Amazon Redshift

# Index

Table of Contents

[Index 2](#_Toc28975043)

[Redshift 3](#_Toc28975044)

[Definition 3](#_Toc28975045)

[Performance 3](#_Toc28975046)

[Deployment Mode 4](#_Toc28975047)

[Distribution Style 5](#_Toc28975048)

[Backup & Recovery 6](#_Toc28975049)

[Miscellaneous Features 7](#_Toc28975050)

# Redshift

## Definition

* Fast, fully managed data warehouse service that makes it simple and cost-effective to analyze data using standard SQL and existing Business Intelligence (BI) tools.
* It allows you to run complex analytic queries against petabytes of structured data, using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel query execution.
* Includes Amazon Redshift Spectrum, allowing you to directly run SQL queries against exabytes of unstructured data in Amazon S3.
  + No loading or transformation is required, and one can use open data formats, including Avro, CSV, Grok, Ion, JSON, ORC, Parquet, RCFile, RegexSerDe, SequenceFile, TextFile, and TSV.
  + It automatically scales query compute capacity based on the data being retrieved, so queries run fast regardless of dataset size.

## Performance

* Columnar Data Storage
  + Amazon Redshift organizes the data by column, instead of rows.
  + Unlike row-based systems, which are ideal for transaction processing, column-based systems are ideal for data warehousing and analytics, where queries often involve aggregates performed over large data sets.
  + Since only the columns involved in the queries are processed and columnar data is stored sequentially on the storage media, column-based systems require far fewer I/Os, greatly improving query performance.
* Advanced Compression
  + Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk.
  + Amazon Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores.
* Zone Maps
  + Essentially, a chunk of data with metadata that can be evaluated to determine if the query even needs to interact with the chunk to execute.
  + For example, MIN and MAX values are maintained to optimize queries by pruning blocks that can’t contain required data.
* Zonal Deployment
  + Redshift is not deployed in multiple AZs to reduce latency.
* Massively Parallel Processing (MPP):
  + Amazon Redshift automatically distributes data and query load across all nodes.
  + It makes It makes easy to add nodes to the data warehouse and enables one to maintain fast query performance as the data warehouse grows.
* Redshift Spectrum:
  + Redshift Spectrum enables you to run queries against exabytes of data in Amazon S3 without requiring an ETL pipeline.
  + When a query is issued, a query plan is generated. The data present in local vs the data required from S3 is determined to minimize unnecessary reads.

## Deployment Mode

* Redshift currently supports single-AZ deployment.
* Redshift cluster can be deployed in two modes:
  + Single-Node configuration
    - Start with a single node, 160GB data warehouse and scale all the way to a petabyte or more.
    - Enables one to get started with Amazon Redshift quickly and cost-effectively and scale up to a multi-node configuration as requirements grow.
  + Multi-Node configuration
    - Leader node
      * It is the entry point for communication with a Redshift cluster and has several special characteristics.
      * Receives queries from client applications, parses the queries and develops execution plans, which are an ordered set of steps to process these queries.
        + These instructions are in the form of compiled C++ codes or binaries.
      * The leader node then coordinates the parallel execution of these plans with the compute nodes, aggregates the intermediate results from these nodes and finally returns the results back to the client applications.
      * It stores cluster metadata that allows it to plan the best ways to query data.
      * It stores and maintains the catalog tables (PG tables).
      * One is not billed for leader node.
    - Compute node
      * Executes the steps specified in the execution plans and transmit data among themselves to serve these queries.
      * The intermediate results are sent back to the leader node for aggregation before being sent back to the client applications.
      * Backs up the data and restores it as needed.
      * Replicates data between nodes.
      * A single Redshift cluster can have up to 128 compute nodes.
      * Components of a compute node:
        + Slices

Essentially a ‘virtual compute node’ that is distributed data rows and acts as a unit of data partitioning

Compute nodes have 2, 16 or 32 slices based on size.

Slices process and load their own data. For better performance, split files into multiple of the number of slices to load in parallel.

* + - * + Disks

Locally attached storage devices.

Compute nodes have 2.5-3x times the advertised storage capacity to make room for mirrored data.

There are 1, 3, 8 or 24 disks per node, each split into two partitions (local storage and mirrored data).

* + - * + Blocks

1 MB immutable chunk of data that contains Zone maps (MIN/MAX of that block) and the location of the next block.

Blocks can have a variety of compression encodings to minimize disk space.

* One can have two types of nodes on Redshift:
  + DC
    - Dense storage node types optimized for large data workloads.
    - Uses HDD storage.
  + DS
    - Dense compute node types optimized for performance-intensive workloads
    - Uses SSD storage.
    - Less storage but much faster I/O operations.

## Distribution Style

Every Redshift table has a designated distribution style used by the leader node to determine what slices to send data to. When selecting the distribution style, the goals are to distribute the data evenly and to minimize data movement during query execution.

The various distribution styles are:

* AUTO distribution
  + The default distribution style.
  + Is upgraded to EVEN distribution style when the data becomes large.
* EVEN distribution
  + The leader node distributes the data in a round-robin fashion across all slices.
  + Appropriate when the table does not participate in joins.
  + Appropriate when there is no clear choice between KEY and ALL.
* KEY distribution
  + The rows are distributed according to the value in one column.
  + The leader node will attempt to put matching values in the same slice.
  + If tables are distributed on joining keys the data sharing joining values will be physically collocated on a slice.
* ALL distribution
  + A copy of the entire table is distributed to every node.
  + EVEN and KEY distributions place a portion of a table on every node, but ALL distributions ensure every row is collocated for every join it participates in.
  + ALL distributions multiply the amount of storage required because data is duplicated multiple times over.
  + This distribution takes longer to load, update or insert, hence is appropriate for tables that are not frequently or extensively updated.
  + Small dimension tables do not benefit much from ALL distribution because of redistribution of the data in such tables is relatively low.

Once the data is distributed, to further store the data in an optimized manner, one can use sort keys to sort the data according to certain columns that are frequently used in queries.

There are different types of sort keys in Redshift:

* Simple Sort Key
  + Single value that is used to sort the data.
  + Better when consistently querying the data with a single filter column.
* Compound Sort Key
  + Multiple values are used to sort in order of priority.
  + Good when the columns one filters on are always the same.
  + Gives performance benefits when filtering on any of the sort keys, with the first key giving the best gain.
* Interleaved Sort Key
  + Weights all columns evenly when sorting.
  + Useful when sorting on many columns because it prioritizes them equally.
  + Requires frequent vacuum operations to clean up old data.

## Backup & Recovery

* Redshift enables automated backups of the data warehouse cluster with a 1-day retention period, by default, which can be extended to a maximum of 35 days.
* Redshift always attempts to maintain at least three copies of the data: the original and replica on the compute nodes and a backup in S3.
* Redshift can also asynchronously replicate the snapshots to S3 in another region for disaster recovery.

## Miscellaneous Features

* In Amazon Redshift, one can 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.
  + A custom WLM would require the creation of a new parameter group, which would then be associated to the Redshift clusters.
* With Amazon Redshift Enhanced VPC Routing enabled, Redshift forces all COPY and UNLOAD traffic between the cluster and data repositories through Amazon VPC, allowing one to use standard VPC features such as VPC security groups, NACLs, VPC endpoints, etc.