Amazon Redshift

Cluster Management Guide
# 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 of Amazon Redshift clusters ...................................................................................... 5
  Clusters and nodes ...................................................................................................................... 5
Use EC2-VPC when you create your cluster .................................................................................. 9
  EC2-VPC ............................................................................................................................... 9
  EC2-Classic ........................................................................................................................... 9
Launch a cluster ............................................................................................................................ 10
Overview of RA3 node types ......................................................................................................... 10
  Working with Amazon Redshift managed storage ................................................................ 11
  Managing RA3 node types ....................................................................................................... 11
  RA3 node type availability in AWS Regions .......................................................................... 11
Upgrading to RA3 node types ........................................................................................................ 12
Upgrading from DC1 node types to DC2 node types .................................................................. 13
Upgrading a DS2 cluster on EC2-Classic to EC2-VPC ............................................................... 14
Region and Availability Zone considerations .................................................................................. 14
Cluster maintenance ....................................................................................................................... 15
  Maintenance windows .............................................................................................................. 15
  Deferring maintenance ............................................................................................................ 16
  Choosing cluster maintenance tracks .................................................................................... 16
  Managing cluster versions ..................................................................................................... 17
  Rolling back the cluster version ............................................................................................. 17
  Determining the cluster maintenance version ...................................................................... 18
Default disk space alarm ................................................................................................................ 19
Cluster status .................................................................................................................................. 20
Managing clusters overview ......................................................................................................... 21
  Resizing clusters ..................................................................................................................... 21
  Pausing and resuming clusters ................................................................................................. 28
  Renaming clusters ................................................................................................................... 30
  Shutting down and deleting clusters ....................................................................................... 30
Working with AQUA ...................................................................................................................... 31
  Managing using the AWS CLI ................................................................................................. 32
  Managing using Amazon Redshift API operations ................................................................. 32
When is AQUA used? ..................................................................................................................... 32
Managing usage limits ................................................................................................................... 34
Managing cluster relocation ......................................................................................................... 35
  Enabling cluster relocation .................................................................................................... 36
  Limitations ............................................................................................................................ 36
  Managing relocation using the console .................................................................................... 37
  Managing relocation using the Amazon Redshift CLI ............................................................ 38
Working with Redshift-managed VPC endpoints ......................................................................... 38
  Considerations ....................................................................................................................... 39
  Managing using the Redshift console ....................................................................................... 40
  Managing using the AWS CLI ................................................................................................. 41
  Managing using Amazon Redshift API operations .................................................................... 41
Managing clusters using the console ............................................................................................ 41
  Creating a cluster .................................................................................................................... 44
  Modifying a cluster .................................................................................................................. 54
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 Guide** – 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. 393).

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. 258).

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. 227).

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 in Amazon Redshift (p. 269).

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-VPC platform for your Amazon Redshift cluster, you must use VPC security groups. We recommend that you launch your cluster in an EC2-VPC platform.

  You cannot move a cluster to a VPC after it has been launched with EC2-Classic. However, you can restore an EC2-Classic snapshot to an EC2-VPC cluster using the Amazon Redshift console. For more information, see Restoring a cluster from a snapshot (p. 245).
If you are using the EC2-Classic platform for your Amazon Redshift cluster, you must use Amazon Redshift 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. 370).

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 cluster that is not encrypted 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. 271).

**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 Configuring security options for connections (p. 144).

**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. 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. 353).

**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. 447).
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. 400).

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. 201).

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 of Amazon Redshift clusters (p. 5)
- Use EC2-VPC when you create your cluster (p. 9)
- Overview of RA3 node types (p. 10)
- Upgrading to RA3 node types (p. 12)
- Upgrading from DC1 node types to DC2 node types (p. 13)
- Upgrading a DS2 cluster on EC2-Classic to EC2-VPC (p. 14)
- Region and Availability Zone considerations (p. 14)
- Cluster maintenance (p. 15)
- Default disk space alarm (p. 19)
- Cluster status (p. 20)
- Overview of managing clusters in Amazon Redshift (p. 21)
- Working with AQUA (Advanced Query Accelerator) (p. 31)
- Managing usage limits in Amazon Redshift (p. 34)
- Managing cluster relocation in Amazon Redshift (p. 35)
- Working with Redshift-managed VPC endpoints in Amazon Redshift (p. 38)
- Managing clusters using the console (p. 41)
- Managing clusters using the AWS CLI and Amazon Redshift API (p. 66)
- Managing clusters using the AWS SDK for Java (p. 67)
- Managing clusters in a VPC (p. 69)
- Cluster version history (p. 80)

Overview of Amazon Redshift clusters

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.

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.

Note
When you create a cluster on the Amazon Redshift console (https://console.aws.amazon.com/redshift/), you can get a recommendation of your cluster configuration based on the size of your data and query characteristics. To use this sizing calculator, look for Help me choose on the console in AWS Regions that support RA3 node types. For more information, see Creating a cluster (p. 44).

When you launch a cluster, one option that you specify is the node type. The node type determines the CPU, RAM, storage capacity, and storage drive type for each node.

Amazon Redshift offers different node types to accommodate your workloads, and we recommend choosing RA3 or DC2 depending on the required performance, data size, and expected data growth.

RA3 nodes with managed storage enable you to optimize your data warehouse by scaling and paying for compute and managed storage independently. With RA3, you choose the number of nodes based on your performance requirements and only pay for the managed storage that you use. Size your RA3 cluster based on the amount of data you process daily. You launch clusters that use the RA3 node types in a virtual private cloud (VPC). You can’t launch RA3 clusters in EC2-Classic. For more information, see Creating a cluster in a VPC (p. 71).

Amazon Redshift managed storage uses large, high-performance SSDs in each RA3 node for fast local storage and Amazon S3 for longer-term durable storage. If the data in a node grows beyond the size of the large local SSDs, Amazon Redshift managed storage automatically offloads that data to Amazon S3. You pay the same low rate for Amazon Redshift managed storage regardless of whether the data sits in high-performance SSDs or Amazon S3. For workloads that require ever-growing storage, managed storage lets you automatically scale your data warehouse storage capacity without adding and paying for additional nodes.

DC2 nodes enable you to have compute-intensive data warehouses with local SSD storage included. You choose the number of nodes you need based on data size and performance requirements. DC2 nodes store your data locally for high performance, and as the data size grows, you can add more compute nodes to increase the storage capacity of the cluster. For datasets under 1 TB (compressed), we recommend DC2 node types for the best performance at the lowest price. If you expect your data to grow, we recommend using RA3 nodes so you can size compute and storage independently to achieve improved price and performance. You launch clusters that use the DC2 node types in a virtual private cloud (VPC). You can’t launch DC2 clusters in EC2-Classic. For more information, see Creating a cluster in a VPC (p. 71).

DS2 nodes enable you to create large data warehouses using hard disk drives (HDDs), and we recommend using RA3 nodes instead. If you are using DS2 nodes, see Upgrading to RA3 node types (p. 12) for upgrade guidelines. If you are using eight or more nodes of ds2.xlarge, or any number of ds2.8xlarge nodes, you can now upgrade to RA3 to get 2x more storage and improved performance for the same on-demand cost.

Node types are available in different sizes. Node size and the number of nodes determine the total storage for a cluster. For more information, see Node type details (p. 7).

Some node types allow one node (single-node) or two or more nodes (multi-node). The minimum number of nodes for clusters of some node types is two nodes. On a single-node cluster, the node is shared for leader and compute functionality. Single-node clusters are not recommended for running production workloads. On a multi-node cluster, the leader node is separate from the compute nodes. The leader node is the same node type as the compute nodes. You only pay for compute nodes.
Amazon Redshift applies quotas to resources for each AWS account in each AWS Region. A *quota* restricts the number of resources that your account can create for a given resource type, such as nodes or snapshots, within an AWS Region. For more information about the default quotas that apply to Amazon Redshift resources, see 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 AWS Region, node type, number of nodes, and whether the nodes are reserved in advance. For more information about the cost of nodes, see the Amazon Redshift pricing page.

### Node type details

The following tables summarize the node specifications for each node type and size. The headings in the tables have these meanings:

- **vCPU** is the number of virtual CPUs for each node.
- **RAM** is the amount of memory in gibibytes (GiB) for each node.
- **Default slices per node** is the number of slices into which a compute node is partitioned when a cluster is created or resized with classic resize.

The number of slices per node might change if the cluster is resized using elastic resize. However the total number of slices on all the compute nodes in the cluster remains the same after elastic resize.

When you create a cluster with the restore from snapshot operation, the number of slices of the resulting cluster might change from the original cluster if you change the node type.

- **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 AWS Region. To request an increase, submit an Amazon Redshift Limit Increase Form.

- **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.

#### RA3 node types

<table>
<thead>
<tr>
<th>Node size</th>
<th>vCPU</th>
<th>RAM (GiB)</th>
<th>Default slices per node</th>
<th>Managed storage quota per node</th>
<th>Node range with create cluster</th>
<th>Total managed storage capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ra3.xlplus</td>
<td>4</td>
<td>32</td>
<td>2</td>
<td>32 TB(^{1})</td>
<td>2–16(^{2})</td>
<td>1024 TB(^{2,4})</td>
</tr>
<tr>
<td>ra3.4xlarge</td>
<td>12</td>
<td>96</td>
<td>4</td>
<td>128 TB(^{1})</td>
<td>2–32(^{3})</td>
<td>8192 TB(^{3,4})</td>
</tr>
<tr>
<td>ra3.16xlarge</td>
<td>48</td>
<td>384</td>
<td>16</td>
<td>128 TB(^{1})</td>
<td>2–128</td>
<td>16,384 TB(^{4})</td>
</tr>
</tbody>
</table>

\(^{1}\) The storage quota for Amazon Redshift managed storage.

\(^{2}\) You can create a cluster with the ra3.xlplus node type that has up to 16 nodes. You can resize it with elastic resize to a maximum of 32 nodes.

\(^{3}\) You can create cluster with the ra3.4xlarge or ra3.16xlarge node type with up to 16 nodes. You can resize it with elastic resize to a maximum of 64 nodes.
Total managed storage quota is the maximum number of nodes times the managed storage quota per node.

### Dense storage node types

<table>
<thead>
<tr>
<th>Node size</th>
<th>vCPU</th>
<th>RAM (GiB)</th>
<th>Default slices per node</th>
<th>Storage per node</th>
<th>Node range</th>
<th>Total capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds2.xlarge</td>
<td>4</td>
<td>31</td>
<td>2</td>
<td>2 TB HDD</td>
<td>1–32</td>
<td>64 TB</td>
</tr>
<tr>
<td>ds2.8xlarge</td>
<td>36</td>
<td>244</td>
<td>16</td>
<td>16 TB HDD</td>
<td>2–128</td>
<td>2 PB</td>
</tr>
</tbody>
</table>

### Dense compute node types

<table>
<thead>
<tr>
<th>Node size</th>
<th>vCPU</th>
<th>RAM (GiB)</th>
<th>Default slices per node</th>
<th>Storage per node</th>
<th>Node range</th>
<th>Total capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc2.large</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>160 GB NVMe-SSD</td>
<td>1–32</td>
<td>5.12 TB</td>
</tr>
<tr>
<td>dc2.8xlarge</td>
<td>32</td>
<td>244</td>
<td>16</td>
<td>2.56 TB NVMe-SSD</td>
<td>2–128</td>
<td>326 TB</td>
</tr>
<tr>
<td>dc1.large¹</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>160 GB SSD</td>
<td>1–32</td>
<td>5.12 TB</td>
</tr>
<tr>
<td>dc1.8xlarge¹</td>
<td>32</td>
<td>244</td>
<td>32</td>
<td>2.56 TB SSD</td>
<td>2–128</td>
<td>326 TB</td>
</tr>
</tbody>
</table>

¹ We recommend DC2 node types over DC1 node types. For more information on how to upgrade, see [Upgrading from DC1 node types to DC2 node types (p. 13)].

### Previous node type names

In previous releases of Amazon Redshift, certain node types had different names. You can use the previous names in the Amazon Redshift API and 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.

<table>
<thead>
<tr>
<th>Current name</th>
<th>Previous names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds2.xlarge</td>
<td>ds1.xlarge, dw.hs1.xlarge, dw1.xlarge</td>
</tr>
<tr>
<td>ds2.8xlarge</td>
<td>ds1.8xlarge, dw.hs1.8xlarge, dw1.8xlarge</td>
</tr>
<tr>
<td>dc1.large</td>
<td>dw2.large</td>
</tr>
<tr>
<td>dc1.8xlarge</td>
<td>dw2.8xlarge</td>
</tr>
</tbody>
</table>

### Determining the number of nodes

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. When you run a cluster with
at least two compute nodes, data on each node is mirrored on disks of another node to reduce the risk of incurring data loss.

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 price and performance for your cluster. 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. 400).

Reserved nodes are appropriate for steady-state production workloads, and offer significant discounts over on-demand nodes. You can purchase reserved nodes after running experiments and proof-of-concepts to validate your production configuration. For more information, see Purchasing Amazon Redshift reserved nodes (p. 258).

When you pause a cluster, you suspend on-demand billing during the time the cluster is paused. During this paused time, you only pay for backup storage. This frees you from planning and purchasing data warehouse capacity ahead of your needs, and enables you to cost-effectively manage environments for development or test purposes.

For information about pricing of on-demand and reserved nodes, see Amazon Redshift pricing.

Use EC2-VPC when you create your cluster

Amazon Redshift clusters run in Amazon EC2 instances that are configured for the Amazon Redshift node type and size that you select. Create your cluster using EC2-VPC. If you are still using EC2-Classic, we recommend you use EC2-VPC to get improved performance and security. For more information about these networking platforms, see Supported Platforms in the Amazon EC2 User Guide for Linux Instances. Your AWS account settings determine whether EC2-VPC or EC2-Classic are available to you.

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. 165).

EC2-VPC

When using EC2-VPC, 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, you control access to your cluster by associating one or more VPC security groups with the cluster. For more information, see 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. 74).

For more information about Amazon Virtual Private Cloud (Amazon VPC), see the Amazon VPC product detail page.

EC2-Classic

In EC2-Classic, your cluster runs in a single, flat network that you share with other AWS customers. If you provision your cluster in the EC2-Classic, you control access to your cluster by associating one or more
Launch a cluster

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

1. Decide which AWS Region you want to deploy a cluster. For a list of AWS Regions in which Amazon Redshift is available, see Amazon Redshift 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, see Supported Platforms in the Amazon EC2 User Guide for Linux Instances.
3. If your account supports both of the platforms, we recommend EC2-VPC. If your account supports only EC2-VPC, you must deploy your cluster in VPC.
4. Launch your Amazon Redshift cluster. You can create a cluster using the Amazon Redshift console, or by using the Amazon Redshift API, AWS CLI, or SDK libraries. For more information about these options and links to the related documentation, see What is Amazon Redshift? (p. 1).

Overview of RA3 node types

We recommend that you upgrade existing workloads running on DS2 node type clusters to RA3 node types to take advantage of improved performance and to get more storage capacity. RA3 nodes provide the following advantages:

- They are flexible to grow your compute capacity without increasing your storage costs. And they scale your storage without over-provisioning compute capacity.
- They use high performance SSDs for your hot data and Amazon S3 for cold data. Thus they provide ease of use, cost-effective storage, and high query performance.
- They use high bandwidth networking built on the AWS Nitro System to further reduce the time taken for data to be offloaded to and retrieved from Amazon S3.

Consider choosing RA3 node types in these cases:

- You need the flexibility to scale and pay for compute separate from storage.
- You query a fraction of your total data.
- Your data volume is growing rapidly or is expected to grow rapidly.
- You want the flexibility to size the cluster based only on your performance needs.

To use RA3 node types, your AWS Region must support RA3. For more information, see RA3 node type availability in AWS Regions (p. 11).

Important
You can use ra3.xlplus node types only with cluster version 1.0.21262 or later. You can view the version of an existing cluster with the Amazon Redshift console. For more information, see Determining the cluster maintenance version (p. 18).
Make sure that you use the new Amazon Redshift console when working with RA3 node types. The original console doesn't support all RA3 operations.
In addition, to use RA3 node types with Amazon Redshift operations that use the maintenance track, the maintenance track value must be set to a cluster version that supports RA3. For more information about maintenance tracks, see Choosing cluster maintenance tracks (p. 16).
Working with Amazon Redshift managed storage

With Amazon Redshift managed storage, you can store and process all your data in Amazon Redshift while getting more flexibility to scale compute and storage capacity separately. You continue to ingest data with the COPY or INSERT command. To optimize performance and manage automatic data placement across tiers of storage, Amazon Redshift takes advantage of optimizations such as data block temperature, data block age, and workload patterns. When needed, Amazon Redshift scales storage automatically to Amazon S3 without requiring any manual action.

For information about storage costs, see Amazon Redshift pricing.

Managing RA3 node types

To take advantage of separating compute from storage, you can create or upgrade your cluster with the RA3 node type. To use the RA3 node types, create your clusters in a virtual private cloud (EC2-VPC).

To change the number of nodes of Amazon Redshift cluster with an RA3 node type, do one of the following:

- Add or remove nodes with the elastic resize operation. In some situations, removing nodes from a RA3 cluster isn't allowed with elastic resize. For example, when a 2:1 node count upgrade puts the number of slices per node at 32. For more information, see Resizing clusters (p. 21). If elastic resize isn't available, use classic resize.
- Add or remove nodes with the classic resize operation. Choose this option when you are resizing to a configuration that isn't available through elastic resize. Elastic resize is quicker than classic resize. For more information, see Resizing clusters (p. 21).

RA3 node type availability in AWS Regions

The RA3 node types are available only in the following AWS Regions:

- US East (N. Virginia) Region (us-east-1)
- US East (Ohio) Region (us-east-2)
- US West (N. California) Region (us-west-1)
- US West (Oregon) Region (us-west-2)
- Asia Pacific (Hong Kong) Region (ap-east-1)
- Asia Pacific (Mumbai) Region (ap-south-1)
- Asia Pacific (Seoul) Region (ap-northeast-2)
- Asia Pacific (Singapore) Region (ap-southeast-1)
- Asia Pacific (Sydney) Region (ap-southeast-2)
- Asia Pacific (Tokyo) Region (ap-northeast-1)
- Canada (Central) Region (ca-central-1)
- Europe (Frankfurt) Region (eu-central-1)
- China (Beijing) Region (cn-north-1)
- China (Ningxia) Region (cn-northwest-1)
- Europe (Ireland) Region (eu-west-1)
- Europe (London) Region (eu-west-2)
- Europe (Paris) Region (eu-west-3)
- Europe (Stockholm) Region (eu-north-1)
- South America (São Paulo) Region (sa-east-1)
• AWS GovCloud (US-East) (us-gov-east-1) – Only ra3.4xlarge and ra3.16xlarge node types are available.
• AWS GovCloud (US-West) (us-gov-west-1) – Only ra3.4xlarge and ra3.16xlarge node types are available.

Upgrading to RA3 node types

To upgrade your existing node type to RA3, you have the following options to change the node type:

• Restore from a snapshot – Amazon Redshift uses the most recent snapshot of your DS2 or DC2 cluster and restores it to create a new RA3 cluster. As soon as the cluster creation is complete (usually within minutes), RA3 nodes are ready to run your full production workload. As compute is separate from storage, hot data is brought in to the local cache at fast speeds thanks to a large networking bandwidth. If you restore from the latest DS2 or DC2 snapshot, RA3 preserves hot block information of the DS2 or DC2 workload and populates its local cache with the hottest blocks. For more information, see Restoring a cluster from a snapshot (p. 231).

To keep the same endpoint for your applications and users, you can rename the new RA3 cluster with the same name as the original DS2 or DC2 cluster. To rename the cluster, modify the cluster in the Amazon Redshift console or ModifyCluster API operation. For more information, see Renaming clusters (p. 30) or ModifyCluster API operation in the Amazon Redshift API Reference.

• Elastic resize – resize the cluster using elastic resize. When you use elastic resize to change node type, Amazon Redshift automatically creates a snapshot, creates a new cluster, deletes the old cluster, and renames the new cluster. The elastic resize operation can be run on-demand or can be scheduled to run at a future time. You can quickly upgrade your existing DS2 or DC2 node type clusters to RA3 with elastic resize. For more information, see Elastic resize (p. 22).

The following table shows recommendations when upgrading to RA3 node types.

<table>
<thead>
<tr>
<th>Existing node type</th>
<th>Range of existing number of nodes</th>
<th>Recommended new node type</th>
<th>Upgrade action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds2.xlarge</td>
<td>1–7</td>
<td>ra3.xlplus</td>
<td>Create 2 nodes of ra3.xlplus for every 3 nodes of ds2.xlarge.</td>
</tr>
<tr>
<td>ds2.xlarge</td>
<td>8–128</td>
<td>ra3.4xlarge</td>
<td>Create 1 node of ra3.4xlarge for every 4 nodes of ds2.xlarge.</td>
</tr>
<tr>
<td>ds2.8xlarge</td>
<td>2–15</td>
<td>ra3.4xlarge</td>
<td>Create 2 nodes of ra3.4xlarge for every 1 node of ds2.8xlarge.</td>
</tr>
<tr>
<td>ds2.8xlarge</td>
<td>16–128</td>
<td>ra3.16xlarge</td>
<td>Create 1 node of ra3.16xlarge for every 2 nodes of ds2.8xlarge.</td>
</tr>
<tr>
<td>dc2.8xlarge</td>
<td>2–15</td>
<td>ra3.4xlarge</td>
<td>Create 2 nodes of ra3.4xlarge for every 1 node of ds2.8xlarge.</td>
</tr>
</tbody>
</table>
Upgrading from DC1 node types to DC2 node types

To take advantage of performance improvements, you can upgrade your DC1 clusters to DC2 node types.

Clusters that use the DC2 node type must be launched in a virtual private cloud (EC2-VPC).

If your DC1 cluster is not in a VPC:

1. Create a snapshot of your DC1 cluster. For more information, see Amazon Redshift snapshots (p. 227).
2. Create a VPC, or choose an existing VPC in your account. For more information, see Managing clusters in a VPC (p. 69).
3. Restore your snapshot to a new DC2 cluster in the VPC. For more information, see Restoring a cluster from a snapshot (p. 231).

If your DC1 cluster is already in a VPC, choose one of the following methods:

- Resize your DC1 cluster and change the node type to DC2 as part of the operation. Your cluster is not available for a period of time during the resize operation. For more information, see Resizing clusters in Amazon Redshift (p. 21).
- Create a snapshot of your DC1 cluster, then restore your snapshot to a DC2 cluster in the VPC. For more information, see Restoring a cluster from a snapshot (p. 231).

1 Extra nodes might be needed depending on workload requirements. Add or remove nodes based on the compute requirements of your required query performance.

The minimum number of nodes for RA3 clusters is 2 nodes. Take this into consideration when creating an RA3 cluster.

If you have already purchased DS2 reserved nodes, contact AWS for help with converting DS2 reserved nodes to RA3 reserved nodes. To contact AWS for more information, see Amazon Redshift RA3 instances with managed storage.
Consider the following when upgrading from DC1 to DC2 node types.

- DC1 clusters that are 100% full might not upgrade to an equivalent number of DC2 nodes. If more disk space is needed, you can:
  - Resize to a configuration with more available disk space.
  - Clean up unneeded data by truncating tables or deleting rows.
- DC2 clusters don't support EC2-Classic networking. If your DC1 cluster is not running in a VPC, create one for your DC2 migration. For more information, see Managing clusters in a VPC (p. 69).
- If you resize the cluster, it might be put into read-only mode for the duration of the operation. For more information, see Resizing clusters in Amazon Redshift (p. 21).
- If you have purchased DC1 reserved nodes, you can upgrade your DC1 reserved nodes to DC2 nodes for the remainder of your term. For more information about how to change your reservation with the AWS CLI, see Upgrading reserved nodes with the AWS CLI (p. 264).
- If you use restore to upgrade from dc1.large to dc2.large, and change the number of nodes, then the snapshot must have been created at cluster version 1.0.10013 or later.
- If you use restore to upgrade from dc1.8xlarge to dc2.8xlarge, then the snapshot must have been created at cluster version 1.0.10013 or later.
- If you use elastic resize to upgrade from DC1 to DC2, and change the number of nodes, then the cluster must be at cluster version 1.0.10013 or later.
- If a snapshot of dc1.8xlarge cluster to upgrade is from a cluster earlier than version 1.0.10013, then first restore the snapshot from the dc1.8xlarge cluster into a new dc1.8xlarge cluster with the same number of nodes. Then use one of the following methods to upgrade the new dc1.8xlarge:
  - Use a snapshot from the new restored cluster to upgrade to dc2.8xlarge.
  - Use elastic resize to upgrade the new restored cluster to dc2.8xlarge.

**Upgrading a DS2 cluster on EC2-Classic to EC2-VPC**

Amazon Redshift clusters run in Amazon EC2 instances that are configured for the Amazon Redshift node type and size that you choose. We recommend that you upgrade your cluster on EC2-Classic to launch in a VPC using EC2-VPC for improved performance and security.

**To upgrade your DS2 cluster on EC2-Classic to EC2-VPC**

1. Create a snapshot of your DS2 cluster. For more information, see Amazon Redshift snapshots (p. 227).
2. Create a VPC, or choose an existing VPC in your account. For more information, see Managing clusters in a VPC (p. 69).
3. Restore your snapshot to a new DS2 cluster in the VPC. For more information, see Restoring a cluster from a snapshot (p. 231).

**Region 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 choose. All the cluster nodes are provisioned in the same Availability Zone.

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

For a list of supported AWS Regions where you can provision an Amazon Redshift cluster, see Amazon Redshift endpoints in the Amazon Web Services General Reference.

Cluster maintenance

Amazon Redshift periodically performs maintenance to apply upgrades to your cluster. During these updates, your Amazon Redshift cluster isn’t available for normal operations. You have several ways to control how we maintain your cluster. For example, you can control when we deploy updates to your clusters. You can also choose whether your cluster runs the most recently released version, or the version released previously to the most recently released version. Finally, you have the option to defer non-mandatory maintenance updates for a period of time.

Topics
- Maintenance windows (p. 15)
- Deferring maintenance (p. 16)
- Choosing cluster maintenance tracks (p. 16)
- Managing cluster versions (p. 17)
- Rolling back the cluster version (p. 17)
- Determining the cluster maintenance version (p. 18)

Maintenance windows

Amazon Redshift assigns a 30-minute maintenance window at random from an 8-hour block of time per AWS Region, occurring on a random day of the week (Monday through Sunday, inclusive).

Default maintenance windows

The following list shows the time blocks for each AWS 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
- Africa (Cape Town) Region: 20:00–04:00 UTC
- Asia Pacific (Hong Kong) Region: 13:00–21:00 UTC
- Asia Pacific (Mumbai) Region: 16:30–00:30 UTC
- Asia Pacific (Osaka) Region: 13:00–21:00 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
- Canada (Central) Region: 03:00–11:00 UTC
- China (Beijing) Region: 13:00–21:00 UTC
• China (Ningxia) Region: 13:00–21:00 UTC
• Europe (Frankfurt) Region: 06:00–14:00 UTC
• Europe (Ireland) Region: 22:00–06:00 UTC
• Europe (London) Region: 22:00–06:00 UTC
• Europe (Milan) Region: 21:00–05:00 UTC
• Europe (Paris) Region: 23:00–07:00 UTC
• Europe (Stockholm) Region: 23:00–07:00 UTC
• Middle East (Bahrain) Region: 13:00–21:00 UTC
• South America (São Paulo) Region: 19:00–03:00 UTC

If a maintenance event is scheduled for a given week, it starts 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. 41).

Deferring maintenance

If you need to reschedule your cluster's maintenance window, you have the option to defer maintenance by up to 45 days. For example, if your cluster's maintenance window is set to Wednesday 8:30 – 9:00 UTC and you need to have access to your cluster at that time, you can defer the maintenance to a later time period. We will not perform any maintenance on your cluster when you have specified a deferment, unless we need to update hardware.

If we need to update hardware or make other mandatory updates during your period of deferment, we notify you and make the required changes. Your cluster isn't available during these updates.

If you defer your cluster's maintenance, the maintenance window following your period of deferment is mandatory. It can't be deferred.

Note
You can't defer maintenance after it has started.

For more information, see Modifying a cluster (p. 54).

Choosing cluster maintenance tracks

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

The maintenance track controls which cluster version is applied during a maintenance window. When Amazon Redshift releases a new cluster version, that version is assigned to the current track, and the previous version is assigned to the trailing track. To set the maintenance track for the cluster, specify one of the following values:

• Current – Use the most current approved cluster version.
• Trailing – Use the cluster version before the current version.
• Preview – Use the cluster version that contains new features available for preview.
For example, suppose that your cluster is currently running version 1.0.2762 and the Amazon Redshift current version is 1.0.3072. If 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 you set the maintenance track value to **Trailing**, your cluster isn't updated until there is a new release after 1.0.3072.

**Preview tracks**

A **Preview** track might not always be available to choose. When you choose a **Preview** track, a track name must also be selected. Preview tracks and its related resources are temporary, have functional limitations, and might not contain all current Amazon Redshift features available in other tracks. When working with preview tracks:

- Use the new Amazon Redshift console when working with preview tracks. For example, when you create a cluster to use with preview features.
- You can't switch a cluster from one preview track to another.
- You can't switch a cluster to a preview track from a current or trailing track.
- You can't restore from a snapshot created from a different preview track.
- You can only use the preview track when creating a new cluster, or when restoring from a snapshot.
- You can't restore from a snapshot created from a different preview track, or with a cluster maintenance version later than the preview track cluster version. For example, when you restore a cluster to a preview track, you can only use a snapshot created from an earlier cluster maintenance version than that of the preview track.

**Switching between maintenance tracks**

Changing tracks for a cluster is generally a one-time decision. You should exercise caution in changing tracks. If you change the maintenance track from **Trailing** to **Current**, we will update the cluster to the **Current** track release version during the next maintenance window. However, if you change the cluster's maintenance track to **Trailing** we won't update your cluster until there is a new release after the **Current** track release version.

**Maintenance tracks and restore**

A snapshot inherits the source cluster’s maintenance track. If you change the source cluster’s maintenance track after you take a snapshot, the snapshot and the source cluster are on different tracks. When you restore from the snapshot, the new cluster will be on the maintenance track that was inherited from the source cluster. You can change the maintenance track after the restore operation completes. Resizing a cluster doesn't affect the cluster's maintenance track.

For more information see, Setting the maintenance track for a cluster (p. 55).

**Managing cluster versions**

A maintenance track is a series of releases. You can decide if your cluster is on the **Current** track or the **Trailing** track. If you put your cluster on the **Current** track, it will always be upgraded to the most recent cluster release version during its maintenance window. If you put your cluster on the **Trailing** track, it will always run the cluster release version that was released immediately before the most recently released version.

The **Release status** column in the Amazon Redshift console list of clusters indicates whether one of your clusters is available for upgrade.

**Rolling back the cluster version**

If your cluster is up to date with the latest cluster version, you can choose to roll it back to the previous version.
For detailed information about features and improvements included with each cluster version, see Cluster version history (p. 80).

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

New Console

To 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. On the navigation menu, choose CLUSTERS.
3. Choose the cluster to roll back.
4. For Actions, choose Roll back cluster version. The Roll back cluster version page appears.
5. If there is a version available for roll back, follow the instructions on the page.
6. Choose Roll back now.

Original Console

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 and choose the Status tab.

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

   ![Cluster Status]

   Cluster: [Cluster Name]
   Configuration
   Status

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

4. Choose Rollback to release (release number).

Determining the cluster maintenance version

You can determine the Amazon Redshift engine and database version with the Amazon Redshift console.

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

To find the version 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. On the navigation menu, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose the Maintenance tab for more details.
4. In the Maintenance section, find Current cluster version.

Note
Although the console displays this information in one field, it's two parameters in the Amazon Redshift API, ClusterVersion and ClusterRevisionNumber. For more information, see Cluster in the Amazon Redshift API Reference.

Original console

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's two parameters in the Amazon Redshift API, ClusterVersion and ClusterRevisionNumber. For more information, see Cluster in the Amazon Redshift API Reference.

To specify whether to automatically upgrade the Amazon Redshift engine in your cluster if a new version of the engine becomes available, use the setting Allow version upgrade. 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. 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 <clusternname>-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. 44) and Creating or editing a disk space alarm (p. 64).

**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.

## Cluster status

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

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>The cluster is running and available.</td>
</tr>
<tr>
<td>available, prep-for-resize</td>
<td>The cluster is being prepared for elastic resize. The cluster is running and available for read and write queries, but cluster operations, such as creating a snapshot, are not available.</td>
</tr>
<tr>
<td>available, resize-cleanup</td>
<td>An elastic resize operation is completing data transfer to the new cluster nodes. The cluster is running and available for read and write queries, but cluster operations, such as creating a snapshot, are not available.</td>
</tr>
<tr>
<td>cancelling-resize</td>
<td>The resize operation is being cancelled.</td>
</tr>
<tr>
<td>creating</td>
<td>Amazon Redshift is creating the cluster. For more information, see Creating a cluster (p. 44).</td>
</tr>
<tr>
<td>deleting</td>
<td>Amazon Redshift is deleting the cluster. For more information, see Deleting a cluster (p. 56).</td>
</tr>
<tr>
<td>final-snapshot</td>
<td>Amazon Redshift is taking a final snapshot of the cluster before deleting it. For more information, see Deleting a cluster (p. 56).</td>
</tr>
<tr>
<td>hardware-failure</td>
<td>The cluster suffered a hardware failure.</td>
</tr>
</tbody>
</table>
Overview of managing clusters in Amazon Redshift

After your cluster is created, there are several operations you can perform on it. The operations include resizing, pausing, resuming, renaming, and deleting.

Resizing clusters in Amazon Redshift

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 use elastic resize to scale your cluster by changing the node type and number of nodes. We recommend using elastic resize, which typically completes in minutes. If your new node configuration isn't available through elastic resize, you can use classic resize. Classic resize takes more time to complete, because it involves provisioning a new cluster and copying data blocks to it. In contrast, elastic resize redistributes data slices, which requires fewer resources.

To resize your cluster, use one of the following approaches:

- **Elastic resize** – Use it to change the node type, number of nodes, or both. Elastic resize works quickly by changing or adding nodes to your existing cluster. If you change only the number of nodes, queries are temporarily paused and connections are held open, if possible. Typically, elastic resize takes 10–15 minutes. During the resize operation, the cluster is read-only.

  We recommend using elastic resize whenever possible, because it completes much more quickly than classic resize.

- **Classic resize** – Use it to change the node type, number of nodes, or both. Classic resize provisions a new cluster and copies the data from the source cluster to the new cluster. Choose this option only when you are resizing to a configuration that isn't available through elastic resize, because it takes considerably more time to complete. An example of when to use it is when resizing to or from a single-node cluster. During the resize operation, the cluster is read-only. Classic resize can take several hours to several days, or longer, depending on the amount of data to transfer and the difference in cluster size and computing resources.

  If your configuration warrants performing a classic resize, you can minimize production impact by making a copy of your existing cluster and subsequently resizing the copy. **Snapshot, restore, and resize (p. 26)** describes the steps.

You can resize (both elastic resize and classic resize) your cluster on a schedule. When you use the new Amazon Redshift console, you can set up a schedule to resize your cluster. For more information, see **Resizing a cluster (p. 59)**. You can also use the AWS CLI or Amazon Redshift API operations to schedule a resize. For more information, see **create-scheduled-action** in the **AWS CLI Command Reference** or **CreateScheduledAction** in the **Amazon Redshift API Reference**.

**Topics**

- Elastic resize (p. 22)
- Classic resize (p. 25)
- Snapshot, restore, and resize (p. 26)
- Details of resizing a cluster (p. 27)

**Elastic resize**

Elastic resize is the fastest method to resize a cluster. You can use elastic resize to add or remove nodes and change node types for an existing cluster.

When a cluster is resized using elastic resize with the same node type, it automatically redistributes the data to the new nodes. Because it doesn't create a new cluster in this scenario, the elastic resize operation completes quickly, usually in a few minutes. You might notice a slight increase in execution time for some queries while the data is redistributed in the background. An elastic resize operation occurs in the following stages:

1. Elastic resize takes a cluster snapshot.

   The snapshot that elastic resize creates includes no-backup tables (p. 230). If your cluster doesn't have a recent snapshot because you disabled automated snapshots, the backup operation takes longer. To minimize the time before the resize operation begins, we recommend that you enable
automated snapshots or create a manual snapshot before starting an elastic resize. When you start an elastic resize and a snapshot operation is currently in progress, then elastic resize might fail if the snapshot operation doesn't complete within a few minutes. For more information, see Amazon Redshift snapshots (p. 227).

2. The cluster is temporarily unavailable while elastic resize migrates cluster metadata.

   This stage is very short, just a few minutes at most. Amazon Redshift holds session connections and queries remain queued. Some sessions and queries might time out.

3. Session connections are reinstated and queries resume.

4. Elastic resize redistributes data to the node slices in the background.

   The cluster is available for read and write operations, but some queries might take longer to execute.

When a cluster is resized using elastic resize to change the node type, a snapshot is created. A new cluster is provisioned for you with the latest data from the snapshot. The cluster is temporarily unavailable for writes when the data is transferred to the new cluster. It is available for reads. The new cluster is populated in the background. After the new cluster is fully populated, queries should reach optimal performance. When the resize process nears completion, Amazon Redshift updates the endpoint of the new cluster, and all connections to the original cluster are terminated.

After the resize completes, Amazon Redshift sends an event notification. You can connect to the new cluster and resume running read and write queries.

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

New console

To monitor the progress of a resize operation using the Amazon Redshift console, choose CLUSTERS, then choose the cluster being resized to see the details.

Original console

To monitor the progress of an elastic resize operation using the Amazon Redshift console, choose the Status tab on the cluster details page.
You can't use elastic resize on single-node clusters.

To run an elastic resize on a cluster that is transferring data from a shared snapshot, at least one backup must be available for the cluster. You can view your backups on the Amazon Redshift console snapshots list, the `describe-cluster-snapshots` CLI command, or the `DescribeClusterSnapshots` API operation.

Elastic resize doesn't sort tables or reclaim disk space, so it isn't a substitute for a vacuum operation. A classic resize copies tables to a new cluster, so it can reduce the need to vacuum. For more information, see Vacuuming tables.

Elastic resize has the following constraints:

- Elastic resize is available only for clusters that use the EC2-VPC platform. For more information, see Use EC2-VPC when you create your cluster (p. 9).
- Make sure that your new node configuration has enough storage for existing data. Even when you add nodes, your new configuration might not have enough storage because of the way that data is redistributed.
- The possible configurations of node number and type that you can resize to is determined by the number of nodes in the original cluster and the target node type of the resized cluster. To determine the possible configurations available, you can use the console. Or you can use the `describe-node-configuration-options` AWS CLI command with the `action-type resize-cluster` option. For more information about the resizing using the Amazon Redshift console, see Resizing a cluster (p. 59).

The following example CLI command describes the configuration options available. In this example, the cluster named `mycluster` is a `dc2.large` 8-node cluster.

```
aws redshift describe-node-configuration-options --cluster-identifier mycluster --region eu-west-1 --action-type resize-cluster
```
This command returns an option list with recommended node types, number of nodes, and disk utilization for each option. The configurations returned can vary based on the specific input cluster. You can choose one of the returned configurations when you specify the options of the `resize-cluster` CLI command.

- Elastic resize has limits on the nodes that you can add to a cluster. For example, a dc2 cluster supports elastic resize up to double the number of nodes. To illustrate, you can add a node to a 4-node dc2.8xlarge cluster to make it a 5-node cluster, or add more nodes until you reach 8.

With some ra3 node types, you can increase the number of nodes up to four times the existing count. Specifically, suppose that your cluster consists of ra3.4xlarge or ra3.16xlarge nodes. You can then use elastic resize to increase the number of nodes in an 8-node cluster to 32. Or you can pick a value below the limit. If your cluster has ra3.xlplus nodes, the limit is double.

All ra3 node types support a decrease in the number of nodes to a quarter of the existing count. For example, you can decrease the size of a cluster with ra3.4xlarge nodes from 12 nodes to 3, or to a number above the minimum.

The following table lists growth and reduction limits for each node type that supports elastic resize.

<table>
<thead>
<tr>
<th>Node type</th>
<th>Growth limit</th>
<th>Reduction limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ra3.16xlarge</td>
<td>4x (from 4 to 16 nodes, for example)</td>
<td>To one-quarter of the number (from 16 to 4 nodes, for example)</td>
</tr>
<tr>
<td>ra3.4xlarge</td>
<td>4x</td>
<td>To one-quarter of the number</td>
</tr>
<tr>
<td>ra3.xlplus</td>
<td>2x (from 4 to 8 nodes, for example)</td>
<td>To one-quarter of the number</td>
</tr>
<tr>
<td>dc2.8xlarge</td>
<td>2x</td>
<td>To one-half the number (from 16 to 8 nodes, for example)</td>
</tr>
<tr>
<td>dc2.large</td>
<td>2x</td>
<td>To one-half the number</td>
</tr>
<tr>
<td>ds2.8xlarge</td>
<td>2x</td>
<td>To one-half the number</td>
</tr>
<tr>
<td>ds2.xlarge</td>
<td>2x</td>
<td>To one-half the number</td>
</tr>
</tbody>
</table>

**Classic resize**

With the classic 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 or longer.

The duration of a classic resize varies based several factors, including:

- The workload on the source cluster.
- The number and size of the tables being transferred.
- How evenly data is distributed across the compute nodes and slices.
- The node configuration in the source and target clusters.

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 can't 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
To resize with minimal production impact, you can use the steps in the following section, Snapshot, restore, and resize (p. 26). You can use these steps 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.

Both the classic resize approach and the snapshot and restore approach copy user tables and data to the new cluster; they don't retain system tables and data. With either classic resize or snapshot and restore, if you have enabled audit logging in your source cluster, you can continue to access the logs in Amazon S3. With these approaches, you can still access the logs after you delete the source cluster. You can keep or delete these logs as your data policies specify. Elastic resize retains the system log tables.

After Amazon Redshift puts the source cluster into read-only mode, it provisions a new cluster, the target cluster. It does so 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 is lost and you must restart the query on the target cluster. You can view the resize progress on the Amazon Redshift console.

Amazon Redshift doesn't 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 aren't transferred, so you need to run only a VACUUM command if your tables need to be resorted. For more information, see Vacuuming tables in the Amazon Redshift Database Developer Guide.

You can cancel a classic resize operation before it completes by choosing Cancel resize from the cluster details in the Amazon Redshift console. The amount of time it takes to cancel a resize depends on the stage of the resize operation when you cancel. The cluster isn't available until the cancel resize operation completes. If the resize operation is in the final stage, you can't cancel the operation.

Snapshot, restore, and resize
As described in the preceding section, the time it takes to resize a cluster with the classic resize operation depends heavily on the amount of data in the cluster.

Elastic resize (p. 22) is the fastest method to resize an Amazon Redshift cluster. If elastic resize isn't an option for you and you require near-constant write access to your cluster, use the snapshot and restore operations with classic resize as 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. Then you 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. However, it minimizes the amount of time that you can't write to the database.

The snapshot, restore, and classic resize approach uses the following process:

1. Take a snapshot of your existing cluster. The existing cluster is the source cluster.
2. Note the time that the snapshot was taken. Doing this means that you can later identify the point when you 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. Choose 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. Be sure to reload the same data in the same order into the target cluster. If you have ongoing
data loads, 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 superuser 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 before the rename. For example, rename the target cluster from preceding to `examplecluster`. From this point on, any applications that use the endpoint containing `examplecluster` connect 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. This approach works if you don't 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 moves 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 don't 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 don't need to update application connection strings because the endpoint doesn't 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. This tie means that you can continue using the same alarms and notifications that you set up for the cluster. This continued use is primarily a concern in production environments where you want the flexibility to resize the cluster without reconfiguring related items, such as alarms and notifications.

**Details of 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 multi-node cluster, or change a multi-node 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 properties.

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 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 a cluster (p. 59).

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 aren't transferred, so you need to run a VACUUM only 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 isn't 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.

---

**Pausing and resuming clusters**

If you have a cluster that only needs to be available at specific times, you can pause the cluster and later resume it. While the cluster is paused, on-demand billing is suspended. Only the cluster's storage incurs charges. For more information about pricing, see the Amazon Redshift pricing page.
When you pause a cluster, Amazon Redshift creates a snapshot, begins terminating queries, and puts the cluster in a pausing state. If you delete a paused cluster without requesting a final snapshot, then you can't restore the cluster. You can't cancel or roll back a pause or resume operation after it's initiated.

You can pause and resume a cluster on the new Amazon Redshift console (not the original console), with the AWS CLI, or with Amazon Redshift API operations.

You can schedule actions to pause and resume a cluster. When you use the new Amazon Redshift console to create a recurring schedule to pause and resume, then two scheduled actions are created for the date range that you choose. The scheduled action names are suffixed with -pause and -resume. The total length of the name must fit within the maximum size of a scheduled action name.

You can't pause the following types of clusters:

- EC2-Classic clusters.
- Clusters that are not active, for example a cluster that is currently modifying.
- Hardware security module (HSM) clusters.
- Clusters that have automated snapshots disabled.

When deciding to pause a cluster, consider the following:

- Connections or queries to the cluster aren't available.
- You can't see query monitoring information of a paused cluster on the Amazon Redshift console.
- You can't modify a paused cluster. Any scheduled actions on the cluster aren't done. These include creating snapshots, resizing clusters, and cluster maintenance operations.
- Hardware metrics aren't created. Update your CloudWatch alarms if you have alarms set on missing metrics.
- You can't copy the latest automated snapshots of a paused cluster to manual snapshots.
- While a cluster is pausing it can't be resumed until the pause operation is complete.
- When you pause a cluster, billing is suspended. However, the pause operation typically completes within 15 minutes, depending upon the size of the cluster.
- Audit logs are archived and not restored on resume.

When you resume a cluster, consider the following:

- The cluster version of the resumed cluster is updated to the maintenance version based on the maintenance window of the cluster.
- If you delete the subnet associated with a paused cluster, you might have an incompatible network. In this case, restore your cluster from the latest snapshot.
- If you delete an Elastic IP address while the cluster is paused, then a new Elastic IP address is requested.
- If Amazon Redshift can't resume the cluster with its previous elastic network interface, then Amazon Redshift tries to allocate a new one.
- When you resume a cluster, your node IP addresses might change. You might need to update your VPC settings to support these new IP addresses for features like COPY from Secure Shell (SSH) or COPY from Amazon EMR.
- If you try to resume a cluster that isn't paused, the resume operation returns an error. If the resume operation is part of a scheduled action, modify or delete the scheduled action to prevent future errors.
- Depending upon the size of the cluster, it can take several minutes to resume a cluster before queries can be processed. In addition, query performance can be impacted for some period of time while the cluster is being re-hydrated after resume completes.
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 changes 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 changes 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 that 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 still has 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. 54).

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. 56).

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
Working with AQUA (Advanced Query Accelerator)

AQUA (Advanced Query Accelerator) is an analytics query accelerator for Amazon Redshift that uses custom-designed hardware to speed up queries that scan large datasets.

AQUA is a cost-effective addition to Amazon Redshift-managed storage that is optimized for secure, transactional, multitenant access and high-throughput analytic queries. It uses high speed Non-Volatile Memory Express (NVMe) solid-state storage elements and Nitro-based acceleration for compression and encryption. It uses acceleration based on a field-programmable gate array (FPGA) to push as much computation as possible into the storage layer. These components are connected together in a unique way, allowing data scanning without using a traditional CPU. At the same time, these components allow intermediate results to be aggregated in high-speed memory. AQUA is a data cache and maintains high-speed connections to Redshift managed storage.

You don’t need to change your databases or applications to use AQUA. Amazon Redshift identifies the scan portions of queries that can benefit from acceleration and push them to AQUA for processing. AQUA automatically optimizes query performance on subsets of the data that require extensive scans, filters, and aggregation. With this approach, you can use AQUA to run queries that scan, filter, and aggregate large datasets. AQUA excels at queries that require processing-intensive scans, filters, and aggregation such as those that contain LIKE and SIMILAR TO predicates.

AQUA supports authentication, encryption, isolation, and compliance to keep your data at rest and data in transit secure. For more information about data security and AQUA, see the following:

- Data protection in Amazon Redshift (p. 270)
- Encryption at rest (p. 271)
- Encryption in transit (p. 285)

For more information about AQUA, see When does Amazon Redshift use AQUA to run queries? (p. 32)

AQUA is available on clusters with ra3.16xlarge and ra3.4xlarge node types.

AQUA is available with release version 1.0.24421 or later in the following AWS Regions:

- US East (N. Virginia) Region (us-east-1)
- US East (Ohio) Region (us-east-2)
- US West (Oregon) Region (us-west-2)
- Asia Pacific (Singapore) Region (ap-southeast-1)
- Asia Pacific (Sydney) Region (ap-southeast-2)
- Asia Pacific (Tokyo) Region (ap-northeast-1)
- Europe (Frankfurt) Region (eu-central-1)
- Europe (Ireland) Region (eu-west-1)

You can activate and manage AQUA for Amazon Redshift clusters on the Amazon Redshift console, with the AWS CLI, or with Amazon Redshift API operations. You can do so when you create a cluster, restore a cluster from a snapshot, or modify an existing cluster. When you activate an existing cluster, make sure to reboot your cluster for the change to take effect.

To activate AQUA on the Amazon Redshift console, navigate to your cluster and choose Configure AQUA for Actions. To view a previously defined AQUA configuration for your cluster, navigate to your cluster, and view the General information section, AQUA information.
You have the following choices when you configure AQUA:

- **Turn on** – You choose to activate AQUA. AQUA can only be activated in certain AWS Regions and for ra3.4xlarge and ra3.16xlarge node types.
- **Turn off** – You choose not to activate AQUA.
- **Automatic** – Amazon Redshift determines whether to use AQUA. This is the default. Currently, AQUA isn’t activated with this option, but this behavior is subject to change.

You can also activate AQUA for the duration of a session. For more information, see activate_aqua in the Amazon Redshift Database Developer Guide.

### Managing AQUA using the AWS CLI

You can use the following AWS CLI commands to activate and manage AQUA. For more information, see the AWS CLI Command Reference.

- create-cluster
- restore-from-cluster-snapshot
- modify-aqua-configuration

### Managing AQUA using Amazon Redshift API operations

You can use the following Amazon Redshift API operations to activate and manage AQUA. For more information, see the Amazon Redshift API Reference.

- CreateCluster
- RestoreFromClusterSnapshot
- ModifyAquaConfiguration

### When does Amazon Redshift use AQUA to run queries?

For each query that scans a table, Amazon Redshift determines whether the scan operation is sent to AQUA or run locally on the Amazon Redshift cluster. Scan and aggregation operations are sent to AQUA when they contain at least one predicate that contains a LIKE or SIMILAR TO expression that AQUA supports. If a scan operation is sent to AQUA, then the entire operation runs on AQUA (not only the LIKE or SIMILAR TO processing).

For example, the following query includes the predicates `str1 LIKE 'a%'` and `num1 > 10` that are sent to AQUA.

```sql
select num2 from tbl where str1 LIKE 'a%' and num1 > 10 GROUP BY num2;
```

In this case, AQUA scans the table `tbl`, filters by the predicates, groups the results by `num2`, and returns the results to Amazon Redshift.

In the following example, Amazon Redshift determines which portions of the query are better run in AQUA and which to run locally on the Amazon Redshift cluster. In this case, there is a LIKE predicate.
on the `customer_name` column that is used to filter the rows returned. The scan operation is sent to AQUA to run that part of the query. Amazon Redshift receives only the results of the scan, which Amazon Redshift then uses to perform a join locally and complete the query.

```
select c.customer_name, p.prod_name, sum(revenue)
from orders o
  join customers c ON c.id = o.customer_id
  join products p ON p.id = o.product_id
where c.customer_name LIKE '%Amazon%Web%'
order by sum(revenue);
```

For more information about LIKE and SIMILAR TO, see LIKE and SIMILAR TO in the Amazon Redshift Database Developer Guide.

### Considerations for AQUA

The following types of SQL queries are currently not supported by AQUA. Queries that include them run locally on your Amazon Redshift cluster.

- Queries that perform writes, such as, INSERT, UPDATE, DELETE, CREATE TABLE AS, COPY, and UNLOAD. For more information, see SQL Commands in the Amazon Redshift Database Developer Guide.
- SELECT queries without a predicate, for example the following.

```
select * from tbl1
```
- Python user-defined functions (UDFs) in scans, including scans on complex views that are defined by Python UDFs.
- Queries that use the following metacharacters in LIKE and SIMILAR TO predicates:
  - A nonconstant regular expression literal in a LIKE expression.
  - Either of two alternatives, for example the following.
    
    ```
    SIMILAR TO '%(cat|dog)%'
    ```
  - Multiple instances of single-character wildcards in VARCHAR columns, for example the following.
    
    ```
    SIMILAR TO 'ab.cd.ef'
    ```
  - Unicode escape sequence, hex-characters, and octal-characters in the match string, for example the following.
    
    ```
    SIMILAR TO '%ab\uc382'
    SIMILAR TO '%ab\U0000c382'
    SIMILAR TO '%ab\x88'
    SIMILAR TO '%ab\127'
    ```
  - Repetition metacharacters such as '*', '+', '?', `{m,n}` applied on a pattern within () and on multibyte characters, for example the following.
    
    ```
    SIMILAR TO 'abc(def)*'
    ```
Managing usage limits in Amazon Redshift

You can define limits to monitor and control your usage and associated cost of some Amazon Redshift features. You can create daily, weekly, and monthly usage limits, and define actions that Amazon Redshift automatically takes if those limits are reached. Actions include such things as logging an event to a system table to record usage exceeding your defined limits. Other possible actions include raising alerts with Amazon SNS and Amazon CloudWatch to notify an administrator and disabling further usage to control costs.

You can define usage limits for each cluster. After your cluster is created, you can define usage limits for the following features:

- Amazon Redshift Spectrum
- Amazon Redshift Concurrency Scaling

Usage limits are available with release version 1.0.14677 or later in the AWS Regions where Amazon Redshift Spectrum and Amazon Redshift Concurrency Scaling are available.

A Redshift Spectrum limit specifies the threshold of the total amount of data scanned in 1-TB increments. A concurrency scaling limit specifies the threshold of the total amount of time used by concurrency scaling in 1-minute increments. A limit can be specified for a daily, weekly, or monthly period (using UTC to determine the start and end of the period). If you create a limit in the middle of a period, then the limit is measured from that point to the end of the period. For example, if you create a monthly limit on March 15, then the first monthly period is measured from March 15 through March 31.

You can define multiple usage limits for each feature. Each limit can have a different action. Possible actions include the following:
Managing cluster relocation

- **Log to system table** – This is the default action. Information is logged to the STL_USAGE_CONTROL table. Logging is helpful when evaluating past usage and in deciding on future usage limits. For more information about what is logged, see STL_USAGE_CONTROL in the Amazon Redshift Database Developer Guide.

- **Alert** – Amazon Redshift emits CloudWatch metrics for available and consumed usage. You can define up to three usage limits for each feature. If you enable the alert action using the Amazon Redshift console, a CloudWatch alarm is automatically created on these metrics. You can optionally attach an Amazon SNS subscription to that alarm. If you are using an AWS CLI or API operation, make sure that you create the CloudWatch alarm manually. When the threshold is reached, events are also logged to a system table.

- **Disable feature** – When the threshold is reached, Amazon Redshift disables the feature until the quota is refreshed for the next time period (daily, weekly, or monthly). Only one limit for each feature can have the disable action. Events are also logged to a system table, and alerts can be emitted.

Usage limits persist until the usage limit definition itself or the cluster is deleted.

You can define and manage usage limits with the new Amazon Redshift console, the AWS CLI, or with Amazon Redshift API operations. To define a limit on the Amazon Redshift console, navigate to your cluster and choose **Configure usage limit for Actions**. To view previously defined usage limits for your cluster, navigate to your cluster, and choose the **Maintenance** tab, **Usage limits** section. To view the amount of usage available and consumed for your cluster, navigate to your cluster. Choose the **Cluster performance** tab, then view the graphs for the usage consumed for a feature.

You can use the following Amazon Redshift CLI operations to manage usage limits. For more information, see the **AWS CLI Command Reference**.

- **create-usage-limit**
- **describe-usage-limits**
- **modify-usage-limit**
- **delete-usage-limit**

You can use the following Amazon Redshift API operations to manage usage limits. For more information, see the **Amazon Redshift API Reference**.

- **CreateUsageLimit**
- **DescribeUsageLimits**
- **ModifyUsageLimit**
- **DeleteUsageLimit**

Watch the following video to learn how to create and monitor usage limits using the Amazon Redshift console: **Cost Controls for Amazon Redshift Spectrum and Concurrency Scaling**.

### Managing cluster relocation in Amazon Redshift

By using relocation in Amazon Redshift, you enable Amazon Redshift to move a cluster to another Availability Zone (AZ) without any loss of data or changes to your applications. With relocation, you can continue operations when there is an interruption of service on your cluster with minimal impact.

When cluster relocation is enabled, Amazon Redshift might choose to relocate clusters in some situations. In particular, this happens where issues in the current Availability Zone prevent optimal cluster operation or to improve service availability. You can also invoke the relocation function in cases
where resource constraints in a given Availability Zone are disrupting cluster operations. An example is the ability to resume or resize a cluster. Amazon Redshift offers the relocation feature at no extra charge.

When an Amazon Redshift cluster is relocated to a new Availability Zone, the new cluster has the same endpoint as the original cluster. Your applications can reconnect to the endpoint and continue operations without modifications or loss of data. However, relocation might not always be possible due to potential resource constraints in a given Availability Zone.

Amazon Redshift cluster relocation is supported for the RA3 instance types only, such as ra3.16xlarge, ra3.4xlarge, and ra3.xlplus. RA3 instance types use Redshift Managed Storage (RMS) as a durable storage layer. The latest copy of a cluster's data is always available in other Availability Zones in an AWS Region. In other words, you can relocate an Amazon Redshift cluster to another Availability Zone without any loss of data.

When you enable your cluster for relocation, Amazon Redshift migrates your cluster to be behind a proxy. Doing this helps implement location-independent access to a cluster's compute resources. The migration causes the cluster to be rebooted. When a cluster is relocated to another Availability Zone, an outage occurs while the new cluster is brought back online in the new Availability Zone. However, you don't have to make any changes to your applications because the cluster endpoint remains unchanged even after the cluster is relocated to the new Availability Zone.

If you enable relocation and you currently use the leader node IP address to access your cluster, make sure to change that access. Instead, use the IP address associated with the cluster's virtual private cloud (VPC) endpoint. To find this cluster IP address, find and use the VPC endpoint in the Network and security section of the cluster details page. To get more details on the VPC endpoint, sign in to the Amazon VPC console.

You can also use the AWS Command Line Interface (AWS CLI) command `describe-vpc-endpoints` to get the elastic network interface associated with the endpoint. You can use the `describe-network-interfaces` command to get the associated IP address. For more information on Amazon Redshift AWS CLI commands, see Available commands in the AWS CLI Command Reference.

Enabling cluster relocation

You can enable and manage cluster relocation from the Amazon Redshift console, AWS CLI, and Amazon Redshift API.

To enable cluster relocation, define a subnet group that includes multiple Availability Zones. If Amazon Redshift identifies more than one accessible Availability Zone, Amazon Redshift automatically chooses from the list of accessible Availability Zones to relocate the cluster.

After relocation is complete, you use the same endpoint to access the cluster. Amazon Redshift deletes the original cluster's compute resources and returns them to the resource pool.

Limitations

When using Amazon Redshift relocation, be aware of the following limitations:

- Cluster relocation might not be possible in all scenarios due to potential resource limitations in a given Availability Zone. If this happens, Amazon Redshift doesn't change the original cluster.
- Relocation isn't supported on DC1, DC2, or the DS2 instance families of products.
- Relocation isn't available for a publicly accessible Amazon Redshift cluster.
- You can't perform a relocation across AWS Regions.
- You can enable relocation only for a cluster that uses the default port setting (5439). Otherwise, enabling relocation fails.
- If you have enabled relocation successfully and later attempt to modify the default port setting, the modify operation fails.
Managing relocation using the console

You can manage the settings for cluster relocation using the Amazon Redshift console.

Enabling relocation when creating a new cluster

Use the following procedure to enable relocation when creating a new cluster.

To enable relocation for a new cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose CLUSTERS.
3. Choose Create cluster to create a new cluster. For more information on how to create a cluster, see Create a sample Amazon Redshift cluster in Amazon Redshift Getting Started Guide.
4. Under Backup, for Cluster relocation, choose Enable. Relocation is disabled by default.
5. Under Network and security, for Publicly accessible, accept the default Disable. If you choose Enable, Amazon Redshift returns an error.
6. Choose Create cluster.

Modifying relocation for an existing cluster

Use the following procedure to change the relocation setting for an existing cluster.

To modify the relocation setting for an existing cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose CLUSTERS. The clusters for your account in the current AWS Region are listed. A subset of properties of each cluster is displayed in columns in the list.
3. Choose the name of the cluster that you want to modify from the list. The cluster details page appears.
4. Choose the Maintenance tab, then in the Backup details section choose Edit.
5. Under Backup, choose Enable. Relocation is disabled by default.
6. Choose the Properties tab, then in the Network and security section make sure to choose Disable for the Publicly accessible option.
7. In the Network and security section, make sure to choose Disable for the Publicly accessible option.
8. Choose Modify cluster.

Relocating a cluster

Use the following procedure to manually relocate a cluster to another Availability Zone. This is especially useful when you want to test your network setup in secondary Availability Zones or when you are running into resource constraints in the current Availability Zone.

To relocate a cluster to another Availability Zone

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose **CLUSTERS**. The clusters for your account in the current AWS Region are listed. A subset of properties of each cluster is displayed in columns in the list.

3. Choose the name of the cluster that you want to move from the list. The cluster details page appears.

4. For **Actions**, choose **Relocate**. The **Relocate cluster** page appears.

5. (Optional) Choose an **Availability Zone**. If you don't choose an Availability Zone, Amazon Redshift chooses one for you.

Amazon Redshift starts the relocation and displays the cluster as relocating. After the relocation completes, the cluster status changes to available.

**Managing relocation using the Amazon Redshift CLI**

You can manage the settings for cluster relocation using the AWS Command Line Interface (CLI).

With the AWS CLI, the following example command creates an Amazon Redshift cluster named `mycluster` that has relocation enabled.

```bash
aws redshift create-cluster --cluster-identifier mycluster --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --node-type ra3.4xlarge --port 5439 --no-publicly-accessible --availability-zone-relocation
```

If your current cluster is using a different port, you have to modify it to use 5439 before modifying it to enable relocation. The following example command modify the port in case your cluster doesn't use 5439.

```bash
aws redshift modify-cluster --cluster-identifier mycluster --port 5439
```

The following example command enables the availability-zone-relocation parameter on the Amazon Redshift cluster.

```bash
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone-relocation
```

The following example command disables the availability-zone-relocation parameter on the Amazon Redshift cluster.

```bash
aws redshift modify-cluster --cluster-identifier mycluster --no-availability-zone-relocation
```

The following example command invokes relocation on the Amazon Redshift cluster.

```bash
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone us-east-1b
```

**Working with Redshift-managed VPC endpoints in Amazon Redshift**

By default, an Amazon Redshift cluster is provisioned in a virtual private cloud (VPC). It can be accessed from another VPC or subnet when you either allow public access or set up an internet gateway, a NAT device, or an AWS Direct Connect connection to route traffic to the cluster. Or you can access a cluster by setting up a Redshift-managed VPC endpoint (powered by AWS PrivateLink).
You set up a Redshift-managed VPC endpoint as a private connection between a VPC that contains a cluster and a VPC that is running a client tool. If the cluster is in another account, then the account owner (grantor) needs to grant access to the account (grantee) that wants to establish a connection. With this approach, you can access the data warehouse without using public IP addresses or routing traffic across the internet.

The following scenarios describe common reasons to allow access to a cluster using a Redshift-managed VPC endpoint:

- AWS account A wants to allow a VPC in AWS account B to have access to a cluster.
- AWS account A wants to allow a VPC that is also in AWS account A to have access to a cluster.
- AWS account A wants to allow a different subnet in the cluster's VPC within AWS account A to have access to a cluster.

The general workflow to set up a Redshift-managed VPC endpoint to access a cluster in another account is as follows:

1. The owner account of the cluster grants access authorization to another account and specifies the AWS account ID and VPC identifier (or all VPCs) of the grantee.
2. The grantee account is notified that they have permission to create a Redshift-managed VPC endpoint.
3. The grantee account creates a Redshift-managed VPC endpoint.
4. The grantee account can now access the cluster of the owner account using the Redshift-managed VPC endpoint.

You can manage this process with the Amazon Redshift console, the AWS CLI, or the Amazon Redshift API.

**Considerations when using Redshift-managed VPC endpoints**

When using Redshift-managed VPC endpoints, keep the following in mind:

- Make sure that the cluster to access is an RA3 node type.
- Make sure that the cluster to access has cluster relocation turned on. For information about requirements to turn on cluster relocation, see Managing cluster relocation in Amazon Redshift (p. 35).
- Make sure that the cluster to access is available through port 5439.
- You can modify the VPC security groups associated with an existing Redshift-managed VPC endpoint. To modify other settings, delete the current Redshift-managed VPC endpoint and create a new one.
- The number of Redshift-managed VPC endpoints that you can create is limited to your VPC endpoint quota.
- The Redshift-managed VPC endpoints aren't accessible from the internet. A Redshift-managed VPC endpoint is accessible only within the VPC where the endpoint is provisioned or any VPCs peered with the VPC where the endpoint is provisioned as permitted by the route tables and security groups.
- You can't use the Amazon VPC console to manage Redshift-managed VPC endpoints.

For information about quotas and naming constraints, see Quotas and limits in Amazon Redshift (p. 468).

For information about pricing, see AWS PrivateLink pricing.
Managing Redshift-managed VPC endpoints using the Amazon Redshift console

You can configure the use of Redshift-managed VPC endpoints by using the Amazon Redshift console.

Granting access to a cluster

If the VPC that you want to access your cluster is in another AWS account, make sure to authorize it from the owner's (grantor's) account.

To allow a VPC in another AWS account to have access to your cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose CLUSTERS.
3. For the cluster that you want to allow access, view the cluster details by choosing the cluster name. Choose the Properties tab of the cluster.
   The Granted accounts section displays the accounts and corresponding VPCs that have access to your cluster.
4. Choose Grant access to display a form to enter Grantee information to add an account.
5. For AWS account ID, enter the ID of the account you are granting access. You can grant access to specific VPCs or all VPCs in the specified account.
6. Choose Grant access to grant access.

Creating a Redshift-managed VPC endpoint

If you own a cluster or you have been granted access to it, you can create a Redshift-managed VPC endpoint for the cluster.

To create a Redshift-managed VPC endpoint

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose CONFIG.
   The Configurations page displays the Redshift-managed VPC endpoints that have been created. To view details for an endpoint, choose its name.
3. Choose Create endpoint to display a form to enter information about the endpoint to add.
4. Enter values for Endpoint name, AWS account ID, Cluster identifier, Virtual private cloud (VPC), Subnet group, and other properties of the endpoint.
   The subnet group in Subnet group defines the subnets and IP addresses where Amazon Redshift deploys the endpoint. Amazon Redshift chooses a subnet that has IP addresses available for the network interface associated with the endpoint.
   The optional security group in Security group defines the ports, protocols, and sources for inbound traffic that you are authorizing for your endpoint. Commonly, you allow access to port 5439 to the security group or the CIDR range where your workloads run.
5. Choose Create endpoint to create the endpoint.

After your endpoint is created, you can access the cluster through the URL shown in Endpoint URL in the configuration settings for your Redshift-managed VPC endpoint.
Managing Redshift-managed VPC endpoints using the AWS CLI

You can use the following Amazon Redshift CLI operations to work with Redshift-managed VPC endpoints. For more information, see the AWS CLI Command Reference.

- authorize-endpoint-access
- revoke-endpoint-access
- create-endpoint-access
- modify-endpoint-access
- delete-endpoint-access
- describe-endpoint-access
- describe-endpoint-authorization

Managing Redshift-managed VPC endpoints using Amazon Redshift API operations

You can use the following Amazon Redshift API operations to work with Redshift-managed VPC endpoints. For more information, see the Amazon Redshift API Reference.

- AuthorizeEndpointAccess
- RevokeEndpointAccess
- CreateEndpointAccess
- ModifyEndpointAccess
- DeleteEndpointAccess
- DescribeEndpointAccess
- DescribeEndpointAuthorization

Managing clusters using the console

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

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

New console

**To view clusters**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS. The clusters for your account in the current AWS Region are listed. A subset of properties of each cluster is displayed in columns in the list. If you don’t have any clusters, choose Create cluster to create one.
3. Choose the cluster name in the list to view more details about a cluster.
Original 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.
Amazon Redshift Cluster Management Guide
Managing clusters using the console

Topics
- Creating a cluster (p. 44)
- Modifying a cluster (p. 54)
- Deleting a cluster (p. 56)
- Rebooting a cluster (p. 58)
- Resizing a cluster (p. 59)
- Upgrading the release version of a cluster (p. 60)
- Getting information about cluster configuration (p. 61)
- Getting an overview of cluster status (p. 62)
- Creating a snapshot of a cluster (p. 63)
- Creating or editing a disk space alarm (p. 64)
- Working with cluster performance data (p. 66)
Creating a cluster

Before you create a cluster, read Overview of Amazon Redshift clusters (p. 5) and Clusters and nodes in Amazon Redshift (p. 5).

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

New console

One way to learn about creating a cluster is to create a cluster using the console.

To create a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS. The clusters for your account in the current AWS Region are listed. A subset of properties of each cluster is displayed in columns in the list.
3. Choose Create cluster to create a cluster.
4. Follow the instructions on the console page to enter the properties for Cluster configuration.

Choose one of the following methods to size your cluster:

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

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

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

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

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

5. Follow the instructions on the console page to enter the properties for Cluster details.

Note
If you are behind a firewall, the database port must be an open port that accepts inbound connections.

6. (Optional) Follow the instructions on the console page to enter properties for Cluster permissions. Provide cluster permissions if your cluster needs to access other AWS services for you, for example to load data from Amazon S3.

7. Choose Create cluster to create the cluster. The cluster might take several minutes to be ready to use.
Additional configurations

When you create a cluster, you can specify additional properties to customize it. You can find more details about some of these properties in the following list.

Virtual private cloud (VPC)

Choose a VPC that has a subnet group. After the cluster is created, the subnet group can’t be changed.

Parameter groups

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

Encryption

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. 271).

- **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. 272).

- **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. 274).

Maintenance track

You can choose whether the cluster version used is the **Current**, **Trailing**, or sometimes **Preview** track.

Monitoring

You can choose whether to create CloudWatch alarms.

Configure cross-region snapshot

You can choose whether to enable cross-Region snapshots.

Original console

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. 46).

• 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 a launch cluster (p. 46).

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 a launch cluster (p. 46).

To create a cluster by using a 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.

Creating a cluster by using a 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. 46).
To create a cluster by using a 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.
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.

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. 271).

   - **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. 272).

   - **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. 274).
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-VPC and EC2-Classic 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. 74).

![Choose a VPC](image1)

- 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.

![Choose a VPC](image2)

**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 enabled.

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 Enhanced VPC routing in Amazon Redshift (p. 194).

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

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.

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. 370).

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

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.

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.

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.
7. 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
- HSM Connection
- HSM Client Certificate
- Maintenance detail
- Snapshot preferences

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. 227).

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

New console

To 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. On the navigation menu, choose CLUSTERS.
3. Choose the cluster to modify.
5. Update the cluster properties. Some of the properties you can modify are:
   - Cluster identifier
   - Snapshot retention
   - Cluster relocation

   To edit settings for Network and security, Maintenance, and Database configurations, the console provides links to the the appropriate cluster details tab.
6. Choose Save changes.
**Original console**

**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 resulting window shows the options available to modify a cluster. Including tabs with options to update **Cluster settings**, **Maintenance settings**, and **Snapshot settings**.

**Setting the maintenance track for a cluster**

You can set the maintenance track for a cluster with the console. For more information, see **Choosing cluster maintenance tracks** (p. 16).

**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. In the navigation pane, choose **Clusters**, and then choose the cluster that you want to modify.
3. Choose the **Maintenance settings** tab.
4. Choose either **Current** or **Trailing**.
5. Choose **Modify**.

**Deferring maintenance**

If you need to reschedule your cluster’s maintenance window, you have the option to defer maintenance by up to 45 days.

**To defer the maintenance window**

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. Choose the **Maintenance settings** tab.
4. Choose **Defer maintenance** and set the date and time to defer maintenance.
5. Choose **Modify**.

### 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. 30).

**Note**

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

To delete a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS.
3. Choose the cluster to delete.
4. For Actions, choose Delete. The Delete cluster page appears.
5. Choose Delete cluster.

Original console

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 creating 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.

![Cluster Status: final-snapshot](image1)

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

![Cluster Status: deleting](image2)

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.

**Note**

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

## New console

To 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. On the navigation menu, choose CLUSTERS.
3. Choose the cluster to reboot.
4. For Actions, choose Reboot cluster. The Reboot cluster page appears.
5. Choose Reboot cluster.

## Original console

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.
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. 21).

**Note**

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

**New console**

**To 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. On the navigation menu, choose **CLUSTERS**.
3. Choose the cluster to resize.
4. For **Actions**, choose **Resize**. The **Resize cluster** page appears.
5. Follow the instructions on the page. You can resize the cluster now, once at a specific time, or increase and decrease the size of your cluster on a schedule.
6. Depending on your choices, choose **Resize now** or **Schedule resize**.

**Original console**

**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.
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**.

![Resize Cluster window](image)

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

![Status tab](image)

You can cancel a classic resize before it's complete by choosing **cancel resize** on the cluster list.

![Clusters list](image)

### Upgrading the release version of a cluster

You can upgrade the release maintenance version of a cluster that has a **Release Status** value of **New release available**. When you upgrade the maintenance version, you can choose to upgrade immediately or upgrade in the next maintenance window.

**Important**

If you upgrade immediately, your cluster is offline until the upgrade completes.

**Note**

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

**New console**

**To upgrade a cluster to a new release version**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at [https://console.aws.amazon.com/redshift/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose CLUSTERS.
3. Choose the cluster to upgrade.
4. For Actions, choose Upgrade cluster version. The Upgrade cluster version page appears.
5. Follow the instructions on the page.

Original console

To upgrade your cluster to a new release 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, and then choose the cluster that you want to upgrade.
3. On the Clusters tab, choose Cluster, and then choose Upgrade release version.
4. In the Upgrade window, you can view version numbers of the Current release and New release, a link to details about the Cluster version history (p. 80), and the Maintenance window schedule. If you choose Yes, Upgrade now, you must acknowledge that the cluster is offline during the upgrade. Otherwise, you can Never mind, upgrade in my maintenance window.

When the upgrade completes, you can see the new status in the Release status column.

Getting information about cluster configuration

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

New console

To display information about a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose each tab to view more details.

Original console

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

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

New console

To view the status 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. On the navigation menu, choose CLUSTERS.
3. View the status of the cluster in the Status column.

Original console

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.
Creating a snapshot of a cluster

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

New console

To create 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. On the navigation menu, choose CLUSTERS.
3. Choose the cluster for which to create a snapshot.
4. For Actions, choose Create snapshot. The Create snapshot page appears.
5. Follow the instructions on the page.
6. Choose Create snapshot.

Original console

You can create a snapshot of your cluster from the Configuration tab of your cluster as shown following. You can also create a snapshot of your cluster from the snapshots part of the Amazon Redshift console. For more information, see Managing snapshots using the console (p. 238).

To create 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, enter 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. 238).

**Creating or editing a disk space alarm**

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

To create a disk space usage alarm for a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose ALARMS.
3. For Actions, choose Create alarm. The Create alarm page appears.
4. Follow the instructions on the page.
5. Choose Create alarm.

Original console

If you created 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.

5. In the Edit Alarm window, edit any values that you want to change, such as the percentage or minutes.
6. 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.

7. Choose Save.

**Working with cluster performance data**

In the new console, you can work with cluster performance on the Cluster performance tab of the cluster details page.

In the original console, 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. 408).

**Managing clusters using the AWS CLI and Amazon Redshift API**

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

- cancel-resize
- create-cluster
- delete-cluster
- describe-clusters
- describe-cluster-versions
- describe-node-configuration-options
- describe-orderable-cluster-options
- describe-resize
- modify-cluster
- pause-cluster
- reboot-cluster
- resize-cluster
- resume-cluster

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

- CancelResize
- CreateCluster
- DeleteCluster
- DescribeClusters
- DescribeClusterVersions
- DescribeNodeConfigurationOptions
- DescribeResize
- DescribeOrderableClusterOptions
- ModifyCluster
- PauseCluster
- RebootCluster
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. 391). You need to update the code and specify a cluster identifier.

Example

```java
package com.amazonaws.services.redshift;

import java.io.IOException;
import com.amazonaws.services.redshift.AmazonRedshift;
import com.amazonaws.services.redshift.AmazonRedshiftClientBuilder;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyCluster {
    // snippet-sourcedescription:[CreateAndModifyCluster demonstrates how to create and modify
    // an Amazon Redshift cluster.]
    // snippet-service:[redshift]
    // snippet-keyword:[Java]
    // snippet-keyword:[Amazon Redshift]
    // snippet-keyword:[Code Sample]
    // snippet-keyword:[CreateCluster]
    // snippet-keyword:[DescribeClusters]
    // snippet-keyword:[ModifyCluster]
    // snippet-keyword:[full-example]
    // snippet-sourcedate:[2019-02-01]
    // snippet-sourceauthor:[AWS]
    // snippet-start:[redshift.java.CreateAndModifyCluster.complete]

    public static void main(String[] args) {
        try {
            // Create a client using the Amazon Redshift service
            AmazonRedshift client = AmazonRedshiftClientBuilder.standard()
                .withCredentials(new InstanceProfileCredentialsProvider())
                .withRegion(Region.US_EAST_1)
                .build();

            // Create a new cluster
            CreateClusterRequest createClusterRequest = new CreateClusterRequest()
                .withClusterIdentifier("my-cluster")
                .withMasterUsername("admin")
                .withMasterUserPassword("password")
                .withNodeType("c3.large")
                .withNodeTypeConfiguration(new NodeTypeConfiguration()
                    .withNodeCount(3)
                    .build());

            // Create the cluster
            CreateClusterResult createClusterResult = client.createCluster(createClusterRequest);

            // List the metadata about the cluster
            DescribeClustersRequest describeClustersRequest = new DescribeClustersRequest()
                .withClusterIdentifier("my-cluster");

            DescribeClustersResult describeClustersResult = client.describeClusters(describeClustersRequest);

            // Modify the preferred maintenance window for the cluster
            ModifyClusterRequest modifyClusterRequest = new ModifyClusterRequest()
                .withClusterIdentifier("my-cluster")
                .withPreferredMaintenanceWindow("00:00-01:00")
                .build();

            // Modify the cluster
            ModifyClusterResult modifyClusterResult = client.modifyCluster(modifyClusterRequest);
        } catch (IOException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // snippet-end:[redshift.java.CreateAndModifyCluster.complete]
}
```
public static AmazonRedshift client;
public static String clusterIdentifier = "***provide a cluster identifier***";
public static long sleepTime = 20;

public static void main(String[] args) throws IOException {
    // Default client using the {@link
    com.amazonaws.auth.DefaultAWSCredentialsProviderChain}
    client = AmazonRedshiftClientBuilder.defaultClient();
    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("ds2.xlarge")
        .withNumberOfNodes(2)
        .withClusterSubnetGroupName("subnetgroup1");

    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);
        }
    }
}

Managing clusters in a VPC

Topics

• Overview (p. 69)
• Creating a cluster in a VPC (p. 71)
• Managing VPC security groups for a cluster (p. 73)
• Amazon Redshift cluster subnet groups (p. 74)

Overview

Amazon Redshift supports both the EC2-VPC and EC2-Classic platforms to launch a cluster in a virtual
private cloud (VPC) based on the Amazon VPC service. For more information, see Use EC2-VPC when
you create your cluster (p. 9).

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 by creating a cluster subnet group. This information includes the VPC ID and a list of
subnets in your VPC. 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. 74).
For more information about setting up VPC, see 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 to
use for the external IP address. An elastic IP address is a static IP address that is associated with your
AWS account. You can use an elastic IP address to connect to your cluster from outside the VPC. An
elastic IP address 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 elastic IP address 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 elastic IP address to use for the VPC. However, the assigned elastic IP address is managed by the Amazon Redshift service and isn't associated with your AWS account. For more information, see Elastic IP addresses in the Amazon EC2 User Guide for Linux Instances.

In some cases, you might have a publicly accessible cluster in a VPC that you want to connect to it by using the private IP address from within the VPC. If so, set the following VPC parameters to true:

- DNS resolution
- DNS hostnames

Suppose that you have a publicly accessible cluster in a VPC but don't set those parameters to true in the VPC. In these cases, connections made from within the VPC resolve to the elastic IP address 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 continue to use the elastic IP address to connect to the cluster until you resize the cluster. This occurs even with the preceding parameters set. Any new clusters follow the new behavior of using the private IP address when connecting to the publicly accessible cluster from within the same VPC.

The elastic IP address is an external IP address for accessing the cluster outside of a VPC. It's not related to the cluster node public IP addresses and private IP addresses that are displayed in the Amazon Redshift console under Connection details. 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. These circumstances occur 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 elastic IP address is available when you create the cluster or restore the cluster from a snapshot. In some cases, you might want to associate the cluster with an elastic IP address or change an elastic IP address that is associated with the cluster. To attach an elastic IP address after the cluster is created, first update the cluster so that it is not publicly accessible, then make it both publicly accessible and add an Elastic IP address in the same operation.

- 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, see 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. 71).

**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. 227).
Creating a cluster in a VPC

The following are the general steps how you can deploy a cluster in your virtual private cloud (VPC).

**New console**

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 Use EC2-VPC when you create your cluster (p. 9). To create a VPC, see Getting started with Amazon VPC in the Amazon VPC User Guide. Make a note of the VPC identifier, subnet, and subnet's Availability Zone. You 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, see Adding a subnet to your VPC in the Amazon VPC User Guide.

2. Create an Amazon Redshift cluster subnet group to specify which subnet your Amazon Redshift cluster can use in the VPC.

   You can create a cluster subnet group using either the Amazon Redshift console or programmatically. For more information, see Amazon Redshift cluster subnet groups (p. 74).

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

   You can enable a client outside the VPC (on the public internet) to connect to the cluster. To do this, you 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, see Security group rules in the Amazon VPC User Guide.

4. Follow the steps in Getting started with Amazon Redshift in the Amazon Redshift Getting Started Guide to create a cluster. Make the following modifications when creating your cluster:

   - To display the Additional configurations section, switch off Use defaults.
   - In the Network and security section, specify the Virtual private cloud (VPC), Cluster subnet group, and VPC security group that you set up.

**Original console**

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 Use EC2-VPC when you create your cluster (p. 9). To create a VPC, see Getting started with Amazon VPC in the Amazon VPC User Guide. Make a note of the VPC identifier, subnet, and subnet's Availability Zone. You 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, see Adding a subnet to your VPC in the Amazon VPC User Guide.
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 a cluster subnet group using either the Amazon Redshift console or programmatically. For more information, see Amazon Redshift cluster subnet groups (p. 74).

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

You can enable a client outside the VPC (on the public internet) to connect to the cluster. To do this, you 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, see 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 Enable if you want the cluster to have a public IP address that can be accessed from the public internet.

  **Note**

  When choosing Enable, your cluster is publicly accessible outside the VPC and subject to security threats.

  Select Disable if you want the cluster to have a private IP addressed that can only be accessed from within the VPC.

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

- **Elastic IP** – Select an elastic IP address 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.

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. You 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. 370).

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 a launch cluster (p. 46) and To modify a cluster (p. 55).

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

<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>The security group ID (sg-xxxxxccc)</td>
<td>All</td>
<td>All</td>
<td>Allow inbound traffic from instances assigned to the same security group</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>All</td>
<td>All</td>
<td>Allow all outbound traffic</td>
</tr>
</tbody>
</table>

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.

You can create up to 100 VPC security groups for a VPC and associate a VPC security group with many clusters. However, you can only associate up to five 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 Amazon EC2 CLI, and the AWS Tools for Windows PowerShell. For more information about working with VPC security groups, see 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, see the 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 API operations 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

You can 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.

You can provision a cluster on one of the subnets that you provide the subnet group. A cluster subnet group enables you to specify a set of subnets in your virtual private cloud (VPC).

Creating a cluster subnet group

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

Note

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

New console

To create 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. On the navigation menu, choose CONFIG, then choose Subnet groups. The list of subnet groups is displayed.
3. Choose Create cluster subnet group to display the create page.
4. Enter information for the subnet group, including which subnets to add.
5. Choose Create cluster subnet group to create the group with the subnets that you chose.

Original console

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

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

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

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

5. Choose Yes, Create.

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

**Modifying a cluster subnet group**

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

To 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. On the navigation menu, choose CONFIG, then choose Subnet groups. The list of subnet groups is displayed.

3. Choose the subnet group to modify.

4. For Actions, choose Modify to display the details of the subnet group.

5. Update information for the subnet group.

6. Choose Save to modify the group.

Original console

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, choose Security.

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

In the example following, subnetgroup2 is the cluster subnet group to modify.

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

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add one or more subnets to the group.</td>
<td>Select an individual subnet by using the Availability Zone and Subnet ID boxes or choose add all the subnets.</td>
</tr>
<tr>
<td>Remove a subnet from the group.</td>
<td>In the lists of subnets in use for the group, choose Remove next to the subnet to remove.</td>
</tr>
</tbody>
</table>

Save.
To delete a cluster subnet group

You can't delete a cluster subnet group that is used by a cluster.

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

**New console**

**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. On the navigation menu, choose **CONFIG**, then choose **Subnet groups**. The list of subnet groups is displayed.
3. Choose the subnet group to delete, then choose **Delete**.

**Original console**

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

In the example following, `my-subnet-group` is the cluster subnet group to delete.

4. In the Delete Cluster Subnet Group dialog box, choose **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. 391). You need to update the code and provide a cluster subnet group name and two subnet identifiers.

Example

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * http://aws.amazon.com/apache2.0/
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[CreateAndModifyClusterSubnetGroup demonstrates how to create
// and modify an Amazon Redshift subnet group.]
// snippet-service:[redshift]
// snippet-keyword:[Java]
// snippet-keyword:[Amazon Redshift]
// snippet-keyword:[Code Sample]
// snippet-keyword:[CreateClusterSubnetGroup]
// snippet-keyword:[DescribeClusterSubnetGroups]
// snippet-keyword:[full-example]
// snippet-sourcedate:[2019-02-01]
// snippet-sourceauthor:[AWS]
// snippet-start:[redshift.java.CreateAndModifyClusterSubnetGroup.complete]
package com.amazonaws.services.redshift;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSubnetGroup {
```
public static AmazonRedshift client;
public static String clusterSubnetGroupName = "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 {
    // Default client using the @link
    com.amazonaws.auth.DefaultAWSCredentialsProviderChain
    client = AmazonRedshiftClientBuilder.defaultClient();
    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)
Manage cluster subnet groups using the AWS CLI and Amazon Redshift 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 API operations to manage cluster subnet groups.

- `CreateClusterSubnetGroup`
- `DeleteClusterSubnetGroup`
- `DescribeClusterSubnetGroups`
- `ModifyClusterSubnetGroup`

Cluster version history

Amazon Redshift periodically releases new cluster versions that are used to update your cluster.

**Important**

For information about available Amazon Redshift cluster versions, and their features, improvements, and fixes, see Amazon Redshift Maintenance Forum Announcements in the Amazon Redshift Discussion Forum.
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.
- 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. 81)
- Scheduling a query on the Amazon Redshift console (p. 87)
- Connecting to an Amazon Redshift cluster using SQL client tools (p. 90)
- Using the Amazon Redshift Data API (p. 167)

### Querying a database using the query editor

Using the query editor is an easy 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.

In February 2021, an updated query editor was deployed and authorization permissions to use the query editor changed. The new query editor uses the Amazon Redshift Data API to run queries. The AmazonRedshiftQueryEditor policy, which is an AWS-managed AWS Identity and Access Management (IAM) policy, was updated to include the necessary permissions. If you have a custom IAM policy, make sure that you update it. Use AmazonRedshiftQueryEditor as a guide. The changes to AmazonRedshiftQueryEditor include the following:

- Permission to manage query editor statement results requires the statement owner user.
- Permission to use Secrets Manager to connect to a database has been added.

For more information, see Permissions required to use the Amazon Redshift console query editor (p. 296).

When you connect to your cluster from the new query editor, you can use one of two authentication methods, as described in Connecting with the query editor (p. 83).

The old query editor is available on the Amazon Redshift console for a limited time. You can switch to it and back on the query editor console page. With the old query editor, use your existing permissions. For more information, see Connecting with the old query editor (p. 84).

Using the query editor, you can do the following:
Considerations

- Run single SQL statement queries.
- Download result sets as large as 100 MB to a comma-separated value (CSV) file.
- Save queries for reuse. You can't save queries in the Europe (Paris) Region, the Asia Pacific (Osaka) Region, the Asia Pacific (Hong Kong) Region, or the Middle East (Bahrain) Region.
- View query runtime details for user-defined tables.
- Schedule queries to run at a future time.
- View a history of queries that you created in the query editor.
- Run queries against clusters using enhanced VPC routing.

Query editor considerations

Consider the following about working with queries when you use the query editor:

- The maximum duration of a query is 24 hours.
- The maximum query result size is 100 MB. If a call returns more than 100 MB of response data, the call is terminated.
- The maximum retention time for query results is 24 hours.
- The maximum query statement size is 100 KB.
- The cluster must be in a virtual private cloud (VPC) based on the Amazon VPC service.
- You can't use transactions in the query editor. For more information about transactions, see BEGIN in the Amazon Redshift Database Developer Guide.
- You can save a query up to 3,000 characters long.

Enabling access to the query editor

To access the query editor, you need permission. To enable access, attach the AmazonRedshiftQueryEditor and AmazonRedshiftReadOnlyAccess AWS-managed policies for IAM permissions to the IAM user that you use to access your cluster. You can use the IAM console (https://console.aws.amazon.com/iam/) to attach IAM policies.

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

The AWS-managed policy AmazonRedshiftQueryEditor allows the action redshift:GetClusterCredentials, which by default gives a database user superuser access to the database. To restrict access, you can do one of the following:

- Create a custom policy that allows calling redshift:GetClusterCredentials and restricts the resource to a given value for DbUser.
- Add a policy to the user that denies permission to redshift:GetClusterCredentials and then requires users of the query editor to sign in with temporary credentials. For example, a denial policy might be similar to the following policy.

```json
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "redshift:GetClusterCredentials",
        "Resource": "*"
    }
}
```
For more information, see Create an IAM role or user role or user with permissions to call GetClusterCredentials (p. 311).

If you are granted access to Amazon Redshift query editor by attaching the AWS-managed policy AmazonRedshiftQueryEditor, then you can list all secrets. However, you can create and retrieve only secrets that are tagged with the key RedshiftQueryOwner and the value ${aws:userid}.

If you create the key from the Amazon Redshift query editor, then the key is automatically tagged.

To use a secret that wasn’t created with the Amazon Redshift query editor, confirm that the secret is tagged with the key RedshiftQueryOwner and a value of your unique IAM user identifier, for example AIDACKCEVSQ6C2EXAMPLE.

To attach the required IAM policies for the query editor

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

Connecting with the query editor

When you connect to a cluster with the query editor, you use one of the following authentication methods. Each method requires a different combination of input from the Amazon Redshift console.

AWS Secrets Manager

With this method, provide a secret value for secret-arn that is stored in AWS Secrets Manager. This secret contains credentials to connect to your database.

Temporary credentials

With this method, provide your database and db-user values.

Storing database credentials in AWS Secrets Manager

When you call the query editor, you can pass credentials for the cluster by using a secret in AWS Secrets Manager. To pass credentials in this way, you specify the name or the Amazon Resource Name (ARN) of the secret.

For more information about the minimum permissions, see Creating and Managing Secrets with AWS Secrets Manager in the AWS Secrets Manager User Guide.

To store your credentials in a secret for an Amazon Redshift cluster

1. Use AWS Secrets Manager to create a secret that contains credentials for the cluster. When you choose Store a new secret, choose Credentials for Redshift cluster. Store a value for User name (the database user), Password, and DB cluster (cluster identifier) in your secret.
Connecting with the old query editor

The following cluster node types support the old query editor:

- dc1.8xlarge
- dc2.large
- dc2.8xlarge
- ds2.8xlarge
- ra3.xlplus
- ra3.4xlarge
- ra3.16xlarge

Be aware of the following considerations when you use the old query editor on the Amazon Redshift console:

- Up to 50 users can connect to a cluster with the query editor at the same time.
- The maximum number of users connecting to a cluster includes those 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.
- The query editor only runs short queries that can complete within 10 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 Enhanced VPC routing in Amazon Redshift (p. 194).
- You can’t use transactions in the query editor. For more information about transactions, see BEGIN in the Amazon Redshift Database Developer Guide.
- You can save a query up to 3,000 characters long.

To access the old query editor, you need permission. To enable access, attach the AWS-managed policies AmazonRedshiftQueryEditor and AmazonRedshiftReadOnlyAccess for IAM permissions to the IAM user that you use to access your cluster.

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

The AWS-managed policy AmazonRedshiftQueryEditor allows the action redshift:GetClusterCredentials, which by default gives a database user superuser access to the database. To restrict access, you can do one of the following:

- Create a custom policy that allows calling redshift:GetClusterCredentials and restricts the resource to a given value for DbUser.
- Add a policy to the user that denies permission to redshift:GetClusterCredentials and then requires users of the query editor to sign in with temporary credentials. For example, a denial policy might be similar to the following policy.

```json
{
}
```
For more information, see Create an IAM role or user role or user with permissions to call GetClusterCredentials (p. 311).

When you connect to a cluster with the old query editor, you use one of the following authentication methods. Each method requires a different combination of input from the Amazon Redshift console.

**Connect with database password**

With this method, provide the master user password for the default database. You provided this password when you created your cluster.

**Temporary credentials**

With this method, provide your `database` and `db-user` values.

---

**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. 44).

**Note**

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

**New console**

**To use the query editor on 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/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose **EDITOR**, then connect to a database in your cluster.
3. For **Schema**, choose **public** to create a new table based on that schema.
4. Enter the following in the query editor window and choose **Run** to create a new table.

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

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

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

7. Choose Clear.

8. Enter the following command in the query editor window and choose Run to query the new table.

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

The Query results displays the results.

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

9. Choose Execution to view the run details.

10. Choose Data, then Export to download the query results as a file.

**Original console**

**To use the query editor**

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

2. In the navigation pane, choose Query Editor.

![Query Editor](image)

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.

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

5. Choose Clear.

6. Enter the following command in the Query Editor window and choose Run query to add rows to the table.
Scheduling a query

7. Choose Clear.
8. Enter the following command in the Query Editor window and choose Run query to query the new table:

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

You should see the following results.

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

Scheduling a query on the Amazon Redshift console

To create a schedule to run a SQL statement, you can use the query editor on the Amazon Redshift console. You can create a schedule to run your SQL statement at the time intervals that match your business needs. When it's time for the scheduled query to run, Amazon EventBridge initiates the query.

**To create a schedule to run a SQL statement on the console**

1. Open the console and query editor as described in Using the query editor (p. 85).
2. Choose Schedule to create a schedule to run an SQL statement.

When you define the schedule, you provide the following information:
• An IAM role that is used to assume the required permissions to run the query. For more information, see Setting up permissions to schedule a query on the Amazon Redshift console (p. 88).
• The authentication values for either AWS Secrets Manager or temporary credentials to authorize access your cluster. For more information, see Authenticating a scheduled query (p. 90).
• The name of the scheduled query and a single SQL statement to be run.
• The schedule frequency and repeat options or a cron formatted value.
• Optionally, you can enable Amazon SNS notifications to monitor the scheduled query. If your query is being run but you don't see messages published in your SNS topic, see My rule is being triggered but I don't see any messages published into my Amazon SNS topic in the Amazon EventBridge User Guide.

You can also manage and update scheduled queries using the Amazon Redshift console. Depending on your version of the console, scheduled queries might be listed in the following places:

• On the Schedules tab of the details page of your cluster.
• On the scheduled queries list that you can reach from the navigation pane. To see the list, on the navigation pane choose QUERIES, Schedule query list.
• On the Scheduled queries tab of the query editor.

If you choose Schedule name from one of these locations, you can view and edit your scheduled query's definition.

**Setting up permissions to schedule a query on the Amazon Redshift console**

To schedule queries, the AWS Identity and Access Management (IAM) user defining the schedule and the IAM role associated with the schedule must be configured as follows.

For the IAM user logged into the Amazon Redshift console, do the following:

• Attach the AmazonEventBridgeFullAccess AWS-managed policy.
• Attach a policy with the sts:AssumeRole permission of the IAM role that you specify when you define the scheduled SQL statement.

The following example shows a policy that assumes a specified IAM role.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeIAMRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::account-id:role/sql-statement-iam-role"
        }
    ]
}
```

For the IAM role that you specify to enable the scheduler to run a query, do the following:

• Ensure that this IAM role specifies the EventBridge service principal (events.amazonaws.com). The following is an example trust relationship.
Setting permissions to schedule a query

For more information about how to create an IAM role for EventBridge events, see Permissions required to use the Amazon EventBridge scheduler (p. 298).

- Attach the AmazonRedshiftDataFullAccess AWS-managed policy to the IAM role.
- To allow IAM users to view schedule history, edit the IAM role to add the sts:AssumeRole permission.

The following is an example of the definition of an IAM role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": ["events.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-id:user/user-name"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The following example snippet allows specific IAM users to view schedule history.

```
{
  "Sid": "AssumeRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:user/testRedshiftUser",
      "arn:aws:iam::account-id:user/myusername"
    ],
    "Action": "sts:AssumeRole"
  }
}
```
Authenticating a scheduled query

When you schedule a query, you use one of the following authentication methods when the query SQL runs. Each method requires a different combination of input from the Amazon Redshift console.

AWS Secrets Manager

With this method, provide a secret value for `secret-arn` that is stored in AWS Secrets Manager. This secret contains credentials to connect to your database. The secret must be tagged with the key `RedshiftDataFullAccess`.

For more information about the minimum permissions, see Creating and Managing Secrets with AWS Secrets Manager in the AWS Secrets Manager User Guide.

Temporary credentials

With this method, provide your `database` and `db-user` values.

The AmazonRedshiftDataFullAccess policy allows the database user named `redshift_data_api_user` permission for `redshift:GetClusterCredentials`. If you want to use a different database user to run the SQL statement, then add a policy to the IAM role to allow `redshift:GetClusterCredentials`. The following example policy allows database users `awsuser` and `myuser`.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "UseTemporaryCredentialsForAllDbUsers",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
                "arn:aws:redshift:*:*:dbuser:*/awsuser",
                "arn:aws:redshift:*:*:dbuser:*/myuser"
            ]
        }
    ]
}
```

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 doesn't provide or install any SQL client tools or libraries. To use these tools or libraries to work with data in your clusters, install them on your client computer or Amazon EC2 instance. You can use most SQL client tools that support JDBC or ODBC drivers.

You can use the following sections to walk through the process of configuring your client computer or Amazon EC2 instance to use a JDBC or ODBC connection. They also discuss related security options for the client connection to the server. Additionally, following you can find information about setting up and connecting from two example third-party SQL client tools, SQL Workbench/J and psql. You can try these tools 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.
Configuring connections in Amazon Redshift

In the following section, you can find 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. It also describes how to use Secure Sockets Layer (SSL) and server certificates to encrypt communication between the client and server.

Topics
- JDBC and ODBC drivers for Amazon Redshift (p. 91)
- Configuring a connection for JDBC driver version 2.0 for Amazon Redshift (p. 92)
- Configuring a JDBC driver version 1.0 connection (p. 122)
- Configuring an ODBC connection (p. 130)

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 API operations, 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 currently use those drivers, we recommend moving to the new Amazon Redshift-specific drivers. For more information about how to download the JDBC and ODBC drivers and configure connections to your cluster, see Configuring a connection for JDBC driver version 2.0 for Amazon Redshift (p. 92), Configuring a JDBC driver version 1.0 connection (p. 122), and Configuring an ODBC connection (p. 130).

For JDBC driver version 1.2.47.1071 or later and ODBC driver version 1.4.16.1000 or later, Amazon Redshift must be version 1.0.17708 or later. To verify the Amazon Redshift version, see VERSION in the Amazon Redshift Database Developer Guide.

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 details page.

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

New console

To find the connection string for a cluster
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **CLUSTERS**, then choose the cluster name from the list to open its details.

3. The **JDBC URL** and **ODBC URL** connection strings are available, along with additional details, in the **General information** section. Each string is based on the AWS Region where the cluster runs. Click the icon next to the appropriate connection string to copy it.

**Original console**

**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, choose 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 (Oregon) Region. If you launch your cluster in a different AWS Region, the connection strings are based that Region's endpoint.

---

**Configuring a connection for JDBC driver version 2.0 for Amazon Redshift**

You can use a JDBC driver version 2.0 connection to connect to your Amazon Redshift cluster from many third-party SQL client tools. To do this, you download a JDBC driver.
To use a JDBC connection, refer to the following sections.

Topics
- Download the Amazon Redshift JDBC driver, version 2.0 (p. 93)
- Installing the Amazon Redshift JDBC driver, version 2.0 (p. 94)
- Referencing the JDBC driver libraries (p. 94)
- Registering the driver class (p. 94)
- Getting the JDBC URL (p. 95)
- Building the connection URL (p. 96)
- Configuring TCP keepalives for your JDBC connection (p. 96)
- Configuring your JDBC connection with Apache Maven (p. 97)
- Configuring authentication and SSL (p. 99)
- Configuring logging (p. 102)
- Converting data types (p. 103)
- Using prepared statement support (p. 104)
- Differences between the 2.x and 1.x versions of the JDBC driver (p. 105)
- Creating initialization (.ini) files for JDBC driver version 2.0 (p. 105)
- Options for JDBC driver version 2.0 configuration (p. 106)
- Previous versions of JDBC driver version 2.0 (p. 121)

Download the Amazon Redshift JDBC driver, version 2.0

Amazon Redshift offers drivers for tools that are compatible with the JDBC 4.2 API. The class name for this driver is com.amazon.redshift.jdbc42.Driver.

For detailed information about how to install the JDBC driver, reference the JDBC driver libraries, and register the driver class, see the following topics.

For each computer where you use the Amazon Redshift JDBC driver version 2.0, make sure that the Java Runtime Environment (JRE) 8.0 is installed.

If you use the Amazon Redshift JDBC driver for database authentication, make sure that you have AWS SDK for Java 1.11.118 or later in your Java class path. If you don't have AWS SDK for Java installed, download the ZIP file with JDBC 4.2–compatible driver and driver dependent libraries for the AWS SDK:

- JDBC 4.2–compatible driver version 2.0 and AWS SDK driver–dependent libraries.

This ZIP file contains the JDBC 4.2–compatible driver version 2.0 and AWS SDK for Java 1.x driver–dependent library files. Unzip the dependent jar files to the same location as the JDBC driver. Only the JDBC driver needs to be in CLASSPATH.

This ZIP file doesn't include the complete AWS SDK for Java 1.x. However, it includes the AWS SDK for Java 1.x driver–dependent libraries that are required for AWS Identity and Access Management (IAM) database authentication.

Use this Amazon Redshift JDBC driver with the AWS SDK that is required for IAM database authentication.

To install the complete AWS SDK for Java 1.x, see AWS SDK for Java 1.x in the AWS SDK for Java Developer Guide.

- JDBC 4.2–compatible driver version 2.0 (without the AWS SDK).
Review the JDBC driver version 2.0 software license and change log file:

- JDBC driver version 2.0 license
- JDBC driver version 2.0 change log

JDBC drivers version 1.2.27.1051 and later support Amazon Redshift stored procedures. For more information, see Creating stored procedures in Amazon Redshift in the Amazon Redshift Database Developer Guide.

If your tool requires a specific previous version of a JDBC driver version 2.0, see Previous versions of JDBC driver version 2.0 (p. 121).

Installing the Amazon Redshift JDBC driver, version 2.0

To install the Amazon Redshift JDBC 4.2–compatible driver version 2.0 and driver–dependent libraries for AWS SDK, extract the files from the ZIP archive to the directory of your choice.

To install the Amazon Redshift JDBC 4.2–compatible driver version 2.0 (without the AWS SDK), copy the JAR file to the directory of your choice.

To access an Amazon Redshift data store using the Amazon Redshift JDBC driver, you need to perform configuration as described following.

Referencing the JDBC driver libraries

The JDBC application or Java code that you use to connect to your data must access the driver JAR files. In the application or code, specify all the JAR files that you extracted from the ZIP archive.

Using the driver in a JDBC application

JDBC applications usually provide a set of configuration options for adding a list of driver library files. Use the provided options to include all the JAR files from the ZIP archive as part of the driver configuration in the application. For more information, see the documentation for your JDBC application.

Using the driver in Java code

You must include all the driver library files in the class path. This is the path that the Java Runtime Environment searches for classes and other resource files. For more information, see the appropriate Java SE documentation to set the class path for your operating system.

- Windows: https://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- Linux and Solaris: https://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html

Registering the driver class

Make sure that you register the appropriate class for your application. You use following classes to connect the Amazon Redshift JDBC driver to Amazon Redshift data stores:

- Driver classes extend java.sql.Driver.
- DataSource classes extend javax.sql.DataSource and javax.sql.ConnectionPoolDataSource.

The driver supports the following fully qualified class names that are independent of the JDBC version:

- com.amazon.redshift.jdbc.Driver
• com.amazon.redshift.jdbc.DataSource

The following example shows how to use the DriverManager class to establish a connection for JDBC 4.2.

```java
private static Connection connectViaDM() throws Exception
{
    Connection connection = null;
    connection = DriverManager.getConnection(CONNECTION_URL);
    return connection;
}
```

The following example shows how to use the DataSource class to establish a connection.

```java
private static Connection connectViaDS() throws Exception
{
    Connection connection = null;
    DataSource ds = new com.amazon.redshift.jdbc.DataSource();
    ds.setURL(CONNECTION_URL);
    connection = ds.getConnection();
    return connection;
}
```

**Getting 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
```

The fields of the preceding format have the following values.

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

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

Be sure to enter the URL values, for example SessionToken values, in URL encoded format.

For information about how to get your JDBC connection, see Finding your cluster connection string (p. 91).
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. 162).

Building the connection URL

Use the connection URL to supply connection information to the data store that you are accessing. The following is the format of the connection URL for the Amazon Redshift JDBC driver version 2.0. Here, [Host] the endpoint of the Amazon Redshift server and [Port] is the number of the Transmission Control Protocol (TCP) port that the server uses to listen for client requests.

```
jdbc:redshift://[Host]:[Port]
```

By default, Amazon Redshift uses port 5439.

The following is the format of a connection URL that specifies some optional settings.

```
jdbc:redshift://[Host]:[Port]/[Schema];[Property1]=[Value];
[Property2]=[Value];
```

For example, suppose that you want to connect to port 9000 on an Amazon Redshift cluster in the US West (N. California) Region on AWS. You also want to access the schema named Default and authenticate the connection using a user name and password. In this case, you use the following connection URL.

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;UID=amazon;PWD=amazon
```

You can use the following characters to separate configuration options in the URL string:

- ;
- ?
- &

For example, the following URL strings are equivalent:

```
jdbc:redshift://my_host:5439/dev;ssl=false;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev?ssl=false?defaultRowFetchSize=100
```

The following URL example specifies a log level of 6 and the path for the logs.

```
jdbc:redshift://redshift.amazonaws.com:5439/dev;DSILogLevel=6;LogPath=/home/user/logs;
```

Don't duplicate properties in the connection URL.

For a complete list of the configuration options that you can specify, see Options for JDBC driver version 2.0 configuration (p. 106).

Configuring TCP keepalives for your JDBC connection

By default, the Amazon Redshift JDBC driver is configured to use TCP keepalives to prevent connections from timing out. You can specify when the driver starts sending keepalive packets or turn off the feature by setting the relevant properties in the connection URL. For more information about the syntax of the connection URL, see Building the connection URL (p. 96).
### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCPKeepAliveMinutes</td>
<td>To specify when the driver sends keepalive packets, set the number of minutes of inactivity before the driver starts sending TCP keepalive packets.</td>
</tr>
<tr>
<td>TCPKeepAlive</td>
<td>To turn off TCP keepalives, set this property to FALSE.</td>
</tr>
</tbody>
</table>

### Configuring your 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](#) in the AWS SDK for Java Developer Guide.

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, add the JDBC driver as a dependency in your project's `pom.xml` file. If you use Maven to build your project and want to use a JDBC connection, take the steps in the following section.

#### Configuring the JDBC driver as a Maven dependency

**To configure the JDBC driver as a Maven dependency**

1. Add either the Amazon repository or the Maven Central repository to the repositories section of your `pom.xml` file.

   **Note**
   
   The URL in the following code example returns an error if used in a browser. Use this URL only in the context of a Maven project.

   For an Amazon Maven repository, use the following.

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

   To connect using Secure Sockets Layer (SSL), add the following repository to your `pom.xml` file.

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

   For a Maven Central repository, add the following to your `pom.xml` file.

   ```xml
   <repositories>
     <repository>
       <id>redshift</id>
       <url>https://repo1.maven.org/maven2</url>
     </repository>
   </repositories>
   ```
2. Declare the version of the driver that you want to use in the dependencies section of your pom.xml file.

Amazon Redshift offers drivers for tools that are compatible with the JDBC 4.2 API. For information about the functionality supported by these drivers, see Download an Amazon Redshift JDBC driver (p. 122).

Add a dependency for the driver as shown following.

Replace \textit{driver-version} in the following example with your driver version, for example \texttt{2.0.0.7}.

For a JDBC 4.2–compatible driver, use the following.

\begin{verbatim}
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42</artifactId>
  <version>driver-version</version>
</dependency>
\end{verbatim}

The class name for this driver is \texttt{com.amazon.redshift.jdbc42.Driver}.

The Amazon Redshift Maven drivers need the following optional dependencies when you use IAM database authentication.

\begin{verbatim}
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-core</artifactId>
  <version>1.12.23</version>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-redshift</artifactId>
  <version>1.12.23</version>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-sts</artifactId>
  <version>1.12.23</version>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>
\end{verbatim}

If your tool requires a specific previous version of a driver, see \textit{Use previous JDBC driver versions with Maven} (p. 130).

\textbf{Upgrading the driver to the latest version}

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

\texttt{mvn clean}
Configuring authentication and SSL

To protect data from unauthorized access, Amazon Redshift data stores require all connections to be authenticated using user credentials. Some data stores also require connections to be made over the Secure Sockets Layer (SSL) protocol, either with or without one-way authentication.

The Amazon Redshift JDBC driver version 2.0 provides full support for these authentication protocols.

The SSL version that the driver supports depends on the JVM version that you are using. For information about the SSL versions that are supported by each version of Java, see Diagnosing TLS, SSL, and HTTPS on the Java Platform Group Product Management Blog.

The SSL version used for the connection is the highest version that is supported by both the driver and the server, which is determined at connection time.

Configure the Amazon Redshift JDBC driver version 2.0 to authenticate your connection according to the security requirements of the Amazon Redshift server that you are connecting to.

You must always provide your Amazon Redshift user name and password to authenticate the connection. Depending on whether SSL is enabled and required on the server, you might also need to configure the driver to connect through SSL. Or you might use one-way SSL authentication so that the client (the driver itself) verifies the identity of the server.

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the connection URL (p. 96).

SSL indicates TLS/SSL, both Transport Layer Security and Secure Sockets Layer. The driver supports industry-standard versions of TLS/SSL.

Using user name and password only

If the server you are connecting to doesn't use SSL, then you only need to provide your user name and password to authenticate the connection.

To configure authentication using your user name and password only

1. Set the UID property to your user name for accessing the Amazon Redshift server.
2. Set the PWD property to the password corresponding to your user name.

Using SSL without identity verification

If the server you are connecting to uses SSL but doesn't require identity verification, then you can configure the driver to use a non-validating SSL factory.

To configure an SSL connection without identity verification

1. Set the UID property to your user name for accessing the Amazon Redshift server.
2. Set the PWD property to the password corresponding to your user name.

Using one-way SSL authentication

If the server you are connecting to uses SSL and has a certificate, then you can configure the driver to verify the identity of the server using one-way authentication.

One-way authentication requires a signed, trusted SSL certificate for verifying the identity of the server. You can configure the driver to use a specific certificate or access a TrustStore that contains the appropriate certificate. If you don't specify a certificate or TrustStore, then the driver uses the default Java TrustStore (typically either jssecacerts or cacerts).
To configure one-way SSL authentication

1. Set the UID property to your user name for accessing the Amazon Redshift server.
2. Set the PWD property to the password corresponding to your user name.
3. Set the SSL property to true.
4. Set the SSLRootCert property to the location of your root CA certificate.
5. If you aren’t using one of the default Java TrustStores, then do one of the following:
   - To specify a server certificate, set the SSLRootCert property to the full path of the certificate.
   - To specify a TrustStore, do the following:
     a. Use the keytool program to add the server certificate to the TrustStore that you want to use.
     b. Specify the TrustStore and password to use when starting the Java application using the driver. For example:

        ```
        -Djavax.net.ssl.trustStore=[TrustStoreName]
        -Djavax.net.ssl.trustStorePassword=[TrustStorePassword]
        ```

6. Choose one:
   - To validate the certificate, set the SSLMode property to verify-ca.
   - To validate the certificate and verify the host name in the certificate, set the SSLMode property to verify-full.

Configuring IAM authentication

If you are connecting to a Amazon Redshift server using IAM authentication, set the following properties as part of your data source connection string.

For more information on IAM authentication, see Identity and access management in Amazon Redshift (p. 287).

To use IAM authentication, use one of the following connection string formats:

<table>
<thead>
<tr>
<th>Connection string</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jdbc:redshift:iam:// [host]:[port]/[db]</td>
<td>A regular connection string. The driver infers the ClusterID and Region from the host.</td>
</tr>
<tr>
<td>jdbc:redshift:iam:// [cluster-id]:[region]:[db]</td>
<td>The driver retrieves host information, given the ClusterID and Region.</td>
</tr>
<tr>
<td>jdbc:redshift:iam:// [host]:[db]</td>
<td>The driver defaults to port 5439, and infers ClusterID and Region from the host.</td>
</tr>
</tbody>
</table>

Specifying profiles

If you are using IAM authentication, you can specify any additional required or optional connection properties under a profile name. By doing this, you can avoid putting certain information directly in the connection string. You specify the profile name in your connection string using the Profile property.

Profiles can be added to the AWS credentials file. The default location for this file is: ~/.aws/credentials
You can change the default value by setting the path in the following environment variable:

AWS_CREDENTIAL_PROFILES_FILE

For more information about profiles, see Working with AWS Credentials in the AWS SDK for Java.

**Using instance profile credentials**

If you are running an application on an Amazon EC2 instance that is associated with an IAM role, you can connect using the instance profile credentials.

To do this, use one of the IAM connection string formats in the preceding table, and set the dbuser connection property to the Amazon Redshift user name that you are connecting as.

For more information about instance profiles, see Access Management in the IAM User Guide.

**Using credential providers**

The driver also supports credential provider plugins from the following services:

- Active Directory Federation Service (ADFS)
- JSON Web Tokens (JWT) Service
- Microsoft Azure Active Directory (AD) Service and Browser Microsoft Azure Active Directory (AD) Service
- Okta Service
- PingFederate Service
- Browser SAML for SAML services such as Okta, Ping, or ADFS

If you use one of these services, the connection URL needs to specify the following properties:

- **Plugin_Name** – The fully-qualified class path for your credentials provider plugin class.
- **IdP_Host** – The host for the service that you are using to authenticate into Amazon Redshift.
- **IdP_Port** – The port that the host for the authentication service listens at. Not required for Okta.
- **User** – The user name for the idp_host server.
- **Password** – The password associated with the idp_host user name.
- **DbUser** – The Amazon Redshift user name you are connecting as.
- **SSL_Insecure** – Indicates whether the IDP server certificate should be verified.
- **Client_ID** – The client ID associated with the user name in the Azure AD portal. Only used for Azure AD.
- **Client_Secret** – The client secret associated with the client ID in the Azure AD portal. Only used for Azure AD.
- **IdP_Tenant** – The Azure AD tenant ID for your Amazon Redshift application. Only used for Azure AD.
- **App_ID** – The Okta app ID for your Amazon Redshift application. Only used for Okta.
- **App_Name** – The optional Okta app name for your Amazon Redshift application. Only used for Okta.
- **Partner_SPID** – The optional partner SPID (service provider ID) value. Only used for PingFederate.

If you are using a browser plugin for one of these services, the connection URL can also include:

- **Login_URL** – The URL for the resource on the identity provider's website when using the Security Assertion Markup Language (SAML) or Azure AD services through a browser plugin. This parameter is required if you are using a browser plugin.

- **Listen_Port** – The port that the driver uses to get the SAML response from the identity provider when using the SAML or Azure AD services through a browser plugin.
• **IdP_Response_Timeout** – The amount of time, in seconds, that the driver waits for the SAML response from the identity provider when using the SAML or Azure AD services through a browser plugin.

For information on additional connection string properties, see Options for JDBC driver version 2.0 configuration (p. 106).

### Configuring logging

You can turn on logging in the driver to assist in diagnosing issues.

You can log driver information by using the following methods:

- To save logged information in .log files, see Using log files (p. 102).
- To send logged information to the LogStream or LogWriter specified in the DriverManager, see Using LogStream or LogWriter (p. 103).

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the connection URL (p. 96).

### Using log files

Only turn on logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

Set the LogLevel key in your connection URL to turn on logging and specify the amount of detail included in log files. The following table lists the logging levels provided by the Amazon Redshift JDBC driver version 2.0, in order from least verbose to most verbose.

<table>
<thead>
<tr>
<th>LogLevel value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log severe error events that will lead the driver to abort.</td>
</tr>
<tr>
<td>2</td>
<td>Log error events that might allow the driver to continue running.</td>
</tr>
<tr>
<td>3</td>
<td>Log events that might result in an error if action is not taken.</td>
</tr>
<tr>
<td>4</td>
<td>Log general information that describes the progress of the driver.</td>
</tr>
<tr>
<td>5</td>
<td>Log detailed information that is useful for debugging the driver.</td>
</tr>
<tr>
<td>6</td>
<td>Log all driver activity.</td>
</tr>
</tbody>
</table>

**To set up logging that uses log files**

1. Set the LogLevel property to the desired level of information to include in log files.
2. Set the LogPath property to the full path to the folder where you want to save log files.

For example, the following connection URL enables logging level 3 and saves the log files in the C:\temp folder: jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;DSILogLevel=3; LogPath=C:\temp

3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.
The Amazon Redshift JDBC driver produces the following log files in the location specified in the LogPath property:

- redshift_jdbc.log file that logs driver activity that is not specific to a connection.
- redshift_jdbc_connection_[Number].log file for each connection made to the database, where [Number] is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If the LogPath value is invalid, then the driver sends the logged information to the standard output stream (System.out)

**Using LogStream or LogWriter**

Only turn on logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

Set the LogLevel key in your connection URL to turn on logging and specify the amount of detail sent to the LogStream or LogWriter specified in the DriverManager.

**To turn on logging that uses the LogStream or LogWriter:**

1. To configure the driver to log general information that describes the progress of the driver, set the LogLevel property to 1 or INFO.
2. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

**To turn off logging that uses the LogStream or LogWriter:**

1. Remove the LogLevel property from the connection URL.
2. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

**Converting data types**

The Amazon Redshift JDBC driver version 2.0 supports many common data formats, converting between Amazon Redshift, SQL, and Java data types.

The following table lists the supported data type mappings.

<table>
<thead>
<tr>
<th>Amazon Redshift type</th>
<th>SQL type</th>
<th>Java type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIGINT</td>
<td>SQL_BIGINT</td>
<td>Long</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>SQL_BIT</td>
<td>Boolean</td>
</tr>
<tr>
<td>CHAR</td>
<td>SQL_CHAR</td>
<td>String</td>
</tr>
<tr>
<td>DATE</td>
<td>SQL_TYPE_DATE</td>
<td>java.sql.Date</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>SQL_NUMERIC</td>
<td>BigDecimal</td>
</tr>
<tr>
<td>DOUBLE PRECISION</td>
<td>SQL_DOUBLE</td>
<td>Double</td>
</tr>
<tr>
<td>GEOMETRY</td>
<td>SQL_LONGVARBINARY</td>
<td>byte[]</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SQL_INTEGER</td>
<td>Integer</td>
</tr>
</tbody>
</table>
Amazon Redshift Cluster Management Guide
Configuring connections in Amazon Redshift

<table>
<thead>
<tr>
<th>Amazon Redshift type</th>
<th>SQL type</th>
<th>Java type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OID</td>
<td>SQL_BIGINT</td>
<td>Long</td>
</tr>
<tr>
<td>SUPER</td>
<td>SQL_LONGVARCHAR</td>
<td>String</td>
</tr>
<tr>
<td>REAL</td>
<td>SQL_REAL</td>
<td>Float</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SQL_SMALLINT</td>
<td>Short</td>
</tr>
<tr>
<td>TEXT</td>
<td>SQL_VARCHAR</td>
<td>String</td>
</tr>
<tr>
<td>TIME</td>
<td>SQL_TYPE_TIME</td>
<td>java.sql.Time</td>
</tr>
<tr>
<td>TIMETZ</td>
<td>SQL_TYPE_TIME</td>
<td>java.sql.Time</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>SQL_TYPE_TIMESTAMP</td>
<td>java.sql.Timestamp</td>
</tr>
<tr>
<td>TIMESTAMPTZ</td>
<td>SQL_TYPE_TIMESTAMP</td>
<td>java.sql.Timestamp</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>SQL_VARCHAR</td>
<td>String</td>
</tr>
</tbody>
</table>

Using prepared statement support

The Amazon Redshift JDBC driver supports prepared statements. You can use prepared statements to improve the performance of parameterized queries that need to be run multiple times during the same connection.

A prepared statement is a SQL statement that is compiled on the server side but not executed immediately. The compiled statement is stored on the server as a PreparedStatement object until you close the object or the connection. While that object exists, you can execute the prepared statement as many times as needed using different parameter values, without having to compile the statement again. This reduced overhead enables the set of queries to be executed more quickly.

For more information about prepared statements, see "Using Prepared Statements" in JDBC Basics tutorial from Oracle.

You can prepare a statement that contains multiple queries. For example, the following prepared statement contains two INSERT queries:

```java
PreparedStatement pstmt = conn.prepareStatement("INSERT INTO MyTable VALUES (1, 'abc'); INSERT INTO CompanyTable VALUES (1, 'abc');");
```

Take care that these queries don't depend on the results of other queries that are specified within the same prepared statement. Because queries don't run during the prepare step, the results have not been returned yet, and aren't available to other queries in the same prepared statement.

For example, the following prepared statement, which creates a table and then inserts values into that newly-created table, is not allowed:

```java
PreparedStatement pstmt = conn.prepareStatement("CREATE TABLE MyTable(col1 int, col2 varchar); INSERT INTO myTable VALUES (1, 'abc');");
```

If you try to prepare this statement, the server returns an error stating that the destination table (myTable) doesn't exist yet. The CREATE query must be executed before the INSERT query can be prepared.
Differences between the 2.x and 1.x versions of the JDBC driver

This section describes the differences in the information returned by the 2.x and 1.x versions of the JDBC driver.

The following table lists the DatabaseMetadata information returned by the getDatabaseProductName() and getDatabaseProductVersion() functions for each version of the JDBC driver. JDBC driver version 2.x obtains the values while establishing the connection. JDBC driver version 1.x obtains the values as a result of a query.

<table>
<thead>
<tr>
<th>JDBC driver version</th>
<th>getDatabaseProductName() result</th>
<th>getDatabaseProductVersion() result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.x</td>
<td>Redshift</td>
<td>8.0.2</td>
</tr>
<tr>
<td>1.x</td>
<td>PostgreSQL</td>
<td>08.00.0002</td>
</tr>
</tbody>
</table>

The following table lists the DatabaseMetadata information returned by the getTypeInfo function for each version of the JDBC driver.

<table>
<thead>
<tr>
<th>JDBC driver version</th>
<th>getTypeInfo result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.x</td>
<td>Consistent with Redshift datatypes</td>
</tr>
<tr>
<td>1.x</td>
<td>Consistent with PostgreSQL datatypes</td>
</tr>
</tbody>
</table>

Creating initialization (.ini) files for JDBC driver version 2.0

By using initialization (.ini) files for Amazon Redshift JDBC driver version 2.0, you can specify system level configuration parameters. For example, federated IdP authentication parameters can vary for each application. The .ini file provides a common location for SQL clients to get the required configuration parameters.

You can create an JDBC driver version 2.0 initialization (.ini) file that contains configuration options for SQL clients. The default name of the file is rsjdbc.ini. The JDBC driver version 2.0 checks for the .ini file in the following locations, listed in order of precedence:

- IniFile parameter in the connection URL or in the connection property dialog box of the SQL client. Be sure that the IniFile parameter contains the full path to the .ini file, including the file name. For information about the IniFile parameter, see IniFile (p. 113). If the IniFile parameter incorrectly specifies the location of the .ini file, an error displays.
- Environment variables such as AMAZON_REDSHIFT_JDBC_INI_FILE with the full path, including the file name. You can use rsjdbc.ini or specify a file name. If the AMAZON_REDSHIFT_JDBC_INI_FILE environment variable incorrectly specifies the location of the .ini file, an error displays.
- Directory where the driver JAR file is located.
- User home directory.
- Temp directory of the system.

You can organize the .ini file into sections, for example [DRIVER]. Each section contains key-value pairs that specify various connection parameters. You can use the IniSection parameter to specify a section in the .ini file. For information about the IniSection parameter, see IniSection (p. 113).

Following is an example of the .ini file format, with sections for [DRIVER], [DEV], [QA], and [PROD]. The [DRIVER] section can apply to any connection.
The JDBC driver version 2.0 loads configuration parameters from the following locations, listed in order of precedence:

- Default configuration parameters in the application code.
- [DRIVER] section properties from the .ini file, if included.
- Custom section configuration parameters, if the IniSection option is provided in the connection URL or in the connection property dialog box of the SQL client.
- Properties from the connection property object specified in the getConnection call.
- Configuration parameters specified in the connection URL.

### Options for JDBC driver version 2.0 configuration

Following, you can find descriptions for the options that you can specify for version 2.0 of the Amazon Redshift JDBC driver.

You can set configuration properties using the connection URL. For more information, see Building the connection URL (p. 96).

**Topics**

- AccessKeyId (p. 107)
- AllowDBUserOverride (p. 108)
- App_ID (p. 108)
- App_Name (p. 108)
- AuthProfile (p. 108)
- AutoCreate (p. 109)
- Client_ID (p. 109)
- Client_Secret (p. 109)
- ClusterID (p. 109)
- databaseMetadataCurrentDbOnly (p. 109)
- DbUser (p. 110)
- DbGroups (p. 110)
- defaultRowFetchSize (p. 110)
- DisableIsValidQuery (p. 110)
- enableFetchRingBuffer (p. 111)
- enableMultiSqlSupport (p. 111)
- fetchRingBufferSize (p. 111)
• ForceLowercase (p. 112)
• IAMDisableCache (p. 112)
• IAMDuration (p. 112)
• IdP_Host (p. 112)
• IdP_Port (p. 113)
• IdP_Tenant (p. 113)
• IdP_Response_Timeout (p. 113)
• IniFile (p. 113)
• IniSection (p. 113)
• Login_URL (p. 114)
• LoginTimeout (p. 114)
• loginToRp (p. 114)
• LogLevel (p. 114)
• LogPath (p. 115)
• Partner_SPID (p. 115)
• Password (p. 116)
• Plugin_Name (p. 116)
• Preferred_Role (p. 116)
• Profile (p. 116)
• PWD (p. 116)
• queryGroup (p. 117)
• ReadOnly (p. 117)
• Region (p. 117)
• reWriteBatchedInsertsSize (p. 117)
• roleArn (p. 118)
• roleSessionName (p. 118)
• SecretAccessKey (p. 118)
• SessionToken (p. 118)
• SocketTimeout (p. 118)
• SSL (p. 118)
• SSL_Insecure (p. 119)
• SSLCert (p. 119)
• SSLFactory (p. 119)
• SSLKey (p. 119)
• SSLMode (p. 120)
• SSLPassword (p. 120)
• SSLRootCert (p. 120)
• StsEndpointUrl (p. 120)
• TCPKeepAlive (p. 120)
• UID (p. 121)
• User (p. 121)
• webIdentityToken (p. 121)

AccessKeyId

• Default Value – None
• **Data Type** – String

You can specify this parameter to enter the IAM access key for the IAM user or role. You can usually locate the key by looking at an existing string or user profile. If you specify this parameter, you must also specify the `SecretAccessKey` parameter.

This parameter is optional.

**AllowDBUserOverride**

• **Default Value** – 0
• **Data Type** – String

This option specifies whether the driver uses the `DbUser` value from the SAML assertion or the value that is specified in the `DbUser` connection property in the connection URL.

This parameter is optional.

1

The driver uses the `DbUser` value from the SAML assertion.

If the SAML assertion doesn’t specify a value for `DbUser`, the driver uses the value specified in the `DbUser` connection property. If the connection property also doesn’t specify a value, the driver uses the value specified in the connection profile.

0

The driver uses the `DbUser` value specified in the `DbUser` connection property.

If the `DbUser` connection property doesn’t specify a value, the driver uses the value specified in the connection profile. If the connection profile also doesn’t specify a value, the driver uses the value from the SAML assertion.

**App_ID**

• **Default Value** – None
• **Data Type** – String

The Okta-provided unique ID associated with your Amazon Redshift application.

This parameter is required if authenticating through the Okta service.

**App_Name**

• **Default Value** – None
• **Data Type** – String

The name of the Okta application that you use to authenticate the connection to Amazon Redshift.

This parameter is optional.

**AuthProfile**

• **Default Value** – None
• **Data Type** – String
The name of the authentication profile to use for connecting to Amazon Redshift.
This parameter is optional.

**AutoCreate**
- **Default Value** – false
- **Data Type** – Boolean

This option specifies whether the driver causes a new user to be created when the specified user doesn't exist.
This parameter is optional.

**true**
If the user specified by either DBUser or unique ID (UID) doesn't exist, a new user with that name is created.

**false**
The driver doesn't cause new users to be created. If the specified user doesn't exist, the authentication fails.

**Client_ID**
- **Default Value** – None
- **Data Type** – String

The client ID to use when authenticating the connection using the Azure AD service.
This parameter is required if authenticating through the Azure AD service.

**Client_Secret**
- **Default Value** – None
- **Data Type** – String

The Client Secret to use when authenticating the connection using the Azure AD service.
This parameter is required if authenticating through the Azure AD service.

**ClusterID**
- **Default Value** – None
- **Data Type** – String

The name of the Amazon Redshift cluster that you want to connect to.
This parameter is optional.

**databaseMetadataCurrentDbOnly**
- **Default Value** – true
- **Data Type** – Boolean
This option specifies whether the metadata API retrieves data from all accessible databases or only from the connected database.

This parameter is optional.

You can specify the following values:

**true**

The application retrieves metadata from a single database.

**false**

The application retrieves metadata from all accessible databases.

**DbUser**

- **Default Value** – None
- **Data Type** – String

The user ID to use with your Amazon Redshift account. You can use an ID that doesn't currently exist if you have enabled the AutoCreate property.

This parameter is optional.

**DbGroups**

- **Default Value** – PUBLIC
- **Data Type** – String

A comma-separated list of existing database group names that `DbUser` joins for the current session.

This parameter is optional.

**defaultRowFetchSize**

- **Default Value** – 0
- **Data Type** – Integer

This option specifies a default value for `getFetchSize`.

This parameter is optional.

You can specify the following values:

**0**

Fetch all rows in a single operation.

**Positive integer**

Number of rows to fetch from the database for each fetch iteration of the ResultSet.

**DisableIsValidQuery**

- **Default Value** – False
- **Data Type** – Boolean
This option specifies whether the driver submits a new database query when using the Connection.isValid() method to determine whether the database connection is active.

This parameter is optional.

true

The driver doesn't submit a query when using Connection.isValid() to determine whether the database connection is active. This may cause the driver to incorrectly identify the database connection as active if the database server has shut down unexpectedly.

false

The driver submits a query when using Connection.isValid() to determine whether the database connection is active.

enableFetchRingBuffer

- Default Value – true
- Data Type – Boolean

This option specifies that the driver fetches rows using a ring buffer on a separate thread. The fetchRingBufferSize parameter specifies the ring buffer size.

This parameter is optional.

enableMultiSqlSupport

- Default Value – true
- Data Type – Boolean

This option specifies whether to process multiple SQL commands separated by semicolons in a Statement.

This parameter is optional.

You can specify the following values:

true

The driver processes multiple SQL commands, separated by semicolons, in a Statement object.

false

The driver returns an error for multiple SQL commands in a single Statement.

fetchRingBufferSize

- Default Value – 1G
- Data Type – String

This option specifies the size of the ring buffer used while fetching the result set. You can specify a size in bytes, for example 1K for 1 KB, 5000 for 5,000 bytes, 1M for 1 MB, 1G for 1 GB, and so on. You can also specify a percentage of heap memory. The driver stops fetching rows upon reaching the limit. Fetching resumes when the application reads rows and frees space in the ring buffer.

This parameter is optional.
ForceLowercase

- **Default Value** – false
- **Data Type** – Boolean

This option specifies whether the driver lowercases all database groups (DbGroups) sent from the identity provider to Amazon Redshift when using SSO authentication.

This parameter is optional.

**true**

The driver lowercases all database groups that are sent from the identity provider.

**false**

The driver doesn't alter database groups.

IAMDisableCache

- **Default Value** – false
- **Data Type** – Boolean

This option specifies whether the IAM credentials are cached.

This parameter is optional.

**true**

The IAM credentials aren't cached.

**false**

The IAM credentials are cached. This improves performance when requests to the API gateway are throttled, for instance.

IAMDuration

- **Default Value** – 900
- **Data Type** – Integer

The length of time, in seconds, until the temporary IAM credentials expire.

- **Minimum value** – 900
- **Maximum value** – 3,600

This parameter is optional.

IdP_Host

- **Default Value** – None
- **Data Type** – String

The IdP (identity provider) host you are using to authenticate into Amazon Redshift. This can be specified in either the connection string or in a profile.
This parameter is optional.

**IdP_Port**

- **Default Value** – None
- **Data Type** – String

The port used by an IdP (identity provider). This can be specified in either the connection string or in a profile.

This parameter is optional.

**IdP_Tenant**

- **Default Value** – None
- **Data Type** – String

The Azure AD tenant ID for your Amazon Redshift application.

This parameter is required if authenticating through the Azure AD service.

**IdP_Response_Timeout**

- **Default Value** – 120
- **Data Type** – Integer

The amount of time, in seconds, that the driver waits for the SAML response from the identity provider when using the SAML or Azure AD services through a browser plugin.

This parameter is optional.

**IniFile**

- **Default Value** – None
- **Data Type** – String

The full path of the .ini file, including file name. For example:

```
IniFile=C:\tools\rsjdbc.ini
```

For information about the .ini file, see Creating initialization (.ini) files for JDBC driver version 2.0 (p. 105).

This parameter is optional.

**IniSection**

- **Default Value** – None
- **Data Type** – String

The name of a section in the .ini file containing the configuration options. For information about the .ini file, see Creating initialization (.ini) files for JDBC driver version 2.0 (p. 105).

The following example specifies the [Prod] section of the .ini file:
IniSection="Prod"

This parameter is optional.

**Login_URL**

- **Default Value** – None
- **Data Type** – String

The URL for the resource on the identity provider's website when using the SAML or Azure AD services through a browser plugin.

This parameter is required if authenticating with the SAML or Azure AD services through a browser plugin.

**LoginTimeout**

- **Default Value** – 0
- **Data Type** – Integer

The number of seconds to wait before timing out when connecting and authenticating to the server. If establishing the connection takes longer than this threshold, then the connection is aborted.

When this property is set to 0, connections don't time out.

This parameter is optional.

**loginToRp**

- **Default Value** – urn:amazon:webservices
- **Data Type** – String

The relying party trust that you want to use for the AD FS authentication type.

This parameter is optional.

**LogLevel**

- **Default Value** – 0
- **Data Type** – Integer

Use this property to turn on or turn off logging in the driver and to specify the amount of detail included in log files.

Enable logging only long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

This parameter is optional.

Set the parameter to one of the following values:

- **0**
  - Disable all logging.
Enable logging on the FATAL level, which logs very severe error events that will lead the driver to abort.

Enable logging on the ERROR level, which logs error events that might still allow the driver to continue running.

Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.

Enable logging on the INFO level, which logs general information that describes the progress of the driver.

Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the driver.

Enable logging on the TRACE level, which logs all driver activity.

When logging is enabled, the driver produces the following log files in the location specified in the LogPath property:

- **redshift_jdbc.log** – File that logs driver activity that is not specific to a connection.
- **redshift_jdbc_connection_[Number].log** – File for each connection made to the database, where [Number] is a number that distinguishes each log file from the others. This file logs driver activity that is specific to the connection.

If the LogPath value is invalid, the driver sends the logged information to the standard output stream, `System.out`.

**LogPath**

- **Default Value** – The current working directory.
- **Data Type** – String

The full path to the folder where the driver saves log files when the DSILogLevel property is enabled.

To be sure that the connection URL is compatible with all JDBC applications, we recommend that you escape the backslashes (\) in your file path by typing another backslash.

This parameter is optional.

**Partner_SPID**

- **Default Value** – None
- **Data Type** – String

The partner SPID (service provider ID) value to use when authenticating the connection using the PingFederate service.
This parameter is optional.

Password

- **Default Value** – None
- **Data Type** – String

When connecting using IAM authentication through an IDP, this is the password for the IDP_Host server. When using standard authentication, this can be used for the Amazon Redshift database password instead of PWD.

This parameter is optional.

Plugin_Name

- **Default Value** – None
- **Data Type** – String

The fully qualified class name to implement a specific credentials provider plugin.

This parameter is optional.

The following provider options are supported:

- **AdfsCredentialsProvider** – Active Directory Federation Service
- **AzureCredentialsProvider** – Microsoft Azure Active Directory (AD) Service
- **BasicJwtCredentialsProvider** – JSON Web Tokens (JWT) Service
- **BrowserAzureCredentialsProvider** – Browser Microsoft Azure Active Directory (AD) Service
- **BrowserSamlCredentialsProvider** – Browser SAML for SAML services such as Okta, Ping, or ADFS
- **OktaCredentialsProvider** – Okta Service
- **PingCredentialsProvider** – PingFederate Service

Preferred_Role

- **Default Value** – None
- **Data Type** – String

The IAM role that you want to assume during the connection to Amazon Redshift.

This parameter is optional.

Profile

- **Default Value** – None
- **Data Type** – String

The name of the profile to use for IAM authentication. This profile contains any additional connection properties not specified in the connection string.

This parameter is optional.

PWD

- **Default Value** – None
• **Data Type** – String

The password corresponding to the Amazon Redshift user name that you provided using the property UID.

This parameter is optional.

**queryGroup**

• **Default Value** – null
• **Data Type** – String

This option assigns a query to a queue at runtime by assigning your query to the appropriate query group. The query group is set for the session. All queries that execute on the connection belong to this query group.

This parameter is optional.

**ReadOnly**

• **Default Value** – false
• **Data Type** – Boolean

This property specifies whether the driver is in read-only mode.

This parameter is optional.

**true**

  The connection is in read-only mode and cannot write to the data store.

**false**

  The connection is not in read-only mode and can write to the data store.

**Region**

• **Default Value** – null
• **Data Type** – String

This option specifies the AWS Region where the cluster is located. If you specify the StsEndPoint option, the Region option is ignored. The Redshift `GetClusterCredentials` API operation also uses the Region option.

This parameter is optional.

**reWriteBatchedInsertsSize**

• **Default Value** – 128
• **Data Type** – Integer

This option enables optimization to rewrite and combine compatible INSERT statements that are batched. This value must increase exponentially by the power of 2.

This parameter is optional.
roleArn

- **Default Value** – None
- **Data Type** – String

The Amazon Resource Name (ARN) of role. Make sure to specify this parameter when you specify BasicJwtCredentialsProvider for the Plugin_Name option. You specify the ARN in the following format:

`arn:partition:service:region:account-id:resource-id`

This parameter is required if you specify BasicJwtCredentialsProvider for the Plugin_Name option.

roleSessionName

- **Default Value** – jwt_redshift_session
- **Data Type** – String

An identifier for the assumed role session. Typically, you pass the name or identifier that is associated with the user of your application. The temporary security credentials that your application uses are associated with that user. You can specify this parameter when you specify BasicJwtCredentialsProvider for the Plugin_Name option.

This parameter is optional.

SecretAccessKey

- **Default Value** – None
- **Data Type** – String

The IAM access key for the user or role. If this is specified, then AccessKeyId must also be specified.

This parameter is optional.

SessionToken

- **Default Value** – None
- **Data Type** – String

The temporary IAM session token associated with the IAM role you are using to authenticate.

This parameter is optional.

SocketTimeout

- **Default Value** – 0
- **Data Type** – Integer

The number of seconds to wait during socket read operations before timing out. If the operation takes longer than this threshold, then the connection is closed. When this property is set to 0, the connection doesn't time out.

This parameter is optional.

SSL

- **Default Value** – TRUE
• **Data Type** – String

Use this property to turn on or turn off SSL for the connection.

This parameter is optional.

You can specify the following values:

**TRUE**

The driver connects to the server through SSL.

**FALSE**

The driver connects to the server without using SSL. This option is not supported with IAM authentication.

Alternatively, you can configure the AuthMech property.

**SSL_Insecure**

• **Default Value** – true

• **Data Type** – String

This property indicates whether the IDP hosts server certificate should be verified.

This parameter is optional.

You can specify the following values:

**true**

The driver doesn't check the authenticity of the IDP server certificate.

**false**

The driver checks the authenticity of the IDP server certificate.

**SSLCert**

• **Default Value** – None

• **Data Type** – String

The full path of a .pem or .crt file containing additional trusted CA certificates for verifying the Amazon Redshift server instance when using SSL.

This parameter is required if SSLKey is specified.

**SSLFactory**

• **Default Value** – None

• **Data Type** – String

The SSL factory to use when connecting to the server through TLS/SSL without using a server certificate.

**SSLKey**

• **Default Value** – None
• **Data Type** – String

The full path of the .der file containing the PKCS8 key file for verifying the certificates specified in SSLCert.

This parameter is required if SSLCert is specified.

**SSLMode**

• **Default Value** – verify-ca

• **Data Type** – String

Use this property to specify how the driver validates certificates when TLS/SSL is enabled.

This parameter is optional.

You can specify the following values:

**verify-ca**

The driver verifies that the certificate comes from a trusted certificate authority (CA).

**verify-full**

The driver verifies that the certificate comes from a trusted CA and that the host name in the certificate matches the host name specified in the connection URL.

**SSLPassword**

• **Default Value** – 0

• **Data Type** – String

The password for the encrypted key file specified in SSLKey.

This parameter is required if SSLKey is specified and the key file is encrypted.

**SSLRootCert**

• **Default Value** – None

• **Data Type** – String

The full path of a .pem or .crt file containing the root CA certificate for verifying the Amazon Redshift Server instance when using SSL.

**StsEndpointUrl**

• **Default Value** – Null

• **Data Type** – String

You can specify an AWS Security Token Service (AWS STS) endpoint. If you specify this option, the Region option is ignored. You can only specify a secure protocol (HTTPS) for this endpoint.

**TCPKeepAlive**

• **Default Value** – TRUE
• **Data Type** – String

Use this property to turn on or turn off TCP keepalives.

This parameter is optional.

You can specify the following values:

**TRUE**

The driver uses TCP keepalives to prevent connections from timing out.

**FALSE**

The driver doesn’t use TCP keepalives.

**UID**

• **Default Value** – None
• **Data Type** – String

The user name that you use to access the database.

This parameter is required.

**User**

• **Default Value** – None
• **Data Type** – String

When connecting using IAM authentication through an IDP, this is the user name for the idp_host server. When using standard authentication this can be used for the Amazon Redshift database user name.

This parameter is optional.

**webIdentityToken**

• **Default Value** – None
• **Data Type** – String

The OAuth 2.0 access token or OpenID Connect ID token that is provided by the identity provider. Your application must get this token by authenticating the user of your application with a web identity provider. Make sure to specify this parameter when you specify BasicJwtCredentialsProvider for the Plugin_Name option.

This parameter is required if you specify BasicJwtCredentialsProvider for the Plugin_Name option.

**Previous versions of JDBC driver version 2.0**

Download a previous version of the Amazon Redshift JDBC driver version 2.0 only if your tool requires a specific version of the driver.

These are JDBC 4.2–compatible JDBC driver version 2.0 drivers:

• [https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.0.0.7/redshift-jdbc42-2.0.0.7.zip](https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.0.0.7/redshift-jdbc42-2.0.0.7.zip).
Configuring connections in Amazon Redshift

Download an Amazon Redshift JDBC driver

Amazon Redshift offers drivers for tools that are compatible with the JDBC 4.2 API. For information about the functionality supported by these drivers, see the Amazon Redshift JDBC driver release notes.

For detailed information about how to install the JDBC driver version 1.0, reference the JDBC driver libraries, and register the driver class, see Amazon Redshift JDBC driver installation and configuration guide.

For each computer where you use the Amazon Redshift JDBC driver, make sure that Java Runtime Environment (JRE) 8.0 is installed.

If you use the Amazon Redshift JDBC driver for database authentication, make sure that you have AWS SDK for Java 1.11.118 or later in your Java class path. If you don’t have AWS SDK for Java installed, download the ZIP file with JDBC 4.2–compatible driver (without the AWS SDK) and driver dependent libraries for the AWS SDK:

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.0.0.2/redshift-jdbc42-2.0.0.2.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.0.0.0/redshift-jdbc42-2.0.0.0.zip.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.0.0.0/redshift-jdbc42-2.0.0.0.jar.

Configuring a JDBC driver version 1.0 connection

You can use a JDBC driver version 1.0 connection to connect to your Amazon Redshift cluster from many third-party SQL client tools. To do this, you download a JDBC driver.

If you want to use a JDBC connection, perform the following steps.

Topics
- Download an Amazon Redshift JDBC driver (p. 122)
- Getting the JDBC URL (p. 123)
- Configure authentication and SSL for JDBC connection (p. 124)
- Configure TCP keepalives for JDBC connection (p. 124)
- Configure logging for JDBC connection (p. 124)
- Configure JDBC connection with Apache Maven (p. 124)
- Configure JDBC driver options (p. 126)
- Use previous JDBC driver version 1.0 driver versions in certain cases (p. 127)
• JDBC 4.2–compatible driver (without the AWS SDK) and driver dependent libraries for AWS SDK files version 1.2.55.

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

This ZIP file contains the JDBC4.2–compatible driver (without the AWS SDK) and its dependent library files. Unzip the dependent jar files to the same location as the JDBC driver. Only the JDBC driver needs to be in the CLASSPATH because the driver manifest file contains all dependent library file names which are located in the same directory as the JDBC driver. For more information about how to install the JDBC driver, see Amazon Redshift JDBC driver installation and configuration guide.

Use this Amazon Redshift JDBC driver with the AWS SDK that is required for IAM database authentication.

• JDBC 4.2–compatible driver (without the AWS SDK) version 1.2.55.

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

Be sure to use ANTLR version 4.8.1. The antlr4-runtime-4.8-1.jar is included in the ZIP download link above with the JDBC 4.2–compatible driver (without the AWS SDK) and driver dependent libraries for the AWS SDK.

For more information about previous driver versions, see Use previous JDBC driver versions with the AWS SDK for Java (p. 129).

Then download and review the Amazon Redshift ODBC and JDBC driver license agreement.

If your tool requires a specific previous version of a driver, see Use previous JDBC driver version 1.0 driver versions in certain cases (p. 127).

**Getting 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` still works.

The fields of the format shown preceding have the following values.

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

The following is an example JDBC URL: `jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`
For information about how to get your JDBC connection, see Finding your cluster connection string (p. 91).

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. 162).

**Configure authentication and SSL for JDBC connection**

Configure the Amazon Redshift JDBC driver to authenticate your connection according to the security requirements of the Amazon Redshift server that you are connecting to.

To authenticate the connection, always provide your Amazon Redshift user name and password. The password is transmitted using a salted MD5 hash of the password. Depending on whether SSL is enabled and required on the server, you might also need to configure the driver to connect through SSL. You might need to use one-way SSL authentication so that the client (the driver itself) verifies the identity of the server.

For information about configuring the JDBC driver to authenticate the connection, see Amazon Redshift JDBC driver installation and configuration guide.

**Configure TCP keepalives for JDBC connection**

By default, the Amazon Redshift JDBC driver is configured to use TCP keepalives to prevent connections from timing out. You can specify when the driver starts sending keepalive packets or disable the feature by setting the relevant properties in the connection URL.

For information about configuring TCP keepalives for the JDBC driver, see Amazon Redshift JDBC driver installation and configuration guide.

**Configure logging for JDBC connection**

To help troubleshoot issues, you can enable logging in the JDBC driver.

For information about configuring logging for JDBC connection, see Amazon Redshift JDBC driver installation and configuration guide.

**Configure 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 in the AWS SDK for Java Developer Guide.

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, add the JDBC driver as a dependency in your project's `pom.xml` file. If you use Maven to build your project and want to use a JDBC connection, take the steps in the following section.

**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.

   ```xml
   <repositories>
   <repository>
   
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
   </repository>
   
   <repository>
   <id>amazon-redshift</id>
   <url>https://aws.amazon.com/maven/</url>
Amazon Redshift Cluster Management Guide
Configuring connections in Amazon Redshift

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

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

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

Amazon Redshift offers drivers for tools that are compatible with the JDBC 4.2 API. For information about the functionality supported by these drivers, see Download an Amazon Redshift JDBC driver (p. 122).

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 before 1.2.1.1001, only version-specific class names are supported.

Replace the `driver-version` in the following example with your driver version. For example, 1.2.45.1069.

- JDBC 4.2–compatible driver:

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

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

3. Download and review the Amazon Redshift ODBC and JDBC driver 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.

Replace the `driver-version` in the following example with your driver version. For example, 1.2.45.1069.

- JDBC 4.2–compatible driver:

```xml
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42-no-awssdk</artifactId>
</dependency>
```
The class name for this driver is `com.amazon.redshift.jdbc42.Driver`.

The Amazon Redshift Maven drivers with no SDKs include the following required dependency that you must include in your project.

```
<dependency>
  <groupId>org.antlr</groupId>
  <artifactId>antlr4</artifactId>
  <version>4.5.3</version>
  <scope>runtime</scope>
  <optional>false</optional>
</dependency>
```

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>
  <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 Use previous JDBC driver versions with Maven (p. 130).

### Upgrading the driver to the latest version

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

```
mvn clean
```

### Configure JDBC driver options

To control the behavior of the Amazon Redshift JDBC driver, you can append configuration options 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. 144).

For information about how to set up JDBC driver configuration options, see Amazon Redshift JDBC driver installation and configuration guide.

Use previous JDBC driver version 1.0 driver versions in certain cases

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 these versions of the drivers, see Download an Amazon Redshift JDBC driver (p. 122).

Then download and review the Amazon Redshift ODBC and JDBC driver license agreement.

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 SSL certificates are managed. 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. 147).

If you use the Amazon Redshift JDBC driver for database authentication, make sure that you have AWS SDK for Java 1.11.118 or later in your Java class path. If you don't have AWS SDK for Java installed, you can use a driver that includes the AWS SDK. For more information, see Use previous JDBC driver versions with the AWS SDK for Java (p. 129).

These are JDBC 4.2–compatible drivers:

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.55.1083/RedshiftJDBC42-no-awssdk-1.2.55.1083.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.54.1082/RedshiftJDBC42-no-awssdk-1.2.54.1082.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.47.1071/RedshiftJDBC42-no-awssdk-1.2.47.1071.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.43.1067/RedshiftJDBC42-no-awssdk-1.2.43.1067.jar.
These ZIP files contain JDBC4.2–compatible drivers and driver–dependent AWS SDK library files. Unzip the dependent jar files to the same location as the JDBC driver. Only the JDBC driver needs to be in the CLASSPATH. For more information about how to install the JDBC driver, see Amazon Redshift JDBC driver installation and configuration guide.

These are previous JDBC 4.1–compatible drivers:

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.43.1065/RedshiftJDBC41-no-awssdk-1.2.43.1067.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.41.1065/RedshiftJDBC41-no-awssdk-1.2.41.1065.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.36.1060/RedshiftJDBC41-no-awssdk-1.2.36.1060.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.34.1058/RedshiftJDBC41-no-awssdk-1.2.34.1058.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.32.1056/RedshiftJDBC41-no-awssdk-1.2.32.1056.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.27.1051/RedshiftJDBC41-no-awssdk-1.2.27.1051.jar.
Configuring connections in Amazon Redshift

These are previous JDBC 4.0–compatible drivers:

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.43.1067/RedshiftJDBC4-no-awssdk-1.2.43.1067.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.41.1065/RedshiftJDBC4-no-awssdk-1.2.41.1065.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.36.1060/RedshiftJDBC4-no-awssdk-1.2.36.1060.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.34.1058/RedshiftJDBC4-no-awssdk-1.2.34.1058.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.32.1056/RedshiftJDBC4-no-awssdk-1.2.32.1056.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.27.1051/RedshiftJDBC4-no-awssdk-1.2.27.1051.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC4-no-awssdk-1.2.15.1025.jar.

Use previous JDBC driver versions with the AWS SDK for Java

If you use the JDBC driver for database authentication, make sure that you have AWS SDK for Java 1.11.118 or later in your Java class path. If you don't have AWS SDK for Java installed, you can use one of the following drivers that include the AWS SDK.

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.43.1067/RedshiftJDBC42-1.2.43.1067.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.41.1065/RedshiftJDBC42-1.2.41.1065.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.37.1061/RedshiftJDBC4-1.2.37.1061.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.36.1060/RedshiftJDBC42-1.2.36.1060.jar.
Configuring connections in Amazon Redshift

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.36.1060/RedshiftJDBC41-1.2.36.1060.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.36.1060/RedshiftJDBC4-1.2.36.1060.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.34.1058/RedshiftJDBC42-1.2.34.1058.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.34.1058/RedshiftJDBC41-1.2.34.1058.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.32.1056/RedshiftJDBC41-1.2.32.1056.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.32.1056/RedshiftJDBC4-1.2.32.1056.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.27.1051/RedshiftJDBC41-1.2.27.1051.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.27.1051/RedshiftJDBC4-1.2.27.1051.jar.
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/1.2.15.1025/RedshiftJDBC42-1.2.15.1025.jar.

Use previous JDBC driver versions with 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 these driver versions, see Download an Amazon Redshift JDBC driver (p. 122). For information about configuring with Maven, see Configure JDBC connection with Apache Maven (p. 124).

Configuring 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, 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 macOS X operating systems. Before you install an ODBC driver, determine whether your SQL client tool is 32-bit or 64-bit. Install the ODBC driver that matches the requirements of your SQL client tool. Otherwise, the connection doesn't 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.

For the latest information about ODBC driver functionality and prerequisites, see Amazon Redshift ODBC driver release notes.

For installation and configuration information for Amazon Redshift ODBC drivers, see Amazon Redshift ODBC connector installation and configuration guide.

If you want to use an ODBC connection, take the following steps.

**Topics**
- Obtain the ODBC URL for your cluster (p. 131)
- Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132)
- Install the Amazon Redshift ODBC driver on Linux (p. 134)
- Install the Amazon Redshift ODBC driver on macOS X (p. 135)
- Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136)
- Configure ODBC driver options (p. 140)
- Use previous ODBC driver versions in certain cases (p. 140)

**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 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
```

The fields of the format shown preceding have the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>The name of the ODBC driver to use. Depending on the driver that you download for your architecture, the values is Amazon Redshift (x86) (for the 32-bit driver) or Amazon Redshift (x64) (for the 64-bit driver).</td>
</tr>
<tr>
<td>Server</td>
<td>The endpoint of the Amazon Redshift cluster.</td>
</tr>
<tr>
<td>Database</td>
<td>The database that you created for your cluster.</td>
</tr>
<tr>
<td>UID</td>
<td>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.</td>
</tr>
<tr>
<td>PWD</td>
<td>The password for the user account to connect to the database.</td>
</tr>
</tbody>
</table>
Configuring connections in Amazon Redshift

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>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.</td>
</tr>
</tbody>
</table>

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
```

For information about how to get your ODBC connection, see Finding your cluster connection string (p. 91).

Install and configure the Amazon Redshift ODBC driver on Microsoft Windows

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 a list of minimum system requirements. For information about minimum system requirements, see Amazon Redshift ODBC connector installation and configuration guide.

Installing the Amazon Redshift driver on Windows operating systems

Use the following procedure to download the Amazon Redshift ODBC drivers for Windows operating systems. Only use a driver other than these if you're running a third-party application that is certified for use with Amazon Redshift and that requires a specific driver.

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 ODBC driver version 1.4.34
     
     The name for this driver is Amazon Redshift (x86).
   
   - 64-bit ODBC driver version 1.4.34
     
     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 and JDBC driver license agreement.

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, add a data source name (DSN) entry to the client computer or Amazon EC2 instance. SQL client tools use this data source to connect to the Amazon Redshift database.

We recommend that you create a system DSN instead of a user DSN. Some applications load the data using a different user account. These applications might not be able to detect user DSNs that are created under another user account.
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. 313).

For information about how to create a system DSN entry, see Amazon Redshift ODBC connector installation and configuration guide.

To create a system DSN entry for an ODBC connection on Windows

1. In the Start menu, open ODBC Data Sources.

   Make sure that you choose the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to Amazon Redshift.

2. In the ODBC Data Source Administrator, choose the Driver tab and locate the driver folder.

   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 have a folder for each driver.

3. Choose the System DSN tab to configure the driver for all users on the computer, or the User DSN tab to configure the driver for your user account only.


5. Choose the Amazon Redshift ODBC driver, and then choose Finish. The Amazon Redshift ODBC Driver DSN Setup window opens.

6. Under Connection Settings, enter the following information:

   Data source name
   Enter 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 Guide, you might type exampleclusterdsn to make it easy to remember the cluster that you 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. 91).

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

   Database
   Enter the name of the Amazon Redshift database. If you launched your cluster without specifying a database name, enter dev. Otherwise, use the name that you chose during the launch process. If you followed the Amazon Redshift Getting Started Guide, enter dev.

7. Under Authentication, specify the configuration options to configure standard or IAM authentication. For information about different authentication options, see "Configuring Authentication on Windows" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

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

   SSL authentication
Choose a mode for handling Secure Sockets Layer (SSL). In a test environment, you might use prefer. However, for production environments and when secure data exchange is required, use verify-ca or verify-full. For more information about using SSL on Windows, see "Configuring SSL Verification on Windows" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

9. Under Additional Options, specify options on how to return query results to your SQL client tool or application. For more information, see "Configuring Additional Options on Windows" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

10. In Logging Options, specify values for the logging option. For more information, see "Configuring Logging Options on Windows" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

   Then choose OK.

11. Under Data Type Options, specify values for data types. For more information, see "Configuring Data Type Options on Windows" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

   Then choose OK.

12. Choose Test. If the client computer can connect to the Amazon Redshift database, you 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. 162).

13. Configure TCP keepalives on Windows to prevent connections from timing out. For information about how to configure TCP keepalives on Windows, see Amazon Redshift ODBC Connector Installation and Configuration Guide.

14. To help troubleshooting, configure logging. For information about how to configure logging on Windows, see Amazon Redshift ODBC Connector Installation and Configuration Guide.

Install the Amazon Redshift ODBC driver on Linux

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 a list of minimum system requirements. For information about minimum system requirements, see Amazon Redshift ODBC connector installation and configuration guide.

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 installs 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 driver version 1.4.34
• 64-bit RPM driver version 1.4.34
• 32-bit Debian driver version 1.4.34
• 64-bit Debian driver version 1.4.34

The name for each 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 and JDBC driver license agreement.

2. Go 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 and CentOS operating systems, run the following command.

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

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

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

• On SLES, run the following command.

  ```bash
  zypper install RPMFileName
  ```

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

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

• On Debian, run the following command.

  ```bash
  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.

  ```bash
  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 Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136).

Install the Amazon Redshift ODBC driver on macOS 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 a list of minimum system requirements. For information about
minimum system requirements, see Amazon Redshift ODBC connector installation and configuration guide.

Installing the Amazon Redshift ODBC driver on macOS X

Use the steps in this section to download and install the Amazon Redshift ODBC driver on a supported version of macOS X. The installation process installs 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 macOS X

1. Download the macOS X driver version 1.4.34. The name for this driver is Amazon Redshift ODBC driver.

   **Important**
   After certificate rotation on September 4, 2020, ODBC Driver version 1.4.8.1000 or earlier on macOS will not be able to establish connections to Amazon Redshift clusters. For more information, see Driver update required for Amazon Redshift ODBC Driver earlier than 1.4.10 on Apple macOS.

   Then download and review the Amazon Redshift ODBC and JDBC driver license agreement.

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. To perform the installation, agree to the terms of the license agreement.

   **Important**
   When you have finished installing the driver, configure it for use on your system. For more information on driver configuration, see Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136).

Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems

On Linux and macOS 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:

- unixODBC driver manager (for Linux operating systems)
- iODBC driver manager (for macOS X operating system)

For more information about the supported ODBC driver managers to configure the Amazon Redshift ODBC drivers, see System requirements (p. 134) for Linux operating systems and System requirements (p. 135) for macOS X operating systems. Also, see "Specifying ODBC Driver Managers on Non- Windows Machines" in Amazon Redshift ODBC connector installation and configuration guide.

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/redshiftdbc/lib/32 (for the 32-bit driver on Linux operating systems)
• /opt/amazon/redshiftdbc/lib/64 (for the 64-bit driver on Linux operating systems)
• /opt/amazon/redshift/lib (for the driver on macOS X)

Additionally, under /opt/amazon/redshiftdbc/Setup on Linux or /opt/amazon/redshift/Setup on macOS X, there are sample odbc.ini and odbcinst.ini files. You can use these files 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. You then lose any changes that you might have made to those files.

To avoid this, copy the amazon.redshiftdbc.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, either use the configuration files in the user’s home directory or create new versions in another directory. By default, your Linux or macOS 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 is indicated by the dot (.) in front of each file name. These files display only when you use the –a flag to list the directory contents.

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

By default, ODBC driver managers are configured to use hidden versions of the odbc.ini and odbcinst.ini configuration files (named .odbc.ini and .odbcinst.ini) located in the home directory. They also are configured to use the amazon.redshiftdbc.ini file in the /lib subfolder of the driver installation directory. If you store these configuration files elsewhere, set the environment variables described following so that the driver manager can locate the files. For more information, see "Specifying the Locations of the Driver Configuration Files" in Amazon Redshift ODBC connector installation and configuration guide.

Creating a data source name on Linux and macOS X operating systems

When connecting to your data store using a data source name (DSN), configure the odbc.ini file to define DSNs. Set the properties in the odbc.ini file to create a DSN that specifies the connection information for your data store.

For information about how to configure the odbc.ini file, see "Creating a Data Source Name on a Non-Windows Machine" in Amazon Redshift ODBC connector installation and configuration guide.

Use the following format on Linux operating systems.

```ini
[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.
Configuring connections in Amazon Redshift

Use the following format on macOS X operating systems.

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/lib/amazonredshiftodbc.dylib
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

The following example shows the configuration for `odbc.ini` on macOS X operating systems.

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

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

Configuring a connection without a DSN on Linux and macOS X operating systems

To connect to your data store through a connection that doesn't have a DSN, define the driver in the `odbcinst.ini` file. Then provide a DSN-less connection string in your application.

For information about how to configure the `odbcinst.ini` file in this case, see "Configuring a DSN-less Connection on a Non-Windows Machine" in Amazon Redshift ODBC connector installation and configuration guide.

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 macOS X operating systems.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=\path/lib/amazonredshiftodbc.dylib
...
```

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

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

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for macOS X
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
```

**Configuring environment variables**

Use the correct ODBC driver manager to load the correct driver. To do this, set the library path environment variable. For more information, see "Specifying ODBC Driver Managers on Non-Windows Machines" in Amazon Redshift ODBC connector installation and configuration guide.

By default, ODBC driver managers are configured to use hidden versions of the odbc.ini and odbcinst.ini configuration files (named .odbc.ini and .odbcinst.ini) located in the home directory. They also are configured to use the amazon.redshiftodbc.ini file in the /lib subfolder of the driver installation directory. If you store these configuration files elsewhere, the environment variables so that the driver manager can locate the files. For more information, see "Specifying the Locations of the Driver Configuration Files" in Amazon Redshift ODBC Connector Installation and Configuration Guide.

**Configuring connection features**

You can configure the following connection features for your ODBC setting:
• Configure the ODBC driver to provide credentials and authenticate the connection to the Amazon Redshift database.
• Configure the ODBC driver to connect to a socket enabled with Secure Sockets Layer (SSL), if you are connecting to an Amazon Redshift server that has SSL enabled.
• Configure the ODBC driver to connect to Amazon Redshift through a proxy server.
• Configure the ODBC driver to use a query processing mode to prevent queries from consuming too much memory.
• Configure the ODBC driver to pass IAM authentication processes through a proxy server.
• Configure the ODBC driver to use TCP keepalives to prevent connections from timing out.

For information about these connection features, see Amazon Redshift ODBC connector installation and configuration guide.

**Configure ODBC driver options**

You can use configuration options 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 (p. 132). For an example of setting driver options in a connection string, see Connect to your cluster programmatically (p. 157).

In Linux and macOS X, you set driver configuration options in your `odbc.ini` and `amazon.redshiftodbc.ini` files, as described in Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136). 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`.

For information about how to set up ODBC driver configuration options, see Amazon Redshift ODBC connector installation and configuration guide.

**Use previous ODBC driver versions in certain cases**

Download a previous version of the Amazon Redshift ODBC driver only if your tool requires a specific version of the driver.

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 SSL certificates are managed. 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. 147).

**Use previous ODBC driver versions for Windows**

The following are the 32-bit drivers:

- [https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC32-1.4.34.1000.msi](https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC32-1.4.34.1000.msi)
- [https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.30.1000/AmazonRedshiftODBC32-1.4.30.1000.msi](https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.30.1000/AmazonRedshiftODBC32-1.4.30.1000.msi)
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.27.1000/AmazonRedshiftODBC32-1.4.27.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.17.1000/AmazonRedshiftODBC32-1.4.17.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.16.1000/AmazonRedshiftODBC32-1.4.16.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.11.1000/AmazonRedshiftODBC32-1.4.11.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.10.1000/AmazonRedshiftODBC32-1.4.10.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.8.1000/AmazonRedshiftODBC32-1.4.8.1000.msi.

The following are the 64-bit drivers:

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC64-1.4.34.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.30.1000/AmazonRedshiftODBC64-1.4.30.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.27.1000/AmazonRedshiftODBC64-1.4.27.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.20.1001/AmazonRedshiftODBC64-1.4.20.1001.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.18.1001/AmazonRedshiftODBC64-1.4.18.1001.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.17.1000/AmazonRedshiftODBC64-1.4.17.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.16.1000/AmazonRedshiftODBC64-1.4.16.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.13.1000/AmazonRedshiftODBC64-1.4.13.1000.msi.
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.8.1000/AmazonRedshiftODBC64-1.4.8.1000.msi.

Use previous ODBC driver versions for Linux

The following are the versions of the 32-bit driver:
The following are the versions of the 64-bit driver:
• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC-64-bit-1.4.34.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC-64-bit-1.4.34.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.30.1000/AmazonRedshiftODBC-64-bit-1.4.30.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.30.1000/AmazonRedshiftODBC-64-bit-1.4.30.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.27.1000/AmazonRedshiftODBC-64-bit-1.4.27.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.27.1000/AmazonRedshiftODBC-64-bit-1.4.27.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.20.1001/AmazonRedshiftODBC-64-bit-1.4.20.1001-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.20.1001/AmazonRedshiftODBC-64-bit-1.4.20.1001-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.18.1001/AmazonRedshiftODBC-64-bit-1.4.18.1001-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.18.1001/AmazonRedshiftODBC-64-bit-1.4.18.1001-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.17.1000/AmazonRedshiftODBC-64-bit-1.4.17.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.17.1000/AmazonRedshiftODBC-64-bit-1.4.17.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.16.1000/AmazonRedshiftODBC-64-bit-1.4.16.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.16.1000/AmazonRedshiftODBC-64-bit-1.4.16.1000-1.x86_64.deb.


• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.13.1000/AmazonRedshiftODBC-64-bit-1.4.13.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.13.1000/AmazonRedshiftODBC-64-bit-1.4.13.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.11.1000/AmazonRedshiftODBC-64-bit-1.4.11.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.11.1000/AmazonRedshiftODBC-64-bit-1.4.11.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.10.1000/AmazonRedshiftODBC-64-bit-1.4.10.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.10.1000/AmazonRedshiftODBC-64-bit-1.4.10.1000-1.x86_64.deb.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.8.1000/AmazonRedshiftODBC-64-bit-1.4.8.1000-1.x86_64.rpm.

• https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.8.1000/AmazonRedshiftODBC-64-bit-1.4.8.1000-1.x86_64.deb.
Use previous ODBC driver versions for macOS X

The following are the versions of the Amazon Redshift ODBC driver for macOS X:

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.34.1000/AmazonRedshiftODBC-1.4.34.1000.dmg.
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.27.1000/AmazonRedshiftODBC-1.4.27.1000.dmg.
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.17.1000/AmazonRedshiftODBC-1.4.17.1000.dmg.
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.16.1000/AmazonRedshiftODBC-1.4.16.1000.dmg.
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.11.1000/AmazonRedshiftODBC-1.4.11.1000.dmg.
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.10.1000/AmazonRedshiftODBC-1.4.10.1000.dmg.

**Important**

After certificate rotation on September 4, 2020, ODBC Driver version 1.4.8.1000 or earlier on macOS will not be able to establish connections to Amazon Redshift clusters. For more information, see Driver update required for Amazon Redshift ODBC Driver earlier than 1.4.10 on Apple macOS.

Configuring 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. You can find the set of Certificate Authorities that you must trust to properly support SSL at https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt. If the certificate bundle doesn't download, open the context (right-click) menu for the link, and then choose Save link as....

**Important**

Amazon Redshift has changed the way that SSL certificates are managed. 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. 147).

In the China Regions on January 5, 2021, Amazon Redshift is replacing the SSL certificates on your clusters with AWS Certificate Manager (ACM) issued certificates. This change affects you...
if your SQL clients or applications connect to Amazon Redshift using SSL with the sslMode connection option set to require, verify-ca, or verify-full. The previous certificate bundle (https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem) will not work after the migration. Use the updated bundle from https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt. The expected MD5 checksum number is a3081eb4a74f882df1c236a9e9a87. The sha256 checksum is df6aeb8a2e6f4eb00b9f72a786d69000916c90218005cc3d92800226c06da8.

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. 201).

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 don't 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 uses 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 Configuring an ODBC connection (p. 130).

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. 147).

The Amazon Redshift certificate authority bundle is stored at https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt

You can verify that the certificate that you downloaded matches the expected MD5 checksum number. To do this, you can use the Md5sum program on Linux operating systems, or another tool on Windows and macOS X operating systems.

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.

You can determine whether SSL is used and server certificates are verified in a connection between the client and the server. To do this, 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:

<table>
<thead>
<tr>
<th>Sslmode (client)</th>
<th>require_SSL (server)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable</td>
<td>false</td>
<td>The connection is not encrypted.</td>
</tr>
<tr>
<td>disable</td>
<td>true</td>
<td>The connection can't be made because the server requires SSL and the client has SSL disabled for the connection.</td>
</tr>
<tr>
<td>allow</td>
<td>true</td>
<td>The connection is encrypted.</td>
</tr>
<tr>
<td>allow</td>
<td>false</td>
<td>The connection is not encrypted.</td>
</tr>
<tr>
<td>prefer or require</td>
<td>true</td>
<td>The connection is encrypted.</td>
</tr>
<tr>
<td>prefer or require</td>
<td>false</td>
<td>The connection is encrypted.</td>
</tr>
<tr>
<td>verify-ca</td>
<td>true</td>
<td>The connection is encrypted and the server certificate is verified.</td>
</tr>
<tr>
<td>verify-ca</td>
<td>false</td>
<td>The connection is encrypted and the server certificate is verified.</td>
</tr>
<tr>
<td>verify-full</td>
<td>true</td>
<td>The connection is encrypted and the server certificate and hostname are verified.</td>
</tr>
<tr>
<td>verify-full</td>
<td>false</td>
<td>The connection is encrypted and the server certificate and hostname are verified.</td>
</tr>
</tbody>
</table>

**Connect using the server certificate with ODBC on Microsoft Windows**

If you want to connect to your cluster using SSL and the server certificate, first download the certificate to your client computer or Amazon EC2 instance. 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 choose **Save**.
For more information about configuring the ODBC DSN, see Configuring an ODBC connection (p. 130).

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 SSL certificates are managed. 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. 147).

To connect using trust CA certificates

You can use the redshift-keytool.jar file to import CA certificates in the Amazon 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 redshift-keytool.jar.
3. Do one of the following:
   • To import the Amazon Redshift Certificate Authority bundle into a Java TrustStore, run the following command.

   ```java
   java -jar redshift-keytool.jar -s
   ```
   • To import the Amazon Redshift Certificate Authority bundle into your private TrustStore, run the following command:

   ```java
   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.
• You aren’t using the Amazon Redshift ODBC or JDBC drivers, or you use Amazon Redshift drivers before ODBC version 1.3.7.1000 or JDBC version 1.2.8.1005.
If this change affects you on commercial Amazon Redshift Regions, 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.

If this change affects you on AWS GovCloud (US) (US) Regions, then you must update your current trust root CA certificates before April 1, 2020 to avoid service interruption. Beginning on this date, clients connecting to Amazon Redshift clusters using SSL encrypted connections need an additional trusted certificate authority (CA). Clients use trusted certificate authorities to confirm the identity of the Amazon Redshift cluster when they connect to it. Your action is required to update your SQL clients and applications to use an updated certificate bundle that includes the new trusted CA.

**Important**

In the China Regions on January 5, 2021, Amazon Redshift is replacing the SSL certificates on your clusters with AWS Certificate Manager (ACM) issued certificates. If this change affects you on China (Beijing) Region or China (Ningxia) Region, then you must update your current trust root CA certificates before January 5, 2021 to avoid service interruption. Beginning on this date, clients connecting to Amazon Redshift clusters using SSL encrypted connections need an additional trusted certificate authority (CA). Clients use trusted certificate authorities to confirm the identity of the Amazon Redshift cluster when they connect to it. Your action is required to update your SQL clients and applications to use an updated certificate bundle that includes the new trusted CA.

- Using the latest Amazon Redshift ODBC or JDBC drivers (p. 148)
- Using earlier Amazon Redshift ODBC or JDBC drivers (p. 148)
- Using other SSL connection types (p. 149)

### 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 Configuring an ODBC connection (p. 130) or Configuring a JDBC driver version 1.0 connection (p. 122).

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 Amazon Redshift certificate authority bundle into a TrustStore (p. 149).

### Using earlier Amazon Redshift ODBC or JDBC drivers

If you must use an Amazon Redshift ODBC driver before 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 before 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 Amazon Redshift certificate authority bundle into a TrustStore (p. 149).
- 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 Amazon Redshift certificate authority bundle into a TrustStore (p. 149).

Importing the Amazon Redshift certificate authority bundle into a TrustStore

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

To import the Amazon Redshift certificate authority bundle into a TrustStore

1. Download redshift-keytool.jar.
2. Do one of the following:
   - To import the Amazon Redshift Certificate Authority bundle into a Java TrustStore, run the following command.
     ```
     java -jar redshift-keytool.jar -s
     ```
   - To import the Amazon 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 Amazon Redshift certificate authority bundle.
2. Place the certificates from the bundle in your root.crt file.
   - On Linux and macOS 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. 150)
- Connect to your cluster by using the psql tool (p. 153)
- Connect to your cluster programmatically (p. 157)
- Using an authentication profile to connect to Amazon Redshift (p. 160)
Connect to your cluster by using SQL Workbench/J

Amazon Redshift doesn't 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 Guide* 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*, choose the Java icon.
   - Windows: In the *Control Panel*, choose the Java icon.
   - Any system: In a command shell, type `java -version`. You can also visit [https://www.java.com](https://www.java.com), choose the *Do I have Java?* link, and choose the *Verify Java* button.

   For information about installing and configuring the Java Runtime Environment, go to [https://www.java.com](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 an Amazon Redshift JDBC driver (p. 122). 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. Choose *File*, and then choose *Connect window*.
3. Choose *Create a new connection profile*.
4. In the *New profile* box, type a name for the profile. For example, examplecluster_jdbc.
5. Choose *Manage Drivers*. The *Manage Drivers* dialog opens. In the *Name* box, type a name for the driver.
Choose the folder icon next to the Library box, navigate to the location of the driver, choose it, and then choose Open.

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

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 Configuring a JDBC driver version 1.0 connection (p. 122).
8. In Username, type the name of the master user.
   
   If you are following the Amazon Redshift Getting Started Guide, type masteruser.
9. In Password, type the password associated with the master user account.
10. Select the Autocommit box.
11. Choose the Save profile list icon, as shown below:
12. Choose **OK**.

![Select Connection Profile](image)

**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 Guide, you can test your connection by typing the following query into the Statement window:

```sql
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](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/

If you have trouble connecting from a Microsoft Windows prompt due to an invalid client_encoding, set the PGCLIENTENCODING environment variable to UTF-8 before running psql.

```
set PGCLIENTENCODING= UTF-8
```

**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. Choose one of the following steps depending on which Amazon Redshift console you are using:
   - (New console) In the navigation pane, choose **CLUSTERS**. Then choose the cluster name from the list to open its details. On the **Properties** tab, in the **Database configurations** section, record the **Database name** and **Port**. View the **Connection details** section and record the **Endpoint** which is in the following form:

   ```
   endpoint:port/databasename
   ```

   - (Original console) In the navigation pane, choose **Clusters**. Choose your cluster to open it. Under **Cluster Database Properties**, record the values of **Endpoint**, **Port**, and **Database Name**.
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 <databasename> -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.
- `<databasename>` 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.<XXXXXXXXXXXX>.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 Configuring security options for connections (p. 144).

For more information about connection information string parameters, see [https://www.postgresql.org/docs/8.4/static/libpq-connect.html](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. Choose one of the following steps depending on which Amazon Redshift console you are using:
   - (New console) In the navigation pane, choose CLUSTERS. Then choose the cluster name from the list to open its details. On the Properties tab, in the Database configurations section, record the Database name and Port. View the Connection details section and record the Endpoint which is in the following form:

   ```
   endpoint:port/databasename
   ```

   - (Original console) In the navigation pane, choose Clusters. Choose your cluster to open it. Under Cluster Database Properties, record the values of Endpoint, Port, and Database Name.
3. At a command prompt, specify the connection information using a connection information string:

```sql
psql "host=<endpoint> user=<userid> dbname=<databasename> 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.
- `<databasename>` 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 macOS X operating systems, the path is

```
~/.postgresql/root.crt
```

On Microsoft Windows, the path is

```
%APPDATA%/postgresql/root.crt
```

For example:
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 Configuring security options for connections (p. 144) 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:

```java
props.setProperty("ssl", "true");
```

For more information about the server certificate, see Configuring security options for connections (p. 144).

For step-by-step instructions to run the following example, see Running Java examples for Amazon Redshift using Eclipse (p. 391).

```
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * 
 * http://aws.amazon.com/apache2.0/
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[ConnectToClusterExample demonstrates how to connect to an
// Amazon Redshift cluster and run a sample query.]
// snippet-service:[redshift]
```
package connection;

import java.sql.*;
import java.util.Properties;

public class ConnectToCluster {
    //Redshift driver: "jdbc:redshift://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{
            }
        }
    }
}
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 Configuring security options for connections (p. 144) 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.

```csharp
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);
}

}

Using an authentication profile to connect to Amazon Redshift

If you have many connections to Amazon Redshift, it can be difficult to manage settings for all of them. Often, each JDBC or ODBC connection uses specific configuration options. By using an authentication profile, you can store connection options together. This way, your users can choose a profile to connect with and avoid managing settings for individual options. Profiles can apply to various scenarios and user types.

After you create an authentication profile, users can add the ready-to-use profile to a connection string. By doing this, they can connect to Amazon Redshift with the right settings for each role and use case.

For Amazon Redshift API information, see CreateAuthenticationProfile.

Creating an authentication profile

Using the AWS CLI, you create an authentication profile with the create-authentication-profile command. This assumes that you have an existing Amazon Redshift cluster and an existing database. Your credentials must have permission to connect to the Amazon Redshift database and rights to fetch the authentication profile. You provide the configuration options as a JSON string, or reference a file containing your JSON string.
create-authentication-profile --authentication-profile-name=ExampleProfileName --authentication-profile-content="{"AllowDBUserOverride":"1","Client_ID":"ExampleClientID","App_ID":"ExampleAppID","AutoCreate":false,"enableFetchRingBuffer":true,"databaseMetadataCurrentDbOnly":true}"

This command creates the profile with the specified JSON settings. The following is returned, which indicates that the profile is created.

```json
{"AuthenticationProfileName": "ExampleProfileName", "AuthenticationProfileContent": "{"AllowDBUserOverride":"1","Client_ID":"ExampleClientID","App_ID":"ExampleAppID","AutoCreate":false,"enableFetchRingBuffer":true,"databaseMetadataCurrentDbOnly":true}" }
```

**Limitations and quotas for creating an authentication profile**

Each customer has a quota of ten (10) authentication profiles.

Certain errors can occur with authentication profiles. Examples are if you create a new profile with an existing name, or if you exceed your profile quota. For more information, see CreateAuthenticationProfile.

You can't store certain option keys and values for JDBC, ODBC, and Python connection strings in the authentication profile store:

- AccessKeyID
- access_key_id
- SecretAccessKey
- secret_access_key_id
- PWD
- Password
- password

You can't store the key or value AuthProfile in the profile store, for JDBC or ODBC connection strings. For Python connections, you can't store auth_profile.

Authentication profiles are stored in Amazon DynamoDB and managed by AWS.

**Working with authentication profiles**

After you create an authentication profile, you can include the profile name as a connection option for JDBC version 2.0 AuthProfile. Using this connection option retrieves the stored settings.

```
```

The following is an example JDBC URL string.
jdbc:redshift:iam://examplecluster:us-west-2/dev?
AuthProfile="ExampleProfile"&AccessKeyID="AKIAIOSFODNN7EXAMPLE"&SecretAccessKey="wJalrXUtMFI/K7MDENG/bPxRfiCYEXAMPLEKEY"

Specify both the AccessKeyID and SecretAccessKey in the JDBC URL, along with the authentication profile name.

You can also separate the configuration options with semicolon delimiters, such as in the following example, which includes options for logging.

jdbc:redshift:iam://my_redshift_end_point:5439/dev?LogLevel=6;LogPath=/tmp;AuthProfile=my_profile;AccessKeyID="AKIAIOSFODNN7EXAMPLE";SecretAccessKey="wJalrXUtMFI/K7MDENG/bPxRfiCYEXAMPLEKEY"

Note
Don't add confidential information to the authentication profile. For example, don't store an AccessKeyID or SecretAccessKey value in an authentication profile. The authentication profile store has rules to prohibit storage of secret keys. You get an error if you try to store a key and value associated with sensitive information.

Getting authentication profiles

To list existing authentication profiles, call the following command.

describe-authentication-profiles --authentication-profile-name <value: String>

The following example shows two retrieved profiles. All profiles are returned if you don't specify a profile name.

```
{ "AuthenticationProfiles": [ { "AuthenticationProfileName": "testProfile1", "AuthenticationProfileContent": "{"AllowDBUserOverride":"1","Client_ID":"ExampleClientID","App_ID":"ExampleAppID","AutoCreate":false,"enableFetchRingBuffer":true,"databaseMetadataCurrentDbOnly":true} " }, { "AuthenticationProfileName": "testProfile2", "AuthenticationProfileContent": "{"AllowDBUserOverride":"1","Client_ID":"ExampleClientID","App_ID":"ExampleAppID","AutoCreate":false,"enableFetchRingBuffer":true,"databaseMetadataCurrentDbOnly":true} " } ] }
```

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 Configuring security options for connections (p. 144).

Important
Amazon Redshift has changed the way that SSL certificates are managed. 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. 147).

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.
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 issue happens when you connect to Amazon Redshift from a machine other than an Amazon EC2 instance. In this case, 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. 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. For more information, see Change TCP/IP timeout settings (p. 163).
- Optionally, set keepalive behavior at the DSN level. For more information, see Change DSN timeout settings (p. 164).

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 keepalive 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 keepalive packets that can be lost before the connection is considered broken.

**KeepAlivesIdle**

The number of seconds of inactivity before the driver sends a TCP keepalive packet.

**KeepAlivesInterval**

The number of seconds between each TCP keepalive 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 macOS, 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 macOS computers, see Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136).

If these parameters don't exist, or if they have a value of 0, the system uses the keepalive parameters specified for TCP/IP to determine DSN keepalive behavior. On Windows, you can find the TCP/IP parameters in the registry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
Troubleshooting connection issues in Amazon Redshift

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.

To connect to the cluster from a client tool outside of the network that the cluster is in, add an ingress rule. Add the rule to the cluster security group for the CIDR/IP that you are connecting from:

- If you created your Amazon Redshift cluster in a virtual private cloud (VPC) based on Amazon VPC, 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 a VPC (p. 69).
- If you created your Amazon Redshift cluster outside a VPC, 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. 370).

If you attempt to connect to the cluster from a client tool in an Amazon EC2 instance, you also add an ingress rule. In this case, add the 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. 370).

In some cases, you might have a layer between your client and server, such as a firewall. In these cases, make sure that the firewall accepts inbound connections over the port that you configured for your cluster.

Client and driver are incompatible

Example error

"The specified DSN contains an architecture mismatch between the Driver and Application."

Possible solution

When you attempt to connect and get an error about an architecture mismatch, this means that the client tool and the driver aren't compatible. This occurs because their system architecture doesn't 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 can't 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. In this case, 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. The rule thus instructs 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. 166).
- If your cluster uses the EC2-Classic platform, or you can't 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. 166).

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 information about Amazon Redshift cluster security group, see Amazon Redshift cluster security groups (p. 370). 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 information about how to set inbound and outbound rules for EC2-Classic or EC2-VPC instances, see Differences between instances in EC2-Classic and a VPC in the Amazon EC2 User Guide for Linux Instances.

For more information about how to add rules to VPC security groups, see Managing VPC security groups for a cluster (p. 73). 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

In some cases, your cluster might use the EC2-Classic platform or you can't allow the custom ICMP rule for inbound traffic. In these cases, we recommend that you adjust the MTU to 1500 on the network interface (NIC) of the EC2 instances you connect to your Amazon Redshift cluster from. This adjustment disables TCP/IP jumbo frames to ensure that connections consistently use the same packet size. However, 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

- Follow instructions on the MacOS support site about How to change the MTU for troubleshooting purposes. For more information, search the support site.

Using the Amazon Redshift Data API

You can access your Amazon Redshift database using the built-in Amazon Redshift Data API. Using this API, you can access Amazon Redshift data with web services–based applications, including AWS Lambda, Amazon SageMaker notebooks, and AWS Cloud9. For more information on these applications, see AWS Lambda, Amazon SageMaker, and AWS Cloud9.

The Data API doesn't require a persistent connection to the cluster. Instead, it provides a secure HTTP endpoint and integration with AWS SDKs. You can use the endpoint to run SQL statements without managing connections. Calls to the Data API are asynchronous.

The Data API uses either credentials stored in AWS Secrets Manager or temporary database credentials. You don't need to pass passwords in the API calls with either authorization method. For more information about AWS Secrets Manager, see What Is AWS Secrets Manager? in the AWS Secrets Manager User Guide.

For more information about the Data API operations, see the Amazon Redshift Data API Reference.

Working with the Amazon Redshift Data API

Before you use the Amazon Redshift Data API, review the following steps:
1. Determine if you, as the caller of the Data API, are authorized. For more information about authorization, see Authorizing access to the Amazon Redshift Data API (p. 168).

2. Determine if you plan to call the Data API with authentication credentials from Secrets Manager or temporary credentials. For more information, see Choosing authentication credentials when calling the Amazon Redshift Data API (p. 171).

3. Set up a secret if you use Secrets Manager for authentication credentials. For more information, see Storing database credentials in AWS Secrets Manager (p. 169).

4. Review the considerations and limitations when calling the Data API. For more information, see Considerations when calling the Amazon Redshift Data API (p. 171).

5. Call the Data API from the AWS Command Line Interface (AWS CLI), from your own code, or using the query editor in the Amazon Redshift console. For examples of calling from the AWS CLI, see Calling the Data API with the AWS CLI (p. 173).

**Authorizing access to the Amazon Redshift Data API**

To access the Data API, a user must be authorized. You can authorize a user to access the Data API by adding a managed policy, which is a predefined AWS Identity and Access Management (IAM) policy, to that user. To see the permissions allowed and denied by managed policies, see the IAM console (https://console.aws.amazon.com/iam/).

Amazon Redshift provides the AmazonRedshiftDataFullAccess managed policy. This policy provides full access to Amazon Redshift Data API operations. This policy also allows scoped access to specific Amazon Redshift, AWS Secrets Manager, and IAM API operations needed to authenticate and access an Amazon Redshift cluster. If you use AWS Secrets Manager to authenticate, the policy allows use of the secretsmanager:GetSecretValue action to retrieve the secret tagged with the key RedshiftDataFullAccess. If you use temporary credentials to authenticate, the policy allows use of the redshift:GetClusterCredentials action to the database user name redshift_data_api_user for any database in the cluster. This user name must have already been created in your database.

You can also create your own IAM policy that allows access to specific resources. To create your policy, use the AmazonRedshiftDataFullAccess policy as your starting template. After you create your policy, add it to each user that requires access to the Data API.

To run a query on a cluster that is owned by another account, the owning account must provide an IAM role that the Data API can assume in the calling account. For example, suppose Account B owns a cluster that Account A needs to access. Account B can attach the AWS-managed policy AmazonRedshiftDataFullAccess to Account B's IAM role. Then Account B trusts Account A using a trust policy such as the following:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::accountID-of-account-A:role/someRoleA"
                ],
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Finally, the Account A IAM role needs to be able to assume the Account B IAM role.
Storing credentials in a secret

```json
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::accountID-of-account-B:role/someRoleB"
  }
}
```

The following links provide more information about AWS Identity and Access Management in the IAM User Guide.

- For information about creating an IAM roles, see Creating IAM roles.
- For information about creating an IAM policy, see Creating IAM policies.
- For information about adding an IAM policy to a user, see Adding and removing IAM identity permissions.

### Storing database credentials in AWS Secrets Manager

When you call the Data API, you can pass credentials for the cluster by using a secret in AWS Secrets Manager. To pass credentials in this way, you specify the name of the secret or the Amazon Resource Name (ARN) of the secret.

To store credentials with Secrets Manager, you need SecretManagerReadWrite managed policy permission. For more information about the minimum permissions, see Creating and Managing Secrets with AWS Secrets Manager in the AWS Secrets Manager User Guide.

**To store your credentials in a secret for an Amazon Redshift cluster**

1. Use AWS Secrets Manager to create a secret that contains credentials for your cluster:
   - When you choose Store a new secret, choose Credentials for Redshift cluster.
   - Store your values for User name (database user), Password, and DB cluster (cluster identifier) in your secret.
   - Tag the secret with the key RedshiftDataFullAccess. The AWS-managed policy AmazonRedshiftDataFullAccess only allows the action secretsmanager:GetSecretValue for secrets tagged with the key RedshiftDataFullAccess.

   For instructions, see Creating a Basic Secret in the AWS Secrets Manager User Guide.

2. Use the AWS Secrets Manager console to view the details for the secret you created, or run the `aws secretsmanager describe-secret` AWS CLI command.

   Note the name and ARN of the secret. You can use these in calls to the Data API.

### Creating an Amazon VPC endpoint (AWS PrivateLink) for the Data API

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources, such as Amazon Redshift clusters and applications, into a virtual private cloud (VPC). AWS PrivateLink provides private connectivity between virtual private clouds (VPCs) and AWS services securely on the Amazon network. Using AWS PrivateLink, you can create VPC endpoints, which you can use connect to services across
different accounts and VPCs based on Amazon VPC. For more information about AWS PrivateLink, see VPC Endpoint Services (AWS PrivateLink) in the Amazon Virtual Private Cloud User Guide.

You can call the Data API with Amazon VPC endpoints. Using an Amazon VPC endpoint keeps traffic between applications in your Amazon VPC and the Data API in the AWS network, without using public IP addresses. Amazon VPC endpoints can help you meet compliance and regulatory requirements related to limiting public internet connectivity. For example, if you use an Amazon VPC endpoint, you can keep traffic between an application running on an Amazon EC2 instance and the Data API in the VPCs that contain them.

After you create the Amazon VPC endpoint, you can start using it without making any code or configuration changes in your application.

To create an Amazon VPC endpoint for the Data API

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose Endpoints, and then choose Create Endpoint.
3. On the Create Endpoint page, for Service category, choose AWS services. For Service Name, choose redshift-data (com.amazonaws.region.redshift-data).
4. For VPC, choose the VPC to create the endpoint in.
   
   Choose the VPC that contains the application that makes Data API calls.
5. For Subnets, choose the subnet for each Availability Zone (AZ) used by the AWS service that is running your application.
   
   To create an Amazon VPC endpoint, specify the private IP address range in which the endpoint is accessible. To do this, choose the subnet for each Availability Zone. Doing so restricts the VPC endpoint to the private IP address range specific to each Availability Zone and also creates an Amazon VPC endpoint in each Availability Zone.
6. For Enable DNS name, select Enable for this endpoint.
   
   Private DNS resolves the standard Data API DNS hostname (https://redshift-data.region.amazonaws.com) to the private IP addresses associated with the DNS hostname specific to your Amazon VPC endpoint. As a result, you can access the Data API VPC endpoint using the AWS CLI or AWS SDKs without making any code or configuration changes to update the Data API endpoint URL.
7. For Security group, choose a security group to associate with the Amazon VPC endpoint.
   
   Choose the security group that allows access to the AWS service that is running your application. For example, if an Amazon EC2 instance is running your application, choose the security group that allows access to the Amazon EC2 instance. The security group enables you to control the traffic to the Amazon VPC endpoint from resources in your VPC.
8. Choose Create endpoint.

After the endpoint is created, choose the link in the AWS Management Console to view the endpoint details.

The endpoint Details tab shows the DNS hostnames that were generated while creating the Amazon VPC endpoint.

You can use the standard endpoint (redshift-data.region.amazonaws.com) or one of the VPC-specific endpoints to call the Data API within the Amazon VPC. The standard Data API endpoint automatically routes to the Amazon VPC endpoint. This routing occurs because the Private DNS hostname was enabled when the Amazon VPC endpoint was created.
When you use an Amazon VPC endpoint in a Data API call, all traffic between your application and the Data API remains in the Amazon VPCs that contain them. You can use an Amazon VPC endpoint for any type of Data API call. For information about calling the Data API, see Considerations when calling the Amazon Redshift Data API (p. 171).

Considerations when calling the Amazon Redshift Data API

Consider the following when calling the Data API:

- The maximum duration of a query is 24 hours.
- The maximum number of active queries (STARTED and SUBMITTED queries) per Amazon Redshift cluster is 200.
- The maximum query result size is 100 MB. If a call returns more than 100 MB of response data, the call is ended.
- The maximum retention time for query results is 24 hours.
- The maximum query statement size is 100 KB.
- The Data API is available to query single-node and multiple-node clusters of the following node types:
  - dc2.large
  - dc2.8xlarge
  - ds2.xlarge
  - ds2.8xlarge
  - ra3.xlarge
  - ra3.4xlarge
  - ra3.16xlarge
- The cluster must be in a virtual private cloud (VPC) based on the Amazon VPC service.
- By default, users with the same IAM role or IAM user as the runner of an ExecuteStatement or BatchExecuteStatement API operation can act on the same statement with CancelStatement, DescribeStatement, GetStatementResult, and ListStatements API operations.
- For a list of AWS Regions where the Data API is available, see Redshift Data API Endpoints in the Amazon Web Services General Reference.

Choosing authentication credentials when calling the Amazon Redshift Data API

When you call the Data API, you use one of the following authentication methods for some API operations. Each method requires a different combination of parameters.

AWS Secrets Manager

With this method, provide the secret-arn secret value that is stored in AWS Secrets Manager. The specified secret contains credentials to connect to your database. You also supply a value for cluster-identifier that matches the cluster identifier in the secret.

Temporary credentials

With this method, provide your cluster-identifier, database, and db-user values.

With either method, you can also supply a region value that specifies the AWS Region where your cluster is located.
### Mapping JDBC data types when calling the Amazon Redshift Data API

The following table maps Java Database Connectivity (JDBC) data types to the data types you specify in Data API calls.

<table>
<thead>
<tr>
<th>JDBC data type</th>
<th>Data API data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER, TINYINT, SMALLINT, BIGINT</td>
<td>LONG</td>
</tr>
<tr>
<td>FLOAT, REAL, DOUBLE</td>
<td>DOUBLE</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>STRING</td>
</tr>
<tr>
<td>BOOLEAN, BIT</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>BLOB, BINARY, LONGVARBINARY, VARBINARY</td>
<td>BLOB</td>
</tr>
<tr>
<td>CLOB</td>
<td>STRING</td>
</tr>
<tr>
<td>Other types (including types related to date and time)</td>
<td>STRING</td>
</tr>
</tbody>
</table>

String values are passed to the Amazon Redshift database and implicitly converted into a database data type.

#### Note
Currently, the Data API doesn't support arrays of universal unique identifiers (UUIDs).

### Running SQL statements with parameters when calling the Amazon Redshift Data API

You can control the SQL text submitted to the database engine by calling the Data API operation using parameters for parts of the SQL statement. Named parameters provide a flexible way to pass in parameters without hardcoding them in the SQL text. They help you reuse SQL text and avoid SQL injection problems.

The following example shows the named parameters of a parameters field of an execute statement AWS CLI command.

```bash
--parameters "[{"name": "id", "value": "1"},{"name": "address", "value": "Seattle"}]
```

Consider the following when using named parameters:

- The named parameters can be in any order and parameters can be used more than one time in the SQL text. The parameters option shown in the previous example, the values 1 and Seattle are inserted into the table columns id and address. In the SQL text, you specify the named parameters as follows:

```bash
--sql "insert into mytable values (:id, :address)"
```

- When the SQL runs, data is implicitly cast to a data type. For more information about data type casting, see Data types in the Amazon Redshift Database Developer Guide.

- You can't set a value to NULL. The Data API interprets it as the literal string NULL. The following example replaces id with the literal string null. Not the SQL NULL value.
Calling the Data API

You can call the Data API or the AWS CLI to run SQL statements on your cluster. The primary operation to run an SQL statement is *ExecuteStatement*. The Data API supports the programming languages that are supported by the AWS SDK. For more information on these, see Tools to Build on AWS.

To see code examples of calling the Data API, see Getting Started with Redshift Data API in GitHub. This repository has examples of using AWS Lambda to access Amazon Redshift data from Amazon EC2, AWS Glue Data Catalog, and Amazon SageMaker. Example programing languages include Python, Go, Java, and Javascript.

Calling the Data API with the AWS CLI

You can call the Data API using the AWS CLI.

The following examples use the AWS CLI to call the Data API. To run the examples, edit the parameter values to match your environment. These examples demonstrate a few of the Data API operations. For more information, see the AWS CLI Command Reference.

Commands in the following examples have been split and formatted for readability.

**To run an SQL statement**

To run an SQL statement, use the `aws redshift-data execute-statement` AWS CLI command.

The following AWS CLI command runs an SQL statement and returns an identifier to fetch the results. This example uses the AWS Secrets Manager authentication method.

```bash
aws redshift-data execute-statement
   --region us-west-2
   --secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hRgPw
   --cluster-identifier mycluster-test
   --sql "select * from stl_query limit 1"
   --database dev
```

The following is an example of the response.

```json
{
   "ClusterIdentifier": "mycluster-test",
   "CreatedAt": 1598323175.823,
   "Database": "dev",
   "Id": null
}
```
The following AWS CLI command runs an SQL statement and returns an identifier to fetch the results. This example uses the temporary credentials authentication method.

```
aws redshift-data execute-statement
  --region us-west-2
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --sql "select * from stl_query limit 1"
```

The following is an example of the response.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Database": "dev",
  "DbUser": "myuser",
  "Id": "d9b6c0c9-0747-4bf4-b142-e883122f766"
}
```

To run an SQL statement with parameters

To run an SQL statement, use the `aws redshift-data execute-statement` AWS CLI command.

The following AWS CLI command runs an SQL statement and returns an identifier to fetch the results. This example uses the AWS Secrets Manager authentication method. The SQL text has named parameters `colname` and `distance`. In this case, the column name in the table is `ratecode` and the distance used in the predicate is `5`. Values for named parameters for the SQL statement are specified in the `parameters` option.

```
aws redshift-data execute-statement
  --region us-west-2
  --secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hKgPWn
  --cluster-identifier mycluster-test
  --sql "SELECT :colname, COUNT(*) FROM demo_table WHERE trip_distance > :distance"
  --parameters "[{"name": "colname", "value": "ratecode"}, {"name": "distance", "value": 5}]"
  --database dev
```

The following is an example of the response.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598323175.823,
  "Database": "dev",
  "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
}
```

The following example uses the `EVENT` table from the sample database. For more information, see `EVENT` table in the Amazon Redshift Database Developer Guide.
If you don't already have the EVENT table in your database, you can create one using the Data API as follows:

```bash
aws redshift-data execute-statement
  --database dev
  --cluster-id my-test-cluster
  --db-user awsuser
  --sql "create table event(eventid integer not null distkey,
  venueid smallint not null,
  catid smallint not null,
  dateid smallint not null sortkey,
  eventname varchar(200),
  starttime timestamp)"
```

The following command inserts one row into the EVENT table.

```bash
aws redshift-data execute-statement
  --database dev
  --cluster-id my-test-cluster
  --db-user awsuser
  --sql "insert into event
  values(:eventid, :venueid::smallint, :catid, :dateid, :eventname, :starttime)"
  --parameters "[{"name": "eventid", "value": "1"},
  {"name": "venueid", "value": "1"},
  {"name": "catid", "value": "1"},
  {"name": "dateid", "value": "1"},
  {"name": "eventname", "value": "event 1"},
  {"name": "starttime", "value": "2022-02-22"}]"
```

The following command inserts a second row into the EVENT table. This example demonstrates the following:

- The parameter named `id` is used four times in the SQL text.
- Implicit type conversion is applied automatically when inserting parameter `starttime`.
- The `venueid` column is type cast to SMALLINT data type.
- Character strings that represent the DATE data type are implicitly converted into the TIMESTAMP data type.
- Comments can be used within SQL text.

```bash
aws redshift-data execute-statement
  --database dev
  --cluster-id my-test-cluster
  --db-user awsuser
  --sql "insert into event values(:id, :id::smallint, :id, :id, :eventname, :starttime) /
  *this is comment, and it won't apply parameterization for :id, :eventname or :starttime here*/
  --parameters "[{"name": "eventname", "value": "event 2"},
  {"name": "starttime", "value": "2022-02-22"},
  {"name": "id", "value": "2"}]"
```

The following shows the two inserted rows:

```
<table>
<thead>
<tr>
<th>eventid</th>
<th>venueid</th>
<th>catid</th>
<th>dateid</th>
<th>eventname</th>
<th>starttime</th>
</tr>
</thead>
<tbody>
<tr>
<td>175</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

175
The following command uses a named parameter in a WHERE clause to retrieve the row where `eventid` is 1.

```bash
aws redshift-data execute-statement
--database dev
--cluster-id my-test-cluster
--db-user awsuser
--sql "select * from event where eventid=:id"
--parameters "[{"name": "id", "value": "1"}]"
```

Run the following command to get the SQL results of the previous SQL statement:

```bash
aws redshift-data get-statement-result --id 7529ad05-b905-4d71-9ec6-8b33836eb5a
```

Provides the following results:

```json
{
  "Records": [
    {
      "longValue": 1
    },
    {
      "longValue": 1
    },
    {
      "longValue": 1
    },
    {
      "longValue": 1
    },
    {
      "stringValue": "event 1"
    },
    {
      "stringValue": "2022-02-22 00:00:00.0"
    }
  ],
  "ColumnMetadata": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "eventid",
      "length": 0,
      "name": "eventid",
      "nullable": 0,
      "precision": 10,
      "scale": 0,
      "schemaName": "public",
      "tableName": "event",
      "typeName": "int4"
    }
  ]
}
```
{"isCaseSensitive": false, "isCurrency": false, "isSigned": true, "label": "venueid", "length": 0, "name": "venueid", "nullable": 0, "precision": 5, "scale": 0, "schemaName": "public", "tableName": "event", "typeName": "int2" },
{"isCaseSensitive": false, "isCurrency": false, "isSigned": true, "label": "catid", "length": 0, "name": "catid", "nullable": 0, "precision": 5, "scale": 0, "schemaName": "public", "tableName": "event", "typeName": "int2" },
{"isCaseSensitive": false, "isCurrency": false, "isSigned": true, "label": "dateid", "length": 0, "name": "dateid", "nullable": 0, "precision": 5, "scale": 0, "schemaName": "public", "tableName": "event", "typeName": "int2" },
{"isCaseSensitive": true, "isCurrency": false, "isSigned": false, "label": "eventname", "length": 0, "name": "eventname", "nullable": 1, "precision": 200, "scale": 0, "schemaName": "public", "tableName": "event", "typeName": "varchar" },
{"isCaseSensitive": true, "isCurrency": false, "isSigned": false, "label": "starttime", "length": 0, "name": "starttime", "nullable": 1, "precision": 29, "scale": 0, "schemaName": "public", "tableName": "event", "typeName": "varchar" },
To run multiple SQL statements

To run multiple SQL statements with one command, use the `aws redshift-data batch-execute-statement` AWS CLI command.

The following AWS CLI command runs three SQL statements and returns an identifier to fetch the results. This example uses the temporary credentials authentication method.

```bash
aws redshift-data batch-execute-statement
  --region us-west-2
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --sqls "set timezone to BST" "select * from mytable" "select * from another_table"
```

The following is an example of the response.

```json
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Database": "dev",
  "DbUser": "myuser",
  "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

To list metadata about SQL statements

To list metadata about SQL statements, use the `aws redshift-data list-statements` AWS CLI command. Authorization to run this command is based on the caller's IAM permissions.

The following AWS CLI command lists SQL statements that ran.

```bash
aws redshift-data list-statements
  --region us-west-2
  --status ALL
```

The following is an example of the response.

```json
{
  "Statements": [ 
    {
      "CreatedAt": 1598306924.632,
      "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
      "QueryString": "select * from stl_query limit 1",
      "Status": "FINISHED",
      "UpdatedAt": 1598306926.667
    },
    {
      "CreatedAt": 1598311717.437,
      "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
    }
  ]
}
```
To describe metadata about an SQL statement

To get descriptions of metadata for an SQL statement, use the `aws redshift-data describe-statement` AWS CLI command. Authorization to run this command is based on the caller's IAM permissions.

The following AWS CLI command describes an SQL statement.

```
aws redshift-data describe-statement
   --id d9b6c0c9-0747-4bf4-b142-e8883122f766
   --region us-west-2
```

The following is an example of the response.

```
{
   "ClusterIdentifier": "mycluster-test",
   "CreatedAt": 1598306924.632,
   "Duration": 1095981511,
   "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
   "QueryString": "select * from stl_query limit 1",
   "RedshiftPid": 20859,
   "RedshiftQueryId": 48879,
   "ResultRows": 1,
   "ResultSize": 4489,
   "Status": "FINISHED",
   "UpdatedAt": 1598306926.667
}
```

The following is an example of a `describe-statement` response after running a `batch-execute-statement` command with multiple SQL statements.

```
{
   "ClusterIdentifier": "mayo",
   "CreatedAt": 1623979777.126,
   "Duration": 6591877,
   "HasResultSet": true,
   "Id": "b2906c76-fa6e-4cdf-8c5f-4d1f9b7652",
   "RedshiftPid": 31459,
   "Status": "ABORTED",
   "UpdatedAt": 1598313685.495
}
```
To fetch the results of an SQL statement

To fetch the result from an SQL statement that ran, use the `redshift-data get-statement-result` AWS CLI command. You can provide an Id that you receive in response to `execute-statement` or `batch-execute-statement`. The Id value for an SQL statement run by `batch-execute-statement` can be retrieved in the result of `describe-statement` and is suffixed by a colon and sequence number such as `b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2`. If you run multiple SQL statements with `batch-execute-statement`, each SQL statement has an Id value as shown in `describe-statement`. Authorization to run this command is based on the caller's IAM permissions.

The following statement returns the result of an SQL statement run by `execute-statement`.

```
aws redshift-data get-statement-result
  --id d9b6c0c9-0747-4bf4-b142-e8883122f766
  --region us-west-2
```

The following statement returns the result of the second SQL statement run by `batch-execute-statement`.

```
aws redshift-data get-statement-result
  --id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
  --region us-west-2
```

The following is an example of the response to a call to `get-statement-result`.

```json
{
  "RedshiftQueryId": 0,
  "ResultRows": 2,
  "ResultSize": 22,
  "Status": "FINISHED",
  "SubStatements": [  
    {
      "CreatedAt": 1623979777.274,
      "Duration": 3396637,
      "HasResultSet": true,
      "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:1",
      "QueryString": "select 1;",
      "RedshiftQueryId": -1,
      "ResultRows": 1,
      "ResultSize": 11,
      "Status": "FINISHED",
      "UpdatedAt": 1623979777.903
    },
    {
      "CreatedAt": 1623979777.274,
      "Duration": 3195240,
      "HasResultSet": true,
      "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2",
      "QueryString": "select 2;",
      "RedshiftQueryId": -1,
      "ResultRows": 1,
      "ResultSize": 11,
      "Status": "FINISHED",
      "UpdatedAt": 1623979778.076
    }
  ],
  "UpdatedAt": 1623979778.183
}
```
"ColumnMetadata": [
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "userid",
    "length": 0,
    "name": "userid",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "query",
    "length": 0,
    "name": "query",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": true,
    "isCurrency": false,
    "isSigned": false,
    "label": "label",
    "length": 0,
    "name": "label",
    "nullable": 0,
    "precision": 320,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "bpchar"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "xid",
    "length": 0,
    "name": "xid",
    "nullable": 0,
    "precision": 19,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int8"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "pid",
    "length": 0,
    "name": "pid",
    "nullable": 0,
    "precision": 10,
"scale": 0,
"schemaName": "",
"tableName": "stll_query",
"typeName": "int4"
},
{
  "isCaseSensitive": true,
  "isCurrency": false,
  "isSigned": false,
  "label": "database",
  "length": 0,
  "name": "database",
  "nullable": 0,
  "precision": 32,
  "scale": 0,
  "schemaName": "",
  "tableName": "stll_query",
  "typeName": "bpchar"
},
{
  "isCaseSensitive": true,
  "isCurrency": false,
  "isSigned": false,
  "label": "querytxt",
  "length": 0,
  "name": "querytxt",
  "nullable": 0,
  "precision": 4000,
  "scale": 0,
  "schemaName": "",
  "tableName": "stll_query",
  "typeName": "bpchar"
},
{
  "isCaseSensitive": false,
  "isCurrency": false,
  "isSigned": false,
  "label": "starttime",
  "length": 0,
  "name": "starttime",
  "nullable": 0,
  "precision": 29,
  "scale": 6,
  "schemaName": "",
  "tableName": "stll_query",
  "type": 93,
  "typeName": "timestamp"
},
{
  "isCaseSensitive": false,
  "isCurrency": false,
  "isSigned": false,
  "label": "endtime",
  "length": 0,
  "name": "endtime",
  "nullable": 0,
  "precision": 29,
  "scale": 6,
  "schemaName": "",
  "tableName": "stll_query",
  "type": 93,
  "typeName": "timestamp"
},
{
  "isCaseSensitive": false,
  "isCurrency": false,
  "isSigned": true,
"label": "aborted",
"length": 0,
"name": "aborted",
"nullable": 0,
"precision": 10,
"scale": 0,
"schemaName": "",
"tableName": "stll_query",
"typeName": "int4"
},
{
"isCaseSensitive": false,
"isCurrency": false,
"isSigned": true,
"label": "insert_pristine",
"length": 0,
"name": "insert_pristine",
"nullable": 0,
"precision": 10,
"scale": 0,
"schemaName": "",
"tableName": "stll_query",
"typeName": "int4"
},
{
"isCaseSensitive": false,
"isCurrency": false,
"isSigned": true,
"label": "concurrency_scaling_status",
"length": 0,
"name": "concurrency_scaling_status",
"nullable": 0,
"precision": 10,
"scale": 0,
"schemaName": "",
"tableName": "stll_query",
"typeName": "int4"
}
],
"Records": [
[
  {
    "longValue": 1
  }
],
{
  "longValue": 3
},
{
  "stringValue": "health"
},
{
  "longValue": 1023
},
{
  "longValue": 15279
},
{
  "stringValue": "dev"
},
{
  "stringValue": "select system_status from stv_gui_status;"
},
{
  "stringValue": "2020-08-21 17:33:51.88712"
}
To describe a table

To get metadata that describes a table, use the `aws redshift-data describe-table` AWS CLI command.

The following AWS CLI command runs an SQL statement and returns metadata that describes a table. This example uses the AWS Secrets Manager authentication method.

```
aws redshift-data describe-table
  --region us-west-2
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
  --table sql_features
  --secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hKgPwn
```

The following is an example of the response.

```
{
  "ColumnList": [
    
  
  ]
}
```
The following AWS CLI command runs an SQL statement that describes a table. This example uses the temporary credentials authentication method.

```
aws redshift-data describe-table
  --region us-west-2
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
  --table sql_features
```

The following is an example of the response.

```
{
  "ColumnList": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_id",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_name",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "sub_feature_id",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    }
  ]
}
```
To list the databases in a cluster

To list the databases in a cluster, use the `aws redshift-data list-databases` AWS CLI command.

The following AWS CLI command runs an SQL statement to list databases. This example uses the AWS Secrets Manager authentication method.

```
aws redshift-data list-databases
  --region us-west-2
  --secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hRgPWN
  --cluster-identifier mycluster-test
```
The following is an example of the response.

```json
{
    "Databases": [
        "dev"
    ]
}
```

The following AWS CLI command runs an SQL statement to list databases. This example uses the temporary credentials authentication method.

```bash
aws redshift-data list-databases
--region us-west-2
--db-user myuser
--cluster-identifier mycluster-test
--database dev
```

The following is an example of the response.

```json
{
    "Databases": [
        "dev"
    ]
}
```

**To list the schemas in a database**

To list the schemas in a database, use the `aws redshift-data list-schemas` AWS CLI command.

The following AWS CLI command runs an SQL statement to list schemas in a database. This example uses the AWS Secrets Manager authentication method.

```bash
aws redshift-data list-schemas
--region us-west-2
--secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hKgPWn
--cluster-identifier mycluster-test
--database dev
```

The following is an example of the response.

```json
{
    "Schemas": [
        "information_schema",
        "pg_catalog",
        "pg_internal",
        "public"
    ]
}
```

The following AWS CLI command runs an SQL statement to list schemas in a database. This example uses the temporary credentials authentication method.

```bash
aws redshift-data list-schemas
```
The following is an example of the response.

```
{
  "Schemas": [
    "information_schema",
    "pg_catalog",
    "pg_internal",
    "public"
  ]
}
```

**To list the tables in a database**

To list the tables in a database, use the `aws redshift-data list-tables` AWS CLI command.

The following AWS CLI command runs an SQL statement to list tables in a database. This example uses the AWS Secrets Manager authentication method.

```
aws redshift-data list-tables
  --region us-west-2
  --secret arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hRgPWh
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
```

The following is an example of the response.

```
{
  "Tables": [
    {
      "name": "sql_features",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    },
    {
      "name": "sql_implementation_info",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

The following AWS CLI command runs an SQL statement to list tables in a database. This example uses the temporary credentials authentication method.

```
aws redshift-data list-tables
  --region us-west-2
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
```
Troubleshooting Data API issues

Use the following sections, titled with common error messages, to help troubleshoot problems that you have with the Data API.

Topics

• Packet for query is too large (p. 189)
• Database response exceeded size limit (p. 189)

Packet for query is too large

If you see an error indicating that the packet for a query is too large, generally the result set returned for a row is too large. The Data API size limit is 64 KB per row in the result set returned by the database.

To solve this issue, make sure that each row in a result set is 64 KB or less.

Database response exceeded size limit

If you see an error indicating that the database response has exceeded the size limit, generally the size of the result set returned by the database was too large. The Data API limit is 100 MB in the result set returned by the database.

To solve this issue, make sure that calls to the Data API return 100 MB of data or less. If you need to return more than 100 MB, you can run multiple statement calls with the LIMIT clause in your query.

Scheduling Amazon Redshift Data API operations with Amazon EventBridge

Amazon EventBridge helps you to respond to state changes in your AWS resources. When your resources change state, they automatically send events into an event stream. Events are sent to the account that contains the Amazon Redshift database. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a predetermined schedule. For more information, see the Amazon EventBridge User Guide.

To schedule Data API operations with EventBridge, the associated IAM role must trust the principal for CloudWatch Events (events.amazonaws.com). This role should have the equivalent of the managed policy AmazonEventBridgeFullAccess attached. It should also have AmazonRedshiftDataFullAccess policy permissions that are managed by the Data API. You can create an IAM role with these permissions.
on the IAM console. When creating a role on the IAM console, choose the AWS service trusted entity for CloudWatch Events. For more information about creating an IAM role, see Creating a Role for an AWS Service (Console) in the IAM User Guide.

The following example uses the AWS CLI to create an EventBridge rule that is used to run an SQL statement.

```
aws events put-rule
  --name test-redshift-data
  --schedule-expression "rate(1 minute)"
```

Then an EventBridge target is created to run on the schedule specified in the rule.

```
aws events put-targets
  --cli-input-json file://data.json
```

The input data.json file is as follows.

```
{
  "Rule": "test-redshift-data",
  "EventBusName": "default",
  "Targets": [
    {
      "Id": "2",
      "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
      "RedshiftDataParameters": {
        "Database": "dev",
        "DbUser": "root",
        "Sql": "select 1;",
        "StatementName": "test-scheduler-statement",
        "WithEvent": true
      }
    }
  ]
}
```

Monitoring the Data API

Monitoring is an important part of maintaining the reliability, availability, and performance of the Data API and your other AWS solutions. AWS provides the following monitoring tools to watch the Data API, report when something is wrong, and take automatic actions when appropriate:

- Amazon EventBridge can be used to automate your AWS services and respond automatically to system events, such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near-real time. You can write simple rules to indicate which events are of interest to you and which automated actions to take when an event matches a rule. For more information, see the Amazon EventBridge User Guide.
- AWS CloudTrail captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics

- Monitoring events for the Amazon Redshift Data API in Amazon EventBridge (p. 191)
- Logging Amazon Redshift Data API calls with AWS CloudTrail (p. 191)
Monitoring events for the Amazon Redshift Data API in Amazon EventBridge

You can monitor Data API events in EventBridge, which delivers a stream of real-time data from your own applications, software-as-a-service (SaaS) applications, and AWS services. EventBridge routes that data to targets such as AWS Lambda and Amazon SNS. These events are the same as those that appear in CloudWatch Events, which delivers a near-real time stream of system events that describe changes in AWS resources. Events are sent to the account that contains the Amazon Redshift database. For example, if you assume a role in another account, events are sent to that account. For more information, see AWS Events in the Amazon EventBridge User Guide.

Data API events are sent when the ExecuteStatement API operation sets the WithEvent option to true. The state field of the event contains one of the following values:

- FAILED – The query run failed.
- FINISHED – The query has finished running.

The following examples show an event for Data API when the ExecuteStatement API operation finishes.

Example for Data API finished event

In the following example event, a statement named test.testtable finished running.

```json
{
    "version": "0",
    "id": "18e7079c-dd4b-dd64-caf9-e2a31640dab0",
    "detail-type": "Redshift Data Statement Status Change",
    "source": "aws.redshift-data",
    "account": "123456789012",
    "time": "2020-10-01T21:14:26Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-cluster-1"
    ],
    "detail": {
        "principal": "arn:aws:iam::123456789012:user/myuser",
        "statementName": "test.testtable",
        "statementId": "dd2e1ec9-2ee3-49a0-819f-905fa7d75a4a",
        "redshiftQueryId": -1,
        "state": "FINISHED",
        "rows": 1,
        "expireAt": 1601673265
    }
}
```

Logging Amazon Redshift Data API calls with AWS CloudTrail

Amazon Redshift Data API is integrated with AWS CloudTrail. CloudTrail is a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Redshift Data API. CloudTrail captures all API calls for Amazon Redshift Data API as events. The calls captured include calls from the Amazon Redshift Data API console and code calls to the Amazon Redshift Data API operations.

If you create a CloudTrail trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Redshift Data API. 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 certain things. These include the request that was made to Amazon
Redshift Data API, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

**Working with Data API information in CloudTrail**

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Redshift Data API, 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 in the AWS CloudTrail User Guide.

For an ongoing record of events in your AWS account, including events for Amazon Redshift Data API, 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 AWS 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 the following in the AWS CloudTrail User Guide:

- 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 Data API actions are logged by CloudTrail and are documented in the Amazon Redshift Data API Reference. For example, calls to the ExecuteStatement, GetStatementResults and CancelStatement 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 CloudTrail userIdentity Element in the AWS CloudTrail User Guide.

**Understanding log file entries for Data API**

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 aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the ExecuteStatement action.

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
    "arn": "arn:aws:sts::123456789012:user/janedoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAI44QH8DH8EXAMPLE",
    "userName": "janedoe"
  },
  "eventSource": "redshift.amazonaws.com",
  "eventTime": "2023-09-28T20:45:45Z",
  "requestParameters": {},
  "responseElements": {
    "status": "OK",
    "resultCode": "Success",
    "columnName": "status",
    "columnValue": "OK"
  },
  "requestID": "0f33c3d4-89e4-40e8-91ca-c6804f97ead4",
  "eventID": "0f33c3d4-89e4-40e8-91ca-c6804f97ead4",
  "eventBusName": "default",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.168.0.1"
}
```
Enhanced VPC routing in Amazon Redshift

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 virtual private cloud (VPC) based on the Amazon VPC service. 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. 195).

- **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. 195)
- Enabling enhanced VPC routing (p. 195)
- Using Amazon Redshift Spectrum with enhanced VPC routing (p. 197)
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. 71).

- 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.

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

New console

To create a cluster with enhanced VPC routing

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose Create cluster and enter the Cluster details properties.
3. To display the Additional configurations section, choose to switch off Use defaults.
4. To enable Enhanced VPC routing select Enabled to force cluster traffic through the VPC.
5. Choose Create cluster to create the cluster. The cluster might take several minutes to be ready to use.

Original console

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. 44).

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. 54).
Amazon Redshift Cluster Management Guide
Using Amazon Redshift Spectrum
with enhanced VPC routing

**Note**
When you modify a cluster to enable enhanced VPC routing, the cluster automatically restarts to apply the change.

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
Considerations for using enhanced VPC routing for Redshift Spectrum

Following are considerations when using Redshift Spectrum enhanced VPC routing:

- Bucket access policies (p. 198)
- Cluster IAM role (p. 199)
- Logging and auditing Amazon S3 access (p. 199)
- Access to AWS Glue or Amazon Athena (p. 200)

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. 344).

To use Redshift Spectrum, no IAM policies blocking the use of presigned URLs can be in place.

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.
Access to AWS Glue or Amazon Athena

Redshift Spectrum accesses your data catalog in AWS Glue or Athena. Another option is to use a dedicated Hive metastore for your data catalog.

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. Alternatively, you can configure an interface VPC endpoint for AWS Glue to access your AWS Glue Data Catalog. When you use a VPC interface endpoint, communication between your VPC and AWS Glue is conducted within the AWS network. For more information, see Creating an Interface Endpoint.

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 Enhanced VPC routing in Amazon Redshift (p. 194).
Amazon Redshift parameter groups

Overview

In Amazon Redshift, you associate a parameter group with each cluster that you create. A 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 date style.

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. In some cases, you might modify parameter values in a parameter group that is already associated with a cluster or associate a different parameter group with a cluster. In these cases, 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. Changes aren't applied if your cluster is restarted during maintenance. For more information, see WLM dynamic and static properties (p. 204).

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.
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.

### Parameter name | Value | More information
--- | --- | ---
auto_analyze | true | auto_analyze in the Amazon Redshift Database Developer Guide

datestyle | ISO, MDY | datestyle in the Amazon Redshift Database Developer Guide

enable_case_sensitive_identifier | false | enable_case_sensitive_identifier in the Amazon Redshift Database Developer Guide

enable_user_activity_logging | false | Database audit logging (p. 353) in this guide

extra_float_digits | 0 | extra_float_digits in the Amazon Redshift Database Developer Guide

max_concurrency_scaling_clusters | 1 | max_concurrency_scaling_clusters in the Amazon Redshift Database Developer Guide

query_group | default | query_group in the Amazon Redshift Database Developer Guide

require_ssl | false | Configuring security options for connections (p. 144) in this guide

search_path | $user, public | search_path in the Amazon Redshift Database Developer Guide

statement_timeout | 0 | statement_timeout in the Amazon Redshift Database Developer Guide

wlm_json_configuration | ["auto_wlm":true] | Configuring workload management (p. 203) 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 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 queue.

#### Configuring workload management using the console

You can manage parameter groups using the console. For more information, see [Managing parameter groups using the console](#).

#### Configuring workload management using the AWS CLI

You can manage WLM using the AWS CLI. For more information, see [Configuring workload management](#).
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 can specify mode and concurrency level (query slots) for the default queue, but you can't 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. To modify the WLM configuration, create a new 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. You can apply dynamic properties to the database without a cluster reboot, but static properties require a cluster reboot for changes to take effect. For more information about static and dynamic properties, see WLM dynamic and static configuration properties.

### Properties for the wlm_json_configuration parameter

You can configure WLM by using the Amazon Redshift console, the AWS CLI, the 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.

**Note**

The following properties appear with their Amazon Redshift console names, with the corresponding JSON property names in the descriptions.

<table>
<thead>
<tr>
<th>WLM property</th>
<th>Automatic WLM</th>
<th>Manual WLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto WLM</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Enable short query acceleration</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum run time for short queries</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Priority</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Queue type</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Queue name</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Concurrency Scaling mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Concurrency</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>User groups</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>User group wildcard</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Query groups</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Query group wildcard</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The following list describes the WLM properties that you can configure for each queue.

**Auto WLM**

Auto WLM set to true enables automatic WLM. Automatic WLM sets the values for Concurrency on main and Memory (%) to Auto. Amazon Redshift manages query concurrency and memory allocation. The default is true.

JSON property: auto_wlm

**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

**Priority**

Priority sets the priority of queries that run in a queue. To set the priority, WLM mode must be set to Auto WLM; that is, auto_wlm must be true. Priority values can be highest, high, normal, low, and lowest. The default is normal.

JSON property: priority

**Queue type**

Queue type designates a queue as used either by Auto WLM or Manual WLM. Set queue_type to either auto or manual. If not specified, the default is manual.

JSON property: queue_type

**Queue name**

The name of the queue. You can set the name of the queue based on your business needs. Queue names must be unique within an WLM configuration, are up to 64 alphanumeric characters, underscores or spaces, and can't contain quotation marks. For example, if you have a queue for your ETL queries, you might name it ETL queue. This name is used in metrics, system table values, and the Amazon Redshift console to identify the queue. Queries and reports that use the name from these sources need to be able to handle changes of the name. Previously, the queue names were generated by Amazon Redshift. The default names of queues are Queue 1, Queue 2, to the last queue named Default queue.
Important
If you change a queue name, the QueueName dimension value of WLM queue metrics (such as, WLMQueueLength, WLMQueueWaitTime, WLMQueriesCompletedPerSecond, WLMQueryDuration, WLMRunningQueries, and so on) also changes. So, if you change the name of a queue, you might need to change CloudWatch alarms you have set up.

JSON property: name

Concurrency Scaling mode
To enable concurrency scaling on a queue, set Concurrency Scaling mode to auto. When the number of queries routed to a queue exceeds the queue's configured concurrency, eligible queries go to the scaling cluster. When slots become available, queries run on the main cluster. The default is off.

JSON property: concurrency_scaling

Concurrency
The number of queries that can run concurrently in a manual WLM queue. This property only applies to manual WLM. If concurrency scaling is enabled, eligible queries go to a scaling cluster when a queue reaches the concurrency level (query slots). If concurrency scaling isn't enabled, queries wait in the queue until a slot becomes available. 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)
WLM timeout (max_execution_time) is deprecated. It is not available when using automatic WLM. Instead, create a query monitoring rule (QMR) using query_execution_time to limit the elapsed execution time for a query. For more information, see WLM query monitoring rules.

The maximum time, in milliseconds, that queries can run before being canceled. In some cases, a read-only query, such as a SELECT statement, might be canceled due to a WLM timeout. In these cases, 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 isn't 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 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. The service can temporarily give this unallocated memory 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 25 rules per queue, and the total limit for all queues is 25 rules.

JSON property: rules

JSON properties hierarchy:

```
rules
  rule_name
  predicate
    metric_name
    operator
    value
  action
    value
```

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 (only available with manual WLM)
  • abort
• change_query_priority (only available with automatic WLM)

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_execution_time",
                "operator": ">",
                "value": 100000
            },
            {
                "metric_name": "query_blocks_read",
                "operator": ">",
                "value": 1000
            }
        ],
        "action": "hop"
    }
]
```

For more information about each of these properties and strategies for configuring query queues, see 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 maximum size of the `wlm_json_configuration` property value is 8000 characters. 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 don't 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 console, see Modifying a parameter group (p. 214).

**Example**

The following example is the default WLM configuration, which defines one queue with automatic WLM.

```
{
    "auto_wlm": true
}
```

**Example**

The following example is a custom WLM configuration, which defines one manual WLM queue with a concurrency level (query slots) 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 manual 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). mode is set to auto, so when the queue's query slots are full eligible queries are sent to a scaling cluster.

- 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. mode is set to off, so all queries sent by members of the admin or dba groups run on the main cluster.

- 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. mode is set to auto.

**Note**
The example is shown on several lines for demonstration purposes. Actual commands should not have line breaks.

```bash
aws redshift modify-cluster-parameter-group
--parameter-group-name example-parameter-group
--parameters
'[
    {
      "query_concurrency": 4,
      "max_execution_time": 20000,
      "memory_percent_to_use": 25,
      "query_group": ["report"],
      "query_group_wild_card": 1,
      "user_group": [],
      "user_group_wild_card": 0,
      "concurrency_scaling": "auto",
      "queue_type": "manual"
    },
    {
      "query_concurrency": 5,
      "memory_percent_to_use": 40,
      "query_group": [],
      "query_group_wild_card": 0,
      "user_group": [
        "admin",
        "dba"
      ],
      "user_group_wild_card": 0,
      "concurrency_scaling": "off",
      "queue_type": "manual"
    },
    {
      "query_concurrency": 5,
      "query_group": [],
      "query_group_wild_card": 0,
      "user_group": [],
      "user_group_wild_card": 0,
      "concurrency_scaling": "auto",
      "queue_type": "manual"
    },
    {"short_query_queue": true}
]
```

The following is an example of configuring WLM query monitoring rules for an automatic WLM configuration. The example creates a parameter group named example-monitoring-rules. The configuration defines the same three queues as the previous example, but the query_concurrency and
memory_percent_to_use are not specified anymore. The configuration also adds the following rules and query priorities:

- 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 priority of this queue is `Normal`.

- 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 `abort`. The priority of this queue is `Highest`.

- The last queue in the configuration is the default queue. The priority of this queue is `Low`.

**Note**

The example is shown on several lines for demonstration purposes. Actual commands should not have line breaks.

```bash
aws redshift modify-cluster-parameter-group
--parameter-group-name example-monitoring-rules
--parameters
'[ {
  "query_group" : [ "report" ],
  "query_group_wild_card" : 1,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "concurrency_scaling" : "auto",
  "rules" : [ {
    "rule_name" : "rule_1",
    "predicate" : [{
      "metric_name" : "query_cpu_time",
      "operator" : ">",
      "value" : 10000000
    },
    { "metric_name" : "query_blocks_read",
      "operator" : ">",
      "value" : 1000
    } ],
    "action" : "log"
  } ],
  "priority" : "normal",
  "queue_type" : "auto"
}],
{
  "query_group" : [ ],
  "query_group_wild_card" : 0,
  "user_group" : [ "admin", "dba" ],
  "user_group_wild_card" : 0,
  "concurrency_scaling" : "off",
  "rules" : [ {
    "rule_name" : "rule_2",
    "predicate" : [
      { "metric_name" : "query_execution_time",
        "operator" : ">",
        "value" : 600000000
      },
      { "metric_name" : "scan_row_count",
        "operator" : ">",
        "value" : 1000000000
      },
      { "metric_name" : "query_cpu_time",
        "operator" : ">",
        "value" : 10000000
      }
    ],
    "action" : "abort",
    "priority" : "high",
    "queue_type" : "auto"
  } ],
  "query_group" : [ ],
  "query_group_wild_card" : 0,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "concurrency_scaling" : "auto"
}],
{
  "query_group" : [ ],
  "query_group_wild_card" : 0,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "concurrency_scaling" : "auto"
}
]'
```
Configuring WLM by using the AWS CLI in the command line with a JSON file

You can modify the `wlm_json_configuration` parameter using the AWS CLI and pass in the value of the `parameters` argument as a JSON file.

```bash
aws redshift modify-cluster-parameter-group --parameter-group-name myclusterparametergroup --parameters file://modify_pg.json
```

The arguments for `--parameters` are stored in file `modify_pg.json`. The file location is specified in the format for your operating system. For more information, see Loading parameters from a file. The following shows examples of the content of the `modify_pg.json` JSON file.

```
[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "["user_group": "example_user_group1", "query_group": "example_query_group1", "query_concurrency": 7], ["query_concurrency": 5]"
  }
]
```

```
[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "["query_group": ["reports"], "query_group_wild_card": 0, "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 the command line on the Linux and macOS X operating systems

Follow these rules to run an AWS CLI command with parameters on one line:

- 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 ({}).

Rules for configuring WLM by using the AWS CLI in Windows PowerShell on Microsoft Windows operating systems

Follow these rules to run an AWS CLI command with parameters on one line:
• 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 ({}).

Rules for configuring WLM by using the command prompt on Windows operating systems

Follow these rules to run an AWS CLI command with parameters on one line:
• 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 ({}).

Managing parameter groups using the console

You can view, create, modify, and delete parameter groups on the Amazon Redshift console.

You can view any of your parameter groups to see a summary of the values for parameters and workload management (WLM) configuration. Group parameters appear on the Parameters tab, and Workload queues appear on the Workload Management tab.
Creating a parameter group

If you want to set parameter values that are different from the default parameter group, you can create your own parameter group,

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

**New console**

**To create 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. On the navigation menu, choose **CONFIG**, then choose **Workload management** to display the Workload management page.
3. Choose **Create** to display the Create parameter group window.
4. Enter a value for **Parameter group name** and **Description**.
5. Choose **Create** to create the parameter group.

**Original console**

**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. Then enter a parameter group name and a parameter group description. For more information about naming constraints for parameter groups, see Quotas and limits in Amazon Redshift (p. 468).
5. Choose **Create**.

**Modifying a parameter group**

You can modify parameters to change the parameter settings and WLM configuration properties.

**Note**
You can't modify the default parameter group.
**Note**
A new console is available for Amazon Redshift. Choose either the **New console** or the **Original console** instructions based on the console that you are using. The **New console** instructions are open by default.

**New console**

**To modify 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. On the navigation menu, choose **CONFIG**, then choose **Workload management** to display the **Workload management** page.
3. Choose the parameter group that you want to modify to display the details page, with tabs for **Parameters** and **Workload management**.
4. Choose the **Parameters** tab to view the current parameter settings.
5. Choose **Edit parameters** to enable changing settings for these parameters:
   - `auto_analyze`
   - `datestyle`
   - `enable_case_sensitive_identifier`
   - `enable_user_activity_logging`
   - `extra_float_digits`
   - `max_concurrency_scaling_clusters`
   - `max_cursor_result_set_size`
   - `query_group`
   - `require_ssl`
   - `search_path`
   - `statement_timeout`
   - `use_fips_ssl`

   For more information about these parameters, see [Amazon Redshift parameter groups](p. 201).
6. Enter your changes and then choose **Save** to update the parameter group.

**To modify the WLM configuration for 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. On the navigation menu, choose **CONFIG**, then choose **Workload management** to display the **Workload management** page.
3. Choose the parameter group that you want to modify to display the details page with tabs for **Parameters** and **Workload management**.
4. Choose the **Workload management** tab to view the current WLM configuration.
5. Choose **Edit workload queues** to edit the WLM configuration,
6. (Optional) Select **Enable short query acceleration** to enable short query acceleration (SQA).

   When you enable SQA, **Maximum run time for short queries (1 to 20 seconds)** is set to **Dynamic** by default. To set the maximum runtime to a fixed value, choose a value of 1–20.
7. Do one or more of the following to modify the queue configuration:
   - Choose **Switch WLM mode** to choose between **Automatic WLM** and **Manual WLM**.
With **Automatic WLM**, the **Memory** and **Concurrency on main** values are set to **auto**.

- To create a queue, choose **Edit workload queues**, then choose **Add Queue**.
- To modify a queue, change property values in the table. Depending on the type of queue, properties can include the following:
  - **Queue name** can be changed.
  - **Memory (%)**
  - **Concurrency on main cluster**
  - **Concurrency scaling mode** can be **off** or **auto**
  - **Timeout (ms)**
  - **User groups**
  - **Query groups**

For more information about these properties, see [Properties for the wlm_json_configuration parameter](p. 204).

**Important**

If you change a queue name, the **QueueName** dimension value of WLM queue metrics (such as, WLMQueueLength, WLMQueueWaitTime, WLMQueriesCompletedPerSecond, WLMQueryDuration, WLMRunningQueries, and so on) also changes. So, if you change the name of a queue, you might need to change CloudWatch alarms you have set up.

- To change the order of queues, choose the **Up** and **Down** arrow buttons.
- To delete a queue, choose **Delete** in the queue's row in the table.

8. (Optional) Select **Defer dynamic changes until reboot** to have the changes applied to clusters after their next reboot.

  **Note**
  
  Some changes require a cluster reboot regardless of this setting. For more information, see [WLM dynamic and static properties](p. 204).

9. Choose **Save**.

**Original console**

**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/](https://console.aws.amazon.com/redshift/).

   In the navigation pane, choose **Parameter Groups**.

2. On the **Parameter Groups** page, in the parameter group list, select the row of the parameter group that you want to modify.

3. To edit the parameters other than the WLM configuration parameter, choose **Edit Parameters**.

   The **Parameters** tab opens and enables you to update parameters in the parameter group. You can update values for parameters such as the following:

   - **auto_analyze**
   - **datestyle**
   - **enable_user_activity_logging**
   - **extra_float_digits**
   - **force_acm**
   - **max_concurrency_scaling_clusters**
   - **query_group**
• require_ssl
• search_path
• statement_timeout
• use_fips_ssl

4. In the Value box that corresponds to the parameter you want to modify, enter a new value. For more information about these parameters, see Amazon Redshift parameter groups (p. 201).

5. 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. 58).

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 can't modify the default parameter group.

4. Choose Edit

5. (Optional) Select Enable short query acceleration to enable short query acceleration (SQA).

   When you enable SQA, Maximum run time for short queries (1 to 20 seconds) is set to Dynamic by default. To set the maximum runtime to a fixed value, choose a value of 1–20.

6. Do one or more of the following to modify the queue configuration:

   • Choose Switch WLM mode to choose between Auto WLM and Manual WLM.

     With Auto WLM, the Memory and Concurrency on main values are set to auto.

     • To create a queue, choose Add Queue.

     • To modify the Max Concurrency Scaling clusters parameter, choose Edit next to the current value that is displayed.

     • To modify a queue, change property values in the table. Depending on the type of queue, properties can include the following:

       • Memory (%)

       • Concurrency on main cluster

       • Concurrency Scaling mode can be off or auto

       • Timeout (ms)

       • User groups

       • Query groups

     • 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.

7. (Optional) Select Defer dynamic changes until reboot to have the changes applied to clusters after their next reboot.

   Note
   Some changes require a cluster reboot regardless of this setting. For more information, see WLM dynamic and static properties (p. 204).
Creating or modifying a query monitoring rule using the console

You can use the Amazon Redshift console to create and modify WLM query monitoring rules. Query monitoring rules are part of the WLM configuration parameter for a parameter group. If you modify a query monitoring rule (QMR), the change happens automatically without the need to modify the cluster. 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.

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

New console

To create 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. On the navigation menu, choose CONFIG, then choose Workload management to display the Workload management page.
3. Choose the parameter group that you want to modify to display the details page with tabs for Parameters and Workload management.
4. Choose the Workload management tab, and choose Edit workload queues to edit the WLM configuration,
5. Add a new rule either by using a predefined template or from scratch.

   To use a predefined template, do the following:
   
   1. Choose Add rule from template in the Query monitoring rules group. The list of rule templates is displayed.
   2. Choose one or more rule templates. When you choose Save, WLM creates one rule for each template that you choose.
   3. Enter or confirm values for the rule, including Rule names, Predicates and Actions.
   4. Choose Save.

   To add a new rule from scratch, do the following:

   1. To add additional predicates, choose Add predicate. You can have up to three predicates for each rule. If all of the predicates are met, WLM triggers the associated action.
   2. Choose an Action. Each rule has one action.
   3. Choose Save.

Amazon Redshift generates your WLM configuration parameter in JSON format and displays it in the JSON section.
Original console

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 (to add a rule), choose Edit.
5. 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 box appears, as shown in the following screenshot.

6. 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.

7. Enter a value for Rule name. The name can be up to 32 alphanumeric characters and must not contain spaces or quotation mark characters. For this example, enter HighI0skew.
8. (Optional) Modify the rule predicates.
9. Choose a value for **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.

10. Choose **Save**.

11. To modify the rules for a queue, choose **Edit**.

12. To add a new rule from scratch, choose **Add Custom Rule**. You can have a maximum of five rules per queue, and a total of eight rules for all queues.

13. Enter a **Rule name**, for example **NestedLoop**.

14. Define a **Predicate** value. 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.

15. 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.

16. 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.

17. Choose **Done Editing**. The queue details collapse.

18. Choose **Save**.

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.
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.

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

New console

To 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. On the navigation menu, choose CONFIG, then choose Workload management to display the Workload management page.
3. For Parameter groups, choose the parameter group that you want to modify.
   
   Note
   You can't delete the default parameter group.
4. Choose Delete and confirm that you want to delete the parameter group.

Original console

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 can't 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 a launch cluster (p. 46) and To modify a cluster (p. 55).

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.

```json
[  
  {  
    "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.
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.

```json
{
  "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. 391). You need to update the code and provide a cluster identifier.

### Example

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * 
 * http://aws.amazon.com/apache2.0/
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[CreateAndModifyClusterParameterGroup demonstrates how to create and modify an Amazon Redshift parameter group.]
// snippet-service:[redshift]
// snippet-keyword:[Java]
// snippet-keyword:[Amazon Redshift]
// snippet-keyword:[Code Sample]
// snippet-keyword:[CreateClusterParameterGroup]
// snippet-keyword:[DescribeClusterParameterGroups]
// snippet-keyword:[ModifyClusterParameterGroup]
// snippet-sourcetype:[full-example]
// snippet-sourcedate:[2019-02-01]
// snippet-sourceauthor:[AWS]
// snippet-start:[redshift.java.CreateAndModifyClusterParameterGroup.complete]

package com.amazonaws.services.redshift;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterParameterGroup {

```
public static AmazonRedshift client;
public static String clusterParameterGroupName = "parametergroup1";
public static String clusterIdentifier = "***provide a cluster identifier***";
public static String parameterGroupFamily = "redshift-1.0";

public static void main(String[] args) throws IOException {
    // Default client using the @link
    com.amazonaws.auth.DefaultAWSCredentialsProviderChain
    client = AmazonRedshiftClientBuilder.defaultClient();
    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",}
Managing parameter groups using the AWS CLI and Amazon Redshift API

You can use the following Amazon Redshift operations in the AWS CLI to manage parameter groups.

- `create-cluster-parameter-group`
Managing parameter groups using the AWS CLI and Amazon Redshift API

- 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 API operations to manage parameter groups.

- CreateClusterParameterGroup
- DeleteClusterParameterGroup
- DescribeClusterParameters
- DescribeClusterParameterGroups
- DescribeDefaultClusterParameters
- ModifyClusterParameterGroup
- ResetClusterParameterGroup
Amazon Redshift snapshots

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.

Amazon Redshift automatically takes incremental snapshots that track changes to the cluster since the previous automated snapshot. Automated snapshots retain all of the data required to restore a cluster from a snapshot. You can create a snapshot schedule to control when automated snapshots are taken, or you can take a manual snapshot any time.

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.

When you launch a cluster, you can set the retention period for automated and manual snapshots. You can change the default retention period for automated and manual snapshots by modifying the cluster. You can change the retention period for a manual snapshot when you create the snapshot or by modifying the 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.

To ensure that your backups are always available to your cluster, Amazon Redshift stores snapshots in an internally managed Amazon S3 bucket that is managed by Amazon Redshift. To manage storage charges, evaluate how many days you need to keep automated snapshots and configure their retention period.
Automated snapshots

When automated snapshots are enabled for a cluster, Amazon Redshift periodically takes snapshots of that cluster. By default Amazon Redshift takes a snapshot about every eight hours or following every 5 GB per node of data changes, or whichever comes first. Alternatively, you can create a snapshot schedule to control when automated snapshots are taken. Automated snapshots are enabled by default when you create a cluster.

Automated 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 or CLI.

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. You can't disable automated snapshots for RA3 node types. You can set an RA3 node type automated retention period from 1–35 days.

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 for the cluster, 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 the retention period, but the corresponding manual snapshot is retained until you manually delete it or until the end of the retention period.

Automated snapshot schedules

To precisely control when snapshots are taken, you can create a snapshot schedule and attach it to one or more clusters. When you modify a snapshot schedule, the schedule is modified for all associated clusters. If a cluster doesn't have a snapshot schedule attached, the cluster uses the default automated snapshot schedule.

A snapshot schedule is a set of schedule rules. You can define a simple schedule rule based on a specified interval, such as every 8 hours or every 12 hours. You can also add rules to take snapshots on certain days of the week, at specific times, or during specific periods. Rules can also be defined using Unix-like cron expressions.

Snapshot schedule format

On the Amazon Redshift console, you can create a snapshot schedule. Then, you can attach a schedule to a cluster to trigger the creation of a system snapshot. A schedule can be attached to multiple clusters, and you can create multiple cron definitions in a schedule to trigger a snapshot.

You can define a schedule for your snapshots using a cron syntax. The definition of these schedules uses a modified Unix-like cron syntax. You specify time in Coordinated universal time (UTC). You can create schedules with a maximum frequency of one hour and minimum precision of one minute.

Amazon Redshift modified cron expressions have 3 required fields, which are separated by white space.
Syntax

cron(\texttt{Minutes Hours Day-of-week})

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
<th>Wildcards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes</td>
<td>0–59</td>
<td></td>
</tr>
<tr>
<td>Hours</td>
<td>0–23</td>
<td>, - */</td>
</tr>
<tr>
<td>Day-of-week</td>
<td>1–7 or SUN-SAT</td>
<td>, - */</td>
</tr>
</tbody>
</table>

Wildcards

- The \texttt{,} (comma) wildcard includes additional values. In the \texttt{Day-of-week} field, \texttt{MON, WED, FRI} would include Monday, Wednesday, and Friday. Total values are limited to 24 per field.
- The \texttt{-} (dash) wildcard specifies ranges. In the \texttt{Hour} field, 1–15 would include hours 1 through 15 of the specified day.
- The \texttt{*} (asterisk) wildcard includes all values in the field. In the \texttt{Hours} field, \texttt{*} would include every hour.
- The \texttt{/} (forward slash) wildcard specifies increments. In the \texttt{Hours} field, you could enter \texttt{1/10} to specify every 10th hour, starting from the first hour of the day (for example, the 01:00, 11:00, and 21:00).

Limits

- Snapshot schedules that lead to backup frequencies less than 1 hour or greater than 24 hours are not supported. If you have overlapping schedules that result in scheduling snapshots within a 1 hour window, a validation error results.

When creating a schedule, you can use the following sample cron strings.

<table>
<thead>
<tr>
<th>Minutes</th>
<th>Hours</th>
<th>Day of week</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14-20/1</td>
<td>TUE</td>
<td>Every hour between 2pm and 8pm on Tuesday.</td>
</tr>
<tr>
<td>0</td>
<td>21</td>
<td>MON-FRI</td>
<td>Every night at 9pm Monday–Friday.</td>
</tr>
<tr>
<td>30</td>
<td>0/6</td>
<td>SAT-SUN</td>
<td>Every 6 hour increment on Saturday and Sunday starting at 30 minutes after midnight (00:30) that day. This results in a snapshot at [00:30, 06:30, 12:30, and 18:30] each day.</td>
</tr>
<tr>
<td>30</td>
<td>12/4</td>
<td>*</td>
<td>Every 4 hour increment starting at 12:30 each day. This resolves to [12:30, 16:30, 20:30].</td>
</tr>
</tbody>
</table>
For example to run on a schedule on an every 2 hour increment starting at 15:15 each day. This resolves to [15:15, 17:15, 19:15, 21:15, 23:15], specify:

```
cron(15 15/2 *)
```

You can create multiple cron schedule definitions within as schedule. For example the following AWS CLI command contains two cron schedules in one schedule.

```
create-snapshot-schedule --schedule-identifier "my-test" --schedule-definition "cron(0 17 SAT,SUN)" "cron(0 9,17 MON-FRI)"
```

### Manual snapshots

You can take a manual snapshot any time. By default, manual snapshots are retained indefinitely, even after you delete your cluster. You can specify the retention period when you create a manual snapshot, or you can change the retention period by modifying the snapshot. For more information about changing the retention period, see Changing the manual snapshot retention period (p. 241).

If a snapshot is deleted, you can't 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 AWS Region. The default quota is listed at Quotas and limits in Amazon Redshift (p. 468).

### Managing snapshot storage

Because snapshots accrue storage charges, it's important that you delete them when you no longer need them. Amazon Redshift deletes automated and manual snapshots at the end of their respective snapshot retention periods. You can also delete manual snapshots using the AWS Management Console or with the `batch-delete-cluster-snapshots` CLI command.

You can change the retention period for a manual snapshot by modifying the manual snapshot settings.

You can get information about how much storage your snapshots are consuming using the Amazon Redshift Console or using the `describe-storage` CLI command.

### 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 AWS Region

You can configure Amazon Redshift to automatically copy snapshots (automated or manual) for a cluster to another AWS Region. When a snapshot is created in the cluster's primary AWS Region, it's copied to a secondary AWS Region. The two AWS Regions are known respectively as the source AWS Region and
destination AWS Region. If you store a copy of your snapshots in another AWS Region, you can restore your cluster from recent data if anything affects the primary AWS Region. You can configure your cluster to copy snapshots to only one destination AWS Region at a time. For a list of Amazon Redshift Regions, see Regions and endpoints in the Amazon Web Services General Reference.

When you enable Amazon Redshift to automatically copy snapshots to another AWS Region, you specify the destination AWS Region to copy the snapshots to. For automated snapshots, you can also specify the retention period to keep them in the destination AWS Region. After an automated snapshot is copied to the destination AWS Region and it reaches the retention time period there, it's deleted from the destination AWS Region. Doing this keeps your snapshot usage low. To keep the automated snapshots for a shorter or longer time in the destination AWS Region, change this retention period.

The retention period that you set for automated snapshots that are copied to the destination AWS Region is separate from the retention period for automated snapshots in the source AWS Region. The default retention period for copied snapshots is seven days. That seven-day period applies only to automated snapshots. In both the source and destination AWS Regions, manual snapshots are deleted at the end of the snapshot retention period or when 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 AWS Region to the destination AWS Region. Any automated snapshots copied to the destination AWS 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 AWS Region, are kept in the destination AWS Region until you manually delete them.

To change the destination AWS Region that you copy snapshots to, first disable the automatic copy feature. Then re-enable it, specifying the new destination AWS Region.

After a snapshot is copied to the destination AWS Region, it becomes active and available for restoration purposes.

To copy snapshots for AWS KMS–encrypted clusters to another AWS Region, create a grant for Amazon Redshift to use a KMS customer master key (CMK) in the destination AWS Region. Then choose that grant when you enable copying of snapshots in the source AWS Region. For more information about configuring snapshot copy grants, see Copying AWS KMS–encrypted snapshots to another AWS Region (p. 273).

Restoring a cluster from a snapshot

A snapshot contains data from any databases that are running on your cluster. It also contains information about your cluster, including the number of nodes, node type, and master user name. If you restore your cluster from a snapshot, Amazon Redshift uses the cluster information to create a new cluster. Then it restores all the databases from the snapshot data.

For the new cluster created from the original snapshot, you can choose the configuration, such as node type and number of nodes. The cluster is restored in the same AWS Region and a random, system-chosen Availability Zone, unless you specify another Availability Zone in your request. When you restore a cluster from a snapshot, you can optionally choose a compatible maintenance track for the new cluster.

Note
When you restore a snapshot to a cluster with a different configuration, the snapshot must have been taken on a cluster with cluster version 1.0.10013, or later.

When a restore is in progress, events are typically emitted in the following order:

1. RESTORE_STARTED – REDSHIFT-EVENT-2008 sent when the restore process begins.
2. RESTORE_SUCCEEDED – REDSHIFT-EVENT-3003 sent when the new cluster has been created.

The cluster is available for queries.
3. **DATA_TRANSFER_COMPLETED** — REDSHIFT-EVENT-3537 sent when data transfer complete.

**Note**
RA3 clusters only emit RESTORE_STARTED and RESTORE_SUCCEEDED events. There is no explicit data transfer to be done after a RESTORE succeeds because RA3 node types store data in Amazon Redshift managed storage. With RA3 nodes, data is continuously transferred between RA3 nodes and Amazon Redshift managed storage as part of normal query processing. RA3 nodes cache hot data locally and keep less frequently queried blocks in Amazon Redshift managed storage automatically.

You can monitor the progress of a restore by either calling the `DescribeClusters` API operation, 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, see `RestoreStatus`.

You can’t 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.

You might want to restore a snapshot to a cluster with a different configuration for these reasons:

- When a cluster is made up of smaller node types and you want to consolidate it into a larger node type with fewer nodes.
- When you have monitored your workload and determined the need to move to a node type with more CPU and storage.
- When you want to measure performance of test workloads with different node types.

Restore has the following constraints:

- The new node configuration must have enough storage for existing data. Even when you add nodes, your new configuration might not have enough storage because of the way that data is redistributed.
- The restore operation checks if the snapshot was created on a cluster version that is compatible with the cluster version of the new cluster. If the new cluster has a version level that is too early, then the restore operation fails and reports more information in an error message.
- The possible configurations (number of nodes and node type) you can restore to is determined by the number of nodes in the original cluster and the target node type of the new cluster. To determine the possible configurations available, you can use the Amazon Redshift console or the `describe-node-configuration-options` AWS CLI command with `action-type restore-cluster`. For more information about the restoring using the Amazon Redshift console, see **Restoring a cluster from a snapshot** (p. 245).

The following steps take a cluster with many nodes and consolidate it into a bigger node type with a smaller number of nodes using the AWS CLI. For this example, we start with a source cluster of 24 `ds2.xlarge` nodes. In this case, suppose that we already created a snapshot of this cluster and want to restore it into a bigger node type.

1. Run the following command to get the details of our 24-node `ds2.xlarge` cluster.

```
aws redshift describe-clusters --region eu-west-1 --cluster-identifier mycluster-123456789012
```

2. Run the following command to get the details of the snapshot.
3. Run the following command to describe the options available for this snapshot.

```bash/aws redshift describe-node-configuration-options --snapshot-identifier mycluster-snapshot --region eu-west-1 --action-type restore-cluster
```

This command returns an option list with recommended node types, number of nodes, and disk utilization for each option. For this example, the preceding command lists the following possible node configurations. We choose to restore into a three-node `ds2.8xlarge` cluster.

```json
{
    "NodeConfigurationOptionList": [
        {
            "EstimatedDiskUtilizationPercent": 65.26134808858235,
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 24
        },
        {
            "EstimatedDiskUtilizationPercent": 32.630674044291176,
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 48
        },
        {
            "EstimatedDiskUtilizationPercent": 65.26134808858235,
            "NodeType": "ds2.8xlarge",
            "NumberOfNodes": 3
        },
        {
            "EstimatedDiskUtilizationPercent": 48.94601106643677,
            "NodeType": "ds2.8xlarge",
            "NumberOfNodes": 4
        },
        {
            "EstimatedDiskUtilizationPercent": 39.156808853149414,
            "NodeType": "ds2.8xlarge",
            "NumberOfNodes": 5
        },
        {
            "EstimatedDiskUtilizationPercent": 32.630674044291176,
            "NodeType": "ds2.8xlarge",
            "NumberOfNodes": 6
        }
    ]
}
```

4. Run the following command to restore the snapshot into the cluster configuration that we chose. After this cluster is restored, we have the same content as the source cluster, but the data has been consolidated into three `ds2.8xlarge` nodes.

```bash/aws redshift restore-from-cluster-snapshot --region eu-west-1 --snapshot-identifier mycluster-snapshot --cluster-identifier mycluster-123456789012-x --node-type ds2.8xlarge --number-of-nodes 3
```
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 the Amazon Redshift 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.

Note

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

New console

To restore a table 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. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to use to restore a table.
3. For Actions, choose Restore table to display the Restore table page.
4. Enter the information about which snapshot, source table, and target table to use, and then choose Restore table.

Original console

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 and choose a cluster.
3. Choose the **Table restore** tab.

![Table restore tab](image)

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.
7. 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:**

1. In the **Table restore** tab, choose an existing table restore status.

2. 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 `my-snapshot-id`. The example restores the snapshot to the `mycluster-example` cluster with a new table name of `my-new-table`.

```bash
code
```

**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
Sharing snapshots

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:

```json
{
   "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:

```json
{
   "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:

```json
{
...
```
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. In this section, you can find how to manage your snapshots from the Amazon Redshift console. For more information about snapshots, see Amazon Redshift snapshots (p. 227).

All snapshot tasks in the Amazon Redshift console start from the snapshot list. You can filter the list by using a time range, the snapshot type, and the cluster associated with the snapshot. In addition, you can sort the list by date, size, and snapshot type. Depending on the snapshot type that you select, you might have different options available for working with the snapshot.

Topics
- Creating a snapshot schedule (p. 238)
- Creating a manual snapshot (p. 240)
- Changing the manual snapshot retention period (p. 241)
- Deleting manual snapshots (p. 243)
- Copying an automated snapshot (p. 244)
- Restoring a cluster from a snapshot (p. 245)
- Sharing a cluster snapshot (p. 248)
- Configuring cross-Region snapshot copy for a nonencrypted cluster (p. 249)
- Configure cross-Region snapshot copy for an AWS KMS–encrypted cluster (p. 250)
- Modifying the retention period for cross-Region snapshot copy (p. 251)

Creating a snapshot schedule

To precisely control when snapshots are taken, you can create a snapshot schedule and attach it to one or more clusters. You can attach a schedule when you create a cluster or by modifying the cluster. For more information, see Automated snapshot schedules (p. 228).

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

To create a snapshot schedule
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/
2. On the navigation menu, choose CLUSTERS, Snapshots, then choose the Snapshot schedules tab. The snapshot schedules are displayed.
3. Choose Add schedule to display the page to add a schedule.
4. Enter the properties of the schedule definition, then choose Add schedule.
5. On the page that appears, you can attach clusters to your new snapshot schedule, then choose OK.

Original console

To create a snapshot schedule
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 Snapshots.
3. Choose Snapshot Schedules.
4. Choose Add Schedule.
5. Under When do you want to take automated snapshots? choose Configure custom automated snapshot rules and then add one or more rules. Or choose Take a snapshot every 8 hours and specify the number of hours.
6. Choose Select a snapshot rule to add and choose a rule template from the list. You can add multiple rules.
7. Modify the template fields to customize your schedule.

8. To view the schedule, choose Preview schedule for all rules.
9. Choose Add Schedule.
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 Creating a snapshot of a cluster (p. 63).

**Note**

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

**New console**

**To create 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. On the navigation menu, choose CLUSTERS, Snapshots, then choose Create snapshot. The snapshot page to create a manual snapshot is displayed.
3. Enter the properties of the snapshot definition, then choose Create snapshot. It might take some time for the snapshot to be available.

**Original console**

**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, choose Snapshots.
3. Choose Create 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.
   c. For Snapshot retention period, enter the number of days to retain the snapshot. To retain the snapshot indefinitely, enter -1.
5. Choose Create.
Changing the manual snapshot retention period

You can change the retention period for a manual snapshot by modifying the snapshot settings.

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

**New console**

**To change the manual snapshot retention period**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, Snapshots, then choose the manual snapshot to change.
3. For Actions, choose Manual snapshot settings to display the properties of the manual snapshot.
4. Enter the revised properties of the snapshot definition, then choose Save.
Original console

**To change the manual snapshot retention period**

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 **Snapshots**.

3. 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, choose a time range that will narrow your search appropriately.
   - In the **Type** box, choose **manual**.
   - In the **Cluster** box, choose a cluster name to list snapshots for a single cluster, or choose **All Clusters** to list snapshots from all clusters.
   - In the **Sort by** field, choose how you want the list ordered.
4. In the snapshot list, select the snapshot that you want to modify.
5. Choose **Manual Snapshot Settings**.

6. For **Snapshot retention period**, enter the number of days to retain the snapshot. To retain the snapshot indefinitely, enter **-1**.
7. Choose Save.

Deleting manual snapshots

You can delete manual snapshots by selecting one or more snapshots in the snapshot list.

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

New console

To 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. On the navigation menu, choose CLUSTERS, Snapshots, then choose the snapshot to delete.
3. For Actions, choose Delete snapshot to delete the snapshot.
4. Confirm the deletion of the listed snapshots, then choose Delete.

Original console

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, choose Snapshots.
3. To filter the list, do any or all of the following:
   • In the Time Range box, choose a time range that will narrow your search appropriately.
   • In the Type box, choose manual.
   • In the Cluster box, choose a cluster name to list snapshots for a single cluster, or choose All Clusters to list snapshots from all clusters.
   • In the Sort by field, choose how you want the list ordered.
4. In the snapshot list, select the rows that contain the snapshots that you want to delete.
6. In the **Delete Manual Snapshot** dialog box, choose **Continue**.

### 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.

**Note**

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

### New console

**To 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/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose **CLUSTERS, Snapshots**, then choose the snapshot to copy.
3. For **Actions**, choose **Copy automated snapshot** to copy the snapshot.
4. Update the properties of the new snapshot, then choose **Copy**.

### Original console

**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/](https://console.aws.amazon.com/redshift/).
2. In the navigation pane, choose **Snapshots**.
3. To filter the list, do any or all of the following:
   • In the Time Range box, choose a time range that will narrow your search appropriately.
   • In the Type box, choose automated.
   • In the Cluster box, choose a cluster name to list snapshots for a single cluster, or choose All Clusters to list snapshots from all clusters.
   • In the Sort by field, choose how you want the list ordered.
4. In the snapshot list, select the snapshot that you want to copy.
5. Choose Copy Automated Snapshot.
6. In the Snapshot Identifier box of the Copy Automated Snapshot dialog box, enter a name for the snapshot copy.
7. For Snapshot retention period, enter the number of days to retain the snapshot. To retain the snapshot indefinitely, enter -1.

8. Choose Continue.

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.

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

New console

To 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. On the navigation menu, choose CLUSTERS, Snapshots, then choose the snapshot to restore.
3. Choose Restore from snapshot to view the Cluster configuration and Cluster details values of the new cluster to be created using the snapshot information.
4. Update the properties of the new cluster, then choose Restore cluster from snapshot.
Original console

Note
You can use these steps to change a cluster platform from EC2-Classic to EC2-VPC and vice versa.

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, choose Snapshots.
3. To filter the list, do any or all of the following:
   - In the Time Range box, choose a time range that will narrow your search appropriately.
   - In the Type box, choose manual or automated.
   - In the Cluster box, choose a cluster name to list snapshots for a single cluster, or choose All Clusters to list snapshots from all clusters.
   - In the Sort by field, choose how you want the list ordered.
4. In the snapshot list, choose the row that contains the snapshot that you want to use.
5. Choose 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.
      - Alphabetic 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. For 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. 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.
      This option only appears if you restore the cluster into the EC2-VPC platform.
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-VPC or EC2-Classic platform.

The option to select a cluster security group or a VPC security group depends on whether you restore the cluster into the EC2-VPC platform or the EC2-Classic platform.

k. In **Maintenance track**, the value of the maintenance track is displayed. In **Change maintenance track to**, optionally choose to restore the cluster using one of the maintenance tracks listed.

The following is an example of restoring a snapshot into a cluster that uses the EC2-VPC platform.

![Restoring a cluster from a snapshot using EC2-VPC platform](image1.png)

The following is an example of restoring a snapshot into a cluster that uses the EC2-Classic platform.

![Restoring a cluster from a snapshot using EC2-Classic platform](image2.png)

7. Choose **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.

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

New console

To share a snapshot with another account
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, Snapshots, then choose the manual snapshot to share.
3. For Actions, choose Manual snapshot settings to display the properties of the manual snapshot.
4. Enter the account or accounts to share with in the Manage access section, then choose Save.

Original console

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, choose 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, choose a time range that will narrow your search appropriately.
   • In the Cluster box, choose the cluster whose snapshot you want to share.
4. In the snapshot list, choose the row that contains the snapshot that you want to use.
5. Choose 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 choose Add Account.
   • To revoke the authorization for a user, choose X beside that user’s AWS account ID.
Configuring cross-Region snapshot copy for a nonencrypted cluster

You can configure Amazon Redshift to copy snapshots for a cluster to another AWS 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 or manual snapshots in the destination AWS Region. When cross-Region copy is enabled for a cluster, all new manual and automated snapshots are copied to the specified AWS Region. Copied snapshot names are prefixed with \texttt{copy}.

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

\textbf{New console}

\textbf{To configure a cross-Region snapshot}

1. Sign in to the AWS Management Console and open the Amazon Redshift console at \url{https://console.aws.amazon.com/redshift/}.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to move snapshots for.
3. For Actions, choose Configure cross-region snapshot.
   
   The Configure cross-Region dialog box appears.
4. For Copy snapshots, choose Yes.
5. In Destination AWS Region, choose the AWS Region to which to copy snapshots.
6. In Automated snapshot retention period (days), choose the number of days for which you want automated snapshots to be retained in the destination AWS Region before they are deleted.
7. In Manual snapshot retention period, choose the value that represents the number of days for which you want manual snapshots to be retained in the destination AWS Region before they are deleted. If you choose Custom value, the retention period must be between 1 to 3653 days.
8. Choose Save.
Original console

To configure cross-Region snapshot copy for a nonencrypted 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.
3. Choose Backup, and then choose Configure Cross-Region Snapshots.
4. In the Configure Cross-Region Snapshots dialog box, for Copy Snapshots choose Yes.
5. In Destination Region, choose the AWS 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 AWS Region before they are deleted.
7. Choose 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 an AWS 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 AWS Region. By doing this, you enable Amazon Redshift to perform encryption operations in the destination AWS 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 AWS Region (p. 273).

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

New console

To configure a cross-Region snapshot for an AWS KMS–encrypted cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to move snapshots for.
3. For Actions, choose Configure cross-region snapshot.

   The Configure cross-Region dialog box appears.

4. For Copy snapshots, choose Yes.
5. In Destination AWS Region, choose the AWS Region to which to copy snapshots.
6. In Automated snapshot retention period (days), choose the number of days for which you want automated snapshots to be retained in the destination AWS Region before they are deleted.
7. In Manual snapshot retention period, choose the value that represents the number of days for which you want manual snapshots to be retained in the destination AWS Region before they are deleted. If you choose Custom value, the retention period must be between 1 to 3653 days.
8. Choose Save.

Original console

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, choose Clusters.
3. In the cluster list, choose a cluster name to open the Configuration view for the cluster.
4. Choose Backup, and then choose Configure Cross-Region Snapshots.
5. In the Configure Cross-Region Snapshots dialog box, for Copy Snapshots choose Yes.
6. In Destination Region, choose the AWS 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 AWS 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 AWS Region. Then choose a grant from Snapshot Copy Grant.
9. Choose 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 can’t modify the destination AWS Region after cross-Region snapshot copy is configured. If you want to copy snapshots to a different AWS Region, first disable cross-Region snapshot copy. Then re-enable it with a new destination AWS Region and retention period. Any copied automated snapshots are deleted after you disable cross-Region snapshot copy. Thus, you should determine if there are any that you want to keep and copy them to manual snapshots before disabling cross-Region snapshot copy.

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

New console

To modify a cross-Region snapshot

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to modify snapshots for.
3. For Actions, choose Configure cross-region snapshot to display the properties of the snapshot.
4. Enter the revised properties of the snapshot definition, then choose Save.
Original console

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, choose Clusters.
3. Choose Backup, and then choose 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 AWS Region.

   If you select a smaller number of days to retain snapshots in the destination AWS Region, any automated snapshots that were taken before the new retention period are deleted. If you select a larger number of days to retain snapshots in the destination AWS Region, the retention period for existing automated snapshots is extended. It lengthens by the difference between the old value and the new value.
5. Choose 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. 227).

For step-by-step instructions to run the following example, see Running Java examples for Amazon Redshift using Eclipse (p. 391). You need to update the code and provide a cluster identifier.

Example

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * *
 * http://aws.amazon.com/apache2.0/
 * *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[CreateAndDescribeSnapshot demonstrates how to create an
// Amazon Redshift cluster snapshot and describe existing snapshots.]
// snippet-service:[redshift]
```
package com.amazonaws.services.redshift;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import com.amazonaws.services.redshift.model.*;

public class CreateAndDescribeSnapshot {

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

    public static void main(String[] args) throws IOException {
        client = AmazonRedshiftClientBuilder.defaultClient();
        try {
            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 Date waitForSnapshotAvailable(String snapshotId) {
        DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshotsRequest()
            .withSnapshotIdentifier(snapshotId);
        DescribeClusterSnapshotsResult result = client.describeClusterSnapshots(request);
        for (Snapshot snapshot : result.getClusterSnapshots()) {
            if (snapshot.getStatus().toString().equals("available")) {
                return snapshot.getSnapshotCreateTime();
            }
            Thread.sleep(1000);
        }
        return null;
    }

    private static Date deleteManualSnapshotsBefore(Date createDate) {
        DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshotsRequest()
            .withClusterIdentifier(clusterIdentifier);
        DescribeClusterSnapshotsResult result = client.describeClusterSnapshots(request);
        for (Snapshot snapshot : result.getClusterSnapshots()) {
            if (snapshot.getSnapshotCreateTime().before(createDate)) {
                client.deleteClusterSnapshot(snapshot.getSnapshotId());
            }
        }
        return createDate;
    }

    private static void printResultSnapshots(DescribeClusterSnapshotsResult result) {
        for (Snapshot snapshot : result.getClusterSnapshots()) {
            System.out.format("Snapshot ID: %s\n", snapshot.getSnapshotIdentifier());
            System.out.format("Snapshot Creation Time: %s\n", snapshot.getSnapshotCreateTime());
            System.out.format("Snapshot Status: %s\n", snapshot.getStatus());
            System.out.format("Cluster ID: %s\n", snapshot.getClusterIdentifier());
            System.out.format(" 통해 스토리지 상태: %s\n", snapshot.getStorageStatus());
            System.out.format(" 스토리지 주소: %s\n", snapshot.getStorageLocation());
        }
    }
}
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("Snapshot 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;
}

// snippet-end:[redshift.java.CreateAndDescribeSnapshot.complete]
• 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. 227).
Integrating Amazon Redshift with an AWS Partner

By working with Amazon Redshift, you can integrate with AWS Partners from the Cluster details page on the Amazon Redshift console. From the Cluster details page, you can speed up your data onboarding into your Amazon Redshift data warehouse with AWS Partner applications. You can also join and analyze data from different sources together with existing data in your cluster. The following AWS Partners can integrate with Amazon Redshift:

- Datacoral
- Etleap
- Fivetran
- Informatica
- SnapLogic
- Matillion (preview)
- Sisense (preview)
- Stitch (preview)

AWS Partners can integrate with Amazon Redshift using the AWS CLI or Amazon Redshift API operations. For more information, see the AWS CLI Command Reference or the Amazon Redshift API Reference.

Integrating with an AWS Partner using the Amazon Redshift console

Use the following procedure to integrate a cluster with an AWS Partner.

To integrate an Amazon Redshift cluster with an AWS Partner

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation pane, choose CLUSTERS.
3. Choose the cluster that you want to integrate.
4. Choose Add partner integration. The Choose partner page opens with details about the available AWS Partners.
5. Choose an AWS Partner, then choose Next.

More details about the chosen AWS Partner appear, along with details about the cluster that you are integrating. The Cluster details section includes information that you provide on the AWS Partner website such as the Cluster identifier, Endpoint, Database name, and User name (which is a database user name). This information is sent to the partner that you chose.

6. Choose Add partner to open the AWS Partner’s website.
7. Configure the integration with your Amazon Redshift cluster on the partner’s website. On the partner’s website, you can select and configure the data sources that are loaded to your Amazon Redshift cluster. You can also define additional extract, load, and transform (ELT) transformations to
process your business data, join it with other datasets, and build consolidated views for analysis and reporting.

You can view and manage AWS Partner integrations from the cluster details Properties tab. The Integrations section lists the Partner name that you can use to link to the AWS Partner website, the Status of the integration, the Database that receives the data, and the Last successful connection that might have updated the cluster.

The possible status values are as follows:

- **Active** – The AWS Partner can connect to the cluster and complete configured tasks.
- **Inactive** – The AWS Partner integration doesn't exist.
- **Runtime failure** – The AWS Partner can connect to the cluster but can't complete configured tasks.
- **Connection failure** – The AWS Partner can't connect to the cluster.

After you delete an AWS Partner integration from Amazon Redshift, data continues to flow into your cluster. Complete the delete on the partner's website.
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.

You may designate a node as a reserved node by calling the `PurchaseReservedNodeOffering` API operation or choosing **Purchase reserved nodes** on the Amazon Redshift console. When you purchase a reserved node, you must specify an AWS Region, node type, term, quantity of nodes, and offering type for the applicable reserved node type. The reserved node may only be used in the designated AWS Region.

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. 258).

### Comparing reserved node offerings

<table>
<thead>
<tr>
<th>Payment option</th>
<th>Payment schedule</th>
<th>Comparative savings</th>
<th>Duration</th>
<th>Upfront charges</th>
<th>Recurring monthly charges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No Upfront</strong></td>
<td>Monthly installments for the duration of the reservation. No upfront payment.</td>
<td>About a 20 percent discount over on-demand rates.</td>
<td>One-year term</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Partial Upfront</strong></td>
<td>Partial upfront payment, and monthly installments for the duration of the reservation.</td>
<td>Up to 41 percent to 73 percent discount depending on duration.</td>
<td>One-year or three-year term</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>All Upfront</strong></td>
<td>Full upfront payment for the reservation. No monthly charges.</td>
<td>Up to 42 percent to 76 percent discount depending on duration.</td>
<td>One-year or three-year term</td>
<td>Yes</td>
<td>None</td>
</tr>
</tbody>
</table>

**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: ds2.xlarge
- Payment Option: No Upfront
Amazon Redshift Cluster Management Guide

Reserved node examples

- Duration: one year
- Number of Reserved Nodes: 16

Example 1

You have one ds2.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 ds2.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 ds2.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 ds2.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 ds2.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 ds2.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.

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. 260).

**Note**

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

**New console**

**To purchase a reserved node**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at [https://console.aws.amazon.com/redshift/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose **CLUSTERS**, then choose **Reserved nodes** to display the list of reserved nodes.
3. Choose **Purchase reserved nodes** to display the page to choose the properties of the node that you want to purchase.
4. Enter the properties of the node, then choose **Purchase reserved nodes**.

**Original console**

**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/](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**.
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](image)

8. On the **Reserved Nodes** page, the reservation displays in the reservations list with a status of **payment-pending**.

**Note**

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

**New console**

To upgrade a reserved node, use the **Original Console** or the AWS CLI.

**Original console**

**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 that the DC1 cluster is shut down** before you upgrade the reserved nodes. The DC2 cluster accrues on-demand pricing until you upgrade the DC1 reserved nodes.

For more information about restoring from a snapshot, see **Amazon Redshift snapshots (p. 227)**. For more information about resizing a cluster, see **Resizing clusters in Amazon Redshift (p. 21)**.
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. On the navigation pane, choose Reserved Nodes.
3. Choose the DC1 reserved node that you want to upgrade.
4. Review the details on the Upgrade reserved node dialog box and choose Upgrade.

Upgrading reserved nodes with the AWS CLI

To upgrade a reserved node reservation with the AWS 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.

```bash
aws redshift get-reserved-node-exchange-offerings --reserved-node-id xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Please note: Migrate your actual dc1 cluster to dc2 by doing a resize or restore operation before exchanging your dc1 reservations for dc2. Learn more

Upgrading reserved nodes with the AWS CLI
"Duration": 31536000,
"ReservedNodeOfferingId": "yyyyyyyy-yyyy-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 that you want to exchange along with the ReservedNodeOfferingID you obtained in the previous step.

The following example illustrates this step.

```bash
aws redshift accept-reserved-node-exchange --reserved-node-id xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx --target-reserved-node-offering-id yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyy {
  "ExchangedReservedNode": {
    "UsagePrice": 0.0,
    "OfferingType": "No Upfront",
    "State": "exchanging",
    "FixedPrice": 0.0,
    "CurrencyCode": "USD",
    "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyyyyy",
    "NodeType": "dc2.large",
    "NodeCount": 1,
    "RecurringCharges": [
      {
        "RecurringChargeFrequency": "Hourly",
        "RecurringChargeAmount": 0.2
      }
    ],
    "ReservedNodeOfferingType": "Regular",
    "StartTime": "2018-06-27T18:02:58Z",
    "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyyyyy",
    "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. 391). 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**

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * http://aws.amazon.com/apache2.0/
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[ListAndPurchaseReservedNodeOffering demonstrates how to list
and purchase Amazon Redshift reserved node offerings.]
// snippet-service:[redshift]
// snippet-keyword:[Java]
// snippet-keyword:[Amazon Redshift]
// snippet-keyword:[DescribeReservedNodeOfferings]
// snippet-keyword:[PurchaseReservedNodeOffering]
// snippet-keyword:[ReservedNode]
// snippet-sourcetype:[full-example]
// snippet-sourcedate:[2019-01-31]
// snippet-sourceauthor:[AWS]
// snippet-start:[redshift.java.ListAndPurchaseReservedNodeOffering.complete]

package com.amazonaws.services.redshift;

import java.io.DataInput;
import java.io.DataInputStream;
import java.io.IOException;
import java.util.ArrayList;
import com.amazonaws.services.redshift.model.*;

public class ListAndPurchaseReservedNodeOffering {
    public static void main(String[] args) throws IOException {
        AmazonRedshift client;
        String nodeTypeToPurchase = "dc2.large";
        Double fixedPriceLimit = 10000.00;
        ArrayList<ReservedNodeOffering> matchingNodes = new ArrayList<ReservedNodeOffering>();
        
        // Purchase a reserved node.
```
// Default client using the {@link
com.amazonaws.auth.DefaultAWSCredentialsProviderChain}
client = AmazonRedshiftClientBuilder.defaultClient();

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("Finding nodes to purchase.");
    for (ReservedNodeOffering offering : result.getReservedNodeOfferings()) {
        if (offering.getNodeType().equals(nodeTypeToPurchase)) {
            if (offering.getFixedPrice() < fixedPriceLimit) {
                matchingNodes.add(offering);
                printOfferingDetails(offering);
                count ++;
            }
        }
    }
    if (count == 0) {
        System.out.println("No reserved node offering matches found.");
    } else {
        System.out.println("Found " + count + " matches.");
    }
}

private static void purchaseReservedNodeOffer() throws IOException {
    if (matchingNodes.size() == 0) {
        return;
    } else {
        System.out.println("Purchasing nodes.");
        for (ReservedNodeOffering offering : matchingNodes) {
            printOfferingDetails(offering);
            System.out.println("WARNING: purchasing this offering will incur costs.");
            System.out.println("Purchase this offering [Y or N]?");
            DataInput in = new DataInputStream(System.in);
            String purchaseOpt = in.readLine();
            if (purchaseOpt.equalsIgnoreCase("y")) {
                try {
                    PurchaseReservedNodeOfferingRequest request = new
                        PurchaseReservedNodeOfferingRequest()
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 API operations to purchase reserved node offerings.

- `PurchaseReservedNodeOffering`
- `DescribeReservedNodeOfferings`
- `DescribeOrderableClusterOptions`
Security in Amazon Redshift

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to Amazon Redshift, see AWS services in scope by compliance program.

- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization’s requirements, and applicable laws and regulations.

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 IAM user or account associated with your AWS security credentials. IAM 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 Identity and access management in Amazon Redshift (p. 287).

- **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. 370).

- **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 in the Amazon Redshift Database Developer Guide.

- **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. 308).

This documentation helps you understand how to apply the shared responsibility model when using Amazon Redshift. The following topics show you how to configure Amazon Redshift to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Redshift resources.
Data protection in Amazon Redshift

The AWS shared responsibility model applies to data protection in Amazon Redshift. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a Name field. This includes when you work with Amazon Redshift or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

AQUA (Advanced Query Accelerator) is a multitenant service running on servers that are separate from the Amazon Redshift cluster. AQUA supports authentication, encryption, isolation, and compliance to keep your data at rest and data in transit secure. The same permissions to access Amazon Redshift are used for AQUA. Queries processed on AQUA servers are run in isolated processes to ensure data protection.

Data encryption

Data protection refers to protecting data while in-transit (as it travels to and from Amazon Redshift at rest (while it is stored on disks in Amazon Redshift data centers). You can protect data in transit by using SSL or by using client-side encryption. You have the following options of protecting data at rest in Amazon Redshift.
• **Use server-side encryption** – You request Amazon Redshift to encrypt your data before saving it on disks in its data centers and decrypt it when you download the objects.

• **Use client-side encryption** – You can encrypt data client-side and upload the encrypted data to Amazon Redshift. In this case, you manage the encryption process, the encryption keys, and related tools.

### Encryption at rest

Server-side encryption is about data encryption at rest—that is, Amazon Redshift optionally encrypts your data as it writes it in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted data.

Amazon Redshift protects data at rest through encryption. Optionally, you can protect all data stored on disks within a cluster and all backups in Amazon S3 with Advanced Encryption Standard AES-256.

To manage the keys used for encrypting and decrypting your Amazon Redshift resources, you use 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. Using AWS KMS, you can create encryption keys and define the policies that control how these keys can be used. AWS KMS supports AWS CloudTrail, so you can audit key usage to verify that keys are being used appropriately. You can use your AWS KMS keys in combination with Amazon Redshift and supported AWS services. For a list of services that support AWS KMS, see How AWS Services Use AWS KMS in the AWS Key Management Service Developer Guide.

AQUA (Advanced Query Accelerator) receives and uses the same database encryption keys (DEKs) used by Amazon Redshift to store the data snapshots in Amazon S3. Amazon Redshift uses envelope encryption, which means that you can rotate keys without having to re-encrypt data blocks. If the Amazon Redshift cluster isn’t encrypted, then AQUA uses its own managed keys to encrypt the data. For more information, see Amazon Redshift database encryption (p. 271).

In some cases, you might revoke AWS KMS access to an IAM role associated with a cluster. If so, after the in-flight queries are complete, no further access to the data cached in AQUA is possible. This is because AQUA doesn’t persist keys beyond a single pushed-down scan or aggregate operation. This lack of persistence also means that you can’t access any metadata about those blocks.

### Topics

• Amazon Redshift database encryption (p. 271)

### 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.

You can enable encryption when you launch your cluster, or you can modify an unencrypted cluster to use AWS Key Management Service (AWS KMS) encryption. To do so, you can use either an AWS-managed key or a customer-managed key (CMK). When you modify your cluster to enable KMS encryption, Amazon Redshift automatically migrates your data to a new encrypted cluster. Snapshots created from the encrypted cluster are also encrypted. You can also migrate an encrypted cluster to an unencrypted cluster by modifying the cluster and changing the Encrypt database option. For more information, see Changing cluster encryption (p. 275).

Though encryption is an optional setting in Amazon Redshift, we recommend that you enable 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. 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.

Topics
- Database encryption for Amazon Redshift using AWS KMS (p. 272)
- Encryption for Amazon Redshift using hardware security modules (p. 274)
- Encryption key rotation in Amazon Redshift (p. 275)
- Changing cluster encryption (p. 275)
- Configuring database encryption using the console (p. 279)
- Configuring database encryption using the Amazon Redshift API and AWS CLI (p. 284)

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 an AWS 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, see 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 Amazon Resource Name (ARN) in Amazon Redshift. For more information about access to keys in AWS KMS, see 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, see 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
Data encryption

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. 44) and Managing clusters using the AWS CLI and Amazon Redshift API (p. 66).

Copying AWS KMS–encrypted snapshots to another AWS Region

AWS KMS keys are specific to an AWS Region. If you enable copying of Amazon Redshift snapshots to another AWS 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 AWS Region. This grant enables Amazon Redshift to encrypt snapshots in the destination AWS Region. For more information about cross-Region snapshot copy, see Copying snapshots to another AWS Region (p. 230).

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 AWS 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 AWS 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, see 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 AWS 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 AWS Region, enable copying of snapshots and specify the name of the snapshot copy grant that you created in the destination AWS 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. 250).

Before the snapshot is copied to the destination AWS Region, Amazon Redshift decrypts the snapshot using the master key in the source AWS 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 AWS Region, decrypts the snapshot using the internally managed RSA key, and then re-encrypts the snapshot using the master key in the destination AWS 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. 284).
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 and RA3 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 is closed to new customers. For more information, see CloudHSM Classic Pricing. AWS CloudHSM Classic isn't available in all AWS Regions. For more information about available AWS Regions, see AWS 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. 279) and Configuring Amazon Redshift to use an HSM using the Amazon Redshift API and AWS CLI (p. 285).
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.

Encryption key rotation 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. 283) and Rotating encryption keys using the Amazon Redshift API and AWS CLI (p. 285).

Changing cluster encryption

You can modify an unencrypted cluster to use AWS Key Management Service (AWS KMS) encryption, using either an AWS-managed key or a customer-managed key (CMK). When you modify your cluster to enable KMS encryption, Amazon Redshift automatically migrates your data to a new encrypted cluster. You can also migrate an unencrypted cluster to an encrypted cluster by modifying the cluster.

During the migration operation, your cluster is available in read-only mode, and the cluster status appears as resizing.

If your cluster is configured to enable cross-AWS Region snapshot copy, you must disable it before changing encryption. For more information, see Copying snapshots to another AWS Region (p. 230) and Configure cross-Region snapshot copy for an AWS KMS–encrypted cluster (p. 250). You can't enable hardware security module (HSM) encryption by modifying the cluster. Instead, create a new, HSM-encrypted cluster and migrate your data to the new cluster. For more information, see Migrating to an HSM-encrypted cluster (p. 277).

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

New console

To modify database encryption on a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to modify encryption.
3. Choose Properties.
4. In the **Database configurations** section, choose **Edit**, then choose **Edit encryption**.

5. Choose one of the encryption options and choose **Save changes**.

**Original console**

**To change cluster encryption using the console**

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

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

3. Choose **Cluster**, and then choose **Modify**.

4. For **Encrypt database**, choose **KMS** to enable encryption, or choose **None** to disable encryption.

5. To use a customer-managed key, for **Master Key** choose **Enter a key ARN** and enter the ARN in the **ARN** field.
6. Choose **Modify**.

**To change cluster encryption using the CLI**

To modify your unencrypted cluster to use KMS, run the `modify-cluster` CLI command and specify `--encrypted`, as shown following. By default, your default KMS key is used. To specify a customer-managed key, include the `--kms-key-id` option.

```bash
aws redshift modify-cluster --cluster-identifier <value> --encrypted --kms-key-id <value>
```

To remove encryption from your cluster, run the following CLI command.

```bash
aws redshift modify-cluster --cluster-identifier <value> --no-encrypted
```

**Migrating to an HSM-encrypted cluster**

To migrate an unencrypted cluster to a cluster encrypted using a hardware security module (HSM), you create a new encrypted cluster and move your data to the new cluster. You can't migrate to an HSM-encrypted cluster by modifying the cluster.

To migrate from an unencrypted cluster to an HSM-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. 271).

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.
   
   ```sql
   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.
   
   ```sql
   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. 409).
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.

   ```sql
   select database, schema,
   ```
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. 271).

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. 409).

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. 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. 44) and Managing clusters using the AWS CLI and Amazon Redshift API (p. 66).

**Note**

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

**New console**

**To modify database encryption on a cluster**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to move snapshots for.
3. For Actions, choose Modify to display the configuration page.
4. In the Database configuration section, choose a setting for Encryption, then choose Modify cluster.

**Original console**

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 navigation pane, choose Security, and then choose the HSM Connections tab.
3. Choose 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.

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, choose Yes and complete the steps in the next procedure. Otherwise, choose Not now to return to the list of HSM connections and complete the remainder of the process at another time.

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 navigation pane, choose Security, and then choose the HSM Certificates tab.
3. Choose **Create HSM Client Certificate**.

![Create HSM Client Certificate](image)

4. On the **Create HSM Client Certificate** page, type a name in the **HSM Client Certificate Identifier** box to identify this client certificate.

![Create HSM Client Certificate](image)

5. Choose **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.

![Create HSM Client Certificate](image)

   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, choose Next.

7. After the HSM client certificate is created and registered, choose 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. 41).

- **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, choose Security, and then choose the HSM Certificates tab.
3. Choose 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. 280)

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 navigation pane, choose Security, and then choose the HSM Connections tab.
3. Choose the HSM connection that you want to delete.
4. In the Delete HSM Connection dialog box, choose Delete to delete the connection from Amazon Redshift, or choose 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, choose Security and select the HSM Certificates tab.
3. In the list, choose the HSM client certificate that you want to delete.
4. In the Delete HSM Client Certificate dialog box, choose Delete to delete the certificate from Amazon Redshift, or choose 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.

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

**To rotate the encryption keys for a cluster**
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to update encryption keys.
3. For Actions, choose Rotate encryption to display the Rotate encryption keys page.

**Original 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, choose Clusters.
3. In the list, choose the cluster for which you want to rotate keys.
4. Choose Database, and then choose Rotate Encryption Keys.
5. Choose 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. 271).

**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
- DescribeSnapshotCopy Grants
- 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

Encryption in transit

You can configure your environment to protect the confidentiality and integrity data in transit.

Encryption of data in transit between an Amazon Redshift cluster and SQL clients over JDBC/ODBC:

• You can connect to Amazon Redshift clusters from SQL client tools over Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) 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. The client connects to the leader node of an Amazon Redshift cluster. For more information, see Configuring security options for connections (p. 144).
• To support SSL connections, Amazon Redshift creates and installs AWS Certificate Manager (ACM) issued certificates on each cluster. For more information, see Transitioning to ACM certificates for SSL connections (p. 147).
• To protect your data in transit within the AWS Cloud, Amazon Redshift uses hardware accelerated SSL to communicate with Amazon S3 or Amazon DynamoDB for COPY, UNLOAD, backup, and restore operations.
Encryption of data in transit between an Amazon Redshift cluster and Amazon S3 or DynamoDB:

- Amazon Redshift uses hardware accelerated SSL to communicate with Amazon S3 or DynamoDB for COPY, UNLOAD, backup, and restore operations.
- Redshift Spectrum supports the Amazon S3 server-side encryption (SSE) using your account’s default key managed by the AWS Key Management Service (KMS).
- Encrypt Amazon Redshift loads with Amazon S3 and AWS KMS. For more information, see Encrypt Your Amazon Redshift Loads with Amazon S3 and AWS KMS.

Encryption and signing of data in transit between AWS CLI, SDK, or API clients and Amazon Redshift endpoints:

- Amazon Redshift provides HTTPS endpoints for encrypting data in transit.
- To protect the integrity of API requests to Amazon Redshift, API calls must be signed by the caller. Calls are signed by an X.509 certificate or the customer’s AWS secret access key according to the Signature Version 4 Signing Process (Sigv4). For more information, see Signature Version 4 Signing Process in the AWS General Reference.
- Use the AWS CLI or one of the AWS SDKs to make requests to AWS. These tools automatically sign the requests for you with the access key that you specify when you configure the tools.

Encryption of data in transit between Amazon Redshift clusters and AQUA:

- Data is transmitted between AQUA and Amazon Redshift clusters over a TLS-encrypted channel. This channel is signed according to the Signature Version 4 Signing Process (Sigv4). For more information, see Signature Version 4 Signing Process in the AWS General Reference.

Key management

You can configure your environment to protect data with keys:

- Amazon Redshift automatically integrates with AWS Key Management Service (AWS KMS) for key management. AWS KMS uses envelope encryption. For more information, see Envelope Encryption.
- When encryption keys are managed in AWS KMS, Amazon Redshift uses a four-tier, key-based architecture for encryption. The architecture consists of randomly generated AES-256 data encryption keys, a database key, a cluster key, and a master key. For more information, see How Amazon Redshift Uses AWS KMS.
- You can create your own customer master key (CMK) in KMS. For more information, see Creating Keys.
- You can also import your own key material for new CMKs. For more information, see Importing Key Material in AWS Key Management Service (AWS KMS).
- Amazon Redshift supports management of encryption keys in external hardware security modules (HSMs). The HSM can be on-premises or can be AWS CloudHSM. When you use an HSM, you must use client and server certificates to configure a trusted connection between Amazon Redshift and your HSM. Amazon Redshift supports only AWS CloudHSM Classic for key management. For more information, see Encryption for Amazon Redshift using hardware security modules (p. 274). For information about AWS CloudHSM, see What is AWS CloudHSM.
- You can rotate encryption keys for encrypted clusters. For more information, see Encryption key rotation in Amazon Redshift (p. 275).

Data tokenization

Tokenization is the process of replacing actual values with opaque values for data security purposes. Security-sensitive applications use tokenization to replace sensitive data such as personally identifiable
Internetwork traffic privacy

To route traffic between Amazon Redshift and clients and applications on a corporate network:

- Set up a private connection between your virtual private cloud (VPC) and your corporate network. Set up either an IPsec VPN connection over the internet or a private physical connection using AWS Direct Connect connection. AWS Direct Connect enables you to establish a private virtual interface from your on-premises network directly to your Amazon VPC, providing you with a private, high-bandwidth network connection between your network and your VPC. With multiple virtual interfaces, you can even establish private connectivity to multiple VPCs while maintaining network isolation. For more information, see What is AWS Site-to-Site VPN? and What is AWS Direct Connect?

To route traffic between an Amazon Redshift cluster in a VPC and Amazon S3 buckets in the same AWS Region:

- Set up an Amazon S3 private VPC endpoint to privately access Amazon S3 data from an ETL load or unload. For more information, see Endpoints for Amazon S3.
- Enable “Enhanced VPC routing” for an Amazon Redshift cluster, specifying a target Amazon S3 VPC endpoint. Traffic generated by Amazon Redshift COPY, UNLOAD, or CREATE LIBRARY commands are then routed through the private endpoint. For more information, see Enabling enhanced VPC routing (p. 195).

Identity and access management in 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. 288)
- Access control (p. 289)
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. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

  - **Federated user access** – Instead of creating an IAM user, you can use existing 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** – A service role is an IAM role that a service assumes to perform actions on your behalf. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. An IAM administrator can create, modify, and delete a service role from within IAM. 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 CLI or 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. 289)
- Using identity-based policies (IAM policies) for Amazon Redshift (p. 294)

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.

<table>
<thead>
<tr>
<th>Resource type</th>
<th>ARN format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster database user group</td>
<td>arn:aws:redshift:region:account-id:dbgroup:cluster-name/database-group-name</td>
</tr>
<tr>
<td>Cluster parameter group</td>
<td>arn:aws:redshift:region:account-id:parametergroup:parameter-group-name</td>
</tr>
<tr>
<td>Cluster security group</td>
<td>arn:aws:redshift:region:account-id:securitygroup:security-group-name</td>
</tr>
</tbody>
</table>
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. 306).

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.

```json
{
  "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. 294). 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. 289)), 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. 289).
- **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. 306).

**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. 293). 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. 306).
Using IAM policy conditions for fine-grained access control

In Amazon Redshift, you can use condition keys to restrict access to resources based on the tags for those resources. The following are common Amazon Redshift condition keys.

<table>
<thead>
<tr>
<th>Condition key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws:RequestTag</td>
<td>Requires users to include a tag key (name) and value whenever they create a resource. For more information, see aws:RequestTag in the IAM User Guide.</td>
</tr>
<tr>
<td>aws:ResourceTag</td>
<td>Restricts user access to resources based on specific tag keys and values. For more information, see aws:ResourceTag in the IAM User Guide.</td>
</tr>
<tr>
<td>aws:TagKeys</td>
<td>Use this key to compare the tag keys in a request with the keys that you specify in the policy. For more information, see aws:TagKeys in the IAM User Guide.</td>
</tr>
</tbody>
</table>

For information on tags, see Tagging overview (p. 474).

For a list of the API actions that support the redshift:RequestTag and redshift:ResourceTag condition keys, see Amazon Redshift API permissions reference (p. 306).

The following condition keys can be used with the Amazon Redshift GetClusterCredentials action.

<table>
<thead>
<tr>
<th>Condition key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>redshift:DurationSeconds</td>
<td>Limits the number of seconds that can be specified for duration.</td>
</tr>
<tr>
<td>redshift:DbName</td>
<td>Restricts database names that can be specified.</td>
</tr>
<tr>
<td>redshift:DbUser</td>
<td>Restricts database user names that can be specified.</td>
</tr>
</tbody>
</table>

Example 1: Restricting access by using the aws: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",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/environment": "test"
      }
    }
  }
}
```

Example 2: Restricting access by using the aws: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. The condition with aws:TagKeys and the ForAllValues modifier specifies that only the keys costcenter and usage can be specified in the request.

```
```
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. 289).

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.

```json
{
    "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": [ 
                "aws:RequestTag/usage": "production",
                "aws:TagKeys": [ 
                    "costcenter",
                    "usage"
                ]
            ],
            "Effect": "Deny"
        }
    ]
}
```
Using identity-based policies (IAM policies)

The policy has two statements:

- The first statement grants permissions for a user to 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.

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 an AWS account.
  
  You can find the AmazonRedshiftReadOnlyAccess policy on the IAM console.

- **AmazonRedshiftFullAccess** – Grants full access to all Amazon Redshift resources for an AWS account.
  
  You can find the AmazonRedshiftFullAccess policy on the IAM console.

- **AmazonRedshiftQueryEditor** – Grants full access to the query editor on the Amazon Redshift console.
  
  You can find the AmazonRedshiftQueryEditor policy on the IAM console.

- **AmazonRedshiftDataFullAccess** – Grants full access to the Amazon Redshift Data API operations and resources for an AWS account.
  
  You can find the AmazonRedshiftDataFullAccess policy on the IAM 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.

Amazon Redshift updates to AWS-managed policies

View details about updates to AWS managed policies for Amazon Redshift since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon Redshift Document history page.
Using identity-based policies (IAM policies)

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmazonRedshiftDataFullAccess – Update to an existing policy</td>
<td>Amazon Redshift added new permissions to allow AuthorizeDataShare.</td>
<td>August 12, 2021</td>
</tr>
<tr>
<td>AmazonRedshiftDataFullAccess – Update to an existing policy</td>
<td>Amazon Redshift added new permissions to allow BatchExecuteStatement.</td>
<td>July 27, 2021</td>
</tr>
<tr>
<td>Amazon Redshift started tracking changes</td>
<td>Amazon Redshift started tracking changes for its AWS-managed policies.</td>
<td>July 27, 2021</td>
</tr>
</tbody>
</table>

Permissions required to use Redshift Spectrum

Amazon Redshift Spectrum requires permissions to other AWS services to access resources. For details about permissions in IAM policies for Redshift Spectrum, see IAM policies for Amazon Redshift Spectrum in the Amazon Redshift Database Developer Guide.

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, Amazon CloudWatch, Amazon SNS, 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 AWS-managed policy to the user. How to do this is described in AWS-managed (predefined) policies for Amazon Redshift (p. 295).

For information to give a user access to the query editor on the Amazon Redshift console, see Permissions required to use the Amazon Redshift console query editor (p. 296).

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.

Permissions required to use the Amazon Redshift console query editor

For a user to work with the Amazon Redshift query editor, that user must have a minimum set of permissions to Amazon Redshift and Amazon Redshift Data API operations. To connect to a database using a secret, you must also have Secrets Manager permissions.

To give a user access to the query editor on the Amazon Redshift console, attach the AmazonRedshiftQueryEditor and AmazonRedshiftReadOnlyAccess AWS-managed policies. The AmazonRedshiftQueryEditor policy allows the user permission to retrieve the results of only their own SQL statements. That is, statements submitted by the same aws:userid as shown in this section of the AmazonRedshiftQueryEditor AWS-managed policy.

```json
{
    "Sid": "DataAPIIAMSessionPermissionsRestriction",
    "Action": [
        "redshift-data:GetStatementResult",
        "redshift-data:CancelStatement",
    ]
}
```
To allow a user to retrieve the results of SQL statements of others in the same IAM role, create your own policy without the condition to limit access to the current user. Also limit access to change a policy to an administrator.

**Permissions required to use the Amazon Redshift scheduler**

When you use the Amazon Redshift scheduler, you set up an IAM role with a trust relationship to the Amazon Redshift scheduler (scheduler.redshift.amazonaws.com) to allow the scheduler to assume permissions on your behalf. You also attach a policy (permissions) to the role for the Amazon Redshift API operations that you want to schedule.

The following example shows the policy document in JSON format to set up a trust relationship with the Amazon Redshift scheduler and Amazon Redshift.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Principal": {
         "Service": [
            "scheduler.redshift.amazonaws.com",
            "redshift.amazonaws.com"
         ],
      "Action": "sts:AssumeRole"
   }
   ]
}
```

For more information about trust entities, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.

You also must add permission for the Amazon Redshift operations you want to schedule.

For the scheduler to use the `ResizeCluster` operation, add a permission that is similar to the following to your IAM policy. Depending on your environment, you might want to make the policy more restrictive.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "redshift:ResizeCluster",
      "Resource": "*"
   }
   ]
}
```
For the steps to create a role for the Amazon Redshift scheduler, see Creating a role for an AWS service (console) in the IAM User Guide. Make these choices when you create a role in the IAM console:

- For **Choose the service that will use this role**: Choose Redshift.
- For **Select your use case**: Choose Redshift - Scheduler.
- Create or attach a policy to the role that allows an Amazon Redshift operation to be scheduled. Choose **Create policy** or modify the role to attach a policy. Enter the JSON policy for the operation that is to be scheduled.
- After you create the role, edit the **Trust Relationship** of the IAM role to include the service redshift.amazonaws.com.

The IAM role you create has trusted entities of scheduler.redshift.amazonaws.com and redshift.amazonaws.com. It also has an attached policy that allows a supported Amazon Redshift API action, such as, "redshift:ResizeCluster".

### Permissions required to use the Amazon EventBridge scheduler

When you use the Amazon EventBridge scheduler, you set up an IAM role with a trust relationship to the EventBridge scheduler (events.amazonaws.com) to allow the scheduler to assume permissions on your behalf. You also attach a policy (permissions) to the role for the Amazon Redshift Data API operations that you want to schedule and a policy for Amazon EventBridge operations.

You use the EventBridge scheduler when you create scheduled queries with the Amazon Redshift query editor on the console.

You can create an IAM role to run scheduled queries on the IAM console. In this IAM role, attach AmazonEventBridgeFullAccess and AmazonRedshiftDataFullAccess.

The following example shows the policy document in JSON format to set up a trust relationship with the EventBridge scheduler.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
        ],
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

For more information about trust entities, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.

For the steps to create a role for the EventBridge scheduler, see Creating a role for an AWS service (console) in the IAM User Guide. Make these choices when you create a role in the IAM console:

- For **Choose the service that will use this role**: Choose CloudWatch Events.
- For **Select your use case**: Choose CloudWatch Events.
- Attach the following permission policies: AmazonEventBridgeFullAccess and AmazonRedshiftDataFullAccess.
The IAM role that you create has a trusted entity of `events.amazonaws.com`. It also has an attached policy that allows supported Amazon Redshift Data API actions, such as, "redshift-data:*".

**Permissions required to use the data sharing API operations**

To control access to the data sharing API operations, use IAM action-based policies. For information about how to manage IAM policies, see Managing IAM policies in the IAM User Guide.

In particular, suppose that a producer cluster administrator needs to use the `AuthorizeDataShare` call to authorize egress for a datashare outside of an AWS account. In this case, you set up an IAM action-based policy to grant this permission. Use the `DeauthorizeDataShare` call to revoke egress.

When using IAM action-based policies, you can also specify an IAM resource in the policy, such as `DataShareARN`. The following shows the format and an example for `DataShareARN`.

| arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/SalesShare |

You can restrict `AuthorizeDataShare` access to a specific datashare by specifying the datashare name in the IAM policy.

| { |
| "Statement": |
| { |
| "Action": |
| [ |
| "redshift:AuthorizeDataShare", |
| ] |
| "Resource": |
| [ |
| "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/SalesShare" |
| ] |
| "Effect": "Deny" |
| } |
| } |

You can also restrict the IAM policy to all datashares owned by a specific producer cluster. To do this, replace the `datashare-name` value in the policy with a wildcard or an asterisk. Keep the cluster's `namespace-guid` value.

| arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/* |

Following is an IAM policy that prevents an entity from calling `AuthorizeDataShare` on the datashares owned by a specific producer cluster.

| { |
| "Statement": |
| { |
| "Action": |
| [ |
| "redshift:AuthorizeDataShare", |
| ] |
| "Resource": |
| [ |
| "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/**" |
| ] |
| "Effect": "Deny" |
| } |
| } |
DataShareARN restricts the access based on both the datashare name and the globally unique ID (GUID) for the owning cluster's namespace. It does this by specifying the name as an asterisk.

**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`, make sure to also allow the actions and permit access to the resources listed in the following table.

<table>
<thead>
<tr>
<th>GetClusterCredentials parameter</th>
<th>Action</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocreate</td>
<td><code>redshift:CreateClusterUser</code></td>
<td></td>
</tr>
<tr>
<td>DbGroups</td>
<td><code>redshift:JoinGroup</code></td>
<td></td>
</tr>
<tr>
<td>DbName</td>
<td>NA</td>
<td><code>dbname</code></td>
</tr>
</tbody>
</table>

For more information about resources, see Amazon Redshift resources and operations (p. 289).

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. 292).

**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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":"AllowRedshift",
      "Action": [
```
Using identity-based policies (IAM policies)

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"],
            "Effect": "Deny",
            "Resource": ["arn:aws:redshift:us-west-2:*"]
        }
    ]
}
```
Using identity-based policies (IAM policies)

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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":"AllowSharedSnapshots",
      "Action": ["redshift:AuthorizeSnapshotAccess"],
      "Effect": "Allow"
    },
    {
      "Sid":"AllowRevokableSnapshot",
      "Action": ["redshift:RevokeSnapshotAccess"],
      "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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":"AllowDescribeSnapshots",
      "Action": ["redshift:DescribeClusterSnapshots"],
      "Resource": ["*"]
    }
  ]
}
```
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"],
         "Effect": "Allow"
      },
      {
         "Sid":"AllowRestoreFromClusterSnapshot",
         "Action": ["redshift:RestoreFromClusterSnapshot"],
         "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": [
      {
         "Effect": "Allow"
      }
   ]
}
```
Example 7: Allow a user to tag resources with the Amazon Redshift console

The following example policy allows a user to tag resources with the Amazon Redshift console using the AWS Resource Groups. This policy can be attached to a user role that invokes the new or original Amazon Redshift console. For more information about tagging, see Tagging resources in Amazon Redshift (p. 474).

```json
{
    "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, CreateClusterUser, 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": [
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid": "AIDIODR4TAW7CSEXAMPLE: ${redshift:DbUser}@yourdomain.com"
        }
      }
    },
    {
      "Sid": "CreateClusterUserStatement",
      "Effect": "Allow",
      "Action": ["redshift:CreateClusterUser"],
      "Resource": [
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid": "AIDIODR4TAW7CSEXAMPLE: ${redshift:DbUser}@yourdomain.com"
        }
      }
    }
  ]
}
```
Amazon Redshift API permissions reference

When you set up Access control (p. 289), you write permission policies that you can attach to an IAM identity (identity-based policies). For detailed reference information, see Actions, resources, and condition keys for Amazon Redshift in the IAM User Guide. This reference contains information about which Amazon Redshift API operations can be used in an IAM policy. 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. 293).

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. To specify an action, use the redshift: prefix followed by the API operation name (for example, redshift:CreateCluster).

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
- ec2:CreateVpcEndpoint
- ec2:DeleteVpcEndpoints
- ec2:DescribeVpcEndpoints
- ec2:ModifyVpcEndpoint

To allow an IAM entity to create AWSServiceRoleForRedshift service-linked roles

Add the following policy statement to the permissions for that IAM entity:

```json
{
  "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:

```json
{
  "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. 30).

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.

Using IAM authentication to generate database user credentials

You can generate temporary database credentials based on permissions granted through an AWS Identity and Access Management (IAM) permissions policy to manage the access that your users have to your Amazon Redshift database.

Commonly, Amazon Redshift database users log in to the database by providing a database user name and password. However, you don't have to maintain user names and passwords in your Amazon Redshift database. As an alternative, you can configure your system to permit users to create user credentials and log in to the database based on their IAM credentials.

For more information, see Identity Providers and Federation in the IAM User Guide.

Topics
- Overview (p. 308)
- Creating temporary IAM user credentials (p. 309)
- Options for providing IAM credentials (p. 318)

Overview

Amazon Redshift provides the GetClusterCredentials API operation 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 operation. 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 operation, retrieve database user credentials, and connect to the database.
If you already manage user identities outside AWS, you can use an identity provider (IdP) compliant with Security Assertion Markup Language (SAML) 2.0 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 in to Amazon Redshift databases.

Your SQL client needs permission to call the GetClusterCredentials operation for you. You manage those permissions by creating an IAM role and attaching an IAM permissions policy that grants or restricts access to the GetClusterCredentials operation 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 operation 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 operation or the parallel `get-cluster-credentials` CLI command, see GetClusterCredentials and get-cluster-credentials.

To manage authentication and authorization centrally, Amazon Redshift supports database authentication with IAM, enabling user authentication through enterprise federation. Instead of creating an IAM user, in this approach you use existing 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 IdP.

To provide federated access to a user or client application in your organization to call Amazon Redshift API operations, you can also use the JDBC or ODBC driver with SAML 2.0 support to request authentication from your organization IdP. In this case, your organization's users don't have direct access to Amazon Redshift.

### Creating temporary IAM user credentials

In this section, you can find how to configure your system to generate temporary IAM-based database user credentials and log in 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. 310)**

   (Optional) You can authenticate users for access to an Amazon Redshift database by integrating IAM authentication and a third-party identity provider (IdP).

2. **Step 2: Configure SAML assertions for your IdP (p. 310)**

   (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. 311)**

   Your SQL client application assumes the IAM role when it calls the GetClusterCredentials operation. 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. 313)**

   (Optional) By default, GetClusterCredentials returns credentials 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. 313)**

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).

For more information, see Identity Providers and Federation in the IAM User Guide.

Before you can use Amazon Redshift IdP authentication, create an AWS SAML identity provider. You create an IdP 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) in the IAM User Guide.

**Step 2: Configure SAML assertions for your IdP**

After you create the IAM role, you define a claim rule in your IdP application to map users or groups in your organization to the IAM role. For more information, see Configuring SAML Assertions for the Authentication Response in the IAM User Guide.

If you choose to use the optional GetClusterCredentials parameters DbUser, AutoCreate, and DbGroups, you have two options. 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. 313).

**Note**

If you use an IAM policy variable ${redshift:DbUser}, as described in Resource policies for GetClusterCredentials (p. 300) 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 by using RoleSessionName. You can find examples of how to set a condition using an IAM policy in Example policy for using GetClusterCredentials (p. 305).

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.amazon.com/SAML/Attributes/DbUser"

  Set the AttributeValue element 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 fewer 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.amazon.com/SAML/Attributes/DbUser">
  <AttributeValue>user-name</AttributeValue>
</Attribute>
```
• An Attribute element with the Name attribute set to "https://redshift.amazon.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.amazon.com/SAML/Attributes/AutoCreate">
  <AttributeValue>true</AttributeValue>
</Attribute>
```

• An Attribute element with the Name attribute set to set to "https://redshift.amazon.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.amazon.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 operation 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 operation. 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, AWS Region, and port, include permission to call the redshift:DescribeClusters operation 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 AWS Region, account, and cluster. For dbuser-name, specify the user name to use to log in to the cluster database.

     arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name

   • (Optional) 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 AWS Region, account, and cluster. For database-name, specify the name of a database that the user will log in to.

• If you use the `DbGroups` option, include permission to call the `redshift:JoinGroup` operation with the Amazon Redshift dbgroup resource in the following format. Replace `region, account-id, and cluster-name` with the values for your AWS Region, account, and cluster. For `dbgroup-name`, specify the name of a user group that the user joins at login.

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

For more information and examples, see Resource policies for GetClusterCredentials (p. 300).

The following example shows a policy that allows the IAM role to call the `GetClusterCredentials` operation. 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",  
  }  
}
```

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",  
  }  
}
```

The following example shows a policy that allows the IAM role to call the `GetClusterCredentials` operation with the option to automatically create a new user and specify groups the user joins at login. 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 in 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 login. When you call the GetClusterCredentials operation, you can specify a list of user group names that the new user joins at login. 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 in 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.

   ```sql
   create user temp_creds_user password disable;
   ```

   The following example disables the password for an existing user.

   ```sql
   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. This driver manages the process of creating database user credentials and establishing a connection between your SQL client and your Amazon Redshift database.

If you use an identity provider for authentication, specify the name of a credential provider plugin. The Amazon Redshift JDBC and ODBC drivers include plugins for the following SAML-based identity providers:

- Active Directory Federation Services (AD FS)
- PingOne
- Okta
- Microsoft Azure AD

For the steps to set up Microsoft Azure AD as an identity provider, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).
To configure a JDBC connection to use IAM credentials

1. Download the latest Amazon Redshift JDBC driver from the Configuring a JDBC driver version 1.0 connection (p. 122) page.

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://
```

Add `cluster-name`, `region`, and `account-id`. The JDBC driver uses your IAM account information and cluster name to retrieve the cluster ID and AWS Region. To do so, your IAM user or role must have permission to call the `redshift:DescribeClusters` operation with the specified cluster. If your IAM user or role doesn't have permission to call the `redshift:DescribeClusters` operation, include the cluster ID, AWS 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. 341).

   The following URL specifies AccessKeyID and SecretAccessKey for an IAM user.

   ```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
AccessKeyID=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtncEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

   The following example specifies a named profile that contains the IAM credentials.

   ```
```

4. Add JDBC options that the JDBC driver uses to call the `GetClusterCredentials` API operation. Don't include these options if you call the `GetClusterCredentials` API operation programmatically.

   The following example includes the JDBC `GetClusterCredentials` options.

   ```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
plugin_name=com.amazon.redshift.plugin.AzureCredentialsProvider&UID=user&PWD=password&idp_tenant=my_tenant&client_secret=my_secret&client_id=my_id
```

To configure an ODBC connection to use IAM credentials

In the following procedure, you can find steps only to configure IAM authentication. For steps to use standard authentication, using a database user name and password, see Configuring an ODBC connection (p. 130).

1. Install and configure the latest Amazon Redshift ODBC driver for your operating system. For more information, see Configuring an ODBC connection (p. 130) 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:

- Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132)
- Use an ODBC driver manager to configure the driver on Linux and macOS X operating systems (p. 136)

3. On Microsoft Windows operating systems, access the Amazon Redshift ODBC Driver DSN Setup window.
   
a. Under **Connection Settings**, enter the following information:

   - **Data Source Name**
   - **Server** (optional)
   - **Port** (optional)
   - **Database**

   If your IAM user or role has permission to call the `redshift:DescribeClusters` operation, only **Data Source Name** and **Database** are required. Amazon Redshift uses **ClusterId** and **Region** to get the server and port by calling the `DescribeCluster` operation.

   If your IAM user or role doesn't have permission to call the `redshift:DescribeClusters` operation, specify **Server** and **Port**. The default port is 5439.

b. Under **Authentication**, choose a value for **Auth Type**.

   For each authentication type, enter values as listed following:

   **AWS Profile**

   Enter the following information:
   
   - **ClusterId**
   - **Region**
   - **Profile name**

   Enter 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. 319).

   (Optional) Provide details for options that the ODBC driver uses to call the `GetClusterCredentials` API operation:

   - **DbUser**
   - **User AutoCreate**
   - **DbGroups**

   For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

   **IAM Credentials**

   Enter the following information:

   - **ClusterId**
   - **Region**
   - **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.

Provide details for options that the ODBC driver uses to call the GetClusterCredentials API operation:
• DbUser (required)
• User AutoCreate (optional)
• DbGroups (optional)

For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

Identity Provider: AD FS

For Windows Integrated Authentication with AD FS, leave User and Password empty.

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

An Amazon Resource Name (ARN) for the IAM role from the multi-valued AttributeValue elements for the Role attribute in the SAML assertion. To find the appropriate value for the preferred role, work with your IdP administrator. For more information, see Configure SAML assertions for your IdP (p. 310).

(Optional) Provide details for options that the ODBC driver uses to call the GetClusterCredentials API operation:
• DbUser
• User AutoCreate
• DbGroups

For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

Identity Provider: PingFederate

For User and Password, enter your IdP user name and password.

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
An Amazon Resource Name (ARN) for the IAM role from the multi-valued AttributeValue elements for the Role attribute in the SAML assertion. To find the appropriate value for the preferred role, work with your IdP administrator. For more information, see Configure SAML assertions for your IdP (p. 310).

(Optional) Provide details for options that the ODBC driver uses to call the GetClusterCredentials API operation:
- DbUser
- User AutoCreate
- DbGroups

For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

Identity Provider: Okta

For User and Password, enter your IdP user name and password.

Provide IdP details:
- IdP Host
  The name of the corporate identity provider host. This name should not include any slashes (/).
- IdP Port
  This value is not used by Okta.
- Preferred Role
  An Amazon Resource Name (ARN) for the IAM role from the AttributeValue elements for the Role attribute in the SAML assertion. To find the appropriate value for the preferred role, work with your IdP administrator. For more information, see Configure SAML assertions for your IdP (p. 310).
- 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.

(Optional) Provide details for options that the ODBC driver uses to call the GetClusterCredentials API operation:
- DbUser
- User AutoCreate
- DbGroups

For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

Identity Provider: Azure AD

For User and Password, enter your IdP user name and password.

For Cluster ID and Region, enter the cluster ID and AWS Region of your Amazon Redshift cluster.

For Database, enter the database that you created for your Amazon Redshift cluster.
Provide IdP details:

- **IdP Tenant**
  
The tenant used for Azure AD.

- **Azure Client Secret**
  
The client secret of the Amazon Redshift enterprise app in Azure.

- **Azure Client ID**
  
The client ID (application ID) of the Amazon Redshift enterprise app in Azure.

(Optional) Provide details for options that the ODBC driver uses to call the `GetClusterCredentials` API operation:

- **DbUser**
- **User AutoCreate**
- **DbGroups**

For more information, see JDBC and ODBC Options for Creating Database User Credentials (p. 341).

### Options for providing IAM credentials

To provide IAM credentials for a JDBC or ODBC connection, choose one of the following options.

- **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. For more information, see Using a Configuration Profile (p. 319).

- **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 JDBC and ODBC options for providing IAM credentials (p. 319).

- **Identity provider federation**
  
  When you use identity provider federation to enable users from an identity provider to authenticate to Amazon Redshift, specify the name of a credential provider plugin. For more information, see Using a credentials provider plugin (p. 320).

  The Amazon Redshift JDBC and ODBC drivers include plugins for the following SAML-based identity federation credential providers:

  - Microsoft Active Identity Federation Services (AD FS)
  - PingOne
  - Okta
  - Microsoft Azure Active Directory (Azure AD)

  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 JDBC driver options (p. 126) and Configure ODBC driver options (p. 140).

For more information, see Configure a JDBC or ODBC connection to use IAM credentials (p. 313).
Using a Configuration Profile

You can supply the IAM credentials options and GetClusterCredentials options as settings in named profiles in your AWS configuration file. To provide the profile name, use the Profile JDBC option. The configuration is stored in a file named config or a file named credentials in a folder named .aws in your home directory.

For a SAML-based credential provider plugin included with an Amazon Redshift JDBC or ODBC driver, you can use the settings described just preceding in Using a credentials provider plugin (p. 320). If plugin_name isn’t used, the other options are ignored.

The following example shows the ~/.aws/credentials file with two profiles.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user2]
aws_access_key_id=AKIAI44QH8DHEXAMPLE
aws_secret_access_key=j7MtcGbClwBF/2Zp9UtK/h3yCo8tvbEXAMPLEKEY
session_token=AwoDYXdzEPT//////////

wEXAMPlE=tc764Bnc9SAPB8M2wDOwXj4fZTwdQWVLw5KHHBuFqwAemxSfSPFeoIYRqTflfKD8YUuwthAx7mSEI/gkPpFIi/kMcGd
QrmGdeehM41C1tBmAGg2wUE8phUZampKsburEDy0KFkxyQDyW7WZowq5VSDvlp75YU
9HFvldR68T6g6fE8YQcHNVXAss1yq6d+xxozKwT38vqT7ZDO0iPpkUL6412bqBAz
+scqKmlz8FDrypNC9Yjc8fPOLn9FX9SYVVKTr4rvx31IS1TJabIQwJ2ICCR/oLxBA==
```

To use the credentials for the user2 example, specify Profile=user2 in the JDBC URL.

For more information on using profiles, see Named Profiles in the AWS Command Line Interface User Guide.

For more information on using profiles for JDBC driver, see Amazon Redshift JDBC driver installation and configuration guide.

For more information on using profiles for ODBC driver, see Amazon Redshift ODBC driver installation and configuration guide.

### JDBC and ODBC options for providing IAM credentials

The following table lists the JDBC and ODBC options for providing IAM credentials.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iam</td>
<td>For use only in an ODBC connection string. Set to 1 to use IAM authentication.</td>
</tr>
<tr>
<td>AccessKeyId</td>
<td>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.</td>
</tr>
<tr>
<td>SecretAccessKey</td>
<td></td>
</tr>
<tr>
<td>SessionToken</td>
<td></td>
</tr>
<tr>
<td>plugin_name</td>
<td>The fully qualified name of a class that implements a credentials provider. The Amazon Redshift JDBC driver includes SAML-based credential provider plugins. If you provide plugin_name, you can also provide other related options. For more information, see Using a credentials provider plugin (p. 320).</td>
</tr>
</tbody>
</table>
Using a credentials provider plugin

Amazon Redshift uses credentials provider plugins for SSO authentication.

To support SSO authentication, Amazon Redshift provides the Azure AD plugin for Microsoft Azure Active Directory. For information on how to configure this plugin, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).

Setting up multi-factor authentication

Setting up multi-factor authentication

To support multi-factor authentication (MFA), Amazon Redshift provides browser-based plugins. Use the browser SAML plugin for Okta, PingOne, and the browser Azure AD plugin for Microsoft Azure Active Directory.

With the browser SAML plugin, SAML authentication flows like this:

1. A user tries to log in.
2. The plugin launches a local server to listen to incoming connections on the localhost.
3. The plugin launches a web browser to request a SAML response over HTTPS from the specified SSO login URL federated identity provider endpoint.
4. The web browser follows the link and prompts the user to enter credentials.
5. After the user authenticates and grants consent, the federated identity provider endpoint returns a SAML response over HTTPS to the URI indicated by redirect_uri.
6. The web browser moves the response message with the SAML response to the indicated redirect_uri.

### Option | Description
--- | ---
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. 319).
With the browser Azure AD plugin, SAML authentication flows like this:

1. A user tries to log in.
2. The plugin launches a local server to listen to incoming connections on the localhost.
3. The plugin launches a web browser to request an authorization code from the Azure AD `oauth2/authorize` endpoint.
4. The web browser follows the generated link over HTTPS and prompts the user to enter credentials. The link is generated using configuration properties, such as tenant and client_id.
5. After the user authenticates and grants consent, the Azure AD `oauth2/authorize` endpoint returns and sends a response over HTTPS with the authorization code to the indicated `redirect_uri`.
6. The web browser moves the response message with the SAML response to the indicated `redirect_uri`.
7. The local server accepts the incoming connection and the plugin requests and retrieves the authorization code and sends a POST request to the Azure AD `oauth2/token` endpoint.
8. The Azure AD `oauth2/token` endpoint returns a response with an access token to the indicated `redirect_uri`.
9. The plugin retrieves the SAML response and passes it to Amazon Redshift.

See the following sections:

• Active Directory Federation Services (AD FS)
For more information, see Setting up JDBC or ODBC Single Sign-on authentication with AD FS (p. 330).

- **PingOne (Ping)**

  Ping is supported only with the predetermined PingOne IdP Adapter using Forms authentication.

  For more information, see Setting up JDBC or ODBC Single Sign-on authentication with Ping Identity (p. 334).

- **Okta**

  Okta is supported only for the Okta-supplied application used with the AWS Management Console.

  For more information, see Setting up JDBC or ODBC Single Sign-on authentication with Okta (p. 337).

- **Microsoft Azure Active Directory**

  For more information, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).

### Configuring plugin options

#### Configuring plugin options

To use a SAML-based credentials provider plugin, specify the following options using JDBC or ODBC options or in a named profile. If `plugin_name` isn't specified, the other options are ignored.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plugin_name</code></td>
<td>For JDBC, the class name that implements a credentials provider. Specify one of the following:</td>
</tr>
<tr>
<td></td>
<td>• For Active Directory Federation Services</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.AdfsCredentialsProvider</td>
</tr>
<tr>
<td></td>
<td>• For Okta</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.OktaCredentialsProvider</td>
</tr>
<tr>
<td></td>
<td>• For PingFederate</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.PingCredentialsProvider</td>
</tr>
<tr>
<td></td>
<td>• For Microsoft Azure Active Directory</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.AzureCredentialsProvider</td>
</tr>
<tr>
<td></td>
<td>• For SAML MFA</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.BrowserSamlCredentialsProvider</td>
</tr>
<tr>
<td></td>
<td>• For Microsoft Azure Active Directory SSO with MFA</td>
</tr>
<tr>
<td></td>
<td>com.amazon.redshift.plugin.BrowserAzureCredentialsProvider</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
**For ODBC, specify one of the following:**
- For Active Directory Federation Services: adfs
- For Okta: okta
- For PingFederate: ping
- For Microsoft Azure Active Directory: azure
- For SAML MFA: browser saml
- For Microsoft Azure Active Directory SSO with MFA: browser azure ad

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 the identity provider. The default is 443. This port is ignored for Okta.

preferred_role | An Amazon Resource Name (ARN) from the AttributeValue elements for the Role attribute in the SAML assertion. To find the appropriate value for the preferred role, work with your IdP administrator. For more information, see Configure SAML assertions for your IdP (p. 310).

user | A corporate user name, including the domain when applicable. For example, for Active Directory, the domain name is required in the format domain\username.

password | The corporate user's password. We recommend not using this option. Instead, use your SQL client to supply the password.

app_id | An ID for an Okta application. Used only with Okta. The value for app_id follows amazon_aws in the Okta application embed link. To get this value, work with your IdP administrator. The following is an example of an application embed link: https://example.okta.com/home/amazon_aws/0oa2hylwrpM8UGehd1t7/272

idp_tenant | A tenant used for Azure AD. Used only with Azure.

client_id | A client ID for the Amazon Redshift enterprise application in Azure AD. Used only with Azure.

Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD

You can use Microsoft Azure AD as an identity provider (IdP) to access your Amazon Redshift cluster. Following, you can find a procedure that describes how to set up a trust relationship for this purpose. For more information about configuring AWS as a service provider for the IdP, see Configuring Your SAML 2.0 IdP with Relying Party Trust and Adding Claims in the IAM User Guide.

Note
To use Azure AD with JDBC, the Amazon Redshift JDBC driver must be version 1.2.37.1061 or later. To use Azure AD with ODBC, the Amazon Redshift ODBC driver must be version 1.4.10.1000 or later.

Watch the following video to learn how to federate Amazon Redshift access with Microsoft Azure AD single sign-on: Federating Amazon Redshift access with Microsoft Azure AD single sign-on.

To set up Azure AD and your AWS account to trust each other

1. Create or use an existing Amazon Redshift cluster for your Azure AD users to connect to. To configure the connection, certain properties of this cluster are needed, such as the cluster identifier. For more information, see Creating a Cluster.
2. Set up an Azure Active Directory, groups, users used for AWS on the Microsoft Azure portal.
3. Add Amazon Redshift as an enterprise application on the Microsoft Azure portal to use for single sign-on to the AWS Console and federated login to Amazon Redshift. Choose Enterprise application.
5. Search for AWS in the search field.
6. Choose Amazon Web Services (AWS) and choose Add. This creates the AWS application.
8. Choose SAML. The Amazon Web Services (AWS) | SAML-based Sign-on page appears.
9. Choose Yes to proceed to the Set up Single Sign-On with SAML page. This page shows the list of pre-configured AWS SSO-related attributes.
10. For Basic SAML Configuration, choose the edit icon and choose Save.
11. When you are configuring for more than one application, provide an identifier value. For example, enter https://signin.aws.amazon.com/saml#2. Note that from the second application onwards, use this format with a # sign to specify a unique SPN value.
12. In the User Attributes and Claims section, choose the edit icon.
   By default, the Unique User Identifier (UID), Role, RoleSessionName, and SessionDuration claims are pre-configured.
13. Choose + Add new claim to add a claim for database users.
   For Name, enter DbUser.
   For Namespace, enter https://redshift.amazon.com/SAML/Attributes.
   For Source, choose Attribute.
   For Source attribute, choose user.userprincipalname. Then, choose Save.
14. Choose + Add new claim to add a claim for AutoCreate.
   For Name, enter AutoCreate.
   For Namespace, enter https://redshift.amazon.com/SAML/Attributes.
   For Source, choose Attribute.
   For Source attribute, choose "true". Then, choose Save.

Here, 123456789012 is your AWS account, AzureSSO is an IAM role you created, and AzureADProvider is the IAM provider.

<table>
<thead>
<tr>
<th>Claim name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique user identifier (name ID)</td>
<td>user.userprincipalname</td>
</tr>
<tr>
<td><a href="https://aws.amazon.com/SAML/Attributes/SessionDuration">https://aws.amazon.com/SAML/Attributes/SessionDuration</a></td>
<td>&quot;900&quot;</td>
</tr>
<tr>
<td><a href="https://aws.amazon.com/SAML/Attributes/RoleSessionName">https://aws.amazon.com/SAML/Attributes/RoleSessionName</a></td>
<td>user.userprincipalname</td>
</tr>
</tbody>
</table>
### Using IAM authentication to generate database user credentials

<table>
<thead>
<tr>
<th>Claim name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/AutoCreate">https://redshift.amazon.com/SAML/Attributes/AutoCreate</a></td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/DbGroups">https://redshift.amazon.com/SAML/Attributes/DbGroups</a></td>
<td>user.assignedroles</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/DbUser">https://redshift.amazon.com/SAML/Attributes/DbUser</a></td>
<td>user.userprincipalname</td>
</tr>
</tbody>
</table>

15. Under App Registration > your-application-name > Authentication, add Mobile And Desktop Application. Specify the URL as http://localhost/redshift/.

16. In the SAML Signing Certificate section, choose Download to download and save the federation metadata XML file for use when you create an IAM SAML identity provider. This file is used to create the AWS SSO federated identity.

17. Create an IAM SAML identity provider on the IAM console. The metadata document that you provide is the federation metadata XML file that you saved when you set up Azure Enterprise Application. For detailed steps, see Creating and Managing an IAM Identity Provider (Console) in the IAM User Guide.

18. Create an IAM role for SAML 2.0 federation on the IAM console. For detailed steps, see Creating a Role for SAML in the IAM User Guide.

19. Create an IAM policy that you can attach to the IAM role that you created for SAML 2.0 federation on the IAM console. For detailed steps, see Creating IAM Policies (Console) in the IAM User Guide.

Modify the following policy (in JSON format) for your environment:

- Substitute the AWS Region of your cluster for `us-west-1`.
- Substitute your AWS account for `123456789012`.
- Substitute your cluster identifier (or `*` for all clusters) for `cluster-identifier`.
- Substitute your database (or `*` for all databases) for `dev`.
- Substitute the unique identifier of your IAM role for `AROAJ2UCCR6DPCEXAMPLE`.
- Substitute your tenant or company email domain for `example.com`.
- Substitute the database group that you plan to assign the user to for `my_dbgroup`.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
            ],
            "Condition": {
                "StringEquals": {
                    "aws:userid": "AROAJ2UCCR6DPCEXAMPLE:${redshift:DbUser}@example.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "redshift:GetDbUserPasssword"
        }
    ]
}
```
This policy grants permissions as follows:

- The first section grants permission to the GetClusterCredentials API operation to get temporary credentials for the specified cluster. In this example, the resource is `cluster-identifier` with database `dev`, in account `123456789012`, and in AWS Region `us-west-1`. The `${redshift:DbUser}` clause allows only users that match the `DbUser` value specified in Azure AD to connect.

- The condition clause enforces that only certain users get temporary credentials. These are users under the role specified by the role unique ID `AROAJ2UCR6DPCEXAMPLE` in the IAM account identified by an email address in your company's email domain. For more information about unique IDs, see Unique IDs in the IAM User Guide.

Your setup with your IdP (in this case, Azure AD) determines how the condition clause is written. If your employee's email is `johndoe@example.com`, first set `${redshift:DbUser}` to the super field that matches the employee's user name `johndoe`. Then, to make this condition work, set the AWS SAML RoleSessionName field to the super field that matches the employee's email `johndoe@example.com`. When you take this approach, consider the following:

- If you set `${redshift:DbUser}` to be the employee's email, then remove the `@example.com` in the example JSON to match the RoleSessionName.

- If you set the RoleSessionId to be just the employee's user name, then remove the `@example.com` in the example to match the RoleSessionName.

- In the example JSON, the `${redshift:DbUser}` and RoleSessionName are both set to the employee's email. This example JSON uses the Amazon Redshift database user name with `@example.com` to sign the user in to access the cluster.

- The second section grants permission to create a `dbuser` name in the specified cluster. In this example JSON, it restricts creation to `${redshift:DbUser}`.

- The third section grants permission to specify which `dbgroup` a user can join. In this example JSON, a user can join the `my_dbgroup` group in the specified cluster.

- The fourth section grants permission to actions the user can do on all resources. In this example JSON, it allows users to call `redshift:DescribeClusters` to get cluster information such as the cluster endpoint, AWS Region, and port. It also allows users to call `iam:ListRoles` to check which roles a user can assume.
To set up JDBC for authentication to Microsoft Azure AD

- Configure your database client to connect to your cluster through JDBC using your Azure AD single sign-on.

You can use any client that uses a JDBC driver to connect using Azure AD single sign-on or use a language like Java to connect using a script. For installation and configuration information, see Configuring a JDBC driver version 1.0 connection (p. 122).

For example, you can use SQLWorkbench/J as the client. When you configure SQLWorkbench/J, the URL of your database uses the following format.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

If you use SQLWorkbench/J as the client, take the following steps:

a. Start SQL Workbench/J. On the Select Connection Profile page, add a Profile Group called AzureAuth.

b. For Connection Profile, enter Azure.

c. Choose Manage Drivers, and choose Amazon Redshift. Choose the Open Folder icon next to Library, then choose the appropriate JDBC .jar file.

d. On the Select Connection Profile page, add information to the connection profile as follows:

   - For User, enter your Microsoft Azure user name. This is the user name of the Microsoft Azure account that you are using for single sign-on that has permission to the cluster that you are trying to authenticate using.
   - For Password, enter your Microsoft Azure password.
   - For Drivers, choose Amazon Redshift (com.amazon.redshift.jdbc.Driver).

e. Choose Extended Properties to add additional information to the connection properties, as described following.

For Azure AD SSO configuration, add additional information as follows:

   - For plugin_name, enter `com.amazon.redshift.plugin.AzureCredentialsProvider`. This value specifies to the driver to use Azure AD Single Sign-On as the authentication method.
   - For idp_tenant, enter `your-idp-tenant`. Used only for Microsoft Azure AD. This is the tenant name of your company configured on Azure AD. This value can either be the tenant name or the tenant unique ID with hyphens.
   - For client_secret, enter `your-azure-redshift-application-client-secret`. Used only for Microsoft Azure AD. This is your client secret of the Amazon Redshift application that you created when setting up your Azure Single Sign-On configuration. This is only applicable to the com.amazon.redshift.plugin.AzureCredentialsProvider plugin.
   - For client_id, enter `your-azure-redshift-application-client-id`. Used only for Microsoft Azure AD. This is the client ID (with hyphens) of the Amazon Redshift application that you created when setting up your Azure Single Sign-On configuration.

For Azure AD SSO with MFA configuration, add additional information to the connection properties as follows:
For **plugin_name**, enter `com.amazon.redshift.plugin.BrowserAzureCredentialsProvider`. This value specifies to the driver to use Azure AD SSO with MFA as the authentication method.

- For **idp_tenant**, enter `your-idp-tenant`. Used only for Microsoft Azure AD. This is the tenant name of your company configured on Azure AD. This value can either be the tenant name or the tenant unique ID with hyphens.

- For **client_id**, enter `your-azure-redshift-application-client-id`. This option is used only for Microsoft Azure AD. This is the client ID (with hyphens) of the Amazon Redshift application that you created when setting up your Azure AD SSO with MFA configuration.

- For **listen_port**, enter `your-listen-port`. This is the port that local server is listening to. The default is 7890.

- For **idp_response_timeout**, enter `the-number-of-seconds`. This is the number of seconds to wait before timing out when the IdP server sends back a response. The minimum number of seconds must be 10. If establishing the connection takes longer than this threshold, then the connection is aborted.

**To set up ODBC for authentication to Microsoft Azure AD**

- Configure your database client to connect to your cluster through ODBC using your Azure AD single sign-on.

Amazon Redshift provides ODBC drivers for Linux, Windows, and macOS operating systems. Before you install an ODBC driver, determine whether your SQL client tool is 32-bit or 64-bit. Install the ODBC driver that matches the requirements of your SQL client tool.

Also install and configure the latest Amazon Redshift ODBC driver for your operating system as follows:

- For Windows, see Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132).
- For macOS, see Install the Amazon Redshift ODBC driver on macOS X (p. 135).
- For Linux, see Install the Amazon Redshift ODBC driver on Linux (p. 134).

On Windows, in the **Amazon Redshift ODBC Driver DSN Setup** page, under **Connection Settings**, enter the following information:

- For **Data Source Name**, enter `your-DSN`. This specifies the data source name used as the ODBC profile name.

- For **Auth type** for Azure AD SSO configuration, choose **Identity Provider: Azure AD**. This is the authentication method that the ODBC driver uses to authenticate using Azure single sign-on.

- For **Auth type** for Azure AD SSO with MFA configuration, choose **Identity Provider: Browser Azure AD**. This is the authentication method that the ODBC driver uses to authenticate using Azure single sign-on with MFA.

- For **Cluster ID**, enter `your-cluster-identifier`.
- For **Region**, enter `your-cluster-region`.
- For **Database**, enter `your-database-name`.
- For **User**, enter `your-azure-username`. This is the user name for the Microsoft Azure account that you are using for single sign-on that has permission to the cluster that you're trying to authenticate using. Use this only for **Auth Type** is **Identity Provider: Azure AD**.

- For **Password**, enter `your-azure-password`. Use this only for **Auth Type** is **Identity Provider: Azure AD**.
• For **IdP Tenant**, enter `your-idp-tenant`. This is the tenant name of your company configured on your IdP (Azure). This value can either be the tenant name or the tenant unique ID with hyphens.

• For **Azure Client Secret**, enter `your-azure-redshift-application-client-secret`. This is the client secret of the Amazon Redshift application that you created when setting up your Azure single sign-on configuration.

• For **Azure Client ID**, enter `your-azure-redshift-application-client-id`. This is the client ID (with hyphens) of the Amazon Redshift application that you created when setting up your Azure single sign-on configuration.

• For **Listen Port**, enter `your-listen-port`. This is the default listen port that local server is listening to. The default is 7890. This applies only to the Browser Azure AD plugin.

• For **Response Timeout**, enter `the-number-of-seconds`. This is the number of seconds to wait before timing out when the IdP server sends back a response. The minimum number of seconds must be 10. If establishing the connection takes longer than this threshold, then the connection is aborted. This option applies only to the Browser Azure AD plugin.

On macOS and Linux, edit the `odbc.ini` file as follows:

**Note**

All entries are case-insensitive.

• For **clusterid**, enter `your-cluster-identifier`. This is the name of the created Amazon Redshift cluster.

• For **region**, enter `your-cluster-region`. This is the AWS Region of the created Amazon Redshift cluster.

• For **database**, enter `your-database-name`. This is the name of the database that you're trying to access on the Amazon Redshift cluster.

• For **locale**, enter `en-us`. This is the language that error messages display in.

• For **iam**, enter `1`. This value specifies to the driver to authenticate using IAM credentials.

• For **plugin_name** for Azure AD SSO configuration, enter `AzureAD`. This specifies to the driver to use Azure Single Sign-On as the authentication method.

• For **plugin_name** for Azure AD SSO with MFA configuration, enter `BrowserAzureAD`. This specifies to the driver to use Azure Single Sign-On with MFA as the authentication method.

• For **uid**, enter `your-azure-username`. This is the user name of the Microsoft Azure account you are using for single sign-on that has permission to the cluster you are trying to authenticate against. Use this only for plugin_name is `AzureAD`.

• For **pwd**, enter `your-azure-password`. Use this only for **plugin_name** is `AzureAD`.

• For **idp_tenant**, enter `your-idp-tenant`. This is the tenant name of your company configured on your IdP (Azure). This value can either be the tenant name or the tenant unique ID with hyphens.

• For **client_secret**, enter `your-azure-redshift-application-client-secret`. This is the client secret of the Amazon Redshift application that you created when setting up your Azure single sign-on configuration.

• For **client_id**, enter `your-azure-redshift-application-client-id`. This is the client ID (with hyphens) of the Amazon Redshift application that you created when setting up your Azure single sign-on configuration.

• For **listen_port**, enter `your-listen-port`. This is the port that local server is listening to. The default is 7890. This applies to the Browser Azure AD plugin.

• For **idp_response_timeout**, enter `the-number-of-seconds`. This is the specified period of time in seconds to wait for response from Azure. This option applies to the Browser Azure AD plugin.

On macOS and Linux, also edit the profile settings to add the following exports.
To troubleshoot issues with the Browser Azure AD plugin

1. To use the Browser Azure AD plugin, you must set the reply URL specified in the request to match the reply URL configured for your application.

   Navigate to the Set up Single Sign-On with SAML page on the Microsoft Azure portal. Then check the Reply URL is set to http://localhost/redshift/.

2. If you get an IdP tenant error, verify that the IdP Tenant name matches the domain name you initially used to set up the Active Directory in Microsoft Azure.

   On Windows, navigate to the Connection Settings section of the Amazon Redshift ODBC DSN Setup page. Then check the tenant name of your company configured on your IdP (Azure) matches the domain name you initially used to set up the Active Directory in Microsoft Azure.

   On macOS and Linux, find the odbc.ini file. Then check the tenant name of your company configured on your IdP (Azure) matches the domain name you initially used to set up the Active Directory in Microsoft Azure.

3. If you get an error that the reply URL specified in the request does not match the reply URLs configured for your application, verify that the Redirect URIs is the same as the reply URL.

   Navigate to the App registration page of your application on the Microsoft Azure portal. Then check the Redirect URIs matches the reply URL.

4. If you get the unexpected response: unauthorized error, verify that you completed the Mobile and desktop applications configuration.

   Navigate to the App registration page of your application on the Microsoft Azure portal. Then navigate to Authentication and check that you configured Mobile and desktop applications to use http://localhost/redshift/ as the redirect URIs.

Setting up JDBC or ODBC Single Sign-on authentication with AD FS

You can use AD FS as an identity provider (IdP) to access your Amazon Redshift cluster. Following, you can find a procedure that describes how to set up a trust relationship for this purpose. For more information about configuring AWS as a service provider for AD FS, see Configuring Your SAML 2.0 IdP with Relying Party Trust and Adding Claims in the IAM User Guide.

To set up AD FS and your AWS account to trust each other

1. Create or use an existing Amazon Redshift cluster for your AD FS users to connect to. To configure the connection, certain properties of this cluster are needed, such as the cluster identifier. For more information, see Creating a Cluster.

2. Set up AD FS to control Amazon Redshift access on the Microsoft Management Console:


   2. On the Select Data Source page, choose Import data about the relying party published online or on a local network.
3. For **Federation metadata address (host name or URL)**, enter \https://signin.aws.amazon.com/saml-metadata.xml. The metadata XML file is a standard SAML metadata document that describes AWS as a relying party.

4. On the **Specify Display Name** page, enter a value for **Display name**.

5. On the **Choose Issuance Authorization Rules** page, choose an issuance authorization rule to either permit or deny all users to access this relying party.

6. On the **Ready to Add Trust** page, review your settings.

7. On the **Finish** page, choose **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes**.

8. On the context (right-click) menu, choose **Relying Party Trusts**.

9. For your relying party, open the context (right-click) menu and choose **Edit Claim Rules**. On the **Edit Claim Rules** page, choose **Add Rule**.

10. For **Claim rule template**, choose **Transform an Incoming Claim**, and then on the **Edit Rule – NameId** page, do the following:
    - For **Claim rule name**, enter **NameId**.
    - For **Incoming claim name**, choose **Windows Account Name**.
    - For **Outgoing claim name**, choose **Name ID**.
    - For **Outgoing name ID format**, choose **Persistent Identifier**.
    - Choose **Pass through all claim values**.

11. On the **Edit Claim Rules** page, choose **Add Rule**. On the **Select Rule Template** page, for **Claim rule template**, choose **Send LDAP Attributes as Claims**.

12. On the **Configure Rule** page, do the following:
    - For **Claim rule name**, enter **RoleSessionName**.
    - For **Attribute store**, choose **Active Directory**.
    - For **LDAP Attribute**, choose **Email Addresses**.
    - For **Outgoing Claim Type**, choose **https://aws.amazon.com/SAML/Attributes/RoleSessionName**.

13. On the **Edit Claim Rules** page, choose **Add Rule**. On the **Select Rule Template** page, for **Claim rule template**, choose **Send Claims Using a Custom Rule**.

14. On the **Edit Rule – Get AD Groups** page, for **Claim rule name**, enter **Get AD Groups**.

15. For **Custom rule**, enter the following.

    ```
c:[Type == identity/claims/windowsaccountname", "http://schemas.microsoft.com/ws/2008/06/Directory", ";;tokenGroups;(0)";
    Issuer == "AD AUTHORITY"] => add(store = "Active
    types = ("http://temp/variable"), query = param = c.Value);
    ```

16. On the **Edit Claim Rules** page, choose **Add Rule**. On the **Select Rule Template** page, for **Claim rule template**, choose **Send Claims Using a Custom Rule**.

17. On the **Edit Rule – Roles** page, for **Claim rule name**, type **Roles**.

18. For **Custom rule**, enter the following.

    ```
    ```
Note the ARNs of the SAML provider and role to assume. In this example, arn:aws:iam:123456789012:saml-provider/ADFS is the ARN of the SAML provider and arn:aws:iam:123456789012:role/ADFS is the ARN of the role.

3. Make sure that you have downloaded the federationmetadata.xml file. Check that the document contents do not have invalid characters. This is the metadata file you use when configuring the trust relationship with AWS.

4. Create an IAM SAML identity provider on the IAM console. The metadata document that you provide is the federation metadata XML file that you saved when you set up Azure Enterprise Application. For detailed steps, see Creating and Managing an IAM Identity Provider (Console) in the IAM User Guide.

5. Create an IAM role for SAML 2.0 federation on the IAM console. For detailed steps, see Creating a Role for SAML in the IAM User Guide.

6. Create an IAM policy that you can attach to the IAM role that you created for SAML 2.0 federation on the IAM console. For detailed steps, see Creating IAM Policies (Console) in the IAM User Guide. For an Azure AD example, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).

To set up JDBC for authentication to AD FS

- Configure your database client to connect to your cluster through JDBC using AD FS SSO.

You can use any client that uses a JDBC driver to connect using AD FS SSO or use a language like Java to connect using a script. For installation and configuration information, see Configuring a JDBC driver version 1.0 connection (p. 122).

For example, you can use SQLWorkbench/J as the client. When you configure SQLWorkbench/J, the URL of your database uses the following format.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

If you use SQLWorkbench/J as the client, take the following steps:

a. Start SQL Workbench/J. In the Select Connection Profile page, add a Profile Group, for example ADFS.

b. For Connection Profile, enter your connection profile name, for example ADFS.

c. Choose Manage Drivers, and choose Amazon Redshift. Choose the Open Folder icon next to Library, then choose the appropriate JDBC .jar file.

d. On the Select Connection Profile page, add information to the connection profile as follows:

   - For User, enter your AD FS user name. This is the user name of the account that you are using for single sign-on that has permission to the cluster that you are trying to authenticate using.
   - For Password, enter your AD FS password.
   - For Drivers, choose Amazon Redshift (com.amazon.redshift.jdbc.Driver).

   e. Choose Extended Properties. For plugin_name, enter `com.amazon.redshift.plugin.AdfsCredentialsProvider`. This value specifies to the driver to use AD FS SSO as the authentication method.

To set up ODBC for authentication to AD FS

- Configure your database client to connect to your cluster through ODBC using AD FS SSO.
Amazon Redshift provides ODBC drivers for Linux, Windows, and macOS operating systems. Before you install an ODBC driver, determine whether your SQL client tool is 32-bit or 64-bit. Install the ODBC driver that matches the requirements of your SQL client tool.

Also install and configure the latest Amazon Redshift ODBC driver for your operating system as follows:

- For Windows, see Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132).
- For macOS, see Install the Amazon Redshift ODBC driver on macOS X (p. 135).
- For Linux, see Install the Amazon Redshift ODBC driver on Linux (p. 134).

On Windows, in the Amazon Redshift ODBC Driver DSN Setup page, under Connection Settings, enter the following information:

- For Data Source Name, enter your-DSN. This specifies the data source name used as the ODBC profile name.
- For Auth type, choose Identity Provider: SAML. This is the authentication method that the ODBC driver uses to authenticate using AD FS SSO.
- For Cluster ID, enter your-cluster-identifier.
- For Region, enter your-cluster-region.
- For Database, enter your-database-name.
- For User, enter your-adfs-username. This is the user name for the AD FS account that you are using for SSO that has permission to the cluster that you're trying to authenticate using. Use this only for Auth type is Identity Provider: SAML.
- For Password, enter your-adfs-password. Use this only for Auth type is Identity Provider: SAML.

On macOS and Linux, edit the odbc.ini file as follows:

**Note**
All entries are case-insensitive.

- For clusterid, enter your-cluster-identifier. This is the name of the created Amazon Redshift cluster.
- For region, enter your-cluster-region. This is the AWS Region of the created Amazon Redshift cluster.
- For database, enter your-database-name. This is the name of the database that you're trying to access on the Amazon Redshift cluster.
- For locale, enter en-us. This is the language that error messages display in.
- For iam, enter 1. This value specifies to the driver to authenticate using IAM credentials.
- For plugin_name, do one of the following:
  - For AD FS SSO with MFA configuration, enter BrowserSAML. This is the authentication method that the ODBC driver uses to authenticate to AD FS.
  - For AD FS SSO configuration, enter ADFS. This is the authentication method that the ODBC driver uses to authenticate using Azure AD SSO.
  - For uid, enter your-adfs-username. This is the user name of the Microsoft Azure account that you are using for SSO that has permission to the cluster you are trying to authenticate against. Use this only for plugin_name is ADFS.
  - For pwd, enter your-adfs-password. Use this only for plugin_name is ADFS.
On macOS and Linux, also edit the profile settings to add the following exports.

```bash
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

---

### Setting up JDBC or ODBC Single Sign-on authentication with Ping Identity

You can use Ping Identity as an identity provider (IdP) to access your Amazon Redshift cluster. Following, you can find a procedure that describes how to set up a trust relationship for this purpose using the PingOne portal. For more information about configuring AWS as a service provider for Ping Identity, see Configuring Your SAML 2.0 IdP with Relying Party Trust and Adding Claims in the IAM User Guide.

**To set up Ping Identity and your AWS account to trust each other**

1. Create or use an existing Amazon Redshift cluster for your Ping Identity users to connect to. To configure the connection, certain properties of this cluster are needed, such as the cluster identifier. For more information, see Creating a Cluster.
2. Add Amazon Redshift as a new SAML application on the PingOne portal. For detailed steps, see the Ping Identity documentation.
   1. Go to My Applications.
   2. Under Add Application, choose New SAML Application.
   3. For Application Name, enter Amazon Redshift.
   4. For Protocol Version, choose SAML v2.0.
   5. For Category, choose your-application-category.
   6. For Assertion Consumer Service (ACS), type your-redshift-local-host-url. This is the local host and port that the SAML assertion redirects to.
   7. For Entity ID, enter urn:amazon:webservices.
   8. For Signing, choose Sign Assertion.
   9. In the SSO Attribute Mapping section, create the claims as shown in the following table.

<table>
<thead>
<tr>
<th>Application attribute</th>
<th>Identity bridge attribute of literal value</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://aws.amazon.com/SAML/Attributes/Role">https://aws.amazon.com/SAML/Attributes/Role</a></td>
<td>arn:aws:iam::123456789012:role/Ping,arn:aws:iam::123456789012:saml-provider/PingProvider</td>
</tr>
<tr>
<td><a href="https://aws.amazon.com/SAML/Attributes/RoleSessionName">https://aws.amazon.com/SAML/Attributes/RoleSessionName</a></td>
<td>email</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/AutoCreate">https://redshift.amazon.com/SAML/Attributes/AutoCreate</a></td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/DbUser">https://redshift.amazon.com/SAML/Attributes/DbUser</a></td>
<td>email</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/DbGroups">https://redshift.amazon.com/SAML/Attributes/DbGroups</a></td>
<td>The groups in the &quot;DbGroups&quot; attributes contain the @directory prefix. To remove this, in Identity bridge, enter memberOf. In Function, choose</td>
</tr>
</tbody>
</table>
3. For **Group Access**, set up the following group access, if needed:
   - https://aws.amazon.com/SAML/Attributes/Role
   - https://aws.amazon.com/SAML/Attributes/RoleSessionName
   - https://redshift.amazon.com/SAML/Attributes/AutoCreate
   - https://redshift.amazon.com/SAML/Attributes/DbUser

4. Review your setup and make changes, if necessary.

5. Use the **Initiate Single Sign-On (SSO) URL** as the login URL for the Browser SAML plugin.

6. Create an IAM SAML identity provider on the IAM console. The metadata document that you provide is the federation metadata XML file that you saved when you set up Ping Identity. For detailed steps, see Creating and Managing an IAM Identity Provider (Console) in the IAM User Guide.

7. Create an IAM role for SAML 2.0 federation on the IAM console. For detailed steps, see Creating a Role for SAML in the IAM User Guide.

8. Create an IAM policy that you can attach to the IAM role that you created for SAML 2.0 federation on the IAM console. For detailed steps, see Creating IAM Policies (Console) in the IAM User Guide. For an Azure AD example, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).

**To set up JDBC for authentication to Ping Identity**

- Configure your database client to connect to your cluster through JDBC using Ping Identity SSO.

You can use any client that uses a JDBC driver to connect using Ping Identity SSO or use a language like Java to connect using a script. For installation and configuration information, see Configuring a JDBC driver version 1.0 connection (p. 122).

For example, you can use SQLWorkbench/J as the client. When you configure SQLWorkbench/J, the URL of your database uses the following format.

```java
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

If you use SQLWorkbench/J as the client, take the following steps:

a. Start SQL Workbench/J. In the **Select Connection Profile** page, add a **Profile Group**, for example Ping.

b. For **Connection Profile**, enter `your-connection-profile-name`, for example Ping.

c. Choose **Manage Drivers**, and choose Amazon Redshift. Choose the **Open Folder** icon next to Library, then choose the appropriate JDBC .jar file.

d. On the **Select Connection Profile** page, add information to the connection profile as follows:

   - **For User**, enter your PingOne user name. This is the user name of the PingOne account that you are using for single sign-on that has permission to the cluster that you are trying to authenticate using.

   - **For Password**, enter your PingOne password.

   - **For Drivers**, choose Amazon Redshift (`com.amazon.redshift.jdbc.Driver`).


e. Choose **Extended Properties** and do one of the following:
• For login_url, enter your-ping-sso-login-url. This value specifies to the URL to use SSO as the authentication to log in.

• For Ping Identity, for plugin_name, enter com.amazon.redshift.plugin.PingCredentialsProvider. This value specifies to the driver to use Ping Identity SSO as the authentication method.

• For Ping Identity with SSO, for plugin_name, enter com.amazon.redshift.plugin.BrowserSamlCredentialsProvider. This value specifies to the driver to use Ping Identity PingOne with SSO as the authentication method.

To set up ODBC for authentication to Ping Identity

• Configure your database client to connect to your cluster through ODBC using Ping Identity PingOne SSO.

Amazon Redshift provides ODBC drivers for Linux, Windows, and macOS operating systems. Before you install an ODBC driver, determine whether your SQL client tool is 32-bit or 64-bit. Install the ODBC driver that matches the requirements of your SQL client tool.

Also install and configure the latest Amazon Redshift ODBC driver for your operating system as follows:

• For Windows, see Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132).

• For macOS, see Install the Amazon Redshift ODBC driver on macOS X (p. 135).

• For Linux, see Install the Amazon Redshift ODBC driver on Linux (p. 134).

On Windows, in the Amazon Redshift ODBC Driver DSN Setup page, under Connection Settings, enter the following information:

• For Data Source Name, enter your-DSN. This specifies the data source name used as the ODBC profile name.

• For Auth type, do one of the following:
  • For Ping Identity configuration, choose Identity Provider: Ping Federate. This is the authentication method that the ODBC driver uses to authenticate using Ping Identity SSO.
  • For Ping Identity with SSO configuration, choose Identity Provider: Browser SAML. This is the authentication method that the ODBC driver uses to authenticate using Ping Identity with SSO.

• For Cluster ID, enter your-cluster-identifier.

• For Region, enter your-cluster-region.

• For Database, enter your-database-name.

• For User, enter your-ping-username. This is the user name for the PingOne account that you are using for SSO that has permission to the cluster that you're trying to authenticate using. Use this only for Auth type is Identity Provider: PingFederate.

• For Password, enter your-ping-password. Use this only for Auth type is Identity Provider: PingFederate.

• For Listen Port, enter your-listen-port. This is the port that local server is listening to. The default is 7890. This applies only to the Browser SAML plugin.

• For Response Timeout, enter the-number-of-seconds. This is the number of seconds to wait before timing out when the IdP server sends back a response. The minimum number of seconds must be 10. If establishing the connection takes longer than this threshold, then the connection is aborted. This applies only to the Browser SAML plugin.

• For Login URL, enter your-login-url. This applies only to the Browser SAML plugin.
On macOS and Linux, edit the odbc.ini file as follows:

**Note**
All entries are case-insensitive.

- For **clusterid**, enter `your-cluster-identifier`. This is the name of the created Amazon Redshift cluster.
- For **region**, enter `your-cluster-region`. This is the AWS Region of the created Amazon Redshift cluster.
- For **database**, enter `your-database-name`. This is the name of the database that you're trying to access on the Amazon Redshift cluster.
- For **locale**, enter `en-us`. This is the language that error messages display in.
- For **iam**, enter 1. This value specifies to the driver to authenticate using IAM credentials.
- For **plugin_name**, do one of the following:
  - For Ping Identity configuration, enter `BrowserSAML`. This is the authentication method that the ODBC driver uses to authenticate to Ping Identity.
  - For Ping Identity with SSO configuration, enter `Ping`. This is the authentication method that the ODBC driver uses to authenticate using Ping Identity with SSO.
  - For **uid**, enter `your-ping-username`. This is the user name of the Microsoft Azure account you are using for SSO that has permission to the cluster you are trying to authenticate against. Use this only for **plugin_name** is `Ping`.
  - For **pwd**, enter `your-ping-password`. Use this only for **plugin_name** is `Ping`.
  - For **login_url**, enter `your-login-url`. This is the Initiate SSO URL that returns the SAML Response. This applies only to the Browser SAML plugin.
  - For **idp_response_timeout**, enter the-number-of-seconds. This is the specified period of time in seconds to wait for response from PingOne Identity. This applies only to the Browser SAML plugin.
  - For **listen_port**, enter `your-listen-port`. This is the port that local server is listening to. The default is 7890. This applies only to the Browser SAML plugin.

On macOS and Linux, also edit the profile settings to add the following exports.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

### Setting up JDBC or ODBC Single Sign-on authentication with Okta

You can use Okta as an identity provider (IdP) to access your Amazon Redshift cluster. Following, you can find a procedure that describes how to set up a trust relationship for this purpose. For more information about configuring AWS as a service provider for Okta, see Configuring Your SAML 2.0 IdP with Relying Party Trust and Adding Claims in the *IAM User Guide*.

**To set up Okta and your AWS account to trust each other**

1. Create or use an existing Amazon Redshift cluster for your Okta users to connect to. To configure the connection, certain properties of this cluster are needed, such as the cluster identifier. For more information, see Creating a Cluster.
2. Add Amazon Redshift as a new application on the Okta portal. For detailed steps, see the Okta documentation.
• Choose Add Application.
• Under Add Application, choose Create New App.
• On the Create a New Add Application Integration page, for Platform, choose Web.
• For Sign on method, choose SAML v2.0.
• On the General Settings page, for App name, enter your-redshift-saml-sso-name. This is the name of your application.
• On the SAML Settings page, for Single sign on URL, enter your-redshift-local-host-url. This is the local host and port that the SAML assertion redirects to, for example http://localhost:7890/redshift/.

3. Use the Single sign on URL value as the Recipient URL and Destination URL.
4. For Signing, choose Sign Assertion.
5. For Audience URI (SP Entity ID), enter urn:amazon:webservices for the claims, as shown in the following table.
6. In the Advanced Settings section, for SAML Issuer ID, enter your-Identity-Provider-Issuer-ID, which you can find in the View Setup Instructions section.
7. In the Attribute Statements section, create the claims as shown in the following table.

<table>
<thead>
<tr>
<th>Claim name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://aws.amazon.com/SAML/Attributes/RoleSessionName">https://aws.amazon.com/SAML/Attributes/RoleSessionName</a></td>
<td>user.email</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/AutoCreate">https://redshift.amazon.com/SAML/Attributes/AutoCreate</a></td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td><a href="https://redshift.amazon.com/SAML/Attributes/DbUser">https://redshift.amazon.com/SAML/Attributes/DbUser</a></td>
<td>user.email</td>
</tr>
</tbody>
</table>

8. In the App Embed Link section, find the URL that you can use as the login URL for the Browser SAML plugin.
9. Create an IAM SAML identity provider on the IAM console. The metadata document that you provide is the federation metadata XML file that you saved when you set up Okta. For detailed steps, see Creating and Managing an IAM Identity Provider (Console) in the IAM User Guide.
10. Create an IAM role for SAML 2.0 federation on the IAM console. For detailed steps, see Creating a Role for SAML in the IAM User Guide.
11. Create an IAM policy that you can attach to the IAM role that you created for SAML 2.0 federation on the IAM console. For detailed steps, see Creating IAM Policies (Console) in the IAM User Guide. For an Azure AD example, see Setting up JDBC or ODBC single sign-on authentication with Microsoft Azure AD (p. 323).

To set up JDBC for authentication to Okta

• Configure your database client to connect to your cluster through JDBC using Okta SSO.

You can use any client that uses a JDBC driver to connect using Okta SSO or use a language like Java to connect using a script. For installation and configuration information, see Configuring a JDBC driver version 1.0 connection (p. 122).
For example, you can use SQLWorkbench/J as the client. When you configure SQLWorkbench/J, the URL of your database uses the following format.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

If you use SQLWorkbench/J as the client, take the following steps:

a. Start SQL Workbench/J. In the Select Connection Profile page, add a Profile Group, for example Okta.

b. For Connection Profile, enter your-connection-profile-name, for example Okta.

c. Choose Manage Drivers, and choose Amazon Redshift. Choose the Open Folder icon next to Library, then choose the appropriate JDBC .jar file.

d. On the Select Connection Profile page, add information to the connection profile as follows:

   - For User, enter your Okta user name. This is the user name of the Okta account that you are using for single sign-on that has permission to the cluster that you are trying to authenticate using.
   - For Password, enter your Okta password.
   - For Drivers, choose Amazon Redshift (com.amazon.redshift.jdbc.Driver).

e. Choose Extended Properties and do one of the following:

   - For login_url, enter `your-okta-sso-login-url`. This value specifies to the URL to use SSO as the authentication to log in to Okta.
   - For Okta SSO, for plugin_name, enter `com.amazon.redshift.plugin.OktaCredentialsProvider`. This value specifies to the driver to use Okta SSO as the authentication method.
   - For Okta SSO with MFA, for plugin_name, enter `com.amazon.redshift.plugin.BrowserSamlCredentialsProvider`. This value specifies to the driver to use Okta SSO with MFA as the authentication method.

To set up ODBC for authentication to Okta

- Configure your database client to connect to your cluster through ODBC using Okta SSO.

Amazon Redshift provides ODBC drivers for Linux, Windows, and macOS operating systems. Before you install an ODBC driver, determine whether your SQL client tool is 32-bit or 64-bit. Install the ODBC driver that matches the requirements of your SQL client tool.

Also install and configure the latest Amazon Redshift ODBC driver for your operating system as follows:

- For Windows, see Install and configure the Amazon Redshift ODBC driver on Microsoft Windows (p. 132).
- For macOS, see Install the Amazon Redshift ODBC driver on macOS X (p. 135).
- For Linux, see Install the Amazon Redshift ODBC driver on Linux (p. 134).

On Windows, in the Amazon Redshift ODBC Driver DSN Setup page, under Connection Settings, enter the following information:

- For Data Source Name, enter `your-DSN`. This specifies the data source name used as the ODBC profile name.
• For **Auth type**, do one of the following:
  • For Okta SSO configuration, choose **Identity Provider: Okta**. This is the authentication method that the ODBC driver uses to authenticate using Okta SSO.
  • For Okta SSO with MFA configuration, choose **Identity Provider: Browser SAML**. This is the authentication method that the ODBC driver uses to authenticate using Okta SSO with MFA.

• For **Cluster ID**, enter *your-cluster-identifier*.
• For **Region**, enter *your-cluster-region*.
• For **Database**, enter *your-database-name*.
• For **User**, enter *your-okta-username*. This is the user name for the Okta account that you are using for SSO that has permission to the cluster that you're trying to authenticate using. Use this only for **Auth type** is **Identity Provider: Okta**.
  • For **Password**, enter *your-okta-password*. Use this only for **Auth type** is **Identity Provider: Okta**.

On macOS and Linux, edit the `odbc.ini` file as follows:

**Note**

All entries are case-insensitive.

• For **clusterid**, enter *your-cluster-identifier*. This is the name of the created Amazon Redshift cluster.
• For **region**, enter *your-cluster-region*. This is the AWS Region of the created Amazon Redshift cluster.
• For **database**, enter *your-database-name*. This is the name of the database that you're trying to access on the Amazon Redshift cluster.
• For **locale**, enter *en-us*. This is the language that error messages display in.
• For **iam**, enter *1*. This value specifies to the driver to authenticate using IAM credentials.
• For **plugin_name**, do one of the following:
  • For Okta SSO with MFA configuration, enter *BrowserSAML*. This is the authentication method that the ODBC driver uses to authenticate to Okta SSO with MFA.
  • For Okta SSO configuration, enter *Okta*. This is the authentication method that the ODBC driver uses to authenticate using Okta SSO.
  • For **uid**, enter *your-okta-username*. This is the user name of the Okta account you are using for SSO that has permission to the cluster you are trying to authenticate against. Use this only for **plugin_name** is **Okta**.
  • For **pwd**, enter *your-okta-password*. Use this only for **plugin_name** is **Okta**.
  • For **login_url**, enter *your-login-url*. This is the Initiate SSO URL that returns the SAML Response. This applies only to the Browser SAML plugin.
  • For **idp_response_timeout**, enter *the-number-of-seconds*. This is the specified period of time in seconds to wait for response from PingOne. This applies only to the Browser SAML plugin.
  • For **listen_port**, enter *your-listen-port*. This is the port that local server is listening to. The default is 7890. This applies only to the Browser SAML plugin.

On macOS and Linux, also edit the profile settings to add the following exports.

```bash
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```
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 login.

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. 310).

Note
If you use an IAM policy variable `${redshift:DbUser}`, as described in Resource policies for GetClusterCredentials (p. 300) 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. 305).

The following table lists the options for creating database user credentials.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbUser</td>
<td>The name of a database user. If a user named DbUser exists in the database, the temporary user credentials have the same permissions as the existing user. If DbUser doesn’t exist in the database and AutoCreate is true, a new user named DbUser is created. Optionally, disable the password for an existing user. For more information, see ALTER_USER.</td>
</tr>
<tr>
<td>AutoCreate</td>
<td>Specify true to create a database user with the name specified for DbUser if one does not exist. The default is false.</td>
</tr>
<tr>
<td>DbGroups</td>
<td>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 PUBLIC.</td>
</tr>
</tbody>
</table>

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 operation. Or you can configure your SQL client with Amazon Redshift JDBC or ODBC drivers that manage the process of calling the GetClusterCredentials operation, 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. 341).

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 operation 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 role or user with permissions to call GetClusterCredentials (p. 311).

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 operation 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 isn't specified, the user can log on to any cluster database.
   - **Database groups** – (Optional) A list of existing database user groups. Upon successful login, the database user is added to the specified user groups. If no group is specified, the user has only PUBLIC permissions. The 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": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM80yxpDhwXVPyJYBDm/gqXzBaq6F3DgTszFg=="
}
```

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 --auto-create --db-name exampledb --db-groups example_group --duration-seconds 3600
```

The result is as follows.

```
{
    "DbUser": "IAMA:temp_creds_user:example_group",
```
5. Establish a Secure Socket Layer (SSL) authentication connection with the Amazon Redshift cluster and send a login 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 Configuring security options for connections (p. 144).

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 S3 bucket. The CREATE EXTERNAL FUNCTION command can invoke an AWS Lambda function using a scalar Lambda user-defined function (UDF). 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 run the Amazon Redshift command. For more information, see Authorizing COPY, UNLOAD, CREATE EXTERNAL FUNCTION, and CREATE EXTERNAL SCHEMA operations using IAM roles (p. 348).

In addition, a superuser can grant the ASSUMEROLE privilege to specific users and groups to provide access to a role for COPY and UNLOAD operations. For information, see GRANT in the Amazon Redshift Database Developer Guide.

### 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. The values used in this section are examples, you can choose values based on your needs.

**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**. The **Attach permissions policy** page appears.
6. For access to Amazon S3 using COPY, as an example, you can use **AmazonS3ReadOnlyAccess** and append. For access to Amazon S3 using COPY or UNLOAD, we suggest that you can create managed policies that restrict access to the desired bucket and prefix accordingly. For both read and write operations, we recommend enforcing the least privileges and restricting to only the Amazon S3 buckets and key prefixes that Amazon Redshift requires.

   For access to invoke Lambda functions for the CREATE EXTERNAL FUNCTION command, add **AWSLambdaRole**.

   For Redshift Spectrum, in addition to Amazon S3 access, add **AWSGlueConsoleFullAccess** or **AmazonAthenaFullAccess**.

   Choose **Next: Tags**.

7. The **Add tags** page appears. You can optionally add tags. Choose **Next: Review**.
8. For **Role name**, type a name for your role, for example **RedshiftCopyUnload**. Choose **Create role**.
9. 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. 344).
10. 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. 349).

   **Note**
   To restrict access to specific data, use an IAM role that grants the least privileges required.

### 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`.
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:

```json
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Principal": {
```
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. The external ID can be any unique string.

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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "sts:ExternalId": [
      ]
    }
  }
  ]
}
```

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**

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:

```json
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "sts:ExternalId": [
      ]
    }
  }
  ]
}
```
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.

```json
{
  "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, Amazon Athena, AWS Glue, and AWS Lambda 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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.us-east-1.amazonaws.com",
          "redshift.us-west-2.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```
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",
      "Action": "sts:AssumeRole",
      "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",
      "Action": "sts:AssumeRole",
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" }
    }
  ]
}
```

**Note**
To restrict role chaining authorization to specific users, define a condition. For more information, see [Restricting access to IAM roles](p. 344).

When you run an UNLOAD, COPY, CREATE EXTERNAL FUNCTION, 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 'exampledb' region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

In the following example, CREATE EXTERNAL FUNCTION uses chained roles to assume the role RoleB.

```
create external function lambda_example(varchar)
returns varchar
volatile
lambda 'exampleLambdaFunction'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

Related topics

- Authorizing COPY, UNLOAD, CREATE EXTERNAL FUNCTION, and CREATE EXTERNAL SCHEMA operations using IAM roles (p. 348)

**Authorizing COPY, UNLOAD, CREATE EXTERNAL FUNCTION, 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. You can use the CREATE EXTERNAL FUNCTION command to create user-defined functions that invoke functions from AWS Lambda.

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 provide security credentials. These credentials authorize your Amazon Redshift cluster to read or write data to and from your target destination, such as an Amazon S3 bucket.

When you run the CREATE EXTERNAL FUNCTION, you provide security credentials using the IAM role parameter. These credentials authorize your Amazon Redshift cluster to invoke Lambda functions from AWS Lambda. 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. 345).
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, CREATE EXTERNAL SCHEMA, or CREATE EXTERNAL FUNCTION 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 for you, you must associate that role with an Amazon Redshift cluster. You must do this 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.

```json
{
    "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-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"
            ]
        }
    ]
}
```
After an IAM user has the appropriate permissions, that user can associate an IAM role with an Amazon Redshift cluster. The IAM role is then ready to 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.

Be aware of 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.

Note

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

New console

To manage IAM role associations

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to update.
3. For Actions, choose Manage IAM roles to display the current list IAM roles associated with the cluster.
4. On the Manage IAM roles page, choose the available IAM roles to add, and then choose Add IAM role.
5. Choose Done to save your changes.

Original console

To manage IAM role associations

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 See IAM Roles.
5. To associate an IAM role with the cluster, choose your IAM role for Available roles. 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.
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"
        }
      ],
    }
  ]
}
```
Logging and monitoring in Amazon Redshift

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Redshift and your AWS solutions. You can collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Amazon Redshift resources and responding to potential incidents:

Amazon CloudWatch Alarms

Using Amazon CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, a notification is sent to an Amazon SNS topic or AWS Auto Scaling policy. CloudWatch alarms do not invoke actions because they are in a particular state. Rather the state must have changed and been maintained for a specified number of periods. For more information, see Managing alarms (p. 443). For a list of metrics, see Monitoring Amazon Redshift using CloudWatch metrics (p. 401).

AWS CloudTrail Logs

CloudTrail provides a record of API operations taken by a IAM user, role, or an AWS service in Amazon Redshift. 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. For more information, see Logging Amazon Redshift API calls with AWS CloudTrail (p. 359).

Database audit logging

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, a process called database auditing. The logs are stored in Amazon S3 buckets. These provide convenient access with data security features for users who are responsible for monitoring activities in the database.

Topics

- Amazon Redshift logs (p. 354)
- Enabling logging (p. 356)
- Managing log files (p. 356)
- Troubleshooting Amazon Redshift audit logging (p. 359)
- Logging Amazon Redshift API calls with AWS CloudTrail (p. 359)
- Amazon Redshift account IDs in AWS CloudTrail logs (p. 364)
• Configuring auditing using the console (p. 366)
• Configuring logging by using the AWS CLI and Amazon Redshift API (p. 368)

Amazon Redshift logs

Amazon Redshift logs information in the following log files:

• Connection log – Logs authentication attempts, connections, and disconnections.
• User log – Logs information about changes to database user definitions.
• User activity log – Logs each query before it's 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. This information might be 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Connection or authentication event.</td>
</tr>
<tr>
<td>recordtime</td>
<td>Time the event occurred.</td>
</tr>
<tr>
<td>remotehost</td>
<td>Name or IP address of remote host.</td>
</tr>
<tr>
<td>remoteport</td>
<td>Port number for remote host.</td>
</tr>
<tr>
<td>pid</td>
<td>Process ID associated with the statement.</td>
</tr>
<tr>
<td>dbname</td>
<td>Database name.</td>
</tr>
<tr>
<td>username</td>
<td>User name.</td>
</tr>
<tr>
<td>authmethod</td>
<td>Authentication method.</td>
</tr>
<tr>
<td>duration</td>
<td>Duration of connection in microseconds.</td>
</tr>
<tr>
<td>sslversion</td>
<td>Secure Sockets Layer (SSL) version.</td>
</tr>
<tr>
<td>Column name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>sslcipher</td>
<td>SSL cipher.</td>
</tr>
<tr>
<td>mtu</td>
<td>Maximum transmission unit (MTU).</td>
</tr>
<tr>
<td>sslcompression</td>
<td>SSL compression type.</td>
</tr>
<tr>
<td>sslexpansion</td>
<td>SSL expansion type.</td>
</tr>
<tr>
<td>iamauthguid</td>
<td>The AWS Identity and Access Management (IAM) authentication ID for the AWS CloudTrail request.</td>
</tr>
<tr>
<td>application_name</td>
<td>The initial or updated name of the application for a session.</td>
</tr>
<tr>
<td>driver_version</td>
<td>The version of ODBC or JDBC driver that connects to your Amazon Redshift cluster from your third-party SQL client tools.</td>
</tr>
<tr>
<td>os_version</td>
<td>The version of the operating system that is on the client machine that connects to your Amazon Redshift cluster.</td>
</tr>
<tr>
<td>plugin_name</td>
<td>The name of the plugin used to connect to your Amazon Redshift cluster.</td>
</tr>
</tbody>
</table>

**User log**

Records details for the following changes to a database user:

- Create user
- Drop user
- Alter user (rename)
- Alter user (alter properties)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid</td>
<td>ID of user affected by the change.</td>
</tr>
<tr>
<td>username</td>
<td>User name of the user affected by the change.</td>
</tr>
<tr>
<td>oldusername</td>
<td>For a rename action, the original user name. For any other action, this field is empty.</td>
</tr>
<tr>
<td>action</td>
<td>Action that occurred. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- Alter</td>
</tr>
<tr>
<td></td>
<td>- Create</td>
</tr>
<tr>
<td></td>
<td>- Drop</td>
</tr>
<tr>
<td></td>
<td>- Rename</td>
</tr>
<tr>
<td>usecreatedb</td>
<td>If true (1), indicates that the user has create database privileges.</td>
</tr>
<tr>
<td>usesuper</td>
<td>If true (1), indicates that the user is a superuser.</td>
</tr>
<tr>
<td>usecatupd</td>
<td>If true (1), indicates that the user can update system catalogs.</td>
</tr>
<tr>
<td>valuntil</td>
<td>Password expiration date.</td>
</tr>
<tr>
<td>pid</td>
<td>Process ID.</td>
</tr>
</tbody>
</table>
Database audit logging

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xid</td>
<td>Transaction ID.</td>
</tr>
<tr>
<td>recordtime</td>
<td>Time in UTC that the query started.</td>
</tr>
</tbody>
</table>

**User activity log**

Logs each query before it is run on the database.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordtime</td>
<td>Time the event occurred.</td>
</tr>
<tr>
<td>db</td>
<td>Database name.</td>
</tr>
<tr>
<td>user</td>
<td>User name.</td>
</tr>
<tr>
<td>pid</td>
<td>Process ID associated with the statement.</td>
</tr>
<tr>
<td>userid</td>
<td>User ID.</td>
</tr>
<tr>
<td>xid</td>
<td>Transaction ID.</td>
</tr>
<tr>
<td>query</td>
<td>A prefix of LOG: followed by the text of the query, including newlines.</td>
</tr>
</tbody>
</table>

**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 time audit logging is enabled 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 log information for only the connection log and user log, but not for the user activity log. The enable_user_activity_logging parameter is not enabled (false) by default. You can set it to true to enable the user activity log. For more information, see Amazon Redshift parameter groups (p. 201).

**Note**

Currently, you can only use Amazon S3-managed keys (SSE-S3) encryption (AES-256) for audit logging.

**Managing log files**

The number and size of Amazon Redshift log files in Amazon S3 depends heavily on the activity in your cluster. If you have an active cluster that is generating a large number of logs, Amazon Redshift might
Database audit logging

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 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. As part of this, determine when the log files 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. Also, the IAM user or IAM role that enables logging must have s3:PutObject permission to the Amazon S3 bucket. 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, you receive an error.

If you have Amazon Redshift create a new bucket for you as part of configuration, correct permissions are 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. You also need the Amazon Redshift account ID that corresponds to your AWS Region from the following table.

<table>
<thead>
<tr>
<th>Region name</th>
<th>Region</th>
<th>Account ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia) Region</td>
<td>us-east-1</td>
<td>193672423079</td>
</tr>
<tr>
<td>US East (Ohio) Region</td>
<td>us-east-2</td>
<td>391106570357</td>
</tr>
<tr>
<td>US West (N. California) Region</td>
<td>us-west-1</td>
<td>262260360010</td>
</tr>
<tr>
<td>US West (Oregon) Region</td>
<td>us-west-2</td>
<td>902366379725</td>
</tr>
<tr>
<td>Africa (Cape Town) Region</td>
<td>af-south-1</td>
<td>365689465814</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong) Region</td>
<td>ap-east-1</td>
<td>313564881002</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai) Region</td>
<td>ap-south-1</td>
<td>865932855811</td>
</tr>
<tr>
<td>Asia Pacific (Osaka) Region</td>
<td>ap-northeast-3</td>
<td>090321488786</td>
</tr>
<tr>
<td>Asia Pacific (Seoul) Region</td>
<td>ap-northeast-2</td>
<td>760740231472</td>
</tr>
<tr>
<td>Asia Pacific (Singapore) Region</td>
<td>ap-southeast-1</td>
<td>361669875840</td>
</tr>
<tr>
<td>Asia Pacific (Sydney) Region</td>
<td>ap-southeast-2</td>
<td>762762565011</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo) Region</td>
<td>ap-northeast-1</td>
<td>404641285394</td>
</tr>
<tr>
<td>Canada (Central) Region</td>
<td>ca-central-1</td>
<td>907379612154</td>
</tr>
<tr>
<td>Europe (Frankfurt) Region</td>
<td>eu-central-1</td>
<td>053454850223</td>
</tr>
<tr>
<td>Europe (Ireland) Region</td>
<td>eu-west-1</td>
<td>210876761215</td>
</tr>
</tbody>
</table>
The bucket policy uses the following format, where *BucketName* is a placeholder for your own value. The *AccountId* is from the table of Amazon Redshift account IDs that corresponds to your AWS Region.

```
{
  "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::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, see 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_LogType_Timestamp.gz
```

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. 357).

- 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. 367).

- 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. 367).

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. These include calls from the Amazon Redshift console and from code calls to the Amazon
Redshift API operations. 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 certain details. These include the request that was made to Amazon Redshift, the IP address it was made from, who made it, when it was made, and other information.

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](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/).

### 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](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/).

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](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)
- [CloudTrail Supported Services and Integrations](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)
- [Configuring Amazon SNS Notifications for CloudTrail](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)
- [Receiving CloudTrail Log Files from Multiple Regions](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)
- [Receiving CloudTrail Log Files from Multiple Accounts](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)

All Amazon Redshift actions are logged by CloudTrail and are documented in the [Amazon Redshift API Reference](https://docs.aws.amazon.com/redshift/latest/api/index.html). 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](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/).

### 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.

```json
{
    "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",
"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",
  "nodeType": "dc1.large",
  "publiclyAccessible": true,
  "vpcSecurityGroupIds": [
    "sg-95f60f6fc"
  ]
},
"responseElements": {
  "nodeType": "dc1.large",
  "preferredMaintenanceWindow": "sat:05:30-sat:06:00",
  "clusterStatus": "creating",
  "vpcId": "vpc-84c22aecd",
  "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-95f60f6fc"
    }
  ]
}
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": "d4c50603e-0032-11e7-b8bf-d7aa466e9920",
  "eventID": "13ba8550-56ac-405b-900a-8a42b0f43c45",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

**Working with data sharing information in CloudTrail**

All Amazon Redshift data sharing API operations are logged by CloudTrail. For example, calls to the `AuthorizeDataShare`, `DeauthorizeDataShare`, and `DescribeDataShares` operations generate entries in the CloudTrail log files. For information about the data sharing API operations, see the Amazon Redshift API Reference.

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 an IAM role or federated user.
- Whether the request was made by another AWS service.
For more information about CloudTrail `userIdentity` element, see CloudTrail `userIdentity` Element.

Understanding log file entries for data sharing

A trail in CloudTrail 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. An event includes information about the requested action, the date and time of the action, or request parameters. CloudTrail log files aren't an ordered stack trace of the public API calls; they don't appear in any specific order.

The following example shows a CloudTrail log entry that illustrates the `AuthorizeDataShare` operation.

```json
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
    "arn": "arn:aws:sts::111122223333:user/janedoe",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH89DBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn": "arn:aws:sts::111122223333:user/janedoe",
        "accountId": "111122223333",
        "userName": "janedoe"
      },
      "attributes": {
        "creationDate": "2021-08-02T23:40:45Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2021-08-02T23:40:58Z",
  "eventSource": "redshift.amazonaws.com",
  "eventName": "AuthorizeDataShare",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "3.227.36.75",
  "userAgent": "aws-cli/1.18.118 Python/3.6.10 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.41",
  "requestParameters": {
    "dataShareArn": "arn:aws:redshift:us-east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
    "consumerIdentifier": "555555555555"
  },
  "responseElements": {
    "dataShareArn": "arn:aws:redshift:us-east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
    "producerNamespaceArn": "arn:aws:redshift:us-east-1:123456789012:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
    "producerArn": "arn:aws:redshift:us-east-1:111122223333:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
    "allowPubliclyAccessibleConsumers": true,
    "dataShareAssociations": [
      {
        "consumerIdentifier": "555555555555",
        "status": "AUTHORIZED",
        "createdDate": "Aug 2, 2021 11:40:56 PM",
        "statusChangeDate": "Aug 2, 2021 11:40:57 PM"
      }
    ]
  },
}
You can use Amazon S3 bucket notification and direct Amazon S3 to publish object-created events to AWS Lambda. When CloudTrail writes logs to your S3 bucket, Amazon S3 can then invoke your Lambda function by passing the Amazon S3 object-created event as a parameter. Your Lambda function can read this log object and process the access records logged by CloudTrail. For more information, see Using AWS Lambda with AWS CloudTrail.

Amazon Redshift account IDs in AWS CloudTrail logs

When Amazon Redshift calls another AWS service for you, the call is logged with an account ID that belongs to Amazon Redshift. It isn't logged with your account ID. For example, suppose that Amazon Redshift calls AWS Key Management Service (AWS KMS) operations such as CreateGrant, Decrypt, Encrypt, and RetireGrant to manage encryption on your cluster. In this case, 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.

<table>
<thead>
<tr>
<th>Region</th>
<th>Region</th>
<th>Account ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia) Region</td>
<td>us-east-1</td>
<td>368064434614</td>
</tr>
<tr>
<td>US East (Ohio) Region</td>
<td>us-east-2</td>
<td>790247189693</td>
</tr>
<tr>
<td>US West (N. California) Region</td>
<td>us-west-1</td>
<td>703715109447</td>
</tr>
<tr>
<td>US West (Oregon) Region</td>
<td>us-west-2</td>
<td>473191095985</td>
</tr>
<tr>
<td>Africa (Cape Town) Region</td>
<td>af-south-1</td>
<td>420376844563</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong) Region</td>
<td>ap-east-1</td>
<td>651179539253</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai) Region</td>
<td>ap-south-1</td>
<td>408097707231</td>
</tr>
<tr>
<td>Asia Pacific (Osaka) Region</td>
<td>ap-northeast-3</td>
<td>398671365691</td>
</tr>
<tr>
<td>Asia Pacific (Seoul) Region</td>
<td>ap-northeast-2</td>
<td>713597048934</td>
</tr>
<tr>
<td>Asia Pacific (Singapore) Region</td>
<td>ap-southeast-1</td>
<td>960118270566</td>
</tr>
<tr>
<td>Asia Pacific (Sydney) Region</td>
<td>ap-southeast-2</td>
<td>485979073181</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo) Region</td>
<td>ap-northeast-1</td>
<td>615915377779</td>
</tr>
<tr>
<td>Canada (Central) Region</td>
<td>ca-central-1</td>
<td>764870610256</td>
</tr>
<tr>
<td>Europe (Frankfurt) Region</td>
<td>eu-central-1</td>
<td>434091160558</td>
</tr>
<tr>
<td>Europe (Ireland) Region</td>
<td>eu-west-1</td>
<td>246478207311</td>
</tr>
<tr>
<td>Europe (London) Region</td>
<td>eu-west-2</td>
<td>885798887673</td>
</tr>
<tr>
<td>Europe (Milan) Region</td>
<td>eu-south-1</td>
<td>041313461515</td>
</tr>
</tbody>
</table>
The following example shows a CloudTrail log entry for the AWS KMS Decrypt operation that was called by Amazon Redshift.

```json
{
   "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",
      }
   },
   "responseElements": null,
   "requestID": "30d2fe51-0035-11e7-ab67-17595a9411c8",
   "eventID": "619bad54-1764-4de4-a786-889bb0af400c",
   "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"
}
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**

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

**New console**

**To enable audit logging for a cluster**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster that you want to update.
3. Choose the Properties tab. Then view the Audit logging in the Database configurations section.
4. Choose Edit, then Edit audit logging.
5. On the Edit audit logging page, choose to Enable audit logging and enter your choices regarding where the logs are stored.
6. Choose Save changes to save your choices.

**Original console**

**To enable audit logging 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, choose Clusters.
3. In the list, choose the cluster for which you want to enable logging.
4. In the cluster details page, choose Database, and then choose Configure Audit Logging.
5. In the Configure Audit Logging dialog box, in the Enable Audit Logging box, choose 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. (Optional) For **S3 Key Prefix**, enter a prefix to add to the S3 bucket.
8. Choose **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**, choose **Go to the S3 console** to navigate to the bucket.

**Modifying the bucket for audit logging**

*(Original console) Modifying the Amazon S3 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/](https://console.aws.amazon.com/redshift/).
2. In the navigation pane, choose **Clusters**.
3. In the list, choose the cluster for which you want to modify the bucket used for audit logging.
4. In the cluster details page, choose **Database**, and then choose **Configure Audit Logging**.
5. For **S3 Bucket**, select an existing bucket or create a new bucket.
6. (Optional) For **S3 Key Prefix**, enter a prefix to add to the S3 bucket.
7. Choose **Save**.
Disabling audit logging using the Console

(Original console) Disabling 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, choose Clusters.
3. In the list, choose the cluster for which you want to disable logging.
4. In the cluster details page, choose Database, and then choose Configure Audit Logging.
5. In the Configure Audit Logging dialog box, in the Enable Audit Logging box, choose No.
6. Choose Save.

Configuring logging by using the AWS CLI and Amazon Redshift 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 operations to configure audit logging:

- DescribeLoggingStatus
- DisableLogging
- EnableLogging

Compliance validation for Amazon Redshift

Third-party auditors assess the security and compliance of Amazon Redshift as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using Amazon Redshift is determined by the sensitivity of your data, your organization's compliance objectives, and applicable laws and regulations. If your use of Amazon Redshift is subject to compliance with standards like HIPAA, PCI, or FedRAMP, AWS provides resources to help:
Resilience in Amazon Redshift

The AWS global infrastructure is built around AWS Regions and Availability Zones (AZs). AWS Regions provide multiple, physically separated and isolated Availability Zones that are connected with low latency, high throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single data center infrastructures or multiple data center infrastructures.

Almost all AWS Regions have multiple Availability Zones and data centers. You can deploy your applications across multiple Availability Zones in the same Region for fault tolerance and low latency.

For more information on AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in Amazon Redshift

As a managed service, Amazon Redshift is protected by the AWS global network security procedures described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access Amazon Redshift through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.
Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

You can call these API operations from any network location. In addition, Amazon Redshift supports resource-based access policies that can include restrictions based on the source IP address.

**Network isolation**

A virtual private cloud (VPC) based on the Amazon VPC service is your private, logically isolated network in the AWS Cloud. You can deploy an Amazon Redshift cluster within a VPC by taking the following steps:

- Create a VPC in an AWS Region. For more information, see What is Amazon VPC? in the *Amazon VPC User Guide*.
- Create two or more private VPC subnets. For more information, see VPCs and subnets in the *Amazon VPC User Guide*.
- Deploy an Amazon Redshift cluster. For more information, see Amazon Redshift cluster subnet groups (p. 74).

An Amazon Redshift cluster is locked down by default upon provisioning. To allow inbound network traffic from Amazon Redshift clients, associate a VPC security group with an Amazon Redshift cluster. For more information, see Amazon Redshift cluster subnet groups (p. 74).

To allow traffic only to or from specific IP address ranges, update the security groups with your VPC. An example is allowing traffic only from or to your corporate network.

Amazon Redshift supports deploying clusters into dedicated tenancy VPCs. For more information, see Dedicated instances in the *Amazon EC2 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-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 a VPC (p. 69).

If you are on the EC2-Classic platform, you define a cluster security group and associate it with a cluster as described in the following sections. We recommend that you launch your cluster in a EC2-VPC platform instead of an EC2-Classic platform. However, you can restore an EC2-Classic snapshot to an EC2-VPC cluster using the Amazon Redshift console. For more information, see Restoring a cluster from a snapshot (p. 245).

**Topics**

- Overview (p. 370)
- Managing cluster security groups using the console (p. 371)
- Managing cluster security groups using the AWS SDK for Java (p. 382)
- Manage cluster security groups using the AWS CLI and Amazon Redshift API (p. 385)

**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.

You can't delete the default cluster security group, but you can modify it by authorizing or revoking ingress access.

Creating a cluster security group

New console

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. On the navigation menu, choose CONFIG, then choose Security groups to display the Cluster security groups page.

Note
You can only manage cluster security groups when logged in with an EC2-Classic AWS account.

3. Choose Create cluster security group to display the Create cluster security group window.
4. For the new security group, enter values for the following:
• Name
• Description
• **CIDR/IP range to authorize** in the form `nnn.nnn.nnn.nn/nn`
• AWS account ID (without hyphens)
• EC2 security group name

5. Choose **Create** to create the security group.

**Original console**

**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.

5. Choose **Create**.

The new group is displayed in the list of cluster security groups.

**Tagging a cluster security group**

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

To 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. On the navigation menu, choose CONFIG, then choose Security groups to display the Cluster security groups page.

   **Note**
   You can only manage cluster security groups when logged in with an EC2-Classic AWS account.
3. Choose a cluster security group, then choose Manage tags to display the Manage tags page.
4. On the Manage tags page, add new tags, and update or delete existing tags. For each new tag, provide information for Key and Value.
5. Choose Apply to save your tags.

Original console

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.
4. In the Manage Tags dialog box, do one of the following:
   - Remove a tag.
     - In the Applied Tags section, choose Delete next to the tag you want to remove.
     - Choose Apply Changes.
• Add a tag.
  • In the Add Tags section, enter a key-value pair for the tag.
  • Choose Apply Changes.
Managing ingress rules for a cluster security group

(Original console) Manage 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**.

5. In the **Add Connection Type** dialog box, do one of the following:
   - 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**.
   - 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**, specify the AWS account ID.
     - For **EC2 Security Group Name**, enter the name of the EC2 security group that you want.
     - Choose **Authorize**.
Revoking ingress rules for a cluster security group

(Original console) Revoke 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 that you want to remove and choose Revoke.
Tagging ingress rules for a cluster security group

(Original console) Tag 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 that you want to tag and choose Manage Tags.
5. In the Manage Tags dialog box, do one of the following:
   - Remove a tag:
     - In the Applied Tags section, choose Delete next to the tag that you want to remove.
     - Choose Apply Changes.
• 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 also, do that separately.

• In the **Add Tags** section, enter a key-value pair for the tag.
• Choose **Apply Changes**.
For more information about tagging an Amazon Redshift resource, see How to manage tags in the Amazon Redshift console (p. 478).

Deleting a cluster security group

If a cluster security group is associated with one or more clusters, you can't delete it.

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

To 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. On the navigation menu, choose CONFIG, then choose Security groups to display the Cluster security groups page.

   **Note**
   You can only manage cluster security groups when logged in with an EC2-Classic AWS account.

3. Choose the security group that you want to delete, then choose Delete.

Original console

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 can't 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

If you are on the EC2-VPC platform, see Managing VPC security groups for a cluster (p. 73) for more information about associating VPC security groups with your cluster. We recommend that you launch your cluster in an EC2-VPC platform. However, you can restore an EC2-Classic snapshot to an EC2-VPC cluster using the Amazon Redshift console. For more information, see Restoring a cluster from a snapshot (p. 245).

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 a launch cluster (p. 46) and To modify a cluster (p. 55).

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. 391). You need to update the code and provide a cluster identifier and AWS account number.

Example

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * 
 * http://aws.amazon.com/apache2.0/
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[CreateAndModifyClusterSecurityGroup demonstrates how to
create and modify an Amazon Redshift security group.]
// snippet-service:[redshift]
// snippet-keyword:[Java]
// snippet-keyword:[Amazon Redshift]
```
package com.amazonaws.services.redshift;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSecurityGroup {

    public static AmazonRedshift client;
    public static String clusterSecurityGroupName = "securitygroup1";
    public static String clusterIdentifier = "***provide a cluster identifier***";
    public static String ownerID = "***provide a 12-digit account number***";

    public static void main(String[] args) throws IOException {
        try {
            createClusterSecurityGroup();
            describeClusterSecurityGroups();
            addIngressRules();
            associateSecurityGroupWithCluster();
        } catch (Exception e) {
            System.err.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'
", 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'
", clusterSecurityGroupName);
    }

    private static void describeClusterSecurityGroups() {
    }

    private static void associateSecurityGroupWithCluster() {
    }
}
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("Associated 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("Describe cluster security groups result is null.");
        return;
    }
    System.out.println("Printing 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 AWS CLI and Amazon Redshift 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 API operations to manage cluster security groups.

- AuthorizeClusterSecurityGroupIngress
- CreateClusterSecurityGroup
- DeleteClusterSecurityGroup
- DescribeClusterSecurityGroups
- RevokeClusterSecurityGroupIngress

Connecting to Amazon Redshift using an interface VPC endpoint

You can connect directly to Amazon Redshift API service using an interface VPC endpoint (AWS PrivateLink) in your virtual private cloud (VPC) instead of connecting over the internet. For information about Amazon Redshift API actions, see Actions in the Amazon Redshift API Reference. For more information about AWS PrivateLink, see Interface VPC endpoints (AWS PrivateLink) in the Amazon VPC User Guide. Note that JDBC/ODBC connection to the cluster is not part of Amazon Redshift API service.

When you use an interface VPC endpoint, communication between your VPC and Amazon Redshift is conducted entirely within the AWS network, which can provide greater security. Each VPC endpoint is represented by one or more elastic network interfaces with private IP addresses in your VPC subnets. For more information on elastic network interfaces, see Elastic network interfaces in the Amazon EC2 User Guide for Linux Instances.

An interface VPC endpoint connects your VPC directly to Amazon Redshift. It doesn't use an internet gateway, network address translation (NAT) device, virtual private network (VPN) connection, or AWS Direct Connect connection. The instances in your VPC don't need public IP addresses to communicate with the Amazon Redshift API.

To use Amazon Redshift through your VPC, you have two options. One is to connect from an instance that is inside your VPC. The other is to connect your private network to your VPC by using an AWS VPN option or AWS Direct Connect. For more information about AWS VPN options, see VPN connections in the Amazon VPC User Guide. For information about AWS Direct Connect, see Creating a Connection in the AWS Direct Connect User Guide.

You can create an interface VPC endpoint to connect to Amazon Redshift using the AWS Management Console or AWS Command Line Interface (AWS CLI) commands. For more information, see Creating an Interface Endpoint.
After you create an interface VPC endpoint, you can enable private DNS host names for the endpoint. When you do, the default Amazon Redshift endpoint (https://redshift.Region.amazonaws.com) resolves to your VPC endpoint.

If you don't enable private DNS host names, Amazon VPC provides a DNS endpoint name that you can use in the following format.

VPC_endpoint_ID.redshift.Region.vpce.amazonaws.com

For more information, see Interface VPC endpoints (AWS PrivateLink) in the Amazon VPC User Guide.

Amazon Redshift supports making calls to all of its API operations inside your VPC.

You can attach VPC endpoint policies to a VPC endpoint to control access for AWS Identity and Access Management (IAM) principals. You can also associate security groups with a VPC endpoint to control inbound and outbound access based on the origin and destination of network traffic. An example is a range of IP addresses. For more information, see Controlling Access to Services with VPC Endpoints in the Amazon VPC User Guide.

Creating a VPC endpoint policy for Amazon Redshift

You can create a policy for VPC endpoints for Amazon Redshift to specify the following:

- The principal that can or can't perform actions
- The actions that can be performed
- The resources on which actions can be performed

For more information, see Controlling access to services with VPC endpoints in the Amazon VPC User Guide.

Following, you can find examples of VPC endpoint policies.

Topics

- Example: VPC endpoint policy to deny all access from a specified AWS account (p. 386)
- Example: VPC endpoint policy to allow VPC access only to a specified IAM principal (user) (p. 387)
- Example: VPC endpoint policy to allow read-only Amazon Redshift operations (p. 387)
- Example: VPC endpoint policy denying access to a specified cluster (p. 388)

Example: VPC endpoint policy to deny all access from a specified AWS account

The following VPC endpoint policy denies the AWS account 123456789012 all access to resources using this endpoint.

```json
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": "123456789012"
    }
  ]
}
```
Example: VPC endpoint policy to allow VPC access only to a specified IAM principal (user)

The following VPC endpoint policy allows full access only to the IAM user `redshiftadmin` in AWS account `123456789012`. All other IAM principals are denied access using the endpoint.

```json
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/redshiftadmin"
                ]
            }
        }
    ]
}
```

Example: VPC endpoint policy to allow read-only Amazon Redshift operations

The following VPC endpoint policy allows only AWS account `123456789012` to perform the specified Amazon Redshift actions.

The actions specified provide the equivalent of read-only access for Amazon Redshift. All other actions on the VPC are denied for the specified account. Also, all other accounts are denied any access. For a list of Amazon Redshift actions, see Actions, Resources, and Condition Keys for Amazon Redshift in the IAM User Guide.

```json
{
    "Statement": [
        {
            "Action": [
                "redshift:DescribeAccountAttributes",
                "redshift:DescribeClusterParameterGroups",
                "redshift:DescribeClusterParameters",
                "redshift:DescribeClusterSecurityGroups",
                "redshift:DescribeClusterSnapshots",
                "redshift:DescribeClusterSubnetGroups",
                "redshift:DescribeClusterVersions",
                "redshift:DescribeDefaultClusterParameters",
                "redshift:DescribeEventCategories",
                "redshift:DescribeEventSubscriptions",
                "redshift:DescribeHsmClientCertificates",
```
Connecting using an interface VPC endpoint

Example: VPC endpoint policy denying access to a specified cluster

The following VPC endpoint policy allows full access for all accounts and principals. At the same time, it denies any access for AWS account 123456789012 to actions performed on the Amazon Redshift cluster with cluster ID my-redshift-cluster. Other Amazon Redshift actions that don’t support resource-level permissions for clusters are still allowed. For a list of Amazon Redshift actions and their corresponding resource type, see Actions, Resources, and Condition Keys for Amazon Redshift in the IAM User Guide.

```json
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": ["123456789012"]
      }
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Principal": {
        "AWS": ["123456789012"]
      }
    }
  ]
}
```
Configuration and vulnerability analysis in Amazon Redshift

AWS handles basic security tasks like guest operating system (OS) and database patching, firewall configuration, and disaster recovery (DR). These procedures have been reviewed by certified third parties. For more information, see Compliance validation for Amazon Redshift (p. 368), the Shared responsibility model, and Best Practices for Security, Identity, and Compliance.

Amazon Redshift automatically applies upgrades and patches your data warehouse so you can focus on your application and not on its administration. Patches and upgrades are applied during a configurable maintenance window. For more information, see Maintenance windows (p. 15).
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 API** – You can call this Amazon Redshift management API by submitting a request. Requests are HTTP or HTTPS requests that use the HTTP verbs `GET` or `POST` with a parameter named `Action`. Calling the Amazon Redshift 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 an Amazon Redshift API request, see Signing an HTTP request (p. 393).
- For information about the Amazon Redshift 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 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 API actions.

- For information about installing and setting up the Amazon Redshift CLI, see Setting up the Amazon Redshift CLI (p. 395).
- 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. 390)
- Signing an HTTP request (p. 393)
- Setting up the Amazon Redshift CLI (p. 395)

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 ds2.xlarge cluster.

```java
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("ds2.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. 382) 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/aws-java-sdk-1.3.18.jar: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. 390). 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. For more information about creating access keys, see Managing access keys for IAM users in the IAM User Guide.

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
&NumberOfNode=2
&NodeType=ds2.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=20121207T000000Z
&x-amz-signedheaders=content-type;host;x-amz-date
```

The canonical form of the request calculated for Task 1: Create a Canonical Request (p. 393) is:

```
POST /content-type:application/x-www-form-urlencoded; charset=utf-8
host:redshift.us-east-1.amazonaws.com
x-amz-date:20121207T000000Z
c
```

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. 393) is:

```
AWS4-HMAC-SHA256
20121207T000000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b50c627cc6c5b5a959b902b5ac2187be80cbac0714
```

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. 393). The service name to use in the credential scope is redshift.

For Task 3: Create a Signature (p. 393), the derived key can be represented as:

```
derived key = HMAC(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 wJalrXUtNPEMI/
X7MDENG/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:

   ```bash
   aws help
   ```

   The following command should display help information for Amazon Redshift:

   ```bash
   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 AWS Command Line Interface (AWS CLI), 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 Guide, but they are focused on the AWS CLI 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 use the AWS 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, see the Amazon Redshift Getting Started Guide.

Step 1.2: Download and install the AWS CLI

If you have not installed the AWS CLI, see Setting up the Amazon Redshift CLI (p. 395).

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, see the Amazon Redshift Getting Started Guide.

Step 2: Launch a cluster

Now you're ready to launch a cluster by using the AWS 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

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Parameter value for this exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Identifier</td>
<td>examplecluster</td>
</tr>
</tbody>
</table>
## Parameter name | Parameter value for this exercise
--- | ---
Master Username | masteruser
Master Password | TopSecret1
Node Type | ds2.xlarge or the node size that you want to use. For more information, see Clusters and nodes in Amazon Redshift (p. 5)
Cluster Type | single-node

To create your cluster, enter the following command.

```
aws redshift create-cluster --cluster-identifier examplecluster --master-username masteruser --master-user-password TopSecret1 --node-type ds2.xlarge --cluster-type single-node
```

The cluster creation process will take several minutes to complete. To check the status, enter the following command.

```
aws redshift describe-clusters --cluster-identifier examplecluster
```

The output will look similar to the following.

```
{
   "Clusters": [
   {
       ...
       "ClusterStatus": "creating",
       "ClusterIdentifier": "examplecluster",
       ...
   }
}
```

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, enter the following command.
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, enter 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 Guide.

**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, enter 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 that you can work with in Amazon Redshift, specifically in the Amazon Redshift console.

Topics

• Overview (p. 400)
• Monitoring Amazon Redshift using CloudWatch metrics (p. 401)
• Working with performance data in the Amazon Redshift console (p. 408)

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 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 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 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 CloudWatch. For example, WriteThroughput, is displayed in GB/s (as compared to bytes/s in 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 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 CloudWatch to access the Amazon Redshift performance data that is exposed as CloudWatch metrics, see What is CloudWatch? in the Amazon CloudWatch User Guide.
Monitoring Amazon Redshift using CloudWatch metrics

Using 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 [CloudWatch concepts](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/clc_concepts.html) in the *Amazon CloudWatch User Guide*.

For a further description of CloudWatch metrics for Amazon Redshift, see the following sections.

**Topics**
- [Amazon Redshift metrics](#)
- [Dimensions for Amazon Redshift metrics](#)
- [Amazon Redshift query and load performance data](#)

## Amazon Redshift metrics

The AWS/Redshift namespace includes the following metrics. Unless stated otherwise, metrics are collected at 1-minute intervals.

### Title

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CommitQueueLength</code></td>
<td>The number of transactions waiting to commit at a given point in time.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: <code>ClusterIdentifier</code></td>
</tr>
<tr>
<td><code>ConcurrencyScalingActiveClusters</code></td>
<td>The number of concurrency scaling clusters that are actively processing queries at any given time.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: <code>ClusterIdentifier</code></td>
</tr>
<tr>
<td><code>ConcurrencyScalingSeconds</code></td>
<td>The number of seconds used by concurrency scaling clusters that have active query processing activity.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Units</strong>: Sum</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier</td>
</tr>
<tr>
<td>CPUUtilization</td>
<td>The percentage of CPU utilization. For clusters, this metric represents an</td>
</tr>
<tr>
<td></td>
<td>aggregation of all nodes (leader and compute) CPU utilization values.</td>
</tr>
<tr>
<td></td>
<td><strong>Units</strong>: Percent</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier</td>
</tr>
<tr>
<td>DatabaseConnections</td>
<td>The number of database connections to a cluster.</td>
</tr>
<tr>
<td></td>
<td><strong>Units</strong>: Count</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier</td>
</tr>
<tr>
<td>HealthStatus</td>
<td>Indicates the health of the cluster. Every minute the cluster connects to</td>
</tr>
<tr>
<td></td>
<td>its database and performs a simple query. If it is able to perform this</td>
</tr>
<tr>
<td></td>
<td>operation successfully, the cluster is considered healthy. Otherwise, the</td>
</tr>
<tr>
<td></td>
<td>cluster is unhealthy. An unhealthy status can occur when the cluster</td>
</tr>
<tr>
<td></td>
<td>database is under extremely heavy load or if there is a configuration</td>
</tr>
<tr>
<td></td>
<td>problem with a database on the cluster.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>In Amazon CloudWatch, this metric is reported as 1 or 0 whereas in the</td>
</tr>
<tr>
<td></td>
<td>Amazon Redshift console, this metric is displayed with the words HEALTHY or</td>
</tr>
<tr>
<td></td>
<td>UNHEALTHY for convenience. When this metric is displayed in the Amazon</td>
</tr>
<tr>
<td></td>
<td>Redshift console, sampling averages are ignored and only HEALTHY or</td>
</tr>
<tr>
<td></td>
<td>UNHEALTHY are displayed. In Amazon CloudWatch, values different than 1 and</td>
</tr>
<tr>
<td></td>
<td>0 might occur because of sampling issue. Any value below 1 for HealthStatus</td>
</tr>
<tr>
<td></td>
<td>is reported as 0 (UNHEALTHY).</td>
</tr>
<tr>
<td></td>
<td><strong>Units</strong>: Count (1/0) (HEALTHY/UNHEALTHY in the Amazon Redshift console)</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier</td>
</tr>
<tr>
<td>MaintenanceMode</td>
<td>Indicates whether the cluster is in maintenance mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>In Amazon CloudWatch, this metric is reported as 1 or 0 whereas in the</td>
</tr>
<tr>
<td></td>
<td>Amazon Redshift console, this metric is displayed with the words ON or OFF</td>
</tr>
<tr>
<td></td>
<td>for convenience. When this metric is displayed in the Amazon Redshift</td>
</tr>
<tr>
<td></td>
<td>console, sampling averages are ignored and only ON or OFF are displayed. In</td>
</tr>
<tr>
<td></td>
<td>Amazon CloudWatch, values different than 1 and 0 might occur because of</td>
</tr>
<tr>
<td></td>
<td>sampling issues. Any value greater than 0 for MaintenanceMode is reported as</td>
</tr>
<tr>
<td></td>
<td>1 (ON).</td>
</tr>
<tr>
<td></td>
<td><strong>Units</strong>: Count (1/0) (ON/OFF in the Amazon Redshift console).</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong>: ClusterIdentifier</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MaxConfiguredConcurrencyScalingClusters</td>
<td>Maximum number of concurrency scaling clusters configured from the parameter group. For more information, see Amazon Redshift parameter groups (p. 201).</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>NetworkReceiveThroughput</td>
<td>The rate at which the node or cluster receives data.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes/Second (MB/s in the Amazon Redshift console)</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>NetworkTransmitThroughput</td>
<td>The rate at which the node or cluster writes data.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes/Second (MB/s in the Amazon Redshift console)</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>PercentageDiskSpaceUsed</td>
<td>The percent of disk space used.</td>
</tr>
<tr>
<td></td>
<td>Units: Percent</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>QueriesCompletedPerSecond</td>
<td>The average number of queries completed per second. Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Count/Second</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, latency</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, wlmid</td>
</tr>
<tr>
<td>QueryDuration</td>
<td>The average amount of time to complete a query. Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Microseconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID, latency</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, latency</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID, wlmid</td>
</tr>
<tr>
<td>QueryRuntimeBreakdown</td>
<td>The total time queries spent running by query stage. Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Milliseconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID, stage</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, stage</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadIOPS</td>
<td>The average number of disk read operations per second.</td>
</tr>
<tr>
<td></td>
<td>Units: Count/Second</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>ReadLatency</td>
<td>The average amount of time taken for disk read I/O operations.</td>
</tr>
<tr>
<td></td>
<td>Units: Seconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>ReadThroughput</td>
<td>The average number of bytes read from disk per second.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes (GB/s in the Amazon Redshift console)</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>TotalTableCount</td>
<td>The number of user tables open at a particular point in time.</td>
</tr>
<tr>
<td></td>
<td>This total doesn't include Amazon Redshift Spectrum tables.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>WLMQueueLength</td>
<td>The number of queries waiting to enter a workload management (WLM) queue.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, service class</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueueName</td>
</tr>
<tr>
<td>WLMQueueWaitTime</td>
<td>The total time queries spent waiting in the workload management (WLM) queue.</td>
</tr>
<tr>
<td></td>
<td>Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Milliseconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueryPriority</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, wlmid</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueueName</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WLMQueriesCompletedPerSecond</td>
<td>The average number of queries completed per second for a workload management (WLM) queue. Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Count/Second</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, wmid</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueueName</td>
</tr>
<tr>
<td>WLMQueryDuration</td>
<td>The average length of time to complete a query for a workload management (WLM) queue. Reported in 5-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>Units: Microseconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, wmid</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueueName</td>
</tr>
<tr>
<td>WLMRunningQueries</td>
<td>The number of queries running from both the main cluster and concurrency scaling cluster per WLM queue.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, wmid</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, QueueName</td>
</tr>
<tr>
<td>WriteIOPS</td>
<td>The average number of write operations per second.</td>
</tr>
<tr>
<td></td>
<td>Units: Count/Second</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>WriteLatency</td>
<td>The average amount of time taken for disk write I/O operations.</td>
</tr>
<tr>
<td></td>
<td>Units: Seconds</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
<tr>
<td>WriteThroughput</td>
<td>The average number of bytes written to disk per second.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes (GB/s in the Amazon Redshift console)</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier, NodeID</td>
</tr>
<tr>
<td></td>
<td>Dimensions: ClusterIdentifier</td>
</tr>
</tbody>
</table>
## Dimensions for Amazon Redshift metrics

Amazon Redshift data can be filtered along any of the dimensions in the table following.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>latency</td>
<td>Possible values are as follows:</td>
</tr>
</tbody>
</table>
Amazon Redshift query and load performance data

In addition to the 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

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Description</td>
</tr>
<tr>
<td>NodeID</td>
<td>Filters requested data that is specific to the nodes of a cluster. NodeID 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. 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.</td>
</tr>
<tr>
<td>ClusterIdentifier</td>
<td>Filters requested data that is specific to the cluster. Metrics that are specific to clusters include HealthStatus, MaintenanceMode, and DatabaseConnections. General metrics for this dimension (for example, ReadIOPS) 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.</td>
</tr>
<tr>
<td>service class</td>
<td>The identifier for a WLM service class.</td>
</tr>
<tr>
<td>stage</td>
<td>The execution stages for a query. The possible values are as follows:</td>
</tr>
<tr>
<td></td>
<td>• QueryPlanning: Time spent parsing and optimizing SQL statements.</td>
</tr>
<tr>
<td></td>
<td>• QueryWaiting: Time spent waiting in the WLM queue.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingRead: Time spent executing read queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingInsert: Time spent executing insert queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingDelete: Time spent executing delete queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingUpdate: Time spent executing update queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingCtas: Time spent executing create table as queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingUnload: Time spent executing unload queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryExecutingCopy: Time spent executing copy queries.</td>
</tr>
<tr>
<td></td>
<td>• QueryCommit: Time spent committing.</td>
</tr>
<tr>
<td>wlmid</td>
<td>The identifier for a workload management queue.</td>
</tr>
<tr>
<td>QueryPriority</td>
<td>The priority of the query. Possible values are CRITICAL, HIGHEST, HIGH, NORMAL, LOW, and LOWEST.</td>
</tr>
<tr>
<td>QueueName</td>
<td>The name of the workload management queue.</td>
</tr>
</tbody>
</table>
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 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 (for more information, 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.

<table>
<thead>
<tr>
<th>Query/Load data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query summary</td>
<td>A list of queries in a specified time period. The list can be sorted on values such as query ID, query runtime, and status. View this data in the Query monitoring tab of the cluster detail page.</td>
</tr>
<tr>
<td>Query detail</td>
<td>Provides details on a particular query including:</td>
</tr>
<tr>
<td></td>
<td>• Query properties such as the query ID, type, cluster the query was run on, and runtime.</td>
</tr>
<tr>
<td></td>
<td>• Details such as the status of the query and the number of errors.</td>
</tr>
<tr>
<td></td>
<td>• The SQL statement that was run.</td>
</tr>
<tr>
<td></td>
<td>• An explain plan if available.</td>
</tr>
<tr>
<td></td>
<td>• Cluster performance data during the query execution (for more information, see Viewing query history data (p. 416)).</td>
</tr>
<tr>
<td>Load summary</td>
<td>Lists all the loads in a specified time period. The list can be sorted on values such as query ID, query runtime, and status. View this data in the Query monitoring tab of the cluster detail page.</td>
</tr>
<tr>
<td>Load detail</td>
<td>Provides details on a particular load operation including:</td>
</tr>
<tr>
<td></td>
<td>• Load properties such as the query ID, type, cluster the query was run on, and runtime.</td>
</tr>
<tr>
<td></td>
<td>• Details such as the status of the load and the number of errors.</td>
</tr>
<tr>
<td></td>
<td>• The SQL statement that was run.</td>
</tr>
<tr>
<td></td>
<td>• A list of loaded files.</td>
</tr>
<tr>
<td></td>
<td>• Cluster performance data during the load operation (for more information, see Viewing query history data (p. 416)).</td>
</tr>
</tbody>
</table>

Working with performance data in the Amazon Redshift console

In this section, you can find 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 Monitoring Amazon Redshift using...
CloudWatch metrics (p. 401), you can also use CloudWatch to further customize your metrics graphs. For example, you can choose longer times or combine metrics across clusters. For more information about working with the CloudWatch console, see Working with performance metrics in the CloudWatch console (p. 445).

Watch the following video to learn how to monitor, isolate, and optimize your queries using the query monitoring features on the Amazon Redshift console: Query Monitoring with Amazon Redshift.

Topics
• Viewing cluster performance data (p. 409)
• Viewing query history data (p. 416)
• Viewing database performance data (p. 419)
• Viewing workload concurrency and concurrency scaling data (p. 423)
• Viewing queries and loads (p. 426)
• Viewing cluster metrics during load operations (p. 437)
• Analyzing workload performance (p. 439)
• Managing alarms (p. 443)
• Working with performance metrics in the CloudWatch console (p. 445)

Viewing cluster performance data

By using cluster metrics in Amazon Redshift, you can do the following common performance tasks:

• 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.

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

New console

To view performance data

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the name of a cluster from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose the Cluster performance tab for performance information including the following:
   • CPU utilization
   • Percentage disk space used
   • Database connections
   • Health status
   • Query duration
   • Query throughput
• Concurrency scaling activity

Many more metrics are available. To see the available metrics and choose which are displayed, choose the Preferences icon.

Original console

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.

<table>
<thead>
<tr>
<th>To do this</th>
<th>Use this filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the time range for which data is displayed</td>
<td>Select a time range from the Time Range list. By default, the last hour is shown.</td>
</tr>
<tr>
<td>Change the period for which data is displayed</td>
<td>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</td>
</tr>
</tbody>
</table>
**To do this** | **Use this filter**
---|---
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.

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 performance graphs**

The following examples show some of the graphs that are displayed in the new Amazon Redshift console.

- **CPU utilization** – Shows the percentage of CPU utilization for all nodes (leader and compute). To find a time when the cluster usage is lowest before scheduling cluster migration or other resource-consuming operations, monitor this chart to see CPU utilization per individual or all of nodes.

- **Maintenance mode** – Shows whether the cluster is in the maintenance mode at a chosen time by using **On** and **Off** indicators. You can see the time when the cluster is undergoing maintenance. You can then correlate this time to operations that are done to the cluster to estimate its future downtimes for recurring events.
• **Percentage disk space used** – Shows the percentage of disk space usage per each compute node, and not for the cluster as a whole. You can explore this chart to monitor the disk utilization. Maintenance operations like VACUUM and COPY use intermediate temporary storage space for their sort operations, so a spike in disk usage is expected.

• **Read throughput** – Shows the average number of megabytes read from disk per second. You can evaluate this chart to monitor the corresponding physical aspect of the cluster. This throughput doesn’t include network traffic between instances in the cluster and its volume.

• **Read latency** – Shows the average amount of time taken for disk read I/O operations per millisecond. You can view the response times for the data to return. When latency is high, it means that the sender spends more time idle (not sending any new packets), which reduces how fast throughput grows.
• **Write throughput** – Shows the average number of megabytes written to disk per second. You can evaluate this metric to monitor the corresponding physical aspect of the cluster. This throughput doesn't include network traffic between instances in the cluster and its volume.

• **Write latency** – Shows the average amount of time in milliseconds taken for disk write I/O operations. You can evaluate the time for the write acknowledgment to return. When latency is high, it means that the sender spends more time idle (not sending any new packets), which reduces how fast throughput grows.

• **Database connections** – Shows the number of database connections to a cluster. You can use this chart to see how many connections are established to the database and find a time when the cluster usage is lowest.
• **Total table count** – Shows the number of user tables open at a particular point in time within a cluster. You can monitor the cluster performance when open table count is high.

![Database connections graph](image)

• **Health status** – Indicates the health of the cluster as **Healthy** or **Unhealthy**. If the cluster can connect to its database and performs a simple query 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.

![Total table count graph](image)

![Health status graph](image)

• **Query duration** – Shows the average amount of time to complete a query in microseconds. You can benchmark the data on this chart to measure I/O performance within the cluster and tune its most time-consuming queries if necessary.
• **Query throughput** – Shows the average number of completed queries per second. You can analyze data on this chart to measure database performance and characterize the ability of the system to support a multiuser workload in a balanced way.

- **Query duration per WLM queue** – Shows the average amount of time to complete a query in microseconds. You can benchmark the data on this chart to measure I/O performance per WLM queue and tune its most time-consuming queries if necessary.

- **Query throughput per WLM queue** – Shows the average number of completed queries per second. You can analyze data on this chart to measure database performance per WLM queue.
• **Concurrency scaling activity** – Shows the number of active concurrency scaling clusters. When concurrency scaling is enabled, Amazon Redshift automatically adds additional cluster capacity when you need it to process an increase in concurrent read queries.

**Viewing query history data**

You can use query history metrics in Amazon Redshift to do the following:

• Isolate and diagnose query performance problems.

• Compare query runtime metrics and cluster performance metrics on the same timeline to see how the two might be related. Doing so helps identify poorly performing queries, look for bottleneck queries, and determine if you need to resize your cluster for your workload.

• Drill down to the details of a specific query by choosing it in the timeline. When *Query ID* and other properties are displayed in a row below the graph, then you can choose the query to see query details. Details include, for example, the query’s SQL statement, execution details, and query plan. For more information, see [Viewing query details](p. 427).

• Determine if your load jobs complete successfully and meet your service level agreements (SLAs).

**To display query history data**

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

2. On the navigation menu, choose **CLUSTERS**, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include **Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance,** and **Properties** tabs.

3. Choose the **Query monitoring** tab for metrics about your queries.
4. In the **Query monitoring** section, choose the **Query history** tab.

Using controls on the window, you can toggle between **Query list** and **Cluster metrics**.

When you choose **Query list**, the tab includes the following graphs:

- **Query runtime** – The query activity on a timeline. Use this graph to see which queries are running in the same timeframe. Choose a query to view more query execution details. The x-axis shows the selected period. You can filter the graphed queries by running, completed, loads, and so on. Each bar represents a query, and the length of the bar represents its runtime from the start of the bar to the end. The queries can include SQL data manipulation statements (such as SELECT, INSERT, DELETE) and loads (such as COPY). By default, the top 100 longest running queries are shown for the selected time period.

- **Queries and loads** – List of queries and loads that ran on the cluster. The window includes an option to **Terminate query** if a query is currently running.

When you choose **Cluster metrics**, the tab includes the following graphs:

- **Query runtime** – The query activity on a timeline. Use this graph to see which queries are running in the same timeframe. Choose a query to view more query execution details.

- **CPU utilization** – The CPU utilization of the cluster by leader node and average of compute nodes.

- **Storage capacity used** – The percent of the storage capacity used.

- **Active database connections** – The number of active database connections to the cluster.

Consider the following when working with the query history graphs:

- Choose a bar that represents a specific query on the **Query runtime** chart to see details about that query. You can also, choose a query ID on **Queries and loads** list to see its details.

- You can swipe to select a section of the **Query runtime** chart to zoom in to display a specific time period.

- On the **Query runtime** chart, to have all data considered by your chosen filter, page forward through all pages listed on the **Queries and loads** list.

- You can change which columns and the number of rows displayed on the **Queries and loads** list using the preferences window displayed by the **settings gear icon**.

- The **Queries and loads** list can also be displayed by navigating from the left navigator **Queries** icon, **Queries and loads**. For more information, see Viewing queries and loads (p. 426).

**Query history graphs**

The following examples show graphs that are displayed in the new Amazon Redshift console.

**Note**

The Amazon Redshift console graphs only contain data for the latest 100,000 queries.

- **Query runtime**
**Viewing query history data**

- Queries and loads

<table>
<thead>
<tr>
<th>Start time</th>
<th>Query</th>
<th>Status</th>
<th>Duration</th>
<th>SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 13th, 2020 01:00:55 PM</td>
<td>69248</td>
<td>Completed</td>
<td>11 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 12:58:07 PM</td>
<td>69199</td>
<td>Completed</td>
<td>11 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 12:54:15 PM</td>
<td>69111,69265,69253</td>
<td>Completed</td>
<td>10 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 12:50:17 PM</td>
<td>68976</td>
<td>Completed</td>
<td>10 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 01:29:23 PM</td>
<td>70089</td>
<td>Completed</td>
<td>10 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 11:18:35 AM</td>
<td>65543</td>
<td>Completed</td>
<td>9 min</td>
<td></td>
</tr>
<tr>
<td>Apr 13th, 2020 12:40:30 PM</td>
<td>68729</td>
<td>Completed</td>
<td>9 min</td>
<td></td>
</tr>
</tbody>
</table>

- CPU utilization
You can use database performance metrics in Amazon Redshift to do the following:

- Analyze the time spent by queries by processing stages. You can look for unusual trends in the amount of time spent in a stage.
• Analyze the number of queries, duration, and throughput of queries by duration ranges (short, medium, long).

• Look for trends in the about of query wait time by query priority (Lowest, Low, Normal, High, Highest, Critical).

• Look for trends in the query duration, throughput, or wait time by WLM queue.

To display database performance data

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

2. On the navigation pane, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, including Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.

3. Choose the Query monitoring tab for metrics about your queries.

4. In the Query monitoring section, choose Database performance tab.

Using controls on the window, you can toggle between Cluster metrics and WLM queue metrics.

When you choose Cluster metrics, the tab includes the following graphs:

• Workload execution breakdown – The time used in query processing stages.

• Queries by duration range – The number of short, medium, and long queries.

• Query throughput – The average number of queries completed per second.

• Query duration – The average amount of time to complete a query.

• Average queue wait time by priority – The total time queries spent waiting in the WLM queue by query priority.

When you choose WLM queue metrics, the tab includes the following graphs:

• Query duration by queue – The average query duration by WLM queue.

• Query throughput by queue – The average number of queries completed per second by WLM queue.

• Query wait time by queue – The average duration of queries spent waiting by WLM queue.

Database performance graphs

The following examples show graphs that are displayed in the new Amazon Redshift console.

• Workload execution breakdown
• Queries by duration range

• Query throughput

• Query duration
• **Average queue wait time by priority**

  ![Average queue wait time by priority](image)

  The total time queries spent waiting in the WLM queue by query priority.

• **Query duration by queue**

  ![Query duration by queue](image)

  The average query duration by WLM queue.

• **Query throughput by queue**

  ![Query throughput by queue](image)
Viewing workload concurrency and concurrency scaling data

By using concurrency scaling metrics in Amazon Redshift, you can do the following:

- Analyze whether you can reduce the number of queued queries by enabling concurrency scaling. You can compare by WLM queue or for all WLM queues.
- View concurrency scaling activity in concurrency scaling clusters. This can tell you if concurrency scaling is limited by the `max_concurrency_scaling_clusters`. If so, you can choose to increase the `max_concurrency_scaling_clusters` in the DB parameter.
- View the total usage of concurrency scaling summed across all concurrency scaling clusters.

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

To display concurrency scaling data

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose the Query monitoring tab for metrics about your queries.
4. In the Query monitoring section, choose Workload concurrency tab.

The tab includes the following graphs:

- **Queued vs. Running queries on the cluster** – The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number of queries waiting in all WLM queues in the cluster.
- **Queued vs. Running queries per queue** – The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number or queries waiting in each WLM queue.
- **Concurrency scaling activity** – The number of concurrency scaling clusters that are actively processing queries.
- **Concurrency scaling usage** – The usage of concurrency scaling clusters that have active query processing activity.

Original console

To view concurrency scaling 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. Choose the cluster for which you want to view concurrency scaling data.
4. Choose the Database Performance tab.

By default, the performance view displays concurrency scaling data over the past hour. If you need to fine-tune the view, you can use filters to change the scope of the data.

Workload concurrency graphs

The following examples show graphs that are displayed in the new Amazon Redshift console.

- **Queued vs. Running queries on the cluster**
Viewing workload concurrency and concurrency scaling data

- Queued vs. Running queries per queue

- Concurrency scaling activity

- Concurrency scaling usage
Viewing queries and loads

The Amazon Redshift console provides information about queries and loads 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 information in the Amazon Redshift console to monitor query processing.

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

**New console**

**To display query performance data**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at [https://console.aws.amazon.com/redshift/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose QUERIES, and then choose Queries and loads to display the list of queries for your account.

   By default, the list displays queries for all your clusters over the past 24 hours. You can change the scope of the displayed date in the console.

   **Important**
   The Queries and loads list displays the longest running queries in the system, up to 100 queries.

**Original console**

**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/](https://console.aws.amazon.com/redshift/).
2. In the navigation pane, choose Clusters.
3. Choose the cluster for which you want to view performance data.
4. By default, the **Queries** list 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 enter a keyword in the **Search** box to search for queries that match your search criteria.

**Ending a running query**

You can also use the **Queries** page to end 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. 291).

*Note*

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

**New console**

**To end 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. On the navigation menu, choose **QUERIES**, and then choose **Queries and loads** to display the list of queries for your account.
3. Choose the running query that you want to end in the list, and then choose **Terminate query**.

**Original console**

**To end 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 analyze query details on the Amazon Redshift console. With a query identifier, you can view details of a query. Details can include, for example, the query's completion status, duration, SQL
statement and whether it’s a user query or one that was rewritten by Amazon Redshift. A user query is a query that is submitted to Amazon Redshift, either from an SQL client or generated by a business intelligence tool. Amazon Redshift might rewrite the query to optimize it, and this can result in multiple rewritten queries. Although the process is done by Amazon Redshift, you see the rewritten queries on the query details page along with the user query.

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

New console

To view a query

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose QUERIES, and then choose Queries to display the list of queries for your account. You might need to change settings on this page to find your query.
3. Choose the Query identifier in the list to display Query details.

The Query details page includes Query details and Query plan tabs with metrics about the query.

Metrics include details about a query such as start time, query ID, status, and duration. Other details include whether a query ran on a main cluster or a concurrency scaling cluster, and if it’s a parent or rewritten query.

Original console

You can view details for a particular query by choosing 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, the duration, and its status. The Executed on property indicates whether the query ran on a main cluster or a concurrency scaling cluster.
- **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. Query execution details aren’t available for queries that ran on a concurrency scaling cluster. For information on using the Query Execution Details section, see Analyzing query execution (p. 429).
- **Cluster Performance During Query Execution.** Displays performance metrics from CloudWatch. For information on using the Cluster Performance During Query Execution section, see Viewing cluster performance as queries run (p. 436).

The Query view looks similar to the following when you open it.

428
Viewing queries and loads

Analyzing query execution

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

To analyze a query

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **QUERIES**, and then choose **Queries and loads** to display the list of queries for your account. You might need to change settings on this page to find your query.
3. Choose the **Query** identifier in the list to display **Query details**.

   The **Query details** page includes **Query details** and **Query plan** tabs with metrics about the query.

   **Note**
   You can also navigate to the **Query details** page from a **Cluster details** page, **Query history** tab when you drill down into a query in a **Query runtime** graph.

The **Query details** page contains the following sections:

- A list of **Rewritten queries**, as shown in the following screenshot.

- A **Query details** section, as shown in the following screenshot.

- A **Query details** tab that contains the SQL that was run and **Execution details** about the run.

- A **Query plan** tab that contains the **Query plan** steps and other information about the query plan. This table also contains graphs about the cluster when the query ran.

- **Cluster health status**
Viewing queries and loads

- CPU utilization
- Storage capacity used
- Active database connections

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.

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.

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. The result is based on the number of tickets sold in 2008 and the query plan for that query.

```sql
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;
```
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. An example is its 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.

<table>
<thead>
<tr>
<th>Hash</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Node</td>
<td>Hash</td>
</tr>
<tr>
<td>Estimated</td>
<td>Actual</td>
</tr>
<tr>
<td>Cost</td>
<td>624.88 (0.00% of total)</td>
</tr>
<tr>
<td>Rows</td>
<td>45</td>
</tr>
<tr>
<td>Width</td>
<td>47 bytes</td>
</tr>
<tr>
<td>Row scanned</td>
<td>Rows returned</td>
</tr>
<tr>
<td>Project</td>
<td>-</td>
</tr>
<tr>
<td>Hash</td>
<td>-</td>
</tr>
</tbody>
</table>

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 cluster performance as queries run

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

New console

To display cluster performance as queries run
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose the Query monitoring tab for more details.

For more information, see Viewing query history data (p. 416).

Original console

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.
4. Choose the Queries tab.
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 queries with the longest run times.
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 act 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. 291).

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

New console

To display cluster performance 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. On the navigation menu, choose CLUSTERS, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include Cluster performance, Query monitoring, Databases, Datashares, Schedules, Maintenance, and Properties tabs.
3. Choose the Query monitoring tab for more details.
4. In the Queries and loads section, choose Loads to view the load operations of a cluster. If a load is running, you can end it by choosing Terminate query.

Original console

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.

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 queries and loads (p. 426). 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 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.

**Note**

A new console is available for Amazon Redshift. Choose either the [New console](#) or the [Original console](#) instructions based on the console that you are using. The [New console](#) instructions are open by default.

**New console**

Following, is an example of the workload execution breakdown chart. In the chart, the y-axis value is the average duration of each stage at the specified time shown as a stacked bar graph.

![Workload execution breakdown chart](image)

The following diagram illustrates how Amazon Redshift aggregates query processing for concurrent sessions.

![Query processing diagram](image)
The y-axis in the diagram is cumulative for all sessions running during the selected time period. The following diagram illustrates how Amazon Redshift aggregates query processing for concurrent sessions.

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.

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

**New console**

**To display the cluster workload breakdown chart**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **CLUSTERS**, then choose the cluster name from the list to open its details. The details of the cluster are displayed, which can include **Cluster performance**, **Query monitoring**, **Databases**, **Datashares**, **Schedules**, **Maintenance**, and **Properties** tabs.
3. Choose the **Query monitoring** tab for metrics about your queries.
4. In the **Query monitoring** section, choose **Database performance**, and choose **Cluster metrics**.

   The following metrics are graphed for the chosen time range as a stacked bar chart:

   - **Plan** time
   - **Wait** time
   - **Commit** time
   - **Execution** time

**Original console**

**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.

**Managing alarms**

Alarms you create in the Amazon Redshift console are 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 **Monitoring Amazon Redshift using CloudWatch metrics** (p. 401). 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.

From **Actions**, you can modify or delete alarms. You can also create a chime or slack alert to send an alert from CloudWatch to Slack or Amazon Chime by specifying a Slack or Amazon Chime webhook URL.

In this section, you can find how to create an alarm using the Amazon Redshift console. You can create an alarm using the CloudWatch console or any other way you work with metrics, such as with the AWS CLI or an AWS SDK.

**Note**

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

**New console**

**To create a CloudWatch alarm with 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. On the navigation menu, choose **ALARMS**, then choose **Create alarm**.
3. On the Create alarm page, enter the properties to create a CloudWatch alarm.
4. Choose Create alarm.

**Original 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.
4. Choose the Events+Alarms tab.
5. Choose Create Alarm.
6. In the Create Alarm dialog box, configure an alarm, and choose Create.

*Note*

The notifications that are displayed in 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 choosing 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`.

7. 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 CloudWatch console.

**Working with performance metrics in the CloudWatch console**

When working with Amazon Redshift metrics in the 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 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 CloudWatch), which is a more relevant unit for the typical storage space of a node.

When working with Amazon Redshift metrics in the CloudWatch console, command line tools, or an Amazon SDK, keep these concepts in mind:

1. First, 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 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 CloudWatch Developer Guide.

2. Then, specify the metric name, such as ReadIOPS.

The following table summarizes the types of Amazon Redshift metric dimensions that are available to you. Depending on the metric, data is available in either 1-minute or 5-minute intervals at no charge. For more information, see Amazon Redshift metrics (p. 401).

<table>
<thead>
<tr>
<th>CloudWatch namespace</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS/Redshift</td>
<td>NodeID</td>
<td>Filters requested data that is specific to the nodes of a cluster. NodeID is either &quot;Leader&quot;, &quot;Shared&quot;, or &quot;Compute-N&quot; where N is 0, 1, ... for the number of nodes in the cluster. &quot;Shared&quot; means that the cluster has only one node, that is the leader node and compute node are combined.</td>
</tr>
<tr>
<td></td>
<td>ClusterIdentifier</td>
<td>Filters requested data that is specific to the cluster. Metrics that are specific to clusters include HealthStatus, MaintenanceMode, and DatabaseConnections. General metrics for this dimension (for example, ReadIOPS) 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.</td>
</tr>
</tbody>
</table>

Working with gateway and volume metrics is similar to working with other service metrics. Many of the common tasks are outlined in the CloudWatch documentation, including the following:

• View available metrics
• Get statistics for a metric
• Creating CloudWatch alarms
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.

Amazon Redshift provides notification in advance for some events. These events have an event category of `pending`. For example, we send an advance notification if a hardware update is required for one of the nodes in your cluster. You can subscribe to pending events the same as other Amazon Redshift events. For more information, see Subscribing to Amazon Redshift event notifications (p. 451).

You can use the Amazon Redshift Management 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. The `Source` column shows the resource name and resource type that triggers a given action.

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. 452)

### Viewing events using the console

**Note**

A new console is available for Amazon Redshift. Choose either the [New console](#) or the [Original console](#) instructions based on the console that you are using. The [New console](#) instructions are open by default.

#### New console

**To view events**

1. Sign in to the AWS Management Console and open the Amazon Redshift console at [https://console.aws.amazon.com/redshift/](https://console.aws.amazon.com/redshift/).
2. On the navigation menu, choose EVENTS.
Original console

To view events in the Amazon Redshift console, choose Events on the navigation pane. In the list of events, you can filter the results using Source Type or a custom Filter value that filters for text in all fields of the list. For example, if you search for 12 Dec 2012, you match Date fields that contain this value.

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 might 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, choose Events.
3. To filter events, do one of the following:
   a. To filter by event type, choose Filter Cluster and choose the source type.
   b. To filter by text in the event description, enter a value in the search box. 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. 391). You need to update the code and specify a cluster identifier and event source type.

Example

```java
/**
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * This file is licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License. A copy of
 * the License is located at
 * http://aws.amazon.com/apache2.0/
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

// snippet-sourcedescription:[ListEvents demonstrates how to list Amazon Redshift events.]
// snippet-service:[redshift]
// snippet-keyword:[Java]
// snippet-keyword:[Amazon Redshift]
// snippet-keyword:[Code Sample]
// snippet-keyword:[DescribeEvents]
// snippet-keyword:[full-example]
// snippet-sourcedate:[2019-01-31]
// snippet-sourceauthor:[AWS]
// snippet-start:[redshift.java.ListEvents.complete]

package com.amazonaws.services.redshift;

import java.util.Date;
import java.io.IOException;

import com.amazonaws.services.redshift.model.*;

public class ListEvents {

    public static AmazonRedshift client;
    public static String clusterIdentifier = "***provide cluster identifier***";
    public static String eventSourceType = "cluster"; // e.g. cluster-snapshot

    public static void main(String[] args) throws IOException {
        // Default client using the (@link com.amazonaws.auth.DefaultAWSCredentialsProviderChain)
        client = AmazonRedshiftClientBuilder.defaultClient();

        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("Event object is null.");
        return;
    }
    System.out.println("Event metadata:
    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());
}

Viewing events using the AWS CLI and Amazon Redshift 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. 447).

Topics
• Subscribing to Amazon Redshift event notifications (p. 451)

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 AWS 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.

<table>
<thead>
<tr>
<th>Amazon Redshift category</th>
<th>Event ID</th>
<th>Event severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1000</td>
<td>INFO</td>
<td>The parameter group [parameter group name] was updated at [time]. If you changed only dynamic parameters, associated clusters are being modified now. If you changed static parameters, all updates, including dynamic parameters, will be applied when you reboot the associated clusters.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1001</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster [cluster name] was modified to use parameter group [parameter group name] at [time].</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1500</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1501</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1502</td>
<td>ERROR</td>
<td>Subnets in cluster subnet group [subnet group name] have no available IP addresses. Cluster [cluster name] could not be created.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1503</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1504</td>
<td>ERROR</td>
<td>The HSM for cluster [cluster name] is unreachable.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1505</td>
<td>ERROR</td>
<td>The HSM for cluster [cluster name] cannot be registered. Try a different configuration.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1506</td>
<td>ERROR</td>
<td>Amazon Redshift exceeded your account's elastic network interface limit. Delete up to [maximum number of elastic network interfaces] elastic network interfaces or request a limit increase of the number of network interfaces per AWS Region with EC2.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1509</td>
<td>ERROR</td>
<td>The Amazon Redshift cluster [cluster name] can't be created because your account's VPC endpoint limit has been reached. Delete unused VPC endpoints or request an increase in the limit of VPC endpoints. For more information, see VPC endpoints in the Amazon VPC User Guide.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2000</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster: [cluster name] has been created and is ready for use.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2001</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster [cluster name] was deleted at [time]. A final snapshot [was / was not] saved.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2002</td>
<td>INFO</td>
<td>Your VPC security group [security group name] was updated at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2003</td>
<td>INFO</td>
<td>Maintenance started on your Amazon Redshift cluster [cluster name] at [time]. The cluster may not be available during maintenance.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2004</td>
<td>INFO</td>
<td>Maintenance completed on your Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2006</td>
<td>INFO</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2007</td>
<td>INFO</td>
<td>The resize for your Amazon Redshift cluster [cluster name] is in progress. Your cluster is in read-only mode.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2008</td>
<td>INFO</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2013</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster [cluster name] was renamed at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2014</td>
<td>INFO</td>
<td>A table restore request for Amazon Redshift cluster [cluster name] has been received.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2015</td>
<td>INFO</td>
<td>Table restore was cancelled for Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2016</td>
<td>INFO</td>
<td>Replacement of your Amazon Redshift cluster [cluster name] was started at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2017</td>
<td>INFO</td>
<td>Customer initiated maintenance started on your Amazon Redshift cluster [cluster name] at [time]. The cluster may not be available during maintenance.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2018</td>
<td>INFO</td>
<td>Customer initiated maintenance completed on your Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2019</td>
<td>ERROR</td>
<td>Customer initiated maintenance failed on your Amazon Redshift cluster [cluster name] at [time]. Returning the cluster back to its original state.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2020</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster [cluster name]’s track has been modified from [from track] to [to track].</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-2021</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-2022</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-2023</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-2024</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-3011</td>
<td>INFO</td>
<td>Elastic resize for Amazon Redshift cluster [cluster name] started at [time]. We will hold the database connections during resize. Some queries and connections may be terminated or timed out during this operation.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT(EVENT)-3012</td>
<td>INFO</td>
<td>We have received an elastic resize request for the cluster [cluster name] started at [time]. We will provide an event notification when resize begins.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pending</td>
<td>REDSHIFT-EVENT-2025</td>
<td>INFO</td>
<td>Your database for cluster &lt;cluster name&gt; will be updated between &lt;start time&gt; and &lt;end time&gt;. Your cluster will not be accessible. Plan accordingly.</td>
</tr>
<tr>
<td>Pending</td>
<td>REDSHIFT-EVENT-2026</td>
<td>INFO</td>
<td>Your cluster &lt;cluster name&gt; will be updated between &lt;start time&gt; and &lt;end time&gt;. Your cluster will not be accessible. Plan accordingly.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-2050</td>
<td>INFO</td>
<td>A hardware issue was detected on Amazon Redshift cluster [cluster name]. A replacement request was initiated at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3000</td>
<td>INFO</td>
<td>Your Amazon Redshift cluster [cluster name] was rebooted at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3001</td>
<td>INFO</td>
<td>A node on your Amazon Redshift cluster: [cluster name] was automatically replaced at [time], and your cluster is operating normally.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3002</td>
<td>INFO</td>
<td>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.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3003</td>
<td>INFO</td>
<td>Amazon Redshift cluster [cluster name] was successfully created from snapshot [snapshot name] and is available for use.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3007</td>
<td>INFO</td>
<td>Your Amazon Redshift snapshot [snapshot name] was copied successfully from [source AWS Region] to [destination AWS Region] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3008</td>
<td>INFO</td>
<td>Table restore started for Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3009</td>
<td>INFO</td>
<td>Table restore completed successfully for Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3010</td>
<td>ERROR</td>
<td>Table restore failed for Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3013</td>
<td>ERROR</td>
<td>The requested elastic resize operation for Amazon Redshift cluster [cluster name] failed at [time] due to [reason].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3014</td>
<td>INFO</td>
<td>Amazon Redshift rebooted cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3500</td>
<td>ERROR</td>
<td>The resize for your Amazon Redshift cluster [cluster name] failed. The resize will be automatically retried in a few minutes.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3501</td>
<td>ERROR</td>
<td>Your restore operation to create Amazon Redshift cluster [cluster name] from snapshot [snapshot name] failed at [time]. Please retry your operation.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3504</td>
<td>ERROR</td>
<td>The Amazon S3 bucket [bucket name] is not valid for logging for cluster [cluster name].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3505</td>
<td>ERROR</td>
<td>The Amazon S3 bucket [bucket name] does not have the correct IAM policies for cluster [cluster name].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3506</td>
<td>ERROR</td>
<td>The Amazon S3 bucket [bucket name] does not exist. Logging cannot continue for cluster [cluster name].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3507</td>
<td>ERROR</td>
<td>The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. This EIP is already in use.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3508</td>
<td>ERROR</td>
<td>The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. The EIP cannot be found.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3509</td>
<td>ERROR</td>
<td>Cross-region snapshot copy is not enabled for cluster [cluster name].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3510</td>
<td>ERROR</td>
<td>Table restore failed to start for Amazon Redshift cluster [cluster name] at [time]. Reason: [reason].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3511</td>
<td>ERROR</td>
<td>Table restore failed for Amazon Redshift cluster [cluster name] at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3512</td>
<td>ERROR</td>
<td>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].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3513</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3514</td>
<td>ERROR</td>
<td>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].</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3515</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3516</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3519</td>
<td>INFO</td>
<td>Cluster [cluster name] began restart at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3520</td>
<td>INFO</td>
<td>Cluster [cluster name] completed restart at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3521</td>
<td>INFO</td>
<td>We detected a connectivity issue on the cluster '[cluster name]'. An automated diagnostics check has been initiated at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3522</td>
<td>INFO</td>
<td>Recovery action on '[cluster name]' cluster failed at [time]. The Amazon Redshift team is working on a solution.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3533</td>
<td>ERROR</td>
<td>Cluster resize on '[cluster name]' was cancelled at [time]. The operation was cancelled because [reason]. [action needed].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3534</td>
<td>INFO</td>
<td>The elastic resize for Amazon Redshift cluster '[cluster name]' completed at [time]. The cluster is now available for read and write operations while we transfer data. Some queries may take longer to finish until data transfer is complete.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3537</td>
<td>INFO</td>
<td>Cluster '[cluster name]' data transfer completed at [time in UTC].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3600</td>
<td>INFO</td>
<td>The requested resize operation for Amazon Redshift cluster '[cluster name]' was cancelled in the past. Rollback was completed at [time].</td>
</tr>
<tr>
<td>Pending</td>
<td>REDSHIFT-EVENT-3601</td>
<td>INFO</td>
<td>A node on your cluster &lt;cluster name&gt; will be replaced between &lt;start time&gt; and &lt;end time&gt;. You can't defer this maintenance. Plan accordingly.</td>
</tr>
<tr>
<td>Pending</td>
<td>REDSHIFT-EVENT-3602</td>
<td>INFO</td>
<td>A node on your cluster &lt;cluster name&gt; is scheduled to be replaced between &lt;start time&gt; and &lt;end time&gt;. Your cluster will not be accessible. Plan accordingly.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3603</td>
<td>INFO</td>
<td>The restore operation to create cluster [cluster name] from snapshot [snapshot name] failed due to an internal error. The cluster is being placed in incompatible restore status and can be deleted. Try to restore the snapshot into a cluster with a different configuration.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3614</td>
<td>INFO</td>
<td>The scheduled action [scheduled action name] was created at [time in UTC]. The first invocation is scheduled at [time in UTC].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3615</td>
<td>INFO</td>
<td>The scheduled action [scheduled action name] is scheduled at [time in UTC].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3616</td>
<td>INFO</td>
<td>The scheduled action [scheduled action name] at [time in UTC] finished with ‘SUCCEEDED' status.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3617</td>
<td>ERROR</td>
<td>The scheduled action [scheduled action name] was skipped at [time in UTC] due to delay.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3618</td>
<td>INFO</td>
<td>The cluster [cluster name] pause operation started at [UTC time]. Pause Started</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3619</td>
<td>INFO</td>
<td>The cluster [cluster name] pause operation failed at [UTC time]. Retry it from the Amazon Redshift console. Pause Failed</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3626</td>
<td>INFO</td>
<td>The scheduled action [scheduled action name] was modified at [time in UTC]. The first invocation is scheduled at [time in UTC].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3627</td>
<td>INFO</td>
<td>The scheduled action [scheduled action name] was deleted at [time in UTC].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3628</td>
<td>ERROR</td>
<td>The scheduled action [scheduled action name] at [time in UTC] finished with ‘FAILED' status.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3629</td>
<td>INFO</td>
<td>Amazon Redshift [cluster name] has received your relocation request. When Availability Zone relocation completes, Amazon Redshift sends an event notification.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3630</td>
<td>INFO</td>
<td>Amazon Redshift cluster [cluster name] was successfully relocated from [availability zone] to [availability zone]. You can use the cluster now.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3631</td>
<td>INFO</td>
<td>Amazon Redshift successfully relocated your Amazon Redshift cluster [cluster name] from [availability zone] to [availability zone] for recovery.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3632</td>
<td>INFO</td>
<td>Amazon Redshift has temporarily disabled cluster relocation for your Amazon Redshift cluster [cluster name] due to configuration changes. Try cluster relocation again later.</td>
</tr>
<tr>
<td>Amazon Redshift category</td>
<td>Event ID</td>
<td>Event severity</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-3652</td>
<td>INFO</td>
<td>The AQUA configuration status changed to [aqua configuration status] on Redshift cluster [cluster name]. This change will take effect after the next cluster reboot.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4000</td>
<td>INFO</td>
<td>Your master credentials for your Amazon Redshift cluster: [cluster name] were updated at [time].</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4001</td>
<td>INFO</td>
<td>The security group [security group name] was modified at [time]. The changes will take place for all associated clusters automatically.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4500</td>
<td>ERROR</td>
<td>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.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4501</td>
<td>ERROR</td>
<td>The security group [security group name] specified in Cluster Security Group [cluster security group name] could not be found. The authorization cannot be completed.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4502</td>
<td>ERROR</td>
<td>The master credentials for Amazon Redshift cluster [cluster name] failed to update at [time] due to concurrent activity. Allow the current workload to complete or reduce the active workload and then retry the operation.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Amazon Redshift category</th>
<th>Event ID</th>
<th>Event severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1002</td>
<td>INFO</td>
<td>The parameter [parameter name] was updated from [value] to [value] at [time].</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1003</td>
<td>INFO</td>
<td>Cluster parameter group [group name] was created.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1004</td>
<td>INFO</td>
<td>Cluster parameter group [group name] was deleted.</td>
</tr>
<tr>
<td>Configuration</td>
<td>REDSHIFT-EVENT-1005</td>
<td>INFO</td>
<td>Cluster parameter group [name] was updated at [time]. If you changed only dynamic parameters, associated clusters are being modified now. If you changed static parameters, all updates, including dynamic parameters,</td>
</tr>
</tbody>
</table>
## Amazon Redshift event categories and event messages

The following tables show 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

<table>
<thead>
<tr>
<th>Amazon Redshift category</th>
<th>Event ID</th>
<th>Event severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4002</td>
<td>INFO</td>
<td>Cluster security group [group name] was created.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4003</td>
<td>INFO</td>
<td>Cluster security group [group name] was deleted.</td>
</tr>
<tr>
<td>Security</td>
<td>REDSHIFT-EVENT-4004</td>
<td>INFO</td>
<td>Cluster security group [group name] was changed at [time]. Changes will be automatically applied to all associated clusters.</td>
</tr>
</tbody>
</table>

The following tables show the event category and a list of events when a snapshot is the source type.

### Categories and events for the snapshot source type

<table>
<thead>
<tr>
<th>Amazon Redshift category</th>
<th>Event ID</th>
<th>Event severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2009</td>
<td>INFO</td>
<td>A user snapshot [snapshot name] for Amazon Redshift Cluster [cluster name] started at [time]. To monitor snapshot progress, please visit the AWS Management Console.</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2010</td>
<td>INFO</td>
<td>The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] was cancelled at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2011</td>
<td>INFO</td>
<td>The user snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was deleted at [time].</td>
</tr>
<tr>
<td>Management</td>
<td>REDSHIFT-EVENT-2012</td>
<td>INFO</td>
<td>The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was started at [time].</td>
</tr>
<tr>
<td>Monitoring</td>
<td>REDSHIFT-EVENT-3004</td>
<td>INFO</td>
<td>The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] completed successfully at [time].</td>
</tr>
</tbody>
</table>
Managing event notifications using the Amazon Redshift console

Topics
- Creating an event notification subscription (p. 462)

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. 450).

This section describes how to manage Amazon Redshift event notification subscriptions from the AWS Management Console.

Creating an event notification subscription

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

New console

To create an event subscription

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose EVENTS.
3. Choose the Event subscription tab, then choose Create event subscriptions.
4. Enter the properties of your event subscription, such as name, source type, category, and severity. You can also enable Amazon SNS topics to get notified of events.
5. Choose Create event subscriptions to create your subscription.

Original console

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:
      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/v3/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.
5. To create the subscription, choose Create. To delete the definition without creating a subscription, choose Cancel. To return to the subscription settings, choose Previous.

(Original console) Managing event subscriptions

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

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. 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

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 modify, and choose Modify.
4. In the Modify Event Subscription dialog box, do the following:
   a. 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:
      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.
      iii. For Source Type, choose a source type.
iv. 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. 465).

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. 465).

**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/](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 AWS CLI and Amazon Redshift 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. 450).
# Quotas and limits in Amazon Redshift

## Amazon Redshift quotas

Amazon Redshift has quotas that limit the use of several resources in your AWS account per AWS Region. There is a default value for each quota and some quotas are adjustable. For adjustable quotas, you can request an increase for your AWS account in an AWS Region by submitting an Amazon Redshift Limit Increase Form.

<table>
<thead>
<tr>
<th>Quota name</th>
<th>AWS default value</th>
<th>Adjustable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS accounts you can authorize to restore a snapshot per snapshot</td>
<td>20</td>
<td>No</td>
<td>The maximum number of AWS accounts that you can authorize to restore a snapshot, per snapshot.</td>
</tr>
<tr>
<td>AWS accounts you can authorize to restore a snapshot per AWS KMS key</td>
<td>100</td>
<td>No</td>
<td>The maximum number of AWS accounts that you can authorize to restore a snapshot, per 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.</td>
</tr>
<tr>
<td>Cluster IAM roles for Amazon Redshift to access other AWS services</td>
<td>10</td>
<td>No</td>
<td>The maximum number of IAM roles that you can associate with a cluster to authorize Amazon Redshift to access other AWS services for the user that owns the cluster and IAM roles.</td>
</tr>
<tr>
<td>Concurrency level (query slots) for all user-defined manual WLM queues</td>
<td>50</td>
<td>No</td>
<td>The maximum query slots for all user-defined queues defined by manual workload management.</td>
</tr>
<tr>
<td>Concurrency scaling clusters</td>
<td>10</td>
<td>Yes</td>
<td>The maximum number of concurrency scaling clusters.</td>
</tr>
<tr>
<td>Quota name</td>
<td>AWS default value</td>
<td>Adjustable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DC2 nodes in a cluster</td>
<td>128</td>
<td>Yes</td>
<td>The maximum number of DC2 nodes that you can allocate to a cluster. For more information about node limits for each node type, see Clusters and nodes in Amazon Redshift (p. 5).</td>
</tr>
<tr>
<td>DS2 nodes in a cluster</td>
<td>128</td>
<td>Yes</td>
<td>The maximum number of DS2 nodes that you can allocate to a cluster. For more information about node limits for each node type, see Clusters and nodes in Amazon Redshift (p. 5).</td>
</tr>
<tr>
<td>Event subscriptions</td>
<td>20</td>
<td>Yes</td>
<td>The maximum number of event subscriptions for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Nodes</td>
<td>200</td>
<td>Yes</td>
<td>The maximum number of nodes across all database instances for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Parameter groups</td>
<td>20</td>
<td>No</td>
<td>The maximum number of parameter groups for this account in the current AWS Region.</td>
</tr>
<tr>
<td>RA3 nodes in a cluster</td>
<td>128</td>
<td>Yes</td>
<td>The maximum number of RA3 nodes that you can allocate to a cluster. For more information about node limits for each node type, see Clusters and nodes in Amazon Redshift (p. 5).</td>
</tr>
<tr>
<td>Redshift-managed VPC endpoints</td>
<td>30</td>
<td>Yes</td>
<td>The maximum number of Redshift-managed VPC endpoints that you can connect to a cluster. For more information about Redshift-managed VPC endpoints, see Working with Redshift-managed VPC endpoints in Amazon Redshift (p. 38).</td>
</tr>
<tr>
<td>Grantees to cluster accessed</td>
<td>5</td>
<td>Yes</td>
<td>The maximum number of grantees that a cluster owner can authorize to create a Redshift-managed VPC endpoint for a cluster. For more information about Redshift-managed VPC endpoints, see Working with Redshift-managed VPC endpoints in Amazon Redshift (p. 38).</td>
</tr>
<tr>
<td>Redshift-managed VPC endpoints per</td>
<td>5</td>
<td>Yes</td>
<td>The maximum number of Redshift-managed VPC endpoints that you can create per authorization. For more information about Redshift-managed VPC endpoints, see Working with Redshift-managed VPC endpoints in Amazon Redshift (p. 38).</td>
</tr>
<tr>
<td>Reserved nodes</td>
<td>200</td>
<td>Yes</td>
<td>The maximum number of reserved nodes for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Schemas in each database per</td>
<td>9,900</td>
<td>No</td>
<td>The maximum number of schemas that you can create in each database, per cluster. However, pg_temp_* schemas do not count towards this quota.</td>
</tr>
<tr>
<td>Quota name</td>
<td>AWS default value</td>
<td>Adjustable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Security groups</td>
<td>20</td>
<td>Yes</td>
<td>The maximum number of security groups for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Single row size when loading by COPY</td>
<td>4</td>
<td>No</td>
<td>The maximum size (in MB) of a single row when loading by using the COPY command.</td>
</tr>
<tr>
<td>Snapshots</td>
<td>20</td>
<td>Yes</td>
<td>The maximum number of user snapshots for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Subnet groups</td>
<td>20</td>
<td>Yes</td>
<td>The maximum number of subnet groups for this account in the current AWS Region.</td>
</tr>
<tr>
<td>Subnets in a subnet group</td>
<td>20</td>
<td>Yes</td>
<td>The maximum number of subnets for a subnet group.</td>
</tr>
</tbody>
</table>
| Tables for large cluster node type             | 9,900             | No         | The maximum number of tables for the large cluster node type. This limit includes permanent tables, temporary tables, datashare tables, and materialized views. External tables are counted as temporary tables. Temporary tables include user-defined temporary tables and temporary tables created by Amazon Redshift during query processing or system maintenance. Views and system tables aren't included in this limit.
| Tables for xlarge cluster node type            | 9,900             | No         | The maximum number of tables for the xlarge cluster node type. This limit includes permanent tables, temporary tables, datashare tables, and materialized views. External tables are counted as temporary tables. Temporary tables include user-defined temporary tables and temporary tables created by Amazon Redshift during query processing or system maintenance. Views and system tables aren't included in this limit.
| Tables for xlplus cluster node type            | 20,000            | No         | The maximum number of tables for the xlplus cluster node type. This limit includes permanent tables, temporary tables, datashare tables, and materialized views. External tables are counted as temporary tables. Temporary tables include user-defined temporary tables and temporary tables created by Amazon Redshift during query processing or system maintenance. Views and system tables aren't included in this limit.
| Tables for 4xlarge cluster node type           | 100,000           | No         | The maximum number of tables for the 4xlarge cluster node type. This limit includes permanent tables, temporary tables, datashare tables, and materialized views. External tables are counted as temporary tables. Temporary tables include user-defined temporary tables and temporary tables created by Amazon Redshift during query processing or system maintenance. Views and system tables aren't included in this limit.
Amazon Redshift Spectrum quotas and limits

Amazon Redshift Spectrum has the following quotas and limits:

- The maximum number of databases per AWS account when using an AWS Glue Data Catalog. For this value, see AWS Glue service quotas in the Amazon Web Services General Reference.
- The maximum number of tables per database when using an AWS Glue Data Catalog. For this value, see AWS Glue service quotas in the Amazon Web Services General Reference.
- The maximum number of partitions per table when using an AWS Glue Data Catalog. For this value, see AWS Glue service quotas in the Amazon Web Services General Reference.
- The maximum number of partitions per AWS account when using an AWS Glue Data Catalog. For this value, see AWS Glue service quotas in the Amazon Web Services General Reference.
- The maximum number of columns for external tables when using an AWS Glue Data Catalog, 1,598 when pseudocolumns are enabled, and 1,600 when pseudocolumns aren't enabled.
- The maximum size of a string value in an ION or JSON file when using an AWS Glue Data Catalog is 16 KB.
- You can add a maximum of 100 partitions using a single ALTER TABLE statement.
- All S3 data must be located in the same AWS Region as the Amazon Redshift cluster.
• Timestamps in ION and JSON must use ISO8601 format.
• 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 filters on the column ship_dtm, but doesn't apply the filter to
  the partition column ship_yyyymm:

  WHERE ship_dtm > '2018-04-01'.

  To skip unneeded partitions you need to add a predicate WHERE ship_yyyymm = '201804'. This
  predicate limits read operations to the partition \ship_yyyymm=201804\.

These limits don't apply to an Apache Hive metastore.

### Naming constraints

The following table describes naming constraints within Amazon Redshift.

<table>
<thead>
<tr>
<th>Cluster identifier</th>
<th>• A cluster identifier must contain only lowercase characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• It must contain 1–63 alphanumeric characters or hyphens.</td>
</tr>
<tr>
<td></td>
<td>• Its first character must be a letter.</td>
</tr>
<tr>
<td></td>
<td>• It cannot end with a hyphen or contain two consecutive</td>
</tr>
<tr>
<td></td>
<td>hyphens.</td>
</tr>
<tr>
<td></td>
<td>• It must be unique for all clusters within an AWS account.</td>
</tr>
<tr>
<td>Database name</td>
<td>• A database name must contain 1–64 alphanumeric</td>
</tr>
<tr>
<td></td>
<td>characters.</td>
</tr>
<tr>
<td></td>
<td>• It must contain only lowercase letters.</td>
</tr>
<tr>
<td></td>
<td>• It cannot be a reserved word. For a list of reserved words,</td>
</tr>
<tr>
<td></td>
<td>see Reserved words in the Amazon Redshift Database</td>
</tr>
<tr>
<td></td>
<td>Developer Guide.</td>
</tr>
<tr>
<td>Endpoint name of a</td>
<td>• An endpoint name must contain 1–30 characters.</td>
</tr>
<tr>
<td>Redshift-managed</td>
<td>• Valid characters are A-Z, a-z, 0-9, and hyphen(-).</td>
</tr>
<tr>
<td>VPC endpoint</td>
<td>• The first character must be a letter.</td>
</tr>
<tr>
<td></td>
<td>• The name can't contain two consecutive hyphens or end</td>
</tr>
<tr>
<td></td>
<td>with a hyphen.</td>
</tr>
<tr>
<td>Master user name</td>
<td>• A master user name must contain only lowercase</td>
</tr>
<tr>
<td></td>
<td>characters.</td>
</tr>
<tr>
<td></td>
<td>• It must contain 1–128 alphanumeric characters.</td>
</tr>
<tr>
<td></td>
<td>• Its first character must be a letter.</td>
</tr>
<tr>
<td></td>
<td>• It cannot be a reserved word. For a list of reserved words,</td>
</tr>
<tr>
<td></td>
<td>see Reserved words in the Amazon Redshift Database</td>
</tr>
<tr>
<td></td>
<td>Developer Guide.</td>
</tr>
<tr>
<td>Master password</td>
<td>• A master password must contain 8–64 characters.</td>
</tr>
<tr>
<td></td>
<td>• It must contain at least one uppercase letter.</td>
</tr>
<tr>
<td></td>
<td>• It must contain at least one lowercase letter.</td>
</tr>
<tr>
<td></td>
<td>• It must contain one number.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Naming constraints</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• It can use any ASCII characters with ASCII codes 33–126,</td>
</tr>
<tr>
<td></td>
<td>except ‘ (single quote), “ (double quote), , /, or @.</td>
</tr>
<tr>
<td>Parameter group name</td>
<td>• A parameter group name must contain 1–255 alphanumeric</td>
</tr>
<tr>
<td></td>
<td>characters or hyphens.</td>
</tr>
<tr>
<td></td>
<td>• It must contain only lowercase characters.</td>
</tr>
<tr>
<td></td>
<td>• Its first character must be a letter.</td>
</tr>
<tr>
<td></td>
<td>• It can't end with a hyphen or contain two consecutive</td>
</tr>
<tr>
<td></td>
<td>hyphens.</td>
</tr>
<tr>
<td>Cluster security group</td>
<td>• A cluster security group name must contain no more than</td>
</tr>
<tr>
<td>name</td>
<td>255 alphanumeric characters or hyphens.</td>
</tr>
<tr>
<td></td>
<td>• It must contain only lowercase characters.</td>
</tr>
<tr>
<td></td>
<td>• It must not be <strong>Default</strong>.</td>
</tr>
<tr>
<td></td>
<td>• It must be unique for all security groups that are created</td>
</tr>
<tr>
<td></td>
<td>by your AWS account.</td>
</tr>
<tr>
<td>Subnet group name</td>
<td>• A subnet group name must contain no more than 255</td>
</tr>
<tr>
<td></td>
<td>alphanumeric characters or hyphens.</td>
</tr>
<tr>
<td></td>
<td>• It must contain only lowercase characters.</td>
</tr>
<tr>
<td></td>
<td>• It must not be <strong>Default</strong>.</td>
</tr>
<tr>
<td></td>
<td>• It must be unique for all security groups that are created</td>
</tr>
<tr>
<td></td>
<td>by your AWS account.</td>
</tr>
<tr>
<td>Cluster snapshot identifier</td>
<td>• A cluster snapshot identifier must contain no more than</td>
</tr>
<tr>
<td></td>
<td>255 alphanumeric characters or hyphens.</td>
</tr>
<tr>
<td></td>
<td>• It must contain only lowercase characters.</td>
</tr>
<tr>
<td></td>
<td>• It must not be <strong>Default</strong>.</td>
</tr>
<tr>
<td></td>
<td>• It must be unique for all security groups that are created</td>
</tr>
<tr>
<td></td>
<td>by your AWS account.</td>
</tr>
</tbody>
</table>
Tagging resources in Amazon Redshift

Topics

- Tagging overview (p. 474)
- Managing resource tags using the console (p. 475)
- Managing tags using the Amazon Redshift API (p. 478)

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
- Amazon EC2 security group
- Hardware security module (HSM) connection
- HSM client certificate
- Parameter group
- Snapshot
- Subnet group
To use tagging from the Amazon Redshift console, your IAM user can attach the AWS-managed policy AmazonRedshiftFullAccess. For an example IAM policy with limited tagging permissions that you can attach to an Amazon Redshift console user, see Example 7: Allow a user to tag resources with the Amazon Redshift console (p. 304). For more information about tagging, see What is AWS Resource Groups?.

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

**Note**

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

**New console**

To manage tags on your Amazon Redshift resources

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **CONFIG**, then choose **Manage tags**.
3. Enter your choices for the resources and choose which tags to add, modify, or delete. Then choose **Manage tags of the resources that you chose**.

   Resources that you can tag include clusters, parameter groups, subnet groups, HSM client certificates, HSM connections, and snapshots.

4. On the **Manage tags** navigation page, choose **Review and apply tag changes**, then choose **Apply** to save your changes.

**Original 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. 475).

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 choosing 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.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster</td>
<td>In the navigation pane, choose Clusters and choose a cluster from the list. Then choose Manage Tags.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>In the navigation pane, choose Snapshots and choose a snapshot from the list. Then choose Manage Tags for Actions.</td>
</tr>
<tr>
<td>Cluster security group</td>
<td>In the navigation pane, choose Security. On the Security Groups tab, choose a security group from the list. Then choose Manage Tags.</td>
</tr>
<tr>
<td>Cluster security group ingress rule</td>
<td>In the navigation pane, choose Security. On the Security Groups tab, choose a security group in the list. On the Security Group Connections page, choose an ingress rule and then choose Manage Tags.</td>
</tr>
<tr>
<td>Subnet group</td>
<td>In the navigation pane, choose Security. On the Subnet Groups tab, choose a subnet group from the list. Then choose Manage Tags.</td>
</tr>
<tr>
<td>HSM connection</td>
<td>In the navigation pane, choose Security. On the HSM Connections tab, choose a connection from the list. Then choose Manage Tags.</td>
</tr>
</tbody>
</table>
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 API operations to manage tags:

- CreateTags
- DeleteTags
- DescribeTags
- Tag
- TaggedResource

---

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSM certificate</td>
<td>In the navigation pane, choose Security. On the HSM Certificates tab, choose a certificate from the list. Then choose Manage Tags.</td>
</tr>
<tr>
<td>Parameter group</td>
<td>In the navigation pane, choose Parameter Groups and choose a parameter group from the list. Then choose Manage Tags.</td>
</tr>
</tbody>
</table>
Also, you can use the following Amazon Redshift API operations to manage and view tags for a specific resource:

- CreateCluster
- CreateClusterParameterGroup
- CreateClusterSecurityGroup
- CreateClusterSnapshot
- CreateClusterSubnetGroup
- CreateHsmClientCertificate
- CreateHsmConfiguration
- DescribeClusters
- DescribeClusterParameterGroups
- DescribeClusterSecurityGroups
- DescribeClusterSnapshots
- DescribeClusterSubnetGroups
- DescribeHsmClientCertificates
- DescribeHsmConfigurations
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: December 11, 2020**

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*.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for the console integration with partners</td>
<td>You can integrate with partners using the Amazon Redshift console.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Support for the ability to move clusters between Availability Zones</td>
<td>You can now move RA3 clusters between Availability Zones.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Support for ra3.xlplus node types</td>
<td>You can now create ra3.xlplus node types.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Support for JDBC driver version 2.0</td>
<td>You can now configure the JDBC driver version 2.0.</td>
<td>November 5, 2020</td>
</tr>
<tr>
<td>Support for Lambda UDFs and tokenization</td>
<td>You can now can write Lambda UDFs to enable external tokenization of data.</td>
<td>October 26, 2020</td>
</tr>
<tr>
<td>Support to schedule the run of an SQL statement</td>
<td>You can now schedule a query on the Amazon Redshift console.</td>
<td>October 22, 2020</td>
</tr>
<tr>
<td>Support for the Data API for Amazon Redshift</td>
<td>Amazon Redshift can now be accessed using the built-in Data API. Documentation updates include an <em>Amazon Redshift Data API Reference</em>.</td>
<td>September 10, 2020</td>
</tr>
<tr>
<td>Support for Amazon Redshift console query monitoring</td>
<td>Updated the guide to describe new query monitoring graphs.</td>
<td>May 7, 2020</td>
</tr>
<tr>
<td>Support for usage limits</td>
<td>Updated the guide to describe usage limits.</td>
<td>April 23, 2020</td>
</tr>
<tr>
<td>Multi-factor authentication</td>
<td>Updated the guide to describe multi-factor authentication support.</td>
<td>April 20, 2020</td>
</tr>
<tr>
<td>Elastic resize now supports node type changes</td>
<td>Updated elastic resize description.</td>
<td>April 6, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Support for ra3.4xlarge node types with managed storage</td>
<td>Updated the guide to include ra3.4xlarge node types.</td>
<td>April 2, 2020</td>
</tr>
<tr>
<td>Support for pause and resume</td>
<td>Updated the guide to describe the pause and resume cluster operations.</td>
<td>March 11, 2020</td>
</tr>
<tr>
<td>Support for Microsoft Azure AD as an identity provider</td>
<td>Updated the guide to describe the steps to use Microsoft Azure AD as an identity provider.</td>
<td>February 10, 2020</td>
</tr>
<tr>
<td>Support for the RA3 node type</td>
<td>Updated the guide to describe the new RA3 node type.</td>
<td>December 3, 2019</td>
</tr>
<tr>
<td>Support for the new console</td>
<td>Updated the guide to describe the new Amazon Redshift console.</td>
<td>November 11, 2019</td>
</tr>
<tr>
<td>Security information updates</td>
<td>Updates to the security information documentation.</td>
<td>June 24, 2019</td>
</tr>
<tr>
<td>Snapshot enhancements</td>
<td>Amazon Redshift now supports several enhancements to managing and scheduling snapshots.</td>
<td>April 4, 2019</td>
</tr>
<tr>
<td>Concurrency scaling</td>
<td>You can configure workload management (WLM) to enable concurrency scaling mode. For more information, see Configuring workload management.</td>
<td>March 21, 2019</td>
</tr>
<tr>
<td>Updated JDBC and ODBC drivers</td>
<td>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.</td>
<td>February 4, 2019</td>
</tr>
<tr>
<td>Deferred maintenance</td>
<td>If you need to reschedule your cluster's maintenance window, you have the option to defer maintenance by up to 14 days. If we need to update hardware or make other mandatory updates during your period of deferment, we notify you and make the required changes. Your cluster isn't available during these updates. For more information, see Deferring maintenance.</td>
<td>November 20, 2018</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Advance notification</td>
<td>Amazon Redshift provides notification in advance for some events. These events have an event category of <em>pending</em>. For example, we send an advance notification if a hardware update is required for one of the nodes in your cluster. You can subscribe to pending events the same as other Amazon Redshift events. For more information, see <em>Subscribing to Amazon Redshift event notifications</em>.</td>
<td>November 20, 2018</td>
</tr>
<tr>
<td>Elastic resize</td>
<td>Elastic resize is the fastest method to resize a cluster. Elastic resize adds or removes nodes on an existing cluster, then automatically redistributes the data to the new nodes. Because it doesn't create a new cluster, the elastic resize operation completes quickly, usually in a few minutes. For more information, see <em>Resizing clusters</em>.</td>
<td>November 15, 2018</td>
</tr>
<tr>
<td>New ODBC drivers</td>
<td>Amazon Redshift ODBC drivers have been updated to version 1.4.3.1000. For more information, see <em>Configure an ODBC connection</em>.</td>
<td>November 8, 2018</td>
</tr>
<tr>
<td>Cancel resize operation</td>
<td>You can now cancel a resize operation while it is in progress. For more information, see <em>Resize operation overview</em>.</td>
<td>November 2, 2018</td>
</tr>
<tr>
<td>Modify cluster to change encryption</td>
<td>You can modify an unencrypted cluster to use AWS Key Management Service (AWS KMS) encryption, using either an AWS-managed key or a customer-managed key (CMK). When you modify your cluster to enable KMS encryption, Amazon Redshift automatically migrates your data to a new encrypted cluster. You can also migrate an unencrypted cluster to an encrypted cluster by modifying the cluster.</td>
<td>October 16, 2018</td>
</tr>
</tbody>
</table>
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

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 choosing 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 Configuring a JDBC driver version 1.0 connection and Configure an 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.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New CloudWatch metrics</td>
<td>New CloudWatch metrics added for monitoring query performance. For more information, see Monitoring Amazon Redshift using CloudWatch metrics (p. 401)</td>
<td>May 17, 2018</td>
</tr>
<tr>
<td>New JDBC and ODBC drivers</td>
<td>Amazon Redshift JDBC drivers have been updated to version 1.2.12.1017. For more information, see Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>March 7, 2018</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Amazon Redshift ODBC drivers have been updated to version 1.4.1.1001. For more information, see Configuring an ODBC connection (p. 130).</td>
<td></td>
</tr>
<tr>
<td>HSM encryption</td>
<td>Amazon Redshift supports only AWS CloudHSM for hardware security module (HSM) key management. For more information, see Amazon Redshift database encryption (p. 271).</td>
<td>March 6, 2018</td>
</tr>
<tr>
<td>IAM Role Chaining</td>
<td>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. 346).</td>
<td>February 23, 2018</td>
</tr>
<tr>
<td>New DC2 node types</td>
<td>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. 5).</td>
<td>October 17, 2017</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>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 Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>October 16, 2017</td>
</tr>
<tr>
<td>ACM certificates</td>
<td>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. 147).</td>
<td>September 18, 2017</td>
</tr>
<tr>
<td>Service-linked roles</td>
<td>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. 306).</td>
<td>September 18, 2017</td>
</tr>
<tr>
<td>IAM database user authentication</td>
<td>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. 308).</td>
<td>August 11, 2017</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>New JDBC and ODBC drivers</td>
<td>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 Configuring a JDBC driver version 1.0 connection (p. 122). Amazon Redshift ODBC drivers have been updated to version 1.3.6.1000. For more information, see Configuring an ODBC connection (p. 130).</td>
<td>August 11, 2017</td>
</tr>
<tr>
<td>Table-level restore supports enhanced VPC routing</td>
<td>Table-level restore is now supported on clusters that use Enhanced VPC routing (p. 194). For more information, see Restoring a table from a snapshot (p. 234).</td>
<td>July 19, 2017</td>
</tr>
<tr>
<td>Query monitoring rules</td>
<td>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. 203).</td>
<td>April 21, 2017</td>
</tr>
<tr>
<td>New JDBC and ODBC drivers</td>
<td>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 Configuring a JDBC driver version 1.0 connection (p. 122). Amazon Redshift ODBC drivers have been updated to version 1.3.1.1000. For more information, see Configuring an ODBC connection (p. 130).</td>
<td>November 18, 2016</td>
</tr>
<tr>
<td>Enhanced VPC routing</td>
<td>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 Enhanced VPC routing in Amazon Redshift (p. 194).</td>
<td>September 15, 2016</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>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 Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>July 5, 2016</td>
</tr>
<tr>
<td>New connection log fields</td>
<td>The Connection log (p. 354) 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: ssicompression and sslexpansion.</td>
<td>May 5, 2016</td>
</tr>
<tr>
<td>New ODBC drivers</td>
<td>Amazon Redshift ODBC drivers have been updated to version 1.2.7.1007. For more information, see Configuring an ODBC connection (p. 130).</td>
<td>March 30, 2016</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>IAM roles for COPY and UNLOAD</td>
<td>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. 343) and Authorizing COPY, UNLOAD, CREATE EXTERNAL FUNCTION, and CREATE EXTERNAL SCHEMA operations using IAM roles (p. 348).</td>
<td>March 29, 2016</td>
</tr>
<tr>
<td>Restore from table</td>
<td>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. 234).</td>
<td>March 10, 2016</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>Amazon Redshift JDBC drivers have been updated to version 1.1.10.1013. For more information, see Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>February 18, 2016</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>You can now set the SSLMode property to specify whether the driver verifies host names when validating TLS/SSL certificates. For more information, see Configure JDBC driver options (p. 126).</td>
<td>November 19, 2015</td>
</tr>
<tr>
<td>Using IAM Condition in policies</td>
<td>You can further restrict access to resources by using the Condition element in IAM policies. For more information, see Using IAM policy conditions for fine-grained access control (p. 293).</td>
<td>December 10, 2015</td>
</tr>
<tr>
<td>Modify publicly accessible</td>
<td>You can modify an existing cluster in a VPC to change whether it is publicly accessible. For more information, see Modifying a cluster (p. 54).</td>
<td>November 20, 2015</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>Amazon Redshift JDBC drivers have been updated to version 1.1.10.1010. For more information, see Configuring a JDBC driver version 1.0 connection (p. 122). Amazon RedshiftODBC drivers have been updated to version 1.2.6.1006. For more information, see Configuring an ODBC connection (p. 130).</td>
<td>November 19, 2015</td>
</tr>
<tr>
<td>Documentation fixes</td>
<td>Published various documentation fixes.</td>
<td>August 28, 2015</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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. 165).</td>
<td>August 25, 2015</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Revised entire section about parameter groups for better organization and clarity. For more information, see Amazon Redshift parameter groups (p. 201).</td>
<td>August 17, 2015</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>Amazon Redshift JDBC drivers have been updated to version 1.1.7. For more information, see Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>August 14, 2015</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>WLM dynamic properties</td>
<td>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. 204) and Amazon Redshift parameter groups (p. 201).</td>
<td>August 3, 2015</td>
</tr>
<tr>
<td>Copy KMS encrypted clusters to another AWS Region</td>
<td>Added content about configuring snapshot copy grants to enable copying of AWS KMS-encrypted clusters to another AWS Region. For more information, see Copying AWS KMS–encrypted snapshots to another AWS Region (p. 273).</td>
<td>July 28, 2015</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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. 271).</td>
<td>July 28, 2015</td>
</tr>
<tr>
<td>New JDBC drivers</td>
<td>Amazon Redshift JDBC drivers have been updated to version 1.1.7. For more information, see Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>July 2, 2015</td>
</tr>
<tr>
<td>New node type</td>
<td>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. 5).</td>
<td>June 9, 2015</td>
</tr>
<tr>
<td>Reserved node offerings</td>
<td>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. 258).</td>
<td>June 9, 2015</td>
</tr>
<tr>
<td>New ODBC drivers</td>
<td>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 Configuring an ODBC connection (p. 130).</td>
<td>June 5, 2015</td>
</tr>
<tr>
<td>Documentation fixes</td>
<td>Published various documentation fixes.</td>
<td>April 30, 2015</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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 Configuring a JDBC driver version 1.0 connection (p. 122).</td>
<td>April 1, 2015</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| Documentation update   | 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 Configuring a JDBC driver version 1.0 connection (p. 122).  
Added cluster security group ingress rules as a taggable resource. For more information, see Tagging resources in Amazon Redshift (p. 474).  
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. 371). | March 16, 2015        |
<p>| 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. 90). | 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 queries and loads (p. 426). | 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. 300). | January 16, 2015      |
| Documentation update   | Updated guidance about setting the maximum transmission unit (MTU) to disable TCP/IP jumbo frames. For more information, see Use EC2-VPC when you create your cluster (p. 9) and Queries appear to hang and sometimes fail to reach the cluster (p. 165). | January 16, 2015      |
| Documentation update   | Revised the content about the wlm_json_configuration 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. 203). | January 13, 2015      |
| Documentation update   | Added missing event notifications and descriptions. For more information, see Amazon Redshift event categories and event messages (p. 452). | 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 in Amazon Redshift (p. 269). | November 21, 2014     |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New feature</td>
<td>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. 271) and Managing clusters using the console (p. 41).</td>
<td>November 12, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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. 474).</td>
<td>November 4, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Increased the maximum node limit to 128 nodes for dw1.8xlarge and dw2.8xlarge node sizes. For more information, see Clusters and nodes in Amazon Redshift (p. 5).</td>
<td>October 30, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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 (p. 132).</td>
<td>October 30, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Added the ability to terminate queries and loads from the Amazon Redshift console. For more information, see Viewing queries and loads (p. 426) and Viewing cluster metrics during load operations (p. 437).</td>
<td>October 28, 2014</td>
</tr>
<tr>
<td>Documentation fixes</td>
<td>Published various documentation fixes.</td>
<td>October 17, 2014</td>
</tr>
<tr>
<td>New content</td>
<td>Added content about shutting down clusters and deleting clusters. For more information, see Shutting down and deleting clusters (p. 30) and Deleting a cluster (p. 56).</td>
<td>August 14, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Clarified the behavior of the Allow Version Upgrade setting for clusters. For more information, see Overview of Amazon Redshift clusters (p. 5).</td>
<td>August 14, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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. 41).</td>
<td>July 11, 2014</td>
</tr>
<tr>
<td>New content</td>
<td>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. 21).</td>
<td>June 27, 2014</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>New feature</td>
<td>Added the ability to rename clusters. For more information, see Renaming clusters (p. 30) and Modifying a cluster (p. 54).</td>
<td>June 2, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>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. 159).</td>
<td>May 15, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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. 245).</td>
<td>May 12, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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. 19).</td>
<td>April 28, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Clarified information about Elliptic curve Diffie—Hellman Exchange (ECDHE) support in Amazon Redshift. For more information, see Connect using SSL (p. 144).</td>
<td>April 22, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Added statement about Amazon Redshift support for the Elliptic curve Diffie—Hellman (ECDH) key agreement protocol. For more information, see Connect using SSL (p. 144).</td>
<td>April 18, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Revised and reorganized the topics in the Connecting to an Amazon Redshift cluster using SQL client tools (p. 90) section. Added more information about JDBC and ODBC connections, and a new troubleshooting section for connection issues.</td>
<td>April 15, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added version in IAM policy examples throughout the guide.</td>
<td>April 3, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added information about how pricing works when you resize a cluster. For more information, see Purchasing Amazon Redshift reserved nodes (p. 258).</td>
<td>April 2, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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. 201).</td>
<td>March 28, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Added explanation about the <strong>Cluster Version</strong> field now including both cluster engine version and database revision number. For more information, see Amazon Redshift clusters (p. 5).</td>
<td>March 21, 2014</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated the resize procedure to show the new resize progress information on the cluster's <strong>Status</strong> tab. For more information, see <em>Resizing a cluster (p. 59)</em>.</td>
<td>March 21, 2014</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Reorganized and updated <em>What is Amazon Redshift? (p. 1)</em> and revised <em>Amazon Redshift management overview (p. 1)</em>. Published various documentation fixes.</td>
<td>February 21, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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 <em>Amazon Redshift clusters (p. 5)</em>.</td>
<td>January 23, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>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 <em>Managing clusters in a VPC (p. 69)</em> and <em>Creating a cluster in a VPC (p. 71)</em>.</td>
<td>December 20, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Added information about the AWS CloudTrail logs for Amazon Redshift. For more information about Amazon Redshift support for CloudTrail, see <em>Logging Amazon Redshift API calls with AWS CloudTrail (p. 359)</em>.</td>
<td>December 13, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Added information about the new user activity log and the <em>enable_user_activity_logging</em> database parameter for the database audit logging feature in Amazon Redshift. For more information about database audit logging, see <em>Database audit logging (p. 353)</em>. For more information about database parameters, see <em>Amazon Redshift parameter groups (p. 201)</em>.</td>
<td>December 6, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to describe configuring Amazon Redshift to automatically copy automated and manual snapshots to a secondary AWS Region. For more information about configuring cross-Region snapshot copy, see <em>Copying snapshots to another AWS Region (p. 230)</em>.</td>
<td>November 14, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>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 <em>Database audit logging (p. 353)</em>.</td>
<td>November 11, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>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 <em>Amazon Redshift database encryption (p. 271)</em>, <em>Encryption for Amazon Redshift using hardware security modules (p. 274)</em>, and <em>Encryption key rotation in Amazon Redshift (p. 275)</em>.</td>
<td>November 11, 2013</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>New feature</td>
<td>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. 450).</td>
<td>November 11, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to describe IAM resource level permissions. For information about Amazon Redshift IAM permissions, see Security in Amazon Redshift (p. 269).</td>
<td>August 9, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to describe restore progress metrics. For more information, see Restoring a cluster from a snapshot (p. 231).</td>
<td>August 9, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to describe cluster snapshot sharing and create snapshot progress metrics. For more information, see Sharing snapshots (p. 236).</td>
<td>July 17, 2013</td>
</tr>
<tr>
<td>Documentation fixes</td>
<td>Published various documentation fixes.</td>
<td>July 8, 2013</td>
</tr>
<tr>
<td>New console screens</td>
<td>Updated the Amazon Redshift Cluster Management Guide to match changes in the Amazon Redshift console.</td>
<td>April 22, 2013</td>
</tr>
<tr>
<td>New guide</td>
<td>This is the first release of the Amazon Redshift Management Guide.</td>
<td>February 14, 2013</td>
</tr>
</tbody>
</table>