example, you can specify an event category without having to specify a source type, source ID, or severity. While you can specify source IDs for resources that are not of the type specified in source type, no notifications will be sent for events from those resources. For example, if you specify a source type of cluster and the ID of a security group, none of the events raised by that security group would match the source type filter criteria, so no notifications would be sent for those events.

Amazon Redshift sends a notification for any event that matches all criteria specified in a subscription. Some examples of the sets of events returned:

- Subscription specifies a source type of cluster, a source ID of `my-cluster-1`, a category of Monitoring, and a severity of ERROR. The subscription will send notifications for only monitoring events with a severity of ERROR from `my-cluster-1`.
- Subscription specifies a source type of cluster, a category of Configuration, and a severity of INFO. The subscription will send notifications for configuration events with a severity of INFO from any Amazon Redshift cluster in the AWS account.
- Subscription specifies a category of Configuration, and a severity of INFO. The subscription will send notifications for configuration events with a severity of INFO from any Amazon Redshift resource in the AWS account.
- Subscription specifies a severity of ERROR. The subscription will send notifications for all events with a severity of ERROR from any Amazon Redshift resource in the AWS account.

If you delete or rename an object whose name is referenced as a source ID in an existing subscription, the subscription will remain active, but will have no events to forward from that object. If you later create a new object with the same name as is referenced in the subscription source ID, the subscription will start sending notifications for events from the new object.

Amazon Redshift publishes event notifications to an Amazon SNS topic, which is identified by its Amazon Resource Name (ARN). When you create an event subscription using the Amazon Redshift console, you can either specify an existing Amazon SNS topic, or request that the console create the topic when it creates the subscription. All Amazon Redshift event notifications sent to the Amazon SNS topic are in turn transmitted to all Amazon SNS consumers that are subscribed to that topic. Use the Amazon SNS console to make changes to the Amazon SNS topic, such as adding or removing consumer subscriptions to the topic. For more information about creating and subscribing to Amazon SNS topics, go to [Getting Started with Amazon Simple Notification Service](#).

The following section lists all categories and events that you can be notified of. It also provides information about subscribing to and working with Amazon Redshift event subscriptions.

Amazon Redshift Event Categories and Event Messages

This section shows the event IDs and categories for each Amazon Redshift source type.

The following table shows the event category and a list of events when a cluster is the source type.

Categories and Events for the Cluster Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Configuration	REDSHIFT-EVENT-1000	INFO	The parameter group [parameter group name] was updated at [time]. Changes will be applied to the associated clusters when they are rebooted.
Configuration	REDSHIFT-EVENT-1001	INFO	Your Amazon Redshift cluster [cluster name] was modified to use parameter group [parameter group name] at [time].
Configuration	REDSHIFT-EVENT-1500	ERROR	The Amazon VPC [VPC name] does not exist. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1501	ERROR	The customer subnets [subnet name] you specified for Amazon VPC [VPC name] do not exist or are invalid. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1502	ERROR	The Amazon VPC [VPC name] has no available IP addresses. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1503	ERROR	The Amazon VPC [VPC name] has no internet gateway attached to it. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1504	ERROR	The HSM for cluster [cluster name] is unreachable.
Configuration	REDSHIFT-EVENT-1505	ERROR	The HSM for cluster [cluster name] cannot be registered. Try a different configuration.
Configuration	REDSHIFT-EVENT-1506	ERROR	Amazon Redshift exceeded your account's EC2 Network Interface (ENI) limit. Please delete up to [maximum number of ENIs] ENIs or request a limit increase of the number of network interfaces per region with EC2.
Management	REDSHIFT-EVENT-2000	INFO	Your Amazon Redshift cluster: [cluster name] has been created and is ready for use.
Management	REDSHIFT-EVENT-2001	INFO	Your Amazon Redshift cluster [cluster name] was deleted at [time]. A final snapshot [was / was not] saved.
Management	REDSHIFT-EVENT-2002	INFO	Your VPC security group [security group name] was updated at [time].

Amazon Redshift Category	Event ID	Event Severity	Description
Management	REDSHIFT-EVENT-2003	INFO	Maintenance started on your Amazon Redshift cluster [cluster name] at [time]. The cluster may not be available during maintenance.
Management	REDSHIFT-EVENT-2004	INFO	Maintenance completed on your Amazon Redshift cluster [cluster name] at [time].
Management	REDSHIFT-EVENT-2006	INFO	A resize for your Amazon Redshift cluster [cluster name] was started at [time]. Your cluster will be in read-only mode during the resize operation.
Management	REDSHIFT-EVENT-2007	INFO	The resize for your Amazon Redshift cluster [cluster name] is in progress. Your cluster is in read-only mode.
Management	REDSHIFT-EVENT-2008	INFO	Your restore operation to create a new Amazon Redshift cluster [cluster name] snapshot [snapshot name] was started at [time]. To monitor restore progress, please visit the AWS Management Console.
Management	REDSHIFT-EVENT-2013	INFO	Your Amazon Redshift cluster [cluster name] was renamed at [time].
Management	REDSHIFT-EVENT-2014	INFO	A table restore request for Amazon Redshift cluster [cluster name] has been received.
Management	REDSHIFT-EVENT-2015	INFO	Table restore was cancelled for Amazon Redshift cluster [cluster name] at [time].
Management	REDSHIFT-EVENT-2016	INFO	Replacement of your Amazon Redshift cluster [cluster name] was started at [time].
Management	REDSHIFT-EVENT-2017	INFO	Customer initiated maintenance started on your Amazon Redshift cluster [cluster name] at [time]. The cluster may not be available during maintenance.
Management	REDSHIFT-EVENT-2018	INFO	Customer initiated maintenance completed on your Amazon Redshift cluster [cluster name] at [time].
Management	REDSHIFT-EVENT-2019	ERROR	Customer initiated maintenance failed on your Amazon Redshift cluster [cluster name] at [time]. Returning the cluster back to its original state.
Management	REDSHIFT-EVENT-2020	INFO	Your Amazon Redshift cluster [cluster name]'s track has been modified from [from track] to [to track].

Amazon Redshift Category	Event ID	Event Severity	Description
Management	REDSHIFT-EVENT-2021	ERROR	The [operation] of Amazon Redshift cluster [cluster name] did not succeed while acquiring capacity from our capacity pool. We are working to acquire capacity but for now, we have cancelled your request. Delete this cluster and retry later.
Management	REDSHIFT-EVENT-2022	ERROR	The [operation] of Amazon Redshift cluster [cluster name] did not succeed while acquiring capacity from our capacity pool. We are working to acquire capacity but for now, we have cancelled your request. Capacity is available in [alternative availability zones]. Delete this cluster and retry in an alternative availability zone.
Management	REDSHIFT-EVENT-2023	ERROR	We have detected hardware failure on your single node Amazon Redshift cluster [cluster name], which may have resulted in failed queries or intermittent availability of the cluster. Replacing the cluster did not succeed while acquiring capacity from our capacity pool. You will need to restore a new cluster from a snapshot. Delete this cluster, select the latest available snapshot, and restore a new cluster from that snapshot. This will automatically provision you on healthy hardware.
Management	REDSHIFT-EVENT-2024	ERROR	We have detected hardware failure on your single node Amazon Redshift cluster [cluster name], which may have resulted in failed queries or intermittent availability of the cluster. Replacing cluster did not succeed while acquiring capacity from our capacity pool. Capacity is available in availability zone: [alternative availability zones]. Delete this cluster, select the latest available snapshot, and restore a new cluster from that snapshot. This will automatically provision you on healthy hardware.
Monitoring	REDSHIFT-EVENT-2050	INFO	A hardware issue was detected on Amazon Redshift cluster [cluster name]. A replacement request was initiated at [time].
Monitoring	REDSHIFT-EVENT-3000	INFO	Your Amazon Redshift cluster [cluster name] was rebooted at [time].
Monitoring	REDSHIFT-EVENT-3001	INFO	A node on your Amazon Redshift cluster: [cluster name] was automatically replaced at [time], and your cluster is operating normally.

Amazon Redshift Category	Event ID	Event Severity	Description
Monitoring	REDSHIFT-EVENT-3002	INFO	The resize for your Amazon Redshift cluster [cluster name] is complete and your cluster is available for reads and writes. The resize was initiated at [time] and took [hours] hours to complete.
Monitoring	REDSHIFT-EVENT-3003	INFO	Amazon Redshift cluster [cluster name] was successfully created from snapshot [snapshot name] and is available for use.
Monitoring	REDSHIFT-EVENT-3007	INFO	Your Amazon Redshift snapshot [snapshot name] was copied successfully from [source region] to [destination region] at [time].
Monitoring	REDSHIFT-EVENT-3008	INFO	Table restore started for Amazon Redshift cluster [cluster name] at [time].
Monitoring	REDSHIFT-EVENT-3009	INFO	Table restore completed successfully for Amazon Redshift cluster [cluster name] at [time].
Monitoring	REDSHIFT-EVENT-3011	ERROR	Table restore failed for Amazon Redshift cluster [cluster name] at [time].
Monitoring	REDSHIFT-EVENT-3500	ERROR	The resize for your Amazon Redshift cluster [cluster name] failed. The resize will be automatically retried in a few minutes.
Monitoring	REDSHIFT-EVENT-3501	ERROR	Your restore operation to create Amazon Redshift cluster [cluster name] from snapshot [snapshot name] failed at [time]. Please retry your operation.
Monitoring	REDSHIFT-EVENT-3504	ERROR	The Amazon S3 bucket [bucket name] is not valid for logging for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3505	ERROR	The Amazon S3 bucket [bucket name] does not have the correct IAM policies for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3506	ERROR	The Amazon S3 bucket [bucket name] does not exist. Logging cannot continue for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3507	ERROR	The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. This EIP is already in use.
Monitoring	REDSHIFT-EVENT-3508	ERROR	The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. The EIP cannot be found.
Monitoring	REDSHIFT-EVENT-3509	ERROR	Cross-region snapshot copy is not enabled for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3510	ERROR	Table restore failed to start for Amazon Redshift cluster [cluster name] at [time]. Reason: [reason].

Amazon Redshift Category	Event ID	Event Severity	Description
Monitoring	REDSHIFT-EVENT-3511	ERROR	Table restore failed for Amazon Redshift cluster [cluster name] at [time].
Monitoring	REDSHIFT-EVENT-3512	ERROR	Amazon Redshift cluster [cluster name] has failed due to a hardware issue. The cluster is being automatically restored from the latest snapshot [snapshot name] created at [time].
Monitoring	REDSHIFT-EVENT-3513	ERROR	Amazon Redshift cluster [cluster name] has failed due to a hardware issue. The cluster is being automatically restored from the latest snapshot [snapshot name] created at [time]. Any database changes made after this time will need to be resubmitted.
Monitoring	REDSHIFT-EVENT-3514	ERROR	Amazon Redshift cluster [cluster name] has failed due to a hardware issue. The cluster is being placed in hardware failure status. Please delete the cluster and restore from the latest snapshot [snapshot name] created at [time].
Monitoring	REDSHIFT-EVENT-3515	ERROR	Amazon Redshift cluster [cluster name] has failed due to a hardware issue. The cluster is being placed in hardware failure status. Please delete the cluster and restore from the latest snapshot [snapshot name] created at [time]. Any database changes made after this time will need to be resubmitted.
Monitoring	REDSHIFT-EVENT-3516	ERROR	Amazon Redshift cluster [cluster name] has failed due to a hardware issue and there are no backups for the cluster. The cluster is being placed in hardware failure status and can be deleted.
Monitoring	REDSHIFT-EVENT-3519	INFO	Cluster [cluster name] began restart at [time].
Monitoring	REDSHIFT-EVENT-3520	INFO	Cluster [cluster name] completed restart at [time].
Monitoring	REDSHIFT-EVENT-3521	INFO	We detected a connectivity issue on the cluster '[cluster name]'. An automated diagnostics check has been initiated at [time].
Monitoring	REDSHIFT-EVENT-3522	INFO	Recovery action on '[cluster name]' cluster failed at [time]. The Amazon Redshift team is working on a solution.
Monitoring	REDSHIFT-EVENT-3533	ERROR	Cluster resize on '[cluster name]' was cancelled at [time]. The operation was cancelled because [reason]. [action needed].
Monitoring	REDSHIFT-EVENT-3600	INFO	The requested resize operation for Amazon Redshift cluster '[cluster name]' was cancelled in the past. Rollback was completed at [time].

Amazon Redshift Category	Event ID	Event Severity	Description
Security	REDSHIFT-EVENT-4000	INFO	Your master credentials for your Amazon Redshift cluster: [cluster name] were updated at [time].
Security	REDSHIFT-EVENT-4001	INFO	The security group [security group name] was modified at [time]. The changes will take place for all associated clusters automatically.
Security	REDSHIFT-EVENT-4500	ERROR	The security group [security group name] you provided is invalid. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Security	REDSHIFT-EVENT-4501	ERROR	The security group [security group name] specified in Cluster Security Group [cluster security group name] could not be found. The authorization cannot be completed.

The following table shows the event category and a list of events when a parameter group is the source type.

Categories and Events for the Parameter Group Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Configuration	REDSHIFT-EVENT-1002	INFO	The parameter [parameter name] was updated from [value] to [value] at [time].
Configuration	REDSHIFT-EVENT-1003	INFO	Cluster parameter group [group name] was created.
Configuration	REDSHIFT-EVENT-1004	INFO	Cluster parameter group [group name] was deleted.
Configuration	REDSHIFT-EVENT-1005	INFO	Cluster parameter group [name] was updated at [time]. Changes will be applied to the associated clusters when they are rebooted.

The following tables shows the event category and a list of events when a security group is the source type.

Categories and Events for the Security Group Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Security	REDSHIFT-EVENT-4002	INFO	Cluster security group [group name] was created.
Security	REDSHIFT-EVENT-4003	INFO	Cluster security group [group name] was deleted.
Security	REDSHIFT-EVENT-4004	INFO	Cluster security group [group name] was changed at [time]. Changes will be automatically applied to all associated clusters.

The following tables shows the event category and a list of events when a snapshot is the source type.

Categories and Events for the Snapshot Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Management	REDSHIFT-EVENT-2009	INFO	A user snapshot [snapshot name] for Amazon Redshift Cluster [cluster name] started at [time]. To monitor snapshot progress, please visit the AWS Management Console.
Management	REDSHIFT-EVENT-2010	INFO	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] was cancelled at [time].
Management	REDSHIFT-EVENT-2011	INFO	The user snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was deleted at [time].
Management	REDSHIFT-EVENT-2012	INFO	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was started at [time].
Monitoring	REDSHIFT-EVENT-3004	INFO	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] completed successfully at [time].
Monitoring	REDSHIFT-EVENT-3005	INFO	The final snapshot [name] for Amazon Redshift cluster [name] completed successfully at [time].
Monitoring	REDSHIFT-EVENT-3006	INFO	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was cancelled at [time].
Monitoring	REDSHIFT-EVENT-3502	ERROR	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] failed at [time]. The team is investigating

Amazon Redshift Category	Event ID	Event Severity	Description
			the issue. Please visit the AWS Management Console to retry the operation.
Monitoring	REDSHIFT-EVENT-3503	ERROR	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] failed at [time]. The team is investigating the issue. Please visit the AWS Management Console to retry the operation.

Managing Event Notifications Using the Amazon Redshift Console

Topics

- [Creating an Event Notification Subscription \(p. 292\)](#)
- [Listing Your Amazon Redshift Event Notification Subscriptions \(p. 295\)](#)
- [Modifying an Amazon Redshift Event Notification Subscription \(p. 295\)](#)
- [Adding a Source Identifier to an Amazon Redshift Event Notification Subscription \(p. 296\)](#)
- [Removing a Source Identifier from an Amazon Redshift Event Notification Subscription \(p. 296\)](#)
- [Deleting an Amazon Redshift Event Notification Subscription \(p. 296\)](#)

You can create an Amazon Simple Notification Service (Amazon SNS) event notification subscription to send notifications when an event occurs for a given Amazon Redshift cluster, snapshot, security group, or parameter group. These notifications are sent to an SNS topic, which in turn transmits messages to any SNS consumers subscribed to the topic. The SNS messages to the consumers can be in any notification form supported by Amazon SNS for an AWS region, such as an email, a text message, or a call to an HTTP endpoint. For example, all regions support email notifications, but SMS notifications can only be created in the US East (N. Virginia) Region. For more information, see [Amazon Redshift Event Notifications \(p. 283\)](#).

This section describes how to manage Amazon Redshift event notification subscriptions from the AWS Management Console.

Creating an Event Notification Subscription

To create an Amazon Redshift event notification subscription

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 **Events**, and then choose the **Subscriptions** tab.
3. In the **Subscriptions** pane, choose **Create Event Subscription**.
4. In the **Create Event Subscription** dialog box, do the following:
 - a. Use the **Subscription Settings** pane to specify the event filter criteria. As you choose the criteria, the **Subscribed Events** list displays the Amazon Redshift events that match the criteria. Do the following:

- a. Use the **Subscription Settings** pane to specify the event filter criteria. As you choose the criteria, the **Subscribed Events** list displays the Amazon Redshift events that match the criteria. Do the following:

- i. For **Categories**, choose one or more event categories. To specify all categories, choose **Category**. To choose a subset of the categories, choose the buttons for the categories to be included.
- ii. For **Severity**, choose an event severity. If you choose **Any**, events with severities of either INFO or ERROR are published. If you choose **Error**, only events with a severity of ERROR are published.
- iii. For **Source Type**, choose a source type. Only events raised by resources of that source type, such as clusters or cluster parameter groups, are published by the event subscription.
- iv. For **Resources**, specify whether events should be published from all resources having the specified source type, or only a subset. Choose **Any** to publish events from all resources of the specified type. Choose **Choose Specific** if you want to choose specific resources.

Note

The name of the **Resource** box changes to match the value specified in **Source Type**. For example, if you choose **Cluster** in **Source Type**, the name of the **Resources** box changes to **Clusters**.

If you choose **Choose Specific**, you can then specify the IDs of the specific resources whose events are published by the event subscription. You specify the resources one at a time and add them to the event subscription. You can only specify resources that are in the same AWS Region as the event subscription. The events you specify are listed below the **Specify IDs** box.

- A. To specify an existing resource, find the resource in the **Specify IDs** box, and choose the **+** button in the **Add** column.
 - B. To specify the ID of a resource before you create it, enter the ID in the box below the **Specify IDs** box and choose **Add**. You can do this to add resources that you plan to create later.
 - C. To remove a resource from the event subscription, choose the **X** box to the right of the resource.
- b. At the bottom of the pane, enter a name for the event notification subscription for **Name**.
 - c. Choose **Yes** to enable the subscription. If you want to create the subscription but to not send notifications yet, choose **No**. A confirmation message is sent when the subscription is created, regardless of this setting.
 - d. Choose **Next** to proceed to specifying the Amazon SNS topic.

- e. Use one of three tabs to specify the Amazon SNS topic that the subscription uses to publish events.
 - i. To choose an existing Amazon SNS topic by from a list, choose the **Use Existing Topic** tab and choose the topic from the list.
 - ii. To specify an existing Amazon SNS topic by its Amazon Resource Name (ARN), choose the **Provide Topic ARN** tab and specify the ARN in the **ARN:** box. You can find the ARN of an Amazon SNS topic by using the Amazon SNS console:
 - A. Sign in to the AWS Management Console and open the Amazon SNS console at <https://console.aws.amazon.com/sns/v2/home>.
 - B. In the navigation pane, expand **Topics**.
 - C. Choose the topic to include in the Amazon Redshift event subscription.
 - D. In the **Topic Details** pane, copy the value of the **Topic ARN:** field.
 - iii. To have the subscription create operation also create a new Amazon SNS topic, choose the **Create New Topic** tab and do the following:
 - A. Type a name for the topic for **Name**.
 - B. For each notification recipient, choose the notification method for **Send**, specify a valid address for **to**, and then choose **Add Recipient**. You can only create **SMS** entries in the US East (N. Virginia) Region.
 - C. To remove a recipient, choose the red X in the **Remove** column.

Create Event Subscription

Use Existing Topic Create New Topic Provide ARN

Name: Provide a name for the SNS topic that will be created.

Specify Recipients: to Add Recipient e.g. user@domain.com

Type	Recipient	Remove
email	myemail@mycompany.com	X
sms	phononenumber	X

*Required

- To create the subscription, choose **Create**. To delete the definition without creating a subscription, choose **Cancel**. To return to the subscription settings, choose **Previous**.

Listing Your Amazon Redshift Event Notification Subscriptions

You can list your current Amazon Redshift event notification subscriptions.

To list your current Amazon Redshift event notification subscriptions

- Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
- In the navigation pane, choose **Events**. The **Subscriptions** tab shows all your event notification subscriptions.

Modifying an Amazon Redshift Event Notification Subscription

After you have created a subscription, you can change the subscription name, source identifier, categories, or topic ARN.

To modify an Amazon Redshift event notification subscription

- Sign in to the AWS Management Console and open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
- In the navigation pane, choose **Events**, and then choose the **Subscriptions** tab.
- In the **Subscriptions** pane, choose the subscription that you want to modify, and choose **Modify**.
- In the **Modify Event Subscription** dialog box, do the following:
 - Use the **Subscription Settings** pane to change the event filter criteria. As you choose the criteria, the **Subscribed Events** list displays the Amazon Redshift events that match the criteria. Do the following:
 - For **Categories**, choose one or more event categories. To specify all categories, choose **Category**. To choose a subset of the categories, choose the buttons for the categories to be included.
 - For **Severity**, choose an event severity.
 - For **Source Type**, choose a source type.
 - For **Resources**, choose the IDs of the resources from the given source type. Only events raised by the specified resources are published by the subscription.

- b. For **Enabled**, choose **Yes** to enable the subscription. Choose **No** to disable the subscription.
- c. Choose **Next** to proceed to changing the Amazon SNS topic.
- d. Use one of three tabs to change the Amazon SNS topic for the subscription to use to publish events.
 - i. To choose an existing Amazon SNS topic from a list, choose the **Use Existing Topic** tab and choose the topic from the list.
 - ii. To specify an existing Amazon SNS topic by its Amazon Resource Name (ARN), choose the **Provide ARN** tab and specify the ARN in the **ARN** box.
 - iii. To have the subscription modify operation also create a new Amazon SNS topic, choose the **Create New Topic** tab and do the following:
 - A. Enter a name for the topic for **Name**.
 - B. For each notification recipient, choose the notification method for **Send**, specify a valid address for **to**, and then choose **Add Recipient**. You can only create **SMS** entries in the US East (N. Virginia) Region.
 - C. To remove a recipient, choose the red X in the **Remove** column.
5. To save your changes, choose **Modify**. To delete your changes without modifying the subscription, choose **Cancel**. To return to the subscription settings, choose **Previous**.

Adding a Source Identifier to an Amazon Redshift Event Notification Subscription

You can add a source identifier (the Amazon Redshift source generating the event) to an existing subscription.

To add a source identifier to an Amazon Redshift event notification subscription

1. You can easily add or remove source identifiers using the Amazon Redshift console by selecting or deselecting them when modifying a subscription. For more information, see [Modifying an Amazon Redshift Event Notification Subscription \(p. 295\)](#).
2. To save your changes, choose **Modify**. To delete your changes without modifying the subscription, choose **Cancel**. To return to the subscription settings, choose **Previous**.

Removing a Source Identifier from an Amazon Redshift Event Notification Subscription

You can remove a source identifier (the Amazon Redshift source generating the event) from a subscription if you no longer want to be notified of events for that source.

To remove a source identifier from an Amazon Redshift event notification subscription

- You can easily add or remove source identifiers using the Amazon Redshift console by selecting or deselecting them when modifying a subscription. For more information, see [Modifying an Amazon Redshift Event Notification Subscription \(p. 295\)](#).

Deleting an Amazon Redshift Event Notification Subscription

You can delete a subscription when you no longer need it. All subscribers to the topic will no longer receive event notifications specified by the subscription.

To delete an Amazon Redshift event notification subscription

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 **Events**, and then choose the **Subscriptions** tab.
3. In the **Subscriptions** pane, choose the subscription that you want to delete.
4. Choose **Delete**.

Managing Event Notifications Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage event notifications.

- [create-event-subscription](#)
- [delete-event-subscription](#)
- [describe-event-categories](#)
- [describe-event-subscriptions](#)
- [describe-events](#)
- [modify-event-subscription](#)

You can use the following Amazon Redshift API actions to manage event notifications.

- [CreateEventSubscription](#)
- [DeleteEventSubscription](#)
- [DescribeEventCategories](#)
- [DescribeEventSubscriptions](#)
- [DescribeEvents](#)
- [ModifyEventSubscription](#)

For more information about Amazon Redshift event notifications, see [Amazon Redshift Event Notifications \(p. 283\)](#).

Database Audit Logging

Topics

- [Overview \(p. 298\)](#)
- [Amazon Redshift Logs \(p. 298\)](#)
- [Enabling Logging \(p. 300\)](#)
- [Managing Log Files \(p. 301\)](#)
- [Troubleshooting Amazon Redshift Audit Logging \(p. 303\)](#)
- [Logging Amazon Redshift API Calls with AWS CloudTrail \(p. 304\)](#)
- [Amazon Redshift Account IDs in AWS CloudTrail Logs \(p. 307\)](#)
- [Configuring Auditing Using the Console \(p. 308\)](#)
- [Configuring Logging by Using the Amazon Redshift CLI and API \(p. 310\)](#)

Overview

Amazon Redshift logs information about connections and user activities in your database. These logs help you to monitor the database for security and troubleshooting purposes, which is a process often referred to as database auditing. The logs are stored in the Amazon Simple Storage Service (Amazon S3) buckets for convenient access with data security features for users who are responsible for monitoring activities in the database.

Amazon Redshift Logs

Amazon Redshift logs information in the following log files:

- *Connection log* — logs authentication attempts, and connections and disconnections.
- *User log* — logs information about changes to database user definitions.
- *User activity log* — logs each query before it is run on the database.

The connection and user logs are useful primarily for security purposes. You can use the connection log to monitor information about the users who are connecting to the database and the related connection information, such as their IP address, when they made the request, what type of authentication they used, and so on. You can use the user log to monitor changes to the definitions of database users.

The user activity log is useful primarily for troubleshooting purposes. It tracks information about the types of queries that both the users and the system perform in the database.

The connection log and user log both correspond to information that is stored in the system tables in your database. You can use the system tables to obtain the same information, but the log files provide an easier mechanism for retrieval and review. The log files rely on Amazon S3 permissions rather than database permissions to perform queries against the tables. Additionally, by viewing the information in log files rather than querying the system tables, you reduce any impact of interacting with the database.

Note

Log files are not as current as the base system log tables, [STL_USERLOG](#) and [STL_CONNECTION_LOG](#). Records that are older than, but not including, the latest record are copied to log files.

Connection Log

Logs authentication attempts, and connections and disconnections. The following table describes the information in the connection log.

Column name	Description
event	Connection or authentication event.
recordtime	Time the event occurred.
remotehost	Name or IP address of remote host.
remoteport	Port number for remote host.
pid	Process ID associated with the statement.
dbname	Database name.
username	User name.
authmethod	Authentication method.
duration	Duration of connection in microseconds.
sslversion	Secure Sockets Layer (SSL) version.
sslcipher	SSL cipher.
mtu	Maximum transmission unit (MTU).
sslcompression	SSL compression type.
sslexpansion	SSL expansion type.

User Log

Records details for the following changes to a database user:

- Create user
- Drop user
- Alter user (rename)
- Alter user (alter properties)

Column name	Description
userid	ID of user affected by the change.
username	User name of the user affected by the change.

Column name	Description
oldusername	For a rename action, the original user name. For any other action, this field is empty.
action	Action that occurred. Valid values: <ul style="list-style-type: none"> • Alter • Create • Drop • Rename
usecreatedb	If true (1), indicates that the user has create database privileges.
usesuper	If true (1), indicates that the user is a superuser.
usecatupd	If true (1), indicates that the user can update system catalogs.
valuntil	Password expiration date.
pid	Process ID.
xid	Transaction ID.
recordtime	Time in UTC that the query started.

User Activity Log

Logs each query before it is run on the database.

Column name	Description
recordtime	Time the event occurred.
db	Database name.
user	User name.
pid	Process ID associated with the statement.
userid	User ID.
xid	Transaction ID.
query	A prefix of LOG: followed by the text of the query, including newlines.

Enabling Logging

Audit logging is not enabled by default in Amazon Redshift. When you enable logging on your cluster, Amazon Redshift creates and uploads logs to Amazon S3 that capture data from the creation of the cluster to the present time. Each logging update is a continuation of the information that was already logged.

Note

Audit logging to Amazon S3 is an optional, manual process. When you enable logging on your cluster, you are enabling logging to Amazon S3 only. Logging to system tables is not optional.

and happens automatically for the cluster. For more information about logging to system tables, see [System Tables Reference](#) in the Amazon Redshift Database Developer Guide.

The connection log, user log, and user activity log are enabled together by using the AWS Management Console, the Amazon Redshift API Reference, or the AWS Command Line Interface (AWS CLI). For the user activity log, you must also enable the `enable_user_activity_logging` database parameter. If you enable only the audit logging feature, but not the associated parameter, the database audit logs will log information for only the connection log and user log, but not for the user activity log. The `enable_user_activity_logging` parameter is disabled (`false`) by default, but you can set it to `true` to enable the user activity log. For more information, see [Amazon Redshift Parameter Groups \(p. 122\)](#).

Managing Log Files

The number and size of Amazon Redshift log files in Amazon S3 will depend heavily on the activity in your cluster. If you have an active cluster that is generating a large number of logs, Amazon Redshift might generate the log files more frequently. You might have a series of log files for the same type of activity, such as having multiple connection logs within the same hour.

Because Amazon Redshift uses Amazon S3 to store logs, you will incur charges for the storage that you use in Amazon S3. Before you configure logging, you should have a plan for how long you need to store the log files, and determine when they can either be deleted or archived based on your auditing needs. The plan that you create depends heavily on the type of data that you store, such as data subject to compliance or regulatory requirements. For more information about Amazon S3 pricing, go to [Amazon Simple Storage Service \(S3\) Pricing](#).

Bucket Permissions for Amazon Redshift Audit Logging

When you enable logging, Amazon Redshift collects logging information and uploads it to log files stored in Amazon S3. You can use an existing bucket or a new bucket. Amazon Redshift requires the following IAM permissions to the bucket:

- `s3:GetBucketAcl` The service requires read permissions to the Amazon S3 bucket so it can identify the bucket owner.
- `s3:PutObject` The service requires put object permissions to upload the logs. Each time logs are uploaded, the service determines whether the current bucket owner matches the bucket owner at the time logging was enabled. If these owners do not match, logging is still enabled but no log files can be uploaded until you select a different bucket.

If you want to use a new bucket, and have Amazon Redshift create it for you as part of the configuration process, the correct permissions will be applied to the bucket. However, if you create your own bucket in Amazon S3 or use an existing bucket, you need to add a bucket policy that includes the bucket name, and the Amazon Redshift Account ID that corresponds to your region from the following table:

Region Name	Region	Account ID
US East (N. Virginia) Region	us-east-1	193672423079
US East (Ohio) Region	us-east-2	391106570357
US West (N. California) Region	us-west-1	262260360010
US West (Oregon) Region	us-west-2	902366379725

Region Name	Region	Account ID
Asia Pacific (Mumbai) Region	ap-south-1	865932855811
Asia Pacific (Seoul) Region	ap-northeast-2	760740231472
Asia Pacific (Singapore) Region	ap-southeast-1	361669875840
Asia Pacific (Sydney) Region	ap-southeast-2	762762565011
Asia Pacific (Tokyo) Region	ap-northeast-1	404641285394
China (Ningxia) Region	cn-northwest-1	660998842044
Canada (Central) Region	ca-central-1	907379612154
EU (Frankfurt) Region	eu-central-1	053454850223
EU (Ireland) Region	eu-west-1	210876761215
EU (London) Region	eu-west-2	307160386991
EU (Paris) Region	eu-west-3	915173422425
South America (São Paulo) Region	sa-east-1	075028567923

The bucket policy uses the following format, where *BucketName* and *AccountId* are placeholders for your own values:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Put bucket policy needed for audit logging",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:user/logs"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::BucketName/*"
    },
    {
      "Sid": "Get bucket policy needed for audit logging",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:user/logs"
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::BucketName"
    }
  ]
}
```

The following example is a bucket policy for the US East (N. Virginia) Region and bucket named AuditLogs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Put bucket policy needed for audit logging",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:user/logs"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::AuditLogs/*"
    }
  ]
}
```

```
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::193672423079:user/logs"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::AuditLogs/*"
},
{
    "Sid": "Get bucket policy needed for audit logging",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::193672423079:user/logs"
    },
    "Action": "s3:GetBucketAcl",
    "Resource": "arn:aws:s3:::AuditLogs"
}
]
```

For more information about creating Amazon S3 buckets and adding bucket policies, go to [Creating a Bucket](#) and [Editing Bucket Permissions](#) in the Amazon Simple Storage Service Console User Guide.

Bucket Structure for Amazon Redshift Audit Logging

By default, Amazon Redshift organizes the log files in the Amazon S3 bucket by using the following bucket and object structure:

`AWSLogs/AccountId/ServiceName/Region/Year/Month/Day/AccountID_ServiceName_Region_ClusterName`

For example: `AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz`

If you provide an Amazon S3 key prefix, the prefix is placed at the start of the key.

For example, if you specify a prefix of myprefix: `myprefix/AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz`

The Amazon S3 key prefix cannot exceed 512 characters. It cannot contain spaces (), double quotation marks ("), single quotation marks ('), a backslash (\). There are also a number of special characters and control characters that are not allowed. The hexadecimal codes for these characters are:

- x00 to x20
- x22
- x27
- x5c
- x7f or larger

Troubleshooting Amazon Redshift Audit Logging

Amazon Redshift audit logging can be interrupted for the following reasons:

- Amazon Redshift does not have permission to upload logs to the Amazon S3 bucket. Verify that the bucket is configured with the correct IAM policy. For more information, see [Bucket Permissions for Amazon Redshift Audit Logging \(p. 301\)](#).
- The bucket owner changed. When Amazon Redshift uploads logs, it verifies that the bucket owner is the same as when logging was enabled. If the bucket owner has changed, Amazon Redshift cannot

upload logs until you configure another bucket to use for audit logging. For more information, see [Modifying the Bucket for Audit Logging \(p. 310\)](#).

- The bucket cannot be found. If the bucket is deleted in Amazon S3, Amazon Redshift cannot upload logs. You either need to recreate the bucket or configure Amazon Redshift to upload logs to a different bucket. For more information, see [Modifying the Bucket for Audit Logging \(p. 310\)](#).

Logging Amazon Redshift API Calls with AWS CloudTrail

Amazon Redshift is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Redshift. CloudTrail captures all API calls for Amazon Redshift as events, including calls from the Amazon Redshift console and from code calls to the Amazon Redshift APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Redshift. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Redshift, the IP address from which the request was made, who made the request, when it was made, and additional details.

You can use CloudTrail independently from or in addition to Amazon Redshift database audit logging.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon Redshift Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Redshift, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon Redshift, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all regions. The trail logs events from all regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Amazon Redshift actions are logged by CloudTrail and are documented in the [Amazon Redshift API Reference](#). For example, calls to the `CreateCluster`, `DeleteCluster`, and `DescribeCluster` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

Understanding Amazon Redshift Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry for a sample CreateCluster call.

```
{  
    "eventVersion": "1.04",  
    "userIdentity": {  
        "type": "IAMUser",  
        "principalId": "AIDAMVNPBQA3EXAMPLE",  
        "arn": "arn:aws:iam::123456789012:user/Admin",  
        "accountId": "123456789012",  
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
        "userName": "Admin",  
        "sessionContext": {  
            "attributes": {  
                "mfaAuthenticated": "false",  
                "creationDate": "2017-03-03T16:51:56Z"  
            }  
        },  
        "invokedBy": "signin.amazonaws.com"  
    },  
    "eventTime": "2017-03-03T16:56:09Z",  
    "eventSource": "redshift.amazonaws.com",  
    "eventName": "CreateCluster",  
    "awsRegion": "us-east-2",  
    "sourceIPAddress": "52.95.4.13",  
    "userAgent": "signin.amazonaws.com",  
    "requestParameters": {  
        "clusterIdentifier": "my-dw-instance",  
        "allowVersionUpgrade": true,  
        "enhancedVpcRouting": false,  
        "encrypted": false,  
        "clusterVersion": "1.0",  
        "masterUsername": "awsuser",  
        "masterUserPassword": "*****",  
        "automatedSnapshotRetentionPeriod": 1,  
        "port": 5439,  
        "dBName": "mydbtest",  
        "clusterType": "single-node",  
        "nodeType": "dc1.large",  
        "publiclyAccessible": true,  
        "vpcSecurityGroupIds": [  
            "sg-95f606fc"  
        ]  
    },  
    "responseElements": {  
        "nodeType": "dc1.large",  
        "preferredMaintenanceWindow": "sat:05:30-sat:06:00",  
        "clusterStatus": "creating",  
        "vpcId": "vpc-84c22aed",  
        "enhancedVpcRouting": false,  
        "masterUsername": "awsuser",  
        "clusterSecurityGroups": [],  
        "pendingModifiedValues": {  
            "masterUserPassword": "*****"  
        }  
    }  
}
```

```

},
"dBName": "mydbtest",
"clusterVersion": "1.0",
"encrypted": false,
"publiclyAccessible": true,
"tags": [],
"clusterParameterGroups": [
    {
        "parameterGroupName": "default.redshift-1.0",
        "parameterApplyStatus": "in-sync"
    }
],
"allowVersionUpgrade": true,
"automatedSnapshotRetentionPeriod": 1,
"numberOfNodes": 1,
"vpcSecurityGroups": [
    {
        "status": "active",
        "vpcSecurityGroupId": "sg-95f606fc"
    }
],
"iamRoles": [],
"clusterIdentifier": "my-dw-instance",
"clusterSubnetGroupName": "default"
},
"requestID": "4c506036-0032-11e7-b8bf-d7aaa466e9920",
"eventID": "13ba5550-56ac-405b-900a-8a42b0f43c45",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

The following example shows a CloudTrail log entry for a sample DeleteCluster call.

```

{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAMVNPBQA3EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Admin",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2017-03-03T16:58:23Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2017-03-03T17:02:34Z",
    "eventSource": "redshift.amazonaws.com",
    "eventName": "DeleteCluster",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "52.95.4.13",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "clusterIdentifier": "my-dw-instance",
        "skipFinalClusterSnapshot": true
    },
    "responseElements": null,
    "requestID": "324cb76a-0033-11e7-809b-1bbbef7710bf",
    "eventID": "59bcc3ce-e635-4cce-b47f-3419a36b3fa5",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}

```

```
}
```

Amazon Redshift Account IDs in AWS CloudTrail Logs

When Amazon Redshift calls another AWS service on your behalf, the call is logged with an account ID that belongs to the Amazon Redshift service instead of your own account ID. For example, when Amazon Redshift calls AWS Key Management Service (AWS KMS) actions such as CreateGrant, Decrypt, Encrypt, and RetireGrant to manage encryption on your cluster, the calls are logged by AWS CloudTrail using an Amazon Redshift account ID.

Amazon Redshift uses the account IDs in the following table when calling other AWS services.

Region	Region	Account ID
US East (N. Virginia) Region	us-east-1	368064434614
US East (Ohio) Region	us-east-2	790247189693
US West (N. California) Region	us-west-1	703715109447
US West (Oregon) Region	us-west-2	473191095985
Asia Pacific (Mumbai) Region	ap-south-1	408097707231
Asia Pacific (Seoul) Region	ap-northeast-2	713597048934
Asia Pacific (Singapore) Region	ap-southeast-1	960118270566
Asia Pacific (Sydney) Region	ap-southeast-2	485979073181
Asia Pacific (Tokyo) Region	ap-northeast-1	615915377779
China (Ningxia) Region	cn-northwest-1	194116488714
Canada (Central) Region	ca-central-1	764870610256
EU (Frankfurt) Region	eu-central-1	434091160558
EU (Ireland) Region	eu-west-1	246478207311
EU (London) Region	eu-west-2	885798887673
EU (Paris) Region	eu-west-3	694668203235
South America (São Paulo) Region	sa-east-1	392442076723

The following example shows a CloudTrail log entry for the AWS KMS Decrypt operation that was called by Amazon Redshift.

```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",

```

```
"principalId": "AROAI5QPCMKLTL4VHFCYY:i-0f53e22dbe5df8a89",
"arn": "arn:aws:sts::790247189693:assumed-role/prod-23264-role-wp/
i-0f53e22dbe5df8a89",
"accountId": "790247189693",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-03-03T16:24:54Z"
    },
    "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAI5QPCMKLTL4VHFCYY",
        "arn": "arn:aws:iam::790247189693:role/prod-23264-role-wp",
        "accountId": "790247189693",
        "userName": "prod-23264-role-wp"
    }
},
"eventTime": "2017-03-03T17:16:51Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-east-2",
"sourceIPAddress": "52.14.143.61",
"userAgent": "aws-internal/3",
"requestParameters": {
    "encryptionContext": {
        "aws:redshift:createtime": "20170303T1710Z",
        "aws:redshift:arn": "arn:aws:redshift:us-east-2:123456789012:cluster:my-dw-
instance-2"
    }
},
"responseElements": null,
"requestID": "30d2fe51-0035-11e7-ab67-17595a8411c8",
"eventID": "619bad54-1764-4de4-a786-8898b0a7f40c",
"readOnly": true,
"resources": [
    {
        "ARN": "arn:aws:kms:us-east-2:123456789012:key/f8f4f94f-e588-4254-
b7e8-078b99270be7",
        "accountId": "123456789012",
        "type": "AWS::KMS::Key"
    }
],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012",
"sharedEventID": "c1daefea-a5c2-4fab-b6f4-d8eaa1e522dc"
}
```

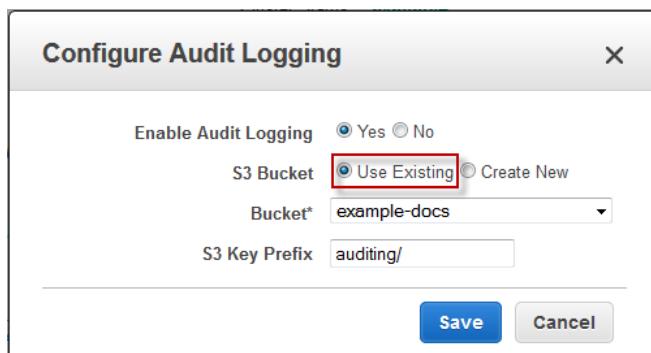
Configuring Auditing Using the Console

You can configure Amazon Redshift to create audit log files and store them in S3.

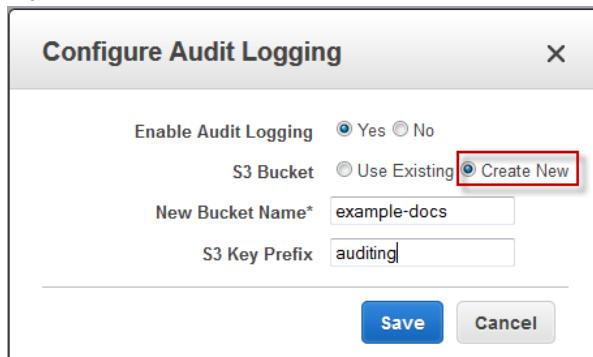
Enabling Audit Logging Using the Console

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, click **Clusters**.
3. In the list, click the cluster for which you want to enable logging.

4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. In the **Configure Audit Logging** dialog box, in the **Enable Audit Logging** box, click **Yes**.
6. For **S3 Bucket**, do one of the following:
 - If you already have an S3 bucket that you want to use, select **Use Existing** and then select the bucket from the **Bucket** list.

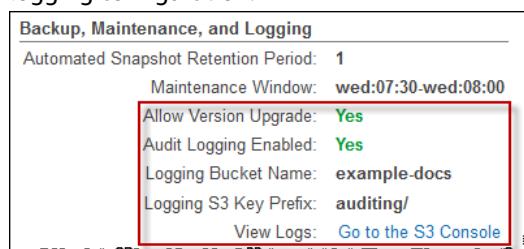


- If you need a new S3 bucket, select **Create New**, and in the **New Bucket Name** box, type a name.

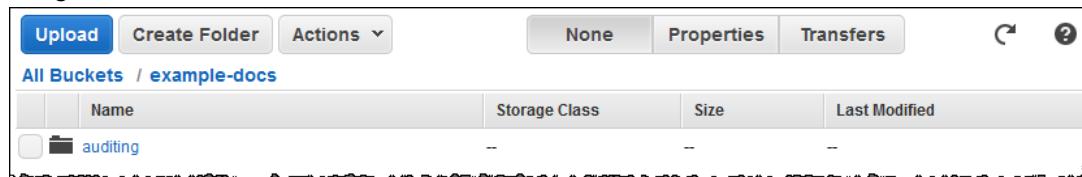


7. Optionally, in the **S3 Key Prefix** box, type a prefix to add to the S3 bucket.
8. Click **Save**.

After you configure audit logging, the **Cluster** details page updates to display information about the logging configuration.



On the **Cluster** details page, under **Backup, Maintenance, and Logging**, click **Go to the S3 console** to navigate to the bucket.

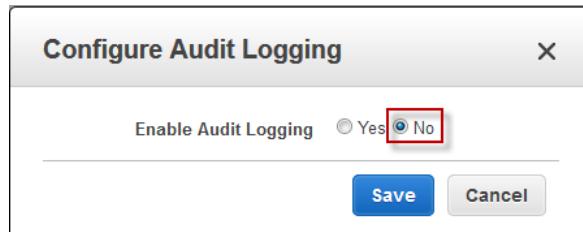


Modifying the Bucket for Audit Logging

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, click **Clusters**.
3. In the list, click the cluster for which you want to modify the bucket used for audit logging.
4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. For **S3 Bucket**, select an existing bucket or create a new bucket.
6. Optionally, in the **S3 Key Prefix** box, type a prefix to add to the S3 bucket.
7. Click **Save**.

Disabling Audit Logging Using the Console

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, click **Clusters**.
3. In the list, click the cluster for which you want to disable logging.
4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. In the **Configure Audit Logging** dialog box, in the **Enable Audit Logging** box, click **No**.



6. Click **Save**.

Configuring Logging by Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to configure audit logging.

- [describe-logging-status](#)
- [disable-logging](#)
- [enable-logging](#)

You can use the following Amazon Redshift API actions to configure audit logging.

- [DescribeLoggingStatus](#)
- [DisableLogging](#)
- [EnableLogging](#)

Resizing Clusters in Amazon Redshift

Topics

- [Overview \(p. 311\)](#)
- [Resize Operation Overview \(p. 311\)](#)
- [Snapshot, Restore, and Resize Operation Overview \(p. 312\)](#)
- [Tutorial: Using the Resize Operation to Resize a Cluster \(p. 313\)](#)
- [Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster \(p. 315\)](#)

Overview

As your data warehousing capacity and performance needs change or grow, you can resize your cluster to make the best use of the computing and storage options that Amazon Redshift provides. You can scale the cluster in or out by changing the number of nodes. Or, you can scale the cluster up or down by specifying a different node type. You can resize your cluster by using one of the following approaches:

- Use the resize operation with an existing cluster.
- Use the snapshot and restore operations to make a copy of an existing cluster. Then, resize the new cluster.

Both the resize approach and the snapshot and restore approach copy user tables and data to the new cluster; they do not do anything with system tables and data. If you have enabled audit logging in your source cluster, you'll be able to continue to access the logs in Amazon Simple Storage Service (Amazon S3) even after you delete the source cluster. You can keep or delete these logs as your data policies specify.

Resize Operation Overview

The resize operation is the preferred method to resize your cluster because it is the simplest method. With the resize operation, your data is copied in parallel from the compute node or nodes in your source cluster to the compute node or nodes in the target cluster. The time that it takes to resize depends on the amount of data and the number of nodes in the smaller cluster. It can take anywhere from a couple of hours to a couple of days.

When you start the resize operation, Amazon Redshift puts the existing cluster into read-only mode until the resize finishes. During this time, you can only run queries that read from the database; you cannot run any queries that write to the database, including read-write queries. For more information, see [Write and read-write operations](#) in the *Amazon Redshift Database Developer Guide*.

Note

If you would like to resize with minimal production impact, you can use the following section, [Snapshot, Restore, and Resize Operation Overview \(p. 312\)](#), to create a copy of your cluster, resize the copy, and then switch the connection endpoint to the resized cluster when the resize is complete.

After Amazon Redshift puts the source cluster into read-only mode, it provisions a new cluster, the target cluster, using the information that you specify for the node type, cluster type, and number of nodes. Then, Amazon Redshift copies the data from the source cluster to the target cluster. When this is complete, all connections switch to use the target cluster. If you have any queries in progress at the time this switch happens, your connection will be lost and you must restart the query on the target cluster. You can view the resize progress on the cluster's **Status** tab on the Amazon Redshift console.

Amazon Redshift does not sort tables during a resize operation, so the existing sort order is maintained. When you resize a cluster, Amazon Redshift distributes the database tables to the new nodes based on their distribution styles and runs an ANALYZE command to update statistics. Rows that are marked for deletion are not transferred, so you will only need to run a VACUUM command if your tables need to be resorted. For more information, see [Vacuuming tables](#) in the *Amazon Redshift Database Developer Guide*.

To walk through the process of resizing an Amazon Redshift cluster using the resize operation, see [Tutorial: Using the Resize Operation to Resize a Cluster \(p. 313\)](#).

Snapshot, Restore, and Resize Operation Overview

As described in the preceding section, the time it takes to resize a cluster with the resize operation depends heavily on the amount of data in the cluster. Because you cannot perform write or read-write operations in the database during the resize, you should determine whether you want to use the resize operation or an alternate method that reduces the amount of time that the cluster is in read-only mode.

If you require near-constant write access to your Amazon Redshift cluster, you can use the snapshot and restore operations described in the following section. This approach requires that any data that is written to the source cluster after the snapshot is taken must be copied manually to the target cluster after the switch. Depending on how long the copy takes, you might need to repeat this several times until you have the same data in both clusters and can make the switch to the target cluster. This process might have a negative impact on existing queries until the full set of data is available in the target cluster, but it does minimize the amount of time that you cannot write to the database.

The snapshot, restore, and resize approach uses the following process:

1. Take a snapshot of your existing cluster. The existing cluster is the source cluster.
2. Make note of the time the snapshot was taken so that you can later identify the point at which you'll need to rerun extract, transact, load (ETL) processes to load any post-snapshot data into the target database.
3. Restore the snapshot into a new cluster. This new cluster is the target cluster. Verify that the sample data exists in the target cluster.
4. Resize the target cluster. Select the new node type, number of nodes, and other settings for the target cluster.
5. Review the loads from your ETL processes that occurred after you took a snapshot of the source cluster. You'll need to reload the same data in the same order into the target cluster. If you have ongoing data loads, you'll need to repeat this process several times until the data is the same in both the source and target clusters.
6. Stop all queries running on the source cluster. To do this, you can reboot the cluster, or you can log on as a super user and use the `PG_CANCEL_BACKEND` and the `PG_TERMINATE_BACKEND` commands. Rebooting the cluster is the easiest way to make sure that the cluster is unavailable.
7. Rename the source cluster. For example, rename it from `examplecluster` to `examplecluster-source`.
8. Rename the target cluster to use the name of the source cluster prior to the rename. For example, rename the target cluster from preceding to `examplecluster`. From this point on, any applications that use the endpoint containing `examplecluster` will be connecting to the target cluster.

9. Delete the source cluster after you switch to the target cluster, and verify that all processes work as expected.

Alternatively, you can rename the source and target clusters before reloading data into the target cluster if you do not have a requirement that any dependent systems and reports be immediately up-to-date with those for the target cluster. In this case, step 6 would be moved to the end of the process described preceding.

The rename process is only required if you want applications to continue using the same endpoint to connect to the cluster. If you do not require this, you can instead update any applications that connect to the cluster to use the endpoint of the target cluster without renaming the cluster.

There are a couple of benefits to reusing a cluster name. First, you do not need to update application connection strings because the endpoint does not change, even though the underlying cluster changes. Second, related items such as Amazon CloudWatch alarms and Amazon Simple Notification Service (Amazon SNS) notifications are tied to the cluster name, so you can continue using the same alarms and notifications that you've set up for the cluster. This continued use is primarily a concern in production environments where you want to have the flexibility to resize the cluster without having to reconfigure related items, such as alarms and notifications.

To walk through the process of resizing an Amazon Redshift cluster using the snapshot, restore, and resize operations, see [Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster \(p. 315\)](#).

Tutorial: Using the Resize Operation to Resize a Cluster

This section walks you through the process of resizing a cluster by using the resize operation in Amazon Redshift. In this example, you'll scale your cluster out by resizing from a single node cluster to a multinode cluster.

Complete this tutorial by performing the steps in the following:

- [Prerequisites \(p. 313\)](#)
- [Step 1: Resize the Cluster \(p. 313\)](#)
- [Step 2: Delete the Sample Cluster \(p. 314\)](#)

Prerequisites

Before you start this tutorial, make sure that you have the following prerequisites:

- A sample cluster. In this example, you'll start with the sample cluster that you created in the [Amazon Redshift Getting Started](#) exercise. If you don't have a sample cluster to use for this tutorial, complete the Getting Started exercise to create one and then return to this tutorial.

Step 1: Resize the Cluster

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from the [Amazon Redshift Getting Started](#) exercise, click **examplecluster**.
3. On the **Configuration** tab of the **Cluster** details page, click **Resize** in the **Cluster** list.



4. In the **Resize Cluster** window, select the following values:

- **Node Type:** dc1.large.
- **Cluster Type:** Multi Node.
- **Number of Nodes:** 2.

Resize Cluster

Choose the number of nodes and optionally a new node type for the resize operation. Note that the available node type and cluster type options may be limited by the cluster's current availability zone.

Node Type: dc1.large Cluster Type: Multi Node Number Of Nodes*: 2

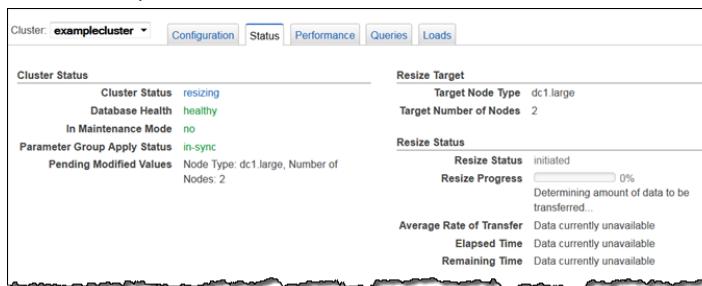
Please make sure the resized cluster is large enough to hold the data that is currently on the cluster, otherwise the resize will fail.

Warning: Resizing the cluster will cause it to be restarted into read-only mode for the duration of the resize operation. All currently executing queries and database connections on the cluster will be terminated when the resize operation begins and again when it is complete. For more information, see [Resizing a Cluster](#).

Cancel **Resize**

5. Click **Resize**.

6. Click **Status**, and review the resize status information to see the resize progress.



Step 2: Delete the Sample Cluster

After you are sure that you no longer need the sample cluster, you can delete it. In a production environment, whether you decide to keep a final snapshot depends on your data policies. In this tutorial, you'll delete the cluster without a final snapshot because you are using sample data.

Important

You are charged for any clusters until they are deleted.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster names from this tutorial, click **examplecluster**.

3. On the **Configuration** tab of the **Cluster** details page, click **Delete** in the **Cluster** list.
4. In the **Delete Cluster** window, click **No** for **Create final snapshot**, and then click **Delete**.

Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster

This section walks you through the process of using the snapshot and restore operations as part of a resize process for an Amazon Redshift cluster. This process is an advanced one that is useful primarily in environments where you are unable or do not want to stop write and read-write operations in the database for the period of time it takes to resize your cluster. If you are unsure how long your cluster takes to resize, you can use this procedure to take a snapshot, restore it into a new cluster, and then resize it to get an estimate. This section takes that process further by switching from the source to the target cluster after the resize of the target cluster completes.

Important

You are charged for any clusters until they are deleted.

Complete this tutorial by performing the steps in the following:

- [Prerequisites \(p. 315\)](#)
- [Step 1: Take a Snapshot \(p. 315\)](#)
- [Step 2: Restore the Snapshot into the Target Cluster \(p. 316\)](#)
- [Step 3: Verify Data in the Target Cluster \(p. 318\)](#)
- [Step 4: Resize the Target Cluster \(p. 318\)](#)
- [Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster \(p. 319\)](#)
- [Step 6: Rename the Source and Target Clusters \(p. 320\)](#)
- [Step 7: Delete the Source Cluster \(p. 321\)](#)
- [Step 8: Clean Up Your Environment \(p. 322\)](#)

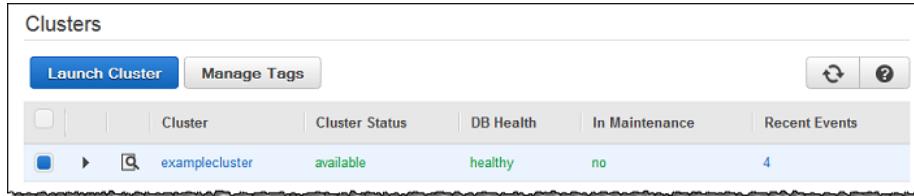
Prerequisites

Before you start this tutorial, make sure that you have the following prerequisites:

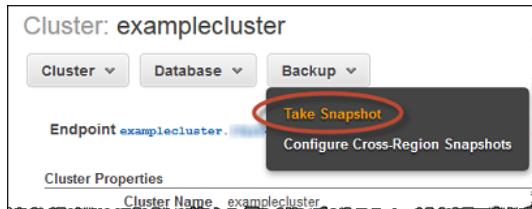
- A sample cluster. In this example, you'll start with the sample cluster that you created in the [Amazon Redshift Getting Started](#) exercise. If you don't have a sample cluster to use for this tutorial, complete the Getting Started exercise to create one and then return to this tutorial.
- A SQL client tool or application to connect to the cluster. This tutorial uses SQL Workbench/J, which you installed if you performed the steps in the [Amazon Redshift Getting Started](#) exercise. If you do not have SQL Workbench/J or another SQL client tool, see [Connect to Your Cluster by Using SQL Workbench/J \(p. 100\)](#).
- Sample data. In this tutorial, you'll take a snapshot of your cluster, and then perform some write queries in the database that cause a difference between the data in the source cluster and the new cluster where you will restore the snapshot. Before you begin this tutorial, load your cluster with the sample data from Amazon S3 as described in the [Amazon Redshift Getting Started](#) exercise.

Step 1: Take a Snapshot

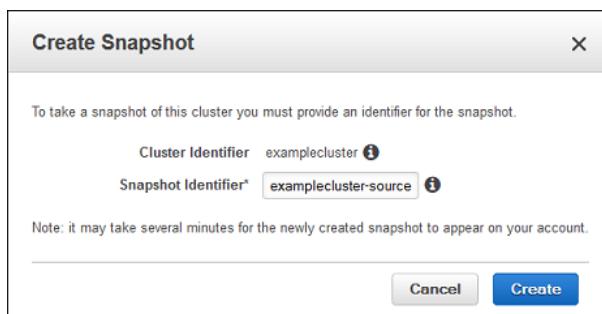
1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from the Amazon Redshift Getting Started exercise, click **examplecluster**.



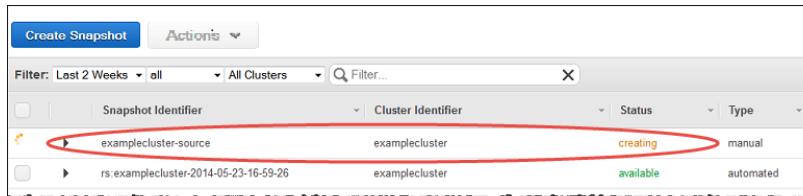
3. On the **Configuration** tab of the **Cluster** details page, click **Take Snapshot** in the **Backup** list.



4. In the **Create Snapshot** window, type *examplecluster-source* in the **Snapshot Identifier** box, and then click **Create**.

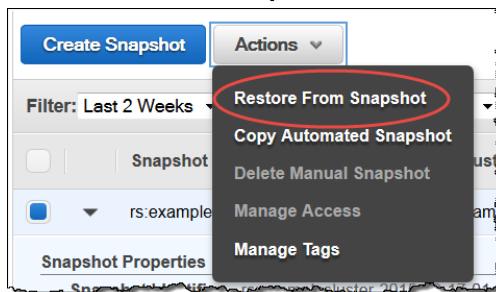


5. In the navigation pane, click **Snapshots** and verify that a new manual snapshot is being created. The snapshot status will be **creating**.



Step 2: Restore the Snapshot into the Target Cluster

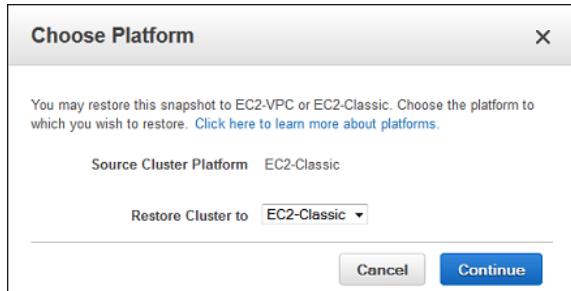
1. In the navigation pane, click **Snapshots**, and then select the **examplecluster-source** snapshot.
2. Click **Restore From Snapshot**.



3. In the **Choose Platform** window, select the platform you want to restore the cluster into. For this example, choose **EC2-VPC**. Then, click **Continue**.

Note

To use EC2-VPC, you must have a cluster subnet group. For more information, see [Creating a Cluster Subnet Group \(p. 40\)](#).



4. In the **Restore Cluster From Snapshot** window, do the following:

- **Snapshot Identifier:** check the snapshot name, **examplecluster-source**.
- **Cluster Identifier:** type **examplecluster-target**.
- **Port:** leave the port number as is.
- **Allow Version Upgrade:** leave this option as **Yes**.
- **Availability Zone:** select an Availability Zone.
- **Cluster Parameter Group:** select a parameter group to use.
- **Cluster Security Group:** select a security group or groups to use.

5. In the navigation pane, click **Clusters**. A new cluster, **examplecluster-target**, will be created from the source cluster's snapshot.

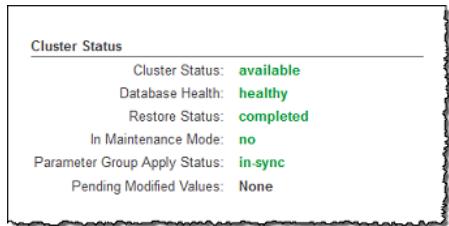
First, the target cluster is created. The **Cluster Status** value is **creating, restoring** at this point.

Clusters				
Launch Cluster				
Cluster	Cluster Status	DB Health	In Maintenance	
examplecluster	available	healthy	no	
examplecluster-target	creating, restoring	unknown	unknown	

After the target cluster is created, the **Cluster Status** value changes to **available, restoring**.

Clusters				
Launch Cluster				
Cluster	Cluster Status	DB Health	In Maintenance	
examplecluster	available	healthy	no	
examplecluster-target	available, restoring	healthy	no	

6. Click **examplecluster-target** to open it. The **Cluster Status** value should display **available**, and the **Restore Status** should display **completed**.



Step 3: Verify Data in the Target Cluster

After the restore operation completes, you can verify that the data in the target cluster meets your expectation of the data that you had in the snapshot from the source. You can use a SQL client tool to connect to the target cluster and run a query to validate the data in the new cluster. For example, you can run the same queries that you ran in the Amazon Redshift Getting Started exercise:

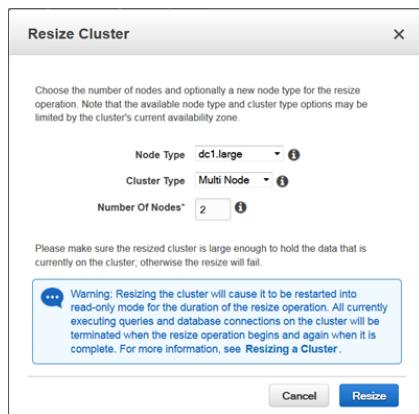
```
-- Get definition for the sales table.  
SELECT *  
FROM pg_table_def  
WHERE tablename = 'sales';  
  
-- Find total sales on a given calendar date.  
SELECT sum(qtysold)  
FROM sales, date  
WHERE sales.dateid = date.dateid  
AND caldate = '2008-01-05';  
  
-- Find top 10 buyers by quantity.  
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 4: Resize the Target Cluster

Once you verify that your target cluster works as expected, you can resize the target cluster. You can continue to allow write and read-write operations in the source cluster, because later in this tutorial you will copy any data that was loaded after your snapshot to the target.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from this tutorial, click **examplecluster-target**.

3. On the **Configuration** tab of the **Cluster** details page, click **Resize** in the **Cluster** list.
4. In the **Resize Cluster** window, select the following values:
 - **Node Type:** dc1.large.
 - **Cluster Type:** Multi Node.
 - **Number of Nodes:** 2.



5. Click **Resize**.
6. Click **Status**, and review the resize status information to see the resize progress.

Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster

For the purposes of this tutorial, this step provides a simple set of COPY statements to load data from Amazon S3 into Amazon Redshift. This step is included to simulate bringing the target cluster up-to-date with the same data as the source cluster. It is not meant to demonstrate an effort to bring an actual production environment into line between the source and target cluster. In production environments, your own ETL process will determine how load your target cluster with all the same data as the source cluster after the snapshot was taken.

If there have been multiple loads after the snapshot was taken, you'll need to make sure that you rerun the loads in the target database in the same order as they were run in the source database. Additionally, if there continue to be loads into the source database while you are working on bringing the target cluster up-to-date, you will need to repeat this process until the target and source match, and find a suitable time to rename the clusters and switch applications to connect to the target database.

In this example, let's suppose that your ETL process loaded data into the source cluster after the snapshot was taken. Perhaps Amazon Redshift was still in the process of restoring the target cluster from the snapshot, or resizing the target cluster. There were some new categories, events, dates, and venues added to the TICKIT database. You now need to get this same data into the target cluster before you switch to use it going forward.

First, you'll use the following COPY statements to load new data from Amazon S3 to the tables in your Amazon Redshift TICKIT database in the target cluster.

The sample data for this tutorial is provided in Amazon S3 buckets that are 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, make sure you have the following for your IAM user:

- Your access key and secret access key. If you do not know these, you can create new ones. For more information, go to [Administering Access Keys for IAM Users](#) in *IAM User Guide*.
- At least LIST and GET permissions to Amazon S3 resources. You can grant your IAM user these permissions by attaching the `AmazonS3ReadOnlyAccess` managed policy to your IAM user or to the group to which your IAM user belongs. For more information about attaching policies, go to [Managing IAM Policies](#) in the *IAM User Guide*.

Note

Without proper permissions to Amazon S3, you receive the following error message when running the COPY command: `S3ServiceException: Access Denied`.

Replace `<access-key-id>` and `<secret-access-key>` with the access key and secret access key for your IAM user. Then run the commands in your SQL client tool.

```
copy venue from 's3://awssampled़/resize/etl_venue_pipe.txt' CREDENTIALS
  'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-Secret-Access-Key>'
  delimiter '|' region 'us-east-1';
copy category from 's3://awssampled़/resize/etl_category_pipe.txt' CREDENTIALS
  'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-Secret-Access-Key>'
  delimiter '|' region 'us-east-1';
copy date from 's3://awssampled़/resize/etl_date_pipe.txt' CREDENTIALS
  'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-Secret-Access-Key>'
  delimiter '|' region 'us-east-1';
copy event from 's3://awssampled़/resize/etl_events_pipe.txt' CREDENTIALS
  'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-Secret-Access-Key>'
  delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-east-1';
```

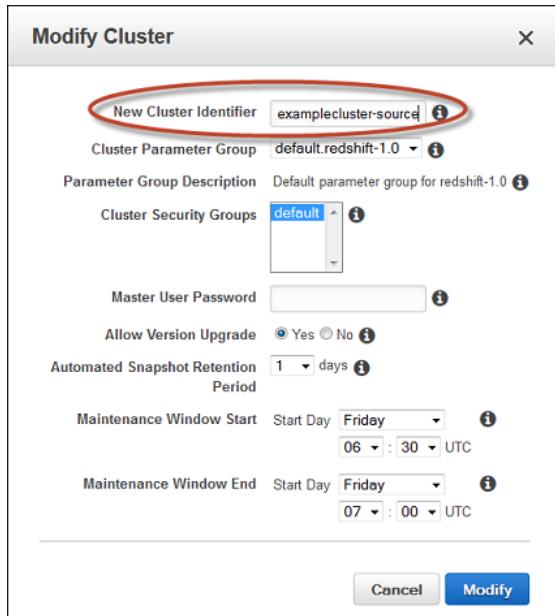
Step 6: Rename the Source and Target Clusters

Once you verify that your target cluster has been brought up to date with any data needed from the ETL process, you can switch to the target cluster. If you need to keep the same name as the source cluster, you'll need to do a few manual steps to make the switch. These steps involve renaming the source and target clusters, during which time they will be unavailable for a short period of time. However, if you are able to update any data sources to use the new target cluster, you can skip this section.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from this tutorial, click `examplecluster`.
3. On the **Configuration** tab of the **Cluster** details page, click **Modify** in the **Cluster** list.



4. In the **Modify Cluster** window, type **examplecluster-source** in the **New Cluster Identifier** box, and then click **Modify**.



5. In the navigation pane, click **Clusters**, and then click **examplecluster-target**.
6. On the **Configuration** tab of the **Cluster** details page, click **Modify** in the **Cluster** list.
7. In the **Modify Cluster** window, type **examplecluster** in the **New Cluster Identifier** box, and then click **Modify**.

If you had any queries running in the source cluster, you'll need to start them over and run them to completion on the target cluster.

Step 7: Delete the Source Cluster

After you are sure that you no longer need the source cluster, you can delete it. In a production environment, whether you decide to keep a final snapshot depends on your data policies. In this tutorial, you'll delete the cluster without a final snapshot because you are using sample data.

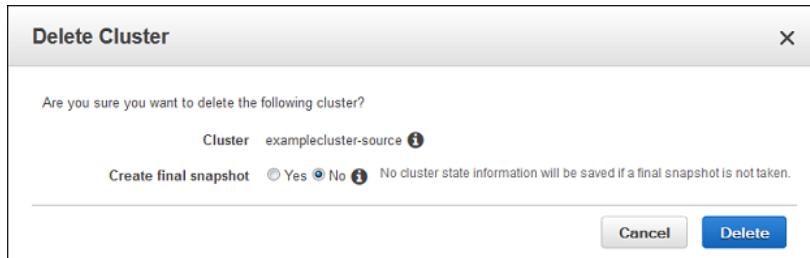
Important

You are charged for any clusters until they are deleted.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster names from this tutorial, click **examplecluster-source**.
3. On the **Configuration** tab of the **Cluster** details page, click **Delete** in the **Cluster** list.



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



Step 8: Clean Up Your Environment

After you have completed this tutorial, you can clean up your environment by deleting the target cluster. To do this, follow the steps in [Step 7: Delete the Source Cluster \(p. 321\)](#) and instead delete the target cluster. Doing this will return your environment back to the state it was in before you started the tutorial. Returning the environment to the original state is important to help reduce any costs associated with having clusters running.

Important

You are charged for any clusters until they are deleted.

Limits in Amazon Redshift

Quotas and Limits

Amazon Redshift has quotas that limit the total number of nodes that you can provision, and the number of snapshots that you can create; these quotas are per AWS account per region. Amazon Redshift has a default quota for each of these, which are listed at [AWS Service Limits](#). If you attempt to exceed any of these quotas, the attempt will fail. To increase these Amazon Redshift quota limits for your account in a region, request a change by submitting an [Amazon Redshift Limit Increase Form](#).

In addition to quotas, Amazon Redshift has limits for the following per-cluster values. These limits cannot be increased:

- The number of nodes that you can allocate per cluster, which is based on the cluster's node type. This limit is separate from the limit for your AWS account per region. For more information about the current node limits for each node type, see [Clusters and Nodes in Amazon Redshift \(p. 6\)](#).
- The maximum number of tables is 9,900 for large and xlarge cluster node types and 20,000 for 8xlarge cluster node types. The limit includes temporary tables. Temporary tables include user-defined temporary tables and temporary tables created by Amazon Redshift during query processing or system maintenance. Views are not included in this limit. For more information about creating a table, see [Create Table Usage Notes](#) in the *Amazon Redshift Database Developer Guide*. For more information about cluster node types, see [Clusters and Nodes in Amazon Redshift \(p. 6\)](#).
- The number of user-defined databases you can create per cluster is 60. For more information about creating a database, see [Create Database](#) in the *Amazon Redshift Database Developer Guide*.
- The number of schemas you can create in each database in a cluster is 9,900, however, pg_temp_* schemas do not count towards this quota. For more information about creating a schema, see [Create Schema](#) in the *Amazon Redshift Database Developer Guide*.
- The number of concurrent user connections that can be made to a cluster is 500. For more information, see [Connecting to an Amazon Redshift Cluster Using SQL Client Tools \(p. 60\)](#) in the *Amazon Redshift Cluster Management Guide*.
- A workload management (WLM) configuration can define a total concurrency level of 50 for all user-defined queues. For more information, see [Defining Query Queues](#) in the *Amazon Redshift Database Developer Guide*.
- The number of AWS accounts you can authorize to restore a snapshot is 20 for each snapshot and 100 for each AWS Key Management Service (AWS KMS) key. That is, if you have 10 snapshots that are encrypted with a single KMS key, then you can authorize 10 AWS accounts to restore each snapshot, or other combinations that add up to 100 accounts and do not exceed 20 accounts for each snapshot. For more information, see [Sharing Snapshots \(p. 150\)](#) in the *Amazon Redshift Cluster Management Guide*.
- The maximum size of a single row loaded by using the COPY command is 4 MB. For more information, see [COPY](#) in the *Amazon Redshift Database Developer Guide*.
- A maximum of 10 IAM roles can be associated with a cluster to authorize Amazon Redshift to access other AWS services on behalf of the user that owns the cluster and IAM role. For more information, see [Authorizing Amazon Redshift to Access Other AWS Services on Your Behalf \(p. 235\)](#).

Spectrum limits

Amazon Redshift Spectrum has the following limits when using the Athena or AWS Glue data catalog:

- A maximum of 10,000 databases per account.
- A maximum of 100,000 tables per database.
- A maximum of 1,000,000 partitions per table.
- A maximum of 10,000,000 partitions per account.
- All S3 data must be located in the same region as the Redshift cluster.
- Timestamps in ION and JSON must use [ISO8601](#) format.
- The maximum allowed size of a row object in a ION/JSON file is 8MB.

Objects in ION or JSON files are the data enclosed by brackets, for example `(`{ "col1": "data1", "col2": "data2" }`)`.

- The maximum allowed size of a string value in an ION or JSON file is 16KB.
- External compression of ORC files is not supported.
- Text, OpenCSV and Regex SERDEs do not support octal delimiters larger than `\177`.
- You must specify a predicate on the partition column to avoid reads from all partitions.

For example, the following predicate will filter on the column `ship_dtm` but will not apply the filter to the partition column `ship_yyyyymm`:

```
WHERE ship_dtm > '2018-04-01'.
```

To skip unneeded partitions you need to add a predicate `WHERE ship_yyyyymm = '201804'`. This predicate will limit read operations to the partition `\ship_yyyyymm=201804\`.

- The column limit for external tables is 1598 with pseudo columns enabled and 1600 without.

You can request a limit increase by contacting AWS Support.

These limits don't apply to a Hive metastore.

Naming Constraints

The following table describes naming constraints within Amazon Redshift.

Cluster identifier	<ul style="list-style-type: none">• A cluster identifier must contain only lowercase characters.• It must contain from 1 to 63 alphanumeric characters or hyphens.• Its first character must be a letter.• It cannot end with a hyphen or contain two consecutive hyphens.• It must be unique for all clusters within an AWS account.
Database name	<ul style="list-style-type: none">• A database name must contain 1 to 64 alphanumeric characters.• It must contain only lowercase letters.• It cannot be a reserved word. For a list of reserved words, see Reserved Words in the <i>Amazon Redshift Database Developer Guide</i>.
Master user name	<ul style="list-style-type: none">• A master user name must contain only lowercase characters.• It must contain from 1 to 128 alphanumeric characters.

	<ul style="list-style-type: none"> • Its first character must be a letter. • It cannot be a reserved word. For a list of reserved words, see Reserved Words in the <i>Amazon Redshift Database Developer Guide</i>.
Master password	<ul style="list-style-type: none"> • A master password must be between 8 and 64 characters in length. • It must contain at least one uppercase letter. • It must contain at least one lowercase letter. • It must contain one number. • It can be any printable ASCII character (ASCII code 33 to 126) except ' (single quotation mark), " (double quotation mark), \, /, @, or space.
Parameter group name	<ul style="list-style-type: none"> • A parameter group name must be 1 to 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • Its first character must be a letter. • It cannot end with a hyphen or contain two consecutive hyphens.
Cluster security group name	<ul style="list-style-type: none"> • A cluster security group name must contain no more than 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • It must not be Default. • It must be unique for all security groups that are created by your AWS account.
Subnet group name	<ul style="list-style-type: none"> • A subnet group name must contain no more than 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • It must not be Default. • It must be unique for all security groups that are created by your AWS account.
Cluster snapshot identifier	<ul style="list-style-type: none"> • A cluster snapshot identifier must contain no more than 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • It must not be Default. • It must be unique for all security groups that are created by your AWS account.

Tagging Resources in Amazon Redshift

Topics

- [Tagging Overview \(p. 326\)](#)
- [Managing Resource Tags Using the Console \(p. 327\)](#)
- [Managing Tags Using the Amazon Redshift API \(p. 329\)](#)

Tagging Overview

In AWS, tags are user-defined labels that consist of key-value pairs. Amazon Redshift supports tagging to provide metadata about resources at a glance, and to categorize your billing reports based on cost allocation. To use tags for cost allocation, you must first activate those tags in the AWS Billing and Cost Management service. For more information about setting up and using tags for billing purposes, see [Use Cost Allocation Tags for Custom Billing Reports](#) and [Setting Up Your Monthly Cost Allocation Report](#).

Tags are not required for resources in Amazon Redshift, but they help provide context. You might want to tag resources with metadata about cost centers, project names, and other pertinent information related to the resource. For example, suppose you want to track which resources belong to a test environment and a production environment. You could create a key named `environment` and provide the value `test` or `production` to identify the resources used in each environment. If you use tagging in other AWS services or have standard categories for your business, we recommend that you create the same key-value pairs for resources in Amazon Redshift for consistency.

Tags are retained for resources after you resize a cluster, and after you restore a snapshot of a cluster within the same region. However, tags are not retained if you copy a snapshot to another region, so you must recreate the tags in the new region. If you delete a resource, any associated tags are deleted.

Each resource has one *tag set*, which is a collection of one or more tags assigned to the resource. Each resource can have up to 50 tags per tag set. You can add tags when you create a resource and after a resource has been created. You can add tags to the following resource types in Amazon Redshift:

- CIDR/IP
- Cluster
- Cluster security group
- Cluster security group ingress rule
- EC2 security group
- HSM connection
- HSM client certificate
- Parameter group
- Snapshot
- Subnet group

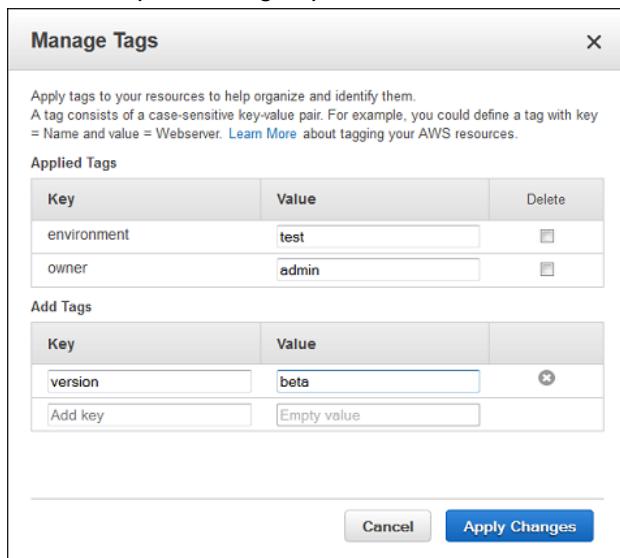
Tagging Requirements

Tags have the following requirements:

- Keys can't be prefixed with aws:.
- Keys must be unique per tag set.
- A key must be between 1 and 128 allowed characters.
- A value must be between 0 and 256 allowed characters.
- Values do not need to be unique per tag set.
- Allowed characters for keys and values are Unicode letters, digits, white space, and any of the following symbols: _ . : / = + - @.
- Keys and values are case sensitive.

Managing Resource Tags Using the Console

The following is an example of the **Manage Tags** window for an Amazon Redshift resource, such as a cluster or a parameter group.

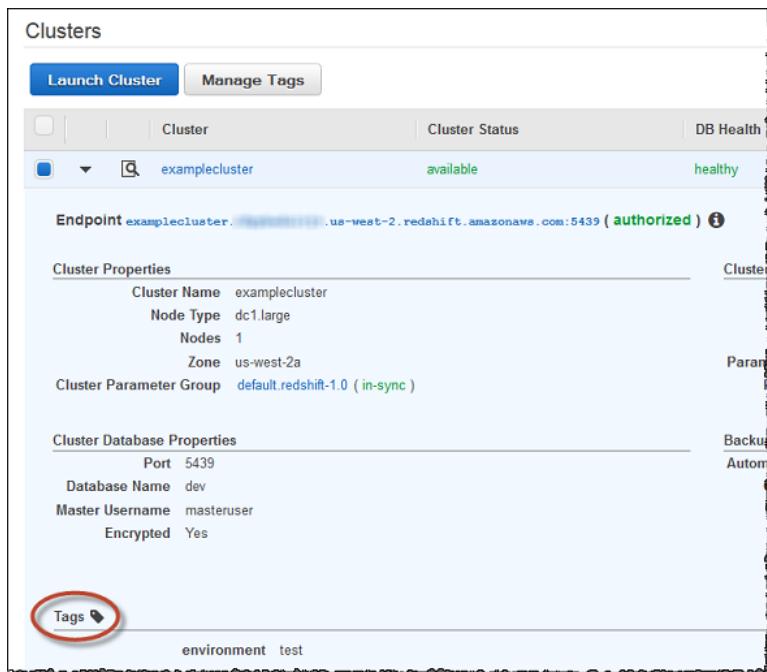


You use the **Add Tags** section to add key pairs to an Amazon Redshift resource. When you begin entering a key pair in the **Add Tags** section, a new row will appear so that you can add another key pair, and so on. For more information about allowed characters for keys and values, see [Tagging Requirements \(p. 327\)](#).

If you decide that you don't want to add a particular tag to the resource, you can remove it from the **Add Tags** section by clicking the X in the row. Once you have specified the key pairs that you want to add, you apply the changes so that they are associated with the resource.

After you add key pairs to a resource, they display in the **Applied Tags** section; this is the tag set for the resource. You can modify a tag value, but you can't modify the key name. You can, however, delete a key if you no longer need it for the resource.

You can view the tags for a resource by reviewing the **Applied Tags** section of the **Manage Tags** window. Alternatively, you can quickly view tags by navigating to a resource type in the navigation pane, and then expanding the resource in the list to view the **Tags** section. The following is an example of a cluster expanded to show various properties, including tags associated with the cluster.



How To Open the Manage Tags Window

The following table describes how to open the **Manage Tags** window for each of the Amazon Redshift resources that support tags.

Resource	Description
Cluster	In the left navigation pane, click Clusters and select a cluster from the list. Then click Manage Tags .
Snapshot	In the left navigation pane, click Snapshots and select a snapshot from the list. Then click Actions , and click Manage Tags from the menu.
Cluster Security Group	In the left navigation pane, click Security . On the Security Groups tab, select a security group from the list. Then click Manage Tags .
Cluster Security Group Ingress Rule	In the left navigation pane, click Security . On the Security Groups tab, click a security group in the list. On the Security Group Connections page, select an ingress rule and then click Manage Tags .
Subnet Group	In the left navigation pane, click Security . On the Subnet Groups tab, select a subnet group from the list. Then click Manage Tags .
HSM Connection	In the left navigation pane, click Security . On the HSM Connections tab, select a connection from the list. Then click Manage Tags .
HSM Certificate	In the left navigation pane, click Security . On the HSM Certificates tab, select a certificate from the list. Then click Manage Tags .
Parameter Group	In the left navigation pane, click Parameter Groups and select a parameter group from the list. Then click Manage Tags .

How to Manage Tags in the Amazon Redshift Console

Use the table in the previous section to navigate to the resource that you want to work with, and then use the procedures in this section to add, modify, delete, and view tags for the resource.

To add tags to a resource

1. Navigate to the resource to which you want to add tags, and open the **Manage Tags** window.
2. Under **Add Tags**, type a key name in the **Key** box and the key value in the **Value** box. For example, type **environment** in the **Key** box and **production** in the **Value** box. Repeat this step to add any additional tags.
3. Click **Apply Changes**.

To modify tags associated with a resource

1. Navigate to the resource for which you want to modify tags, and open the **Manage Tags** window.
2. Under **Applied Tags**, locate the key that you want to modify. In the **Value** box, type a new key value. Repeat for any other tags that you want to modify.
3. Click **Apply Changes**.

To delete tags associated with a resource

1. Navigate to the resource from which you want to delete tags, and open the **Manage Tags** window.
2. Under **Applied Tags**, locate the key that you want to delete. Select the **Delete** check box. Repeat for any other tags that you want to delete.
3. Click **Apply Changes**.

Managing Tags Using the Amazon Redshift API

You can use the following AWS CLI operations to manage tags in Amazon Redshift.

- [create-tags](#)
- [delete-tags](#)
- [describe-tags](#)

You can use the following Amazon Redshift APIs to manage tags:

- [CreateTags](#)
- [DeleteTags](#)
- [DescribeTags](#)
- [Tag](#)
- [TaggedResource](#)

Additionally, you can use the following Amazon Redshift APIs to manage and view tags for a specific resource:

- [CreateCluster](#)
- [CreateClusterParameterGroup](#)
- [CreateClusterSecurityGroup](#)

- [CreateClusterSnapshot](#)
- [CreateClusterSubnetGroup](#)
- [CreateHsmClientCertificate](#)
- [CreateHsmConfiguration](#)
- [DescribeClusters](#)
- [DescribeClusterParameterGroups](#)
- [DescribeClusterSecurityGroups](#)
- [DescribeClusterSnapshots](#)
- [DescribeClusterSubnetGroups](#)
- [DescribeHsmClientCertificates](#)
- [DescribeHsmConfigurations](#)

Document History

The following table describes the important changes in each release of the *Amazon Redshift Cluster Management Guide* after June 2018. For notification about updates to this documentation, you can subscribe to an RSS feed.

API version: 2012-12-01

Latest documentation update: October 10, 2018

For a list of the changes to the *Amazon Redshift Database Developer Guide*, see [Amazon Redshift Database Developer Guide Document History](#).

For more information about new features, including a list of fixes and the associated cluster version numbers for each release, see [Cluster Version History](#).

update-history-change	update-history-description	update-history-date
Cluster release version 1.0.4349	Release notes for cluster release version 1.0.4349 are available. For more information, see Version 1.0.4349 .	October 10, 2018
Amazon Redshift Spectrum supports Enhanced VPC Routing	You can now use Redshift Spectrum with Enhanced VPC Routing enabled for your cluster. You might need to perform additional configuration steps. For more information, see Using Amazon Redshift Spectrum with Enhanced VPC Routing .	October 10, 2018
Query Editor	You can now run SQL queries from the Amazon Redshift management console.	October 4, 2018
Workload Execution Breakdown Chart	You can now get a detailed view of your workload's performance by looking at the Workload Execution Breakdown chart in the console. For more information, see Analyzing Workload Performance .	July 30, 2018
Amazon Redshift Maintenance Tracks	You can now determine if your cluster will always be updated to the latest version of Amazon Redshift or to a previous version by selecting a Maintenance Track. For more information, see Choosing Cluster Maintenance Tracks .	July 26, 2018

Updated JDBC and ODBC drivers	Amazon Redshift now supports new versions of the JDBC and ODBC drivers. For more information, see Configure a JDBC Connection and Configure and ODBC Connection	July 13, 2018
On-demand cluster release versions	You can now upgrade your cluster to the latest release version as soon as it is available. For more information, see Manage Cluster Versions .	June 29, 2018

The following table describes the important changes to the *Amazon Redshift Cluster Management Guide* before July 2018.

Change	Description	Release Date
New CloudWatch metrics	New CloudWatch metrics added for monitoring query performance. For more information, see Amazon Redshift Performance Data (p. 256)	May 17, 2018
New JDBC and ODBC drivers	Amazon Redshift JDBC drivers have been updated to version 1.2.12.1017. For more information, see Configure a JDBC Connection (p. 62) . Amazon Redshift ODBC drivers have been updated to version 1.4.1.1001. For more information, see Configure an ODBC Connection (p. 77) .	March 7, 2018
HSM encryption	Amazon Redshift supports only AWS CloudHSM for hardware security module (HSM) key management. For more information, see Amazon Redshift Database Encryption (p. 163) .	March 6, 2018
IAM Role Chaining	If an IAM role attached to your cluster doesn't have access to the necessary resources, you can chain another role, possibly belonging to another account. Your cluster then temporarily assumes the chained role to access the data. You can also grant cross-account access by chaining roles. Each role in the chain assumes the next role in the chain, until the cluster assumes the role at the end of chain. You can chain a maximum of 10 roles. For more information, see Chaining IAM Roles in Amazon Redshift (p. 238) .	February 23, 2018
New DC2 node types	The new generation of dense compute (DC) node types offer much better performance at the same price as DC1. To take advantage of performance improvements, you can migrate your DC1 cluster to the newer DC2 node types. For more information, see Clusters and Nodes in Amazon Redshift (p. 6) .	October 17, 2017
New JDBC drivers	Amazon Redshift JDBC drivers have been updated to version 1.2.10.1009. Also, JDBC version 4.2 drivers are now supported. For more information, see Configure a JDBC Connection (p. 62) .	October 16, 2017

Change	Description	Release Date
ACM certificates	Amazon Redshift is replacing the SSL certificates on your clusters with AWS Certificate Manager (ACM) issued certificates. ACM is a trusted public certificate authority (CA) that is trusted by most current systems. You might need to update your current trust root CA certificates to continue to connect to your clusters using SSL. For more information, see Transitioning to ACM Certificates for SSL Connections (p. 98) .	September 18, 2017
Service-linked roles	A service-linked role is a unique type of IAM role that is linked directly to Amazon Redshift. Service-linked roles are predefined by Amazon Redshift and include all the permissions that the service requires to call AWS services on behalf of your Amazon Redshift cluster. For more information, see Using Service-Linked Roles for Amazon Redshift (p. 201) .	September 18, 2017
IAM database user authentication	You can configure your system to permit users to create user credentials and log on to the database based on their IAM credentials. You can also configure your system to let users sign on using federated single sign-on (SSO) through a SAML 2.0-compliant identity provider. For more information, see Using IAM Authentication to Generate Database User Credentials (p. 217) .	August 11, 2017
New JDBC and ODBC drivers	The new JDBC and ODBC drivers support IAM database user authentication. Amazon Redshift JDBC drivers have been updated to version 1.2.7.1003. For more information, see Configure a JDBC Connection (p. 62) . Amazon Redshift ODBC drivers have been updated to version 1.3.6.1000. For more information, see Configure an ODBC Connection (p. 77) .	August 11, 2017
Table-level restore supports Enhanced VPC Routing	Table-level restore is now supported on clusters that use Enhanced VPC Routing (p. 115) . For more information, see Restoring a Table from a Snapshot (p. 147) .	July 19, 2017
Query Monitoring Rules	Using WLM query monitoring rules, you can define metrics-based performance boundaries for WLM queues and specify what action to take when a query goes beyond those boundaries—log, hop, or abort. You define query monitoring rules as part of your workload management (WLM) configuration. For more information, see Configuring Workload Management (p. 124) .	April 21, 2017
New JDBC and ODBC drivers	Amazon Redshift JDBC drivers have been updated to version 1.2.1.1001. Also, JDBC version 4.2 drivers are now supported. For more information, see Configure a JDBC Connection (p. 62) . Amazon Redshift ODBC drivers have been updated to version 1.3.1.1000. For more information, see Configure an ODBC Connection (p. 77) .	November 18, 2016

Change	Description	Release Date
Enhanced VPC Routing	When you use Amazon Redshift Enhanced VPC Routing, Amazon Redshift forces all COPY and UNLOAD traffic between your cluster and your data repositories through your Amazon VPC. For more information, see Amazon Redshift Enhanced VPC Routing (p. 115) .	September 15, 2016
New JDBC Drivers	Amazon Redshift JDBC drivers have been updated to version 1.1.17.1017. Also, JDBC version 4.2 drivers are now supported. For more information, see Configure a JDBC Connection (p. 62) .	July 5, 2016
New Connection Log fields	The Connection Log (p. 299) audit log has two new fields to track SSL connections. If you routinely load audit logs to an Amazon Redshift table, you will need to add the following new columns to the target table: <code>sslcompression</code> and <code>sslexpansion</code> .	May 5, 2016
New ODBC drivers	Amazon Redshift ODBC drivers have been updated to version 1.2.7.1007. For more information, see Configure an ODBC Connection (p. 77) .	March 30, 2016
IAM Roles for COPY and UNLOAD	You can now specify one or more AWS Identity and Access Management (IAM) roles that your cluster can use for authentication to access other AWS services. IAM roles provide a more secure alternative to provide authentication with COPY, UNLOAD, or CREATE LIBRARY commands. For more information, see Authorizing Amazon Redshift to Access Other AWS Services on Your Behalf (p. 235) and Authorizing COPY, UNLOAD, and CREATE EXTERNAL SCHEMA Operations Using IAM Roles (p. 240) .	March 29, 2016
Restore from Table	You can restore a table from a cluster snapshot to a new table in an active cluster. For more information, see Restoring a Table from a Snapshot (p. 147) .	March 10, 2016
New JDBC Drivers	Amazon Redshift JDBC drivers have been updated to version 1.1.10.1013. For more information, see Configure a JDBC Connection (p. 62) . You can now set the <code>SSLMode</code> property to specify whether the driver verifies host names when validating TLS/SSL certificates. For more information, see JDBC Driver Configuration Options (p. 64) .	February 18, 2016
Using IAM Condition in policies	You can further restrict access to resources by using the <code>Condition</code> element in IAM policies. For more information, see Using IAM Policy Conditions for Fine-Grained Access Control (p. 192) .	December 10, 2015
Modify Publicly Accessible	You can modify an existing cluster in a VPC to change whether it is publicly accessible. For more information, see Modifying a Cluster (p. 25) .	November 20, 2015

Change	Description	Release Date
New JDBC Drivers	Amazon Redshift JDBC drivers have been updated to version 1.1.10.1010. For more information, see Configure a JDBC Connection (p. 62) . Amazon Redshift ODBC drivers have been updated to version 1.2.6.1006. For more information, see Configure an ODBC Connection (p. 77) .	November 19, 2015
Documentation Fixes	Published various documentation fixes.	August 28, 2015
Documentation Update	Updated troubleshooting guidance about configuring network settings to ensure that hosts with different maximum transmission unit (MTU) sizes can determine the packet size for a connection. For more information, see Queries Appear to Hang and Sometimes Fail to Reach the Cluster (p. 112) .	August 25, 2015
Documentation Update	Revised entire section about parameter groups for better organization and clarity. For more information, see Amazon Redshift Parameter Groups (p. 122) .	August 17, 2015
New JDBC Drivers	Amazon Redshift JDBC drivers have been updated to version 1.1.7. For more information, see Configure a JDBC Connection (p. 62) .	August 14, 2015
WLM Dynamic Properties	The WLM configuration parameter now supports applying some properties dynamically. Other properties remain static changes and require that associated clusters be rebooted so that the configuration changes can be applied. For more information, see WLM Dynamic and Static Properties (p. 125) and Amazon Redshift Parameter Groups (p. 122) .	August 3, 2015
Copy KMS Encrypted Custers to Another Region	Added content about configuring snapshot copy grants to enable copying of AWS KMS-encrypted clusters to another region. For more information, see Copying AWS KMS-Encrypted Snapshots to Another Region (p. 164) .	July 28, 2015
Documentation Update	Updated the database encryption section to better explain how Amazon Redshift uses AWS KMS or HSMs for managing keys, and how the encryption process works with each of these options. For more information, see Amazon Redshift Database Encryption (p. 163) .	July 28, 2015
New JDBC Drivers	Amazon Redshift JDBC drivers have been updated to version 1.1.7. For more information, see Configure a JDBC Connection (p. 62) .	July 2, 2015
New Node Type	Amazon Redshift now offers a new node type, DS2. Updated documentation references to existing node types to use new names introduced in this release. Also revised the section to better explain the node type combinations and clarify default quota limits. For more information, see Clusters and Nodes in Amazon Redshift (p. 6) .	June 9, 2015

Change	Description	Release Date
Reserved Node Offerings	Added content about new reserved node offerings. Also revised the section to better explain and compare the available offerings, and provided examples to demonstrate how on-demand and reserved node pricing affect billing. For more information, see Overview (p. 176) .	June 9, 2015
New ODBC Drivers	Amazon Redshift ODBC driver have been updated. Added a section for previous versions of these drivers and a link to release notes for the drivers. For more information, see Configure an ODBC Connection (p. 77) .	June 5, 2015
Documentation Fixes	Published various documentation fixes.	April 30, 2015
Documentation Update	Updated the download links to new versions of the Amazon Redshift JDBC drivers, and added a section for previous versions of these drivers. Also added a link to release notes for the drivers. For more information, see Configure a JDBC Connection (p. 62) .	April 1, 2015
Documentation Update	<p>Added downloads for new versions of the Amazon Redshift JDBC drivers. Also updated the format of the Amazon Redshift JDBC URL. For more information, see Configure a JDBC Connection (p. 62).</p> <p>Added cluster security group ingress rules as a taggable resource. For more information, see Tagging Resources in Amazon Redshift (p. 326).</p> <p>Updated the instructions for adding a cluster security group ingress rule, and added instructions for tagging a cluster security group ingress rule. For more information, see Managing Cluster Security Groups Using the Console (p. 204).</p>	March 16, 2015
New Feature	This release of Amazon Redshift introduces new ODBC and JDBC drivers optimized for use with Amazon Redshift. For more information, see Connecting to an Amazon Redshift Cluster Using SQL Client Tools (p. 60) .	February 26, 2015
New Feature	This release of Amazon Redshift introduces cluster performance metrics that allow you to view and analyze query execution details. For more information, see Viewing Query Performance Data (p. 264) .	February 26, 2015
Documentation Update	Added a new example policy that demonstrates granting permission to common AWS service actions and resources on which Amazon Redshift relies. For more information, see Customer Managed Policy Examples (p. 195) .	January 16, 2015
Documentation Update	Updated guidance about setting the maximum transmission unit (MTU) to disable TCP/IP jumbo frames. For more information, see Supported Platforms to Launch Your Cluster (p. 10) and Queries Appear to Hang and Sometimes Fail to Reach the Cluster (p. 112) .	January 16, 2015

Change	Description	Release Date
Documentation Update	Revised the content about the <code>wlm_json_configuration</code> parameter, and provided example syntax to configure this parameter by using the AWS CLI on the Linux, Mac OS X, and Microsoft Windows operating systems. For more information, see Configuring Workload Management (p. 124) .	January 13, 2015
Documentation Update	Added missing event notifications and descriptions. For more information, see Amazon Redshift Event Categories and Event Messages (p. 284) .	January 8, 2015
Documentation Update	Updated guidance about IAM policies for Amazon Redshift actions and resources. Revised the section to improve organization and clarity. For more information, see Security (p. 187) .	November 21, 2014
New Feature	This release of Amazon Redshift introduces the ability to encrypt clusters using encryption keys from AWS Key Management Service (AWS KMS). AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. For more information about AWS KMS and encryption options for Amazon Redshift, see Amazon Redshift Database Encryption (p. 163) and Managing Clusters Using the Console (p. 15) .	November 12, 2014
New Feature	This release of Amazon Redshift introduces the ability to tag resources, such as clusters and snapshots. Tags enable you to provide user-defined metadata to categorize your billing reports based on cost allocation, and to help you better identify resources at a glance. For more information, see Tagging Resources in Amazon Redshift (p. 326) .	November 4, 2014
New Feature	Increased the maximum node limit to 128 nodes for <code>dw1.8xlarge</code> and <code>dw2.8xlarge</code> node sizes. For more information, see Clusters and Nodes in Amazon Redshift (p. 6) .	October 30, 2014
Documentation Update	Added links to the Microsoft Visual C++ 2010 Redistributable Packages that are required for Amazon Redshift to use PostgreSQL ODBC drivers. For more information, see Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems (p. 79) .	October 30, 2014
New Feature	Added the ability to terminate queries and loads from the Amazon Redshift console. For more information, see Viewing Query Performance Data (p. 264) and Viewing Cluster Metrics During Load Operations (p. 272) .	October 28, 2014
Documentation Fixes	Published various documentation fixes.	October 17, 2014
New Content	Added content about shutting down clusters and deleting clusters. For more information, see Shutting Down and Deleting Clusters (p. 14) and Deleting a Cluster (p. 27) .	August 14, 2014

Change	Description	Release Date
Documentation Update	Clarified the behavior of the Allow Version Upgrade setting for clusters. For more information, see Overview (p. 5) .	August 14, 2014
Documentation Update	Revised procedures, screenshots, and organization of topic about working with clusters in Amazon Redshift console. For more information, see Managing Clusters Using the Console (p. 15) .	July 11, 2014
New Content	Added a new tutorial about resizing Amazon Redshift clusters, including how to resize a cluster while minimizing the amount of time that the cluster is in read-only mode. For more information, see Resizing Clusters in Amazon Redshift (p. 311) .	June 27, 2014
New Feature	Added the ability to rename clusters. For more information, see Renaming Clusters (p. 13) and Modifying a Cluster (p. 25) .	June 2, 2014
Documentation Update	Updated the .NET code example to use the ODBC data provider when connecting to a cluster programmatically by using .NET. For more information, see Connecting to a Cluster by Using .NET (p. 108) .	May 15, 2014
New Feature	Added options to select a different parameter group and security group when you restore a cluster from a snapshot. For more information, see Restoring a Cluster from a Snapshot (p. 154) .	May 12, 2014
New Feature	Added new section to describe how to configure a default Amazon CloudWatch alarm to monitor the percentage of disk space used in an Amazon Redshift cluster. This alarm is a new option in the cluster creation process. For more information, see Default Disk Space Alarm (p. 12) .	April 28, 2014
Documentation Update	Clarified information about Elliptic curve Diffie—Hellman Exchange (ECDHE) support in Amazon Redshift. For more information, see Connect Using SSL (p. 95) .	April 22, 2014
New Feature	Added statement about Amazon Redshift support for the Elliptic curve Diffie—Hellman (ECDH) key agreement protocol. For more information, see Connect Using SSL (p. 95) .	April 18, 2014
Documentation Update	Revised and reorganized the topics in the Connecting to an Amazon Redshift Cluster Using SQL Client Tools (p. 60) section. Added more information about JDBC and ODBC connections, and a new troubleshooting section for connection issues.	April 15, 2014
Documentation Update	Added version in IAM policy examples throughout the guide.	April 3, 2014
Documentation Update	Added information about how pricing works when you resize a cluster. For more information, see Purchasing Amazon Redshift Reserved Nodes (p. 176) .	April 2, 2014

Change	Description	Release Date
New Feature	Added a section about a new parameter, <code>max_cursor_result_set_size</code> , which sets the maximum result set size, in megabytes, that can be stored per individual cursor. This parameter value also affects the number of concurrently active cursors for the cluster. For more information, see Amazon Redshift Parameter Groups (p. 122) .	March 28, 2014
New Feature	Added explanation about the Cluster Version field now including both cluster engine version and database revision number. For more information, see Amazon Redshift Clusters (p. 5) .	March 21, 2014
New Feature	Updated the resize procedure to show the new resize progress information on the cluster's Status tab. For more information, see Resizing a Cluster (p. 29) .	March 21, 2014
Documentation Update	Reorganized and updated What Is Amazon Redshift? (p. 1) and revised Amazon Redshift Management Overview (p. 1) . Published various documentation fixes.	February 21, 2014
New Feature	Added new node types and sizes for Amazon Redshift clusters, and rewrote the related cluster overview topic for better organization and clarity based on feedback. For more information, see Amazon Redshift Clusters (p. 5) .	January 23, 2014
New Feature	Added information about using elastic IP (EIP) addresses for publicly-accessible Amazon Redshift clusters in virtual private clouds. For more information about EIP in Amazon Redshift, see Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 36) and Creating a Cluster in a VPC (p. 37) .	December 20, 2013
New Feature	Added information about the AWS CloudTrail logs for Amazon Redshift. For more information about Amazon Redshift support for CloudTrail, see Logging Amazon Redshift API Calls with AWS CloudTrail (p. 304) .	December 13, 2013
New Feature	Added information about the new user activity log and the <code>enable_user_activity_logging</code> database parameter for the database audit logging feature in Amazon Redshift. For more information about database audit logging, see Database Audit Logging (p. 298) . For more information about database parameters, see Amazon Redshift Parameter Groups (p. 122) .	December 6, 2013
New Feature	Updated to describe configuring Amazon Redshift to automatically copy automated and manual snapshots to a secondary region. For more information about configuring cross-region snapshot copy, see Copying Snapshots to Another Region (p. 146) .	November 14, 2013

Change	Description	Release Date
New Feature	Added section to describe Amazon Redshift audit logging for connection and user activity, and storing these logs in Amazon S3. For more information about database audit logging, see Database Audit Logging (p. 298) .	November 11, 2013
New Feature	Added section to describe Amazon Redshift encryption with new features for managing encryption keys in a hardware security module (HSM) and rotating encryption keys. For more information about encryption, HSM, and key rotation, see Amazon Redshift Database Encryption (p. 163) , Encryption for Amazon Redshift Using Hardware Security Modules (p. 165) , and About Rotating Encryption Keys in Amazon Redshift (p. 166) .	November 11, 2013
New Feature	Updated to describe publishing notifications of Amazon Redshift events by using Amazon SNS. For information about Amazon Redshift event notifications, see Amazon Redshift Event Notifications (p. 283) .	November 11, 2013
New Feature	Updated to describe IAM resource level permissions. For information about Amazon Redshift IAM permissions, see Security (p. 187) .	August 9, 2013
New Feature	Updated to describe restore progress metrics. For more information, see Restoring a Cluster from a Snapshot (p. 147) .	August 9, 2013
New Feature	Updated to describe cluster snapshot sharing and create snapshot progress metrics. For more information, see Sharing Snapshots (p. 150) .	July 17, 2013
Documentation Fixes	Published various documentation fixes.	July 8, 2013
New Console Screens	Updated the <i>Amazon Redshift Cluster Management Guide</i> to match changes in the Amazon Redshift console.	April 22, 2013
New Guide	This is the first release of the <i>Amazon Redshift Management Guide</i> .	February 14, 2013