

Amazon Redshift: Cost Optimization

Cost Optimization Guidelines for Amazon Redshift

August 2020



Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Contents

- Introduction 1
- Sizing Considerations..... 1
- Trusted Advisor..... 2
- Cost Explorer 3
- Amazon Redshift Advisor..... 3
 - Compressing Amazon S3 file objects loaded by COPY 5
 - Cluster Resize 6
- Conclusion 8
- Contributors 9
- Document Revisions..... 9

Introduction

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. It is optimized for datasets ranging from a few hundred gigabytes to a petabyte or more and costs less than a tenth the cost of most traditional data warehousing solutions.

There are a number of best practices you can follow to ensure you're getting the best value with Amazon Redshift. This document shares the most common cost optimization methods adopted across our customer base. This document is intended for advanced Amazon Redshift users. For more information, [see Amazon Redshift](#).

Sizing Considerations

Cost optimization starts with choosing the right node type, instance type, and payment structure to meet your cloud data warehouse requirements such as, CPU, RAM, storage capacity and type, and availability. When you select your instance type consider that Amazon Redshift compresses data up to four times. When you start using Amazon Redshift for the first time, you will receive a recommendation for the best node type based on your needs. You can easily scale up or down if your business need changes.

Amazon Redshift 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 pay only for the managed storage that you use. You should size your RA3 cluster based on the amount of data you process daily. There's a recommendation engine built into the console to help you make the right selection. If you're upgrading from a previous version of Amazon Redshift, you can also view the guidelines in the [documentation](#).

Amazon Redshift managed storage uses large, high-performance SSDs in each RA3 node for fast local storage and Amazon Simple Storage Service (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.

Previous generation nodes include DC2 (Compute intensive), DS2 (Storage Intensive). Reserved instances (RI) (also called reserved nodes in the Amazon Redshift console) can provide up to 75% savings vs on-demand pricing. The table below provides discount percentages for all up-front commitments in us-east-1 region. For more information, see <https://aws.amazon.com/redshift/pricing/>.

Instance type	Size	Memory	CPUs	RI Discount	
				1-Year	3-Year
RA3 4xlarge	Scales to 64 TB	96 GB	12	34%	63%
RA3 16xlarge	Scales to 64 TB	384 GB	48	34%	63%
DC2 large	160 GB	16 GB	2	37%	62%
DC2 8xlarge	2.56 TB	244 GB	32	34%	69%
DS2 xlarge	2 TB	32 GB	4	42%	75%
DS2 8xlarge	16 TB	244 GB	36	42%	75%

Trusted Advisor

The Trusted Advisor application (available under management and governance) runs automated checks against your Amazon Redshift resources in your account to notify you about cost optimization opportunities. Checks include following:

- **Redshift Reserved Node Optimization:** Checks usage to provide recommendations about when to purchase reserved nodes to help reduce costs.

- **Recommended Action:** Evaluate and identify clusters that will benefit from purchasing reserved nodes. Moving from on-demand will result in between 60-75% cost savings.
- **Underutilized Redshift Clusters:** Checks for clusters that appear to be underutilized (< 5% average CPU utilization for 99% of last 7 days).
 - **Recommended Action:** Shutting down the cluster and taking a final snapshot or downsizing will save costs.

Cost Explorer

[AWS Cost Explorer](#) helps you visualize, understand, and manage your AWS costs and usage over time. It provides the following features, insights, and alerts to manage your Amazon Redshift cluster by breaking down its usage across linked accounts, regions, usage groups, and tags from the last 12 months.

- **Budgets:** Create budgets based on cost, usage, Reserved node utilization/coverage. Amazon Redshift customers can create budgets based on usage type (paid snapshots, node hours, and data scanned in TB), or usage type groups (Amazon Redshift running hours) and schedule automated alerts.
- **Cost and Usage Reports:** Amazon Redshift cost and usage reports include usage by an account and AWS Identity and Access Management (IAM) users in hourly or daily line items, as well as tags for cost allocation. It can provide complex insights and aggregations. It integrates with Amazon Athena, Amazon Redshift, and Amazon Quicksight. It supports compression types such as Gzip, zip, and Parquet.
- **Reservations:** Provides recommendations on RI purchase for Amazon Redshift cluster based the last 30 to 60 days. These recommendations include potential savings (monthly/yearly) based on payment terms (no upfront/partial upfront/all upfront). RI coverage and utilization reports give insights on the cluster usage to help with decisions to purchase reservations for Amazon Redshift.

Amazon Redshift Advisor

When it comes to assisting with not only the operating costs of your Amazon Redshift cluster, but also improving the performance, Amazon Redshift offers its own *Trusted Advisor*. The [Amazon Redshift Advisor](#) tool is built directly in the Amazon Redshift console. It identifies undesirable end-user behaviors for resolutions by providing

recommendations to improve performance and reduce cost. Amazon Redshift Advisor generates automatic observations and recommendations based on your cluster workload. Two of the most important cost recommendations provided are alerts to idle clusters that could be deleted or down sized, and alerts to excessive uncompressed storage. However there are many other additional [Advisor checks](#) available.

Table Compression

On the topic of cost optimization, one of the first things that should come to mind when thinking of a data warehouse (DW) is compressing the table data. When you don't use compression, data consumes additional space and requires additional disk I/O. That additional space and I/O will directly correlate to increases in your Amazon Redshift bill. Advisor tracks uncompressed storage and reviews storage metadata associated with large uncompressed columns that aren't sort key columns. Advisor offers a recommendation to rebuild tables with uncompressed columns when the total amount of uncompressed storage exceeds 15 percent of total storage space, or at the following node-specific thresholds. When it comes time to select the correct compression, use the `ANALYZE COMPRESSION` command to suggest a compression.

Encoding type	Keyword in CREATE TABLE and ALTER TABLE	Data Types
Raw (no compression)	RAW	ALL
AZ64	AZ64	SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, TIMESTAMPTZ
Byte Dictionary	BYTEDICT	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Delta	DELTA DELTA3K	SMALLINT, INT, BIGINT, DATE, TIMESTAMP, DECIMAL INT, BIGINT, DATE, TIMESTAMP, DECIMAL

Encoding type	Keyword in CREATE TABLE and ALTER TABLE	Data Types
LZO	LZO	SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Mostlyn	MOSTLY8	SMALLINT, INT, BIGINT, DECIMAL
	MOSTLY16	INT, BIGINT, DECIMAL
	MOSTLY32	BIGINT, DECIMAL
Run-Length	RUNLENGTH	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Text	TEXT255	VARCHAR only
	TEXT32K	
Zstandard	ZSTD	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ

Compressing Amazon S3 file objects loaded by COPY

The `COPY` command integrates with the massively parallel processing (MPP) architecture in Amazon Redshift to read and load data in parallel from Amazon S3,

Amazon DynamoDB, and text output. The Advisor analysis identifies `COPY` commands that load large uncompressed datasets. In this case, Advisor generates a recommendation to implement compression on the source files in Amazon S3. This greatly reduces data transfer costs. The ideal object size is 1 to 128 MB after compression. You can also use the `COPY` command with `COMPUPDATE` set to `ON` to analyze and [apply compression automatically](#).

You can use [automatic compression](#) when you create and load a brand new table. The `COPY` command performs a compression analysis. You can also perform a compression analysis without loading data or changing the compression on a table by running the `ANALYZE COMPRESSION` command on an already populated table. You can also compress files with Gzip, lzop, or bzip2 format.

Pause and Resume

You may choose to use the [pause and resume](#) feature your cluster to quickly suspend the on demand billing of your cluster while it is in downtime. While production workloads often run 24/7, development works can be easily shut on and off enabling you to optimize your spend for the hours the dev cluster is not in use. When the cluster is turned off you are no longer charged for the clusters compute. You will still incur associated storage charges. In order to invoke pause and resume you can either use the Amazon Redshift console or the Amazon Redshift API.

Cluster Resize

As your Amazon Redshift capacity or performance changes throughout its lifecycle, you may find yourself needing to [resize](#) in order to make the best use of your cluster. If you find yourself with an oversized cluster, you have three options to resize for the appropriate workload.

Elastic Resize

[Elastic Resize](#) is the optimal and fastest way to quickly add or remove nodes or change node types from an existing cluster. It [automates](#) the steps of taking a snapshot, creating a new cluster, deleting the old cluster and renaming the new cluster into a simple, quick and familiar operation. The elastic resize operation can be run at any time or can be scheduled to run at a future time. Customers can quickly upgrade their existing DS2 or DC2 node type-based cluster to the new RA3 node type with elastic resize.

This leads to serious cost control as you can downsize if you are working with oversized cluster or you can start with a small, cost effective data warehouse and scale up on-demand as your needs grow. When a cluster is resized using elastic resize with the same node type, it automatically redistributes the data to the new nodes. The process only takes ten to fifteen minutes to complete as the resize does not create a new cluster. A snapshot would be created and a new cluster is provisioned for you with the latest data from the snapshot. The cluster is temporarily unavailable for writes (reads will be available) when the data is transferred to the new cluster. Amazon Redshift will send an event notification once the resize completes.

For dc2.large or ds2.xlarge node types, you can double the size or half the size of the number of nodes of the original cluster.

For dc2.8xlarge, ds2.8xlarge, ra3.4xlarge, or ra3.16xlarge node types, you can change the number of nodes to half the current number to double the current number of nodes. A 4-node cluster can be resized to 2, 3, 5, 6, 7, or 8 nodes.

Classic Resize

[Classic resize](#) is the manual approach to resizing your cluster and is the predecessor to Elastic Resize. Elastic resize is the recommended option, this document will not dive into the classic. For more information, see [classic resize](#) in the *Amazon Redshift Cluster Management Guide*.

Concurrency Scaling

While resizing your cluster is fit for known workloads, for spiky workloads you should consider using the [concurrency scaling](#) feature. Concurrency scaling is a cost-effective way to pay only for additional capacity during large workload spikes, as opposed to adding persistent nodes in the cluster that will incur extra costs during downtime. Users can right-size to a more cost-effective cluster and Amazon Redshift will automatically add additional cluster capacity when you need it to process an increase in concurrent read queries (write operations continue as normal). Each cluster earns up to one hour of free concurrency scaling credits per day, which is sufficient capacity for almost of all workload types. For the small chance you go over your allotted free credits, you simply pay a per-second on-demand rate for the usage that exceeds those credits.

To implement the concurrency scaling, the user will route queries to concurrency scaling clusters by enabling a workload manager (WLM) queue as a concurrency scaling queue. To enable concurrency scaling on a queue, you will set the

concurrency_scaling mode value to auto. For more information, see the [pricing page](#) and [usage controls](#) for concurrency scaling.

Amazon Redshift Spectrum

Amazon Redshift and the [Amazon Redshift Spectrum](#) feature powers the [lake house architecture](#) – enabling you to query data across your data warehouse, data lake, and operational databases to gain faster and deeper insights not possible otherwise. With a lake house architecture, you can store data in open file formats in your Amazon S3 data lake.

With an Amazon Redshift lake house architecture, you can:

- Easily query data in your data lake and write data back to your data lake in open formats.
- Use familiar SQL statements to combine and process data across all your data stores.
- Queries span both the frequently accessed hot data stored locally in Amazon Redshift and the warm or cold data stored cost-effectively in Amazon S3, using views [with no schema binding](#) for external tables.

Conclusion

Amazon Redshift has many advantages when it comes to cost management. You can start small at just \$0.25 per hour, and scale to 1TB for just under \$1,000TB per year. You can take advantage of reserved instances, which save up to 75% compared to on-demand prices when you commit to using Amazon Redshift for a 1- or 3-year term. It starts with choosing the appropriate cluster configuration, leveraging tools to keep track of the costs during its operationalization, following advisor recommendations for its management, pause and resume for specified downtimes, and resizing options for changing demands and use cases. Amazon Redshift also provides predictability in month-to-month cost even when you have unpredictable or highly concurrent workloads because each Amazon Redshift cluster earns up to an hour of free concurrency scaling credits per day, which can be used to offset the cost of the transient clusters that are automatically added to handle high concurrency. As the size of data grows, infrequently accessed data can also be stored cost-effectively in Amazon S3 and still be queried with Amazon Redshift.

Contributors

Contributors to this document include:

- Srijit Mohapatra, Technical Account Manager, Analytics Community
- Stephen Perri, Technical Account Manager, Cost Optimization Community

Document Revisions

Date	Description
August 2020	First publication