HEVO

EBOOK

REDSHIFT VS SNOWFLAKE
AN IN-DEPTH COMPARISON

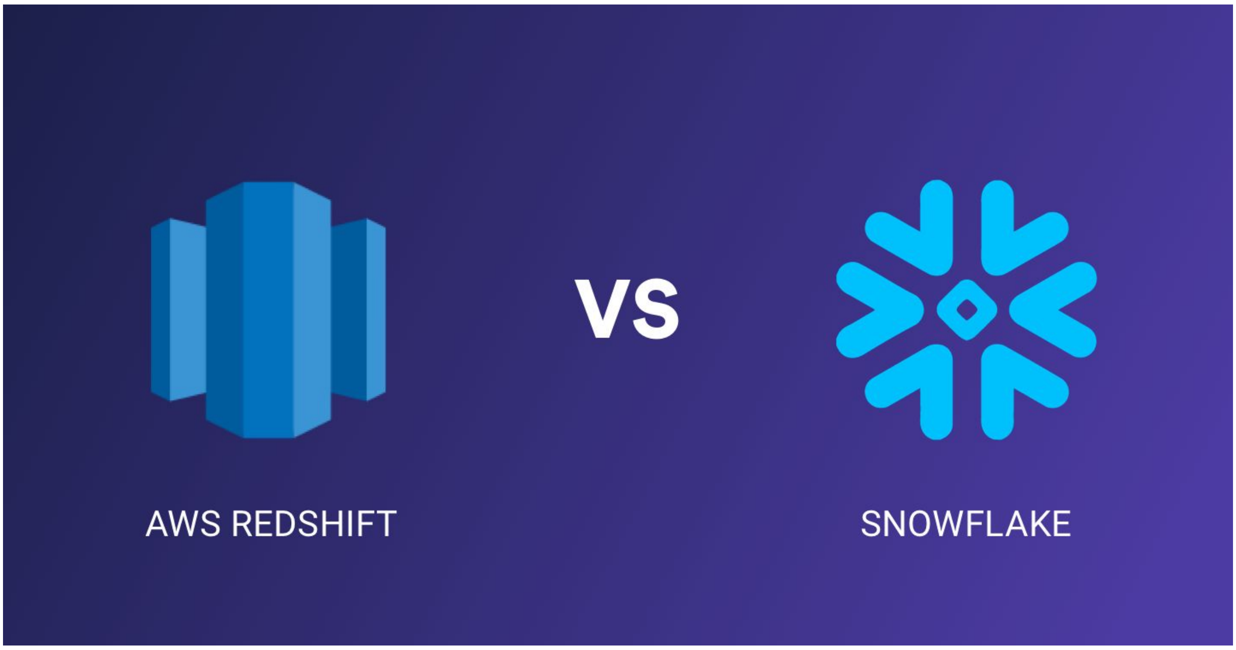
VS

AMAZON REDSHIFT

SNOWFLAKE

Table of Contents

Introduction



Redshift

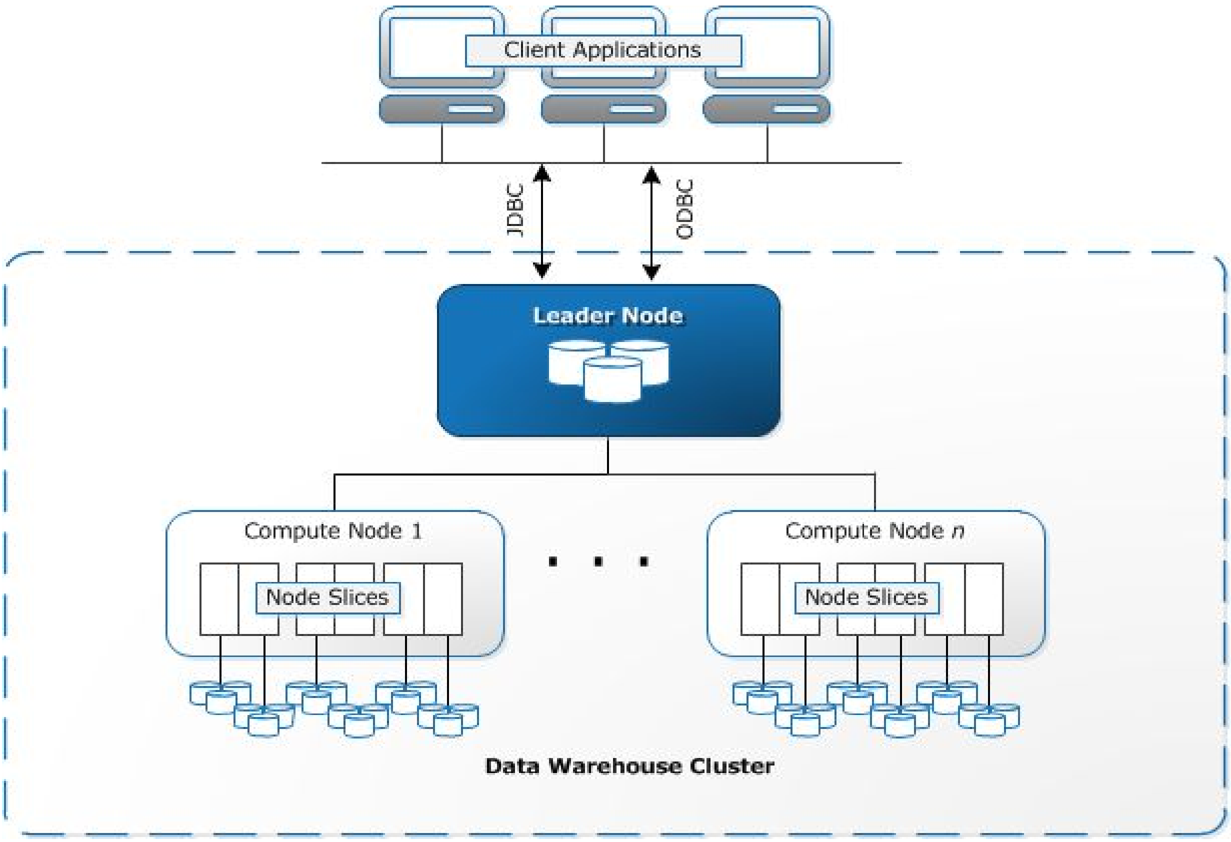
Amazon Redshift is an enterprise-level, petabyte scale, columnar, and fully managed data warehousing service from AWS. Its massive parallel processing and columnar compression make it one of the most efficient data warehousing services. It also supports standard SQL and is fast enough when compared to any traditional data warehouse. It also provides a quick option to load massive data sets.

Snowflake

Snowflake is a simple, affordable, and high-performance data warehousing service on cloud built using AWS. It stores data in controlled columnar fashion. It provides broad support for standard SQL queries(update, delete, and joins). It was also built keeping in mind the challenges faced by conventional data warehousing systems. The management cost and effort is almost zero in this solution as there is no infrastructure to manage. It automatically handles security, optimization, infrastructure, etc.

Architectural Difference

Redshift



- **Integrations**

Amazon Redshift can be integrated with various ETL tools like Hevo, BI reporting like Power BI, and other analytics tools. Redshift follows industry-standard PostgreSQL hence most existing SQL client applications would work with least changes.

- **Connections**

Amazon Redshift communicates with applications by using PostgreSQL JDBC and ODBC drivers.

- **Clusters**

The core component of the data warehouse is a cluster on Redshift. A cluster can have one or more compute nodes. The various nodes in Redshift cluster are following:

1. Leader Node

The leader node interacts with client programs and does all the communication with compute nodes. It communicates steps to obtain a certain result in the most efficient way, assigning data storage to all to compute nodes. It does not store any data and acts as a leader instructing all the compute nodes for the actions.

2. Compute Nodes

The leader node compiles code for the request and assigns the code to individual compute nodes. Now, all the compute nodes will execute the compiled code and send results back to the leader.

Each compute node has its own CPU, memory, and disk storage, which are configured by the node type from AWS console login or CLI.

Node Slices

A compute node is made up of slices (partitions). Each slice has a portion of the node's memory and some disk space, where it processes the workload assigned to the node. The number of slices per node is defined by the node size of the cluster.

Internal network

Redshift uses complex protocols (VPC) to provide highly secure and high-speed network communication between leader and compute nodes without hampering the performance.

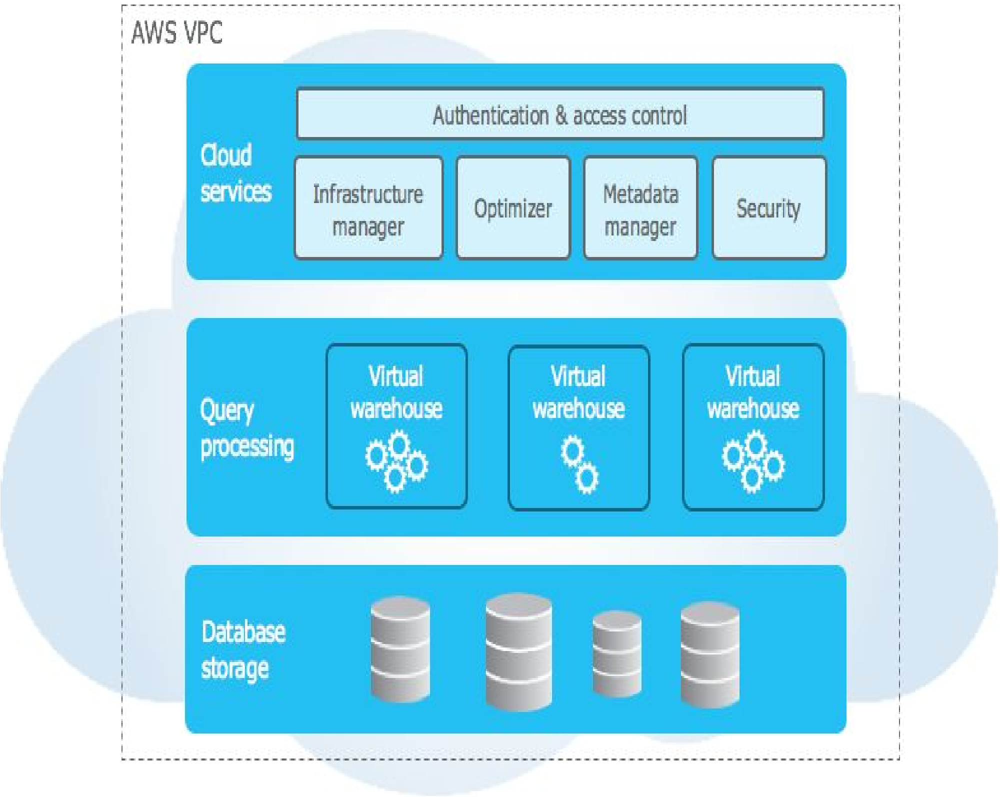
- **Databases**

A cluster contains one or more databases. User data is stored on the compute nodes. The SQL client requests the leader node, which in turn asks query execution with the compute nodes in a systematic way.

Amazon Redshift is an RDBMS, therefore it's compatible with most of the RDBMS applications and it is customized for high-performance analysis and reporting/KPI's of large datasets.

Amazon Redshift resides on PostgreSQL 8.0.2. However, Amazon Redshift and PostgreSQL have few important differences that need to be accounted as you develop your data warehouse applications.

Snowflake



Snowflake is a combination of shared disk database and shared-nothing database architectures. It fits for both structured and semi-structured data. As a feature of shared-disk architectures Snowflake has central data storage for prolonged data that can be accessed by all compute nodes in the warehouse. While as a feature of shared-nothing architectures, Snowflake queries are performed in parallel which is termed as MPP (Massively Parallel Processing). Snowflake uses micro-partitions to securely and efficiently store data. When loaded into Snowflake, data is automatically split into micro-partitions, and metadata is extracted to enable efficient query processing. This makes data management simple and the management cost comes to zero which is not in the case of Redshift. It also holds performance and scale-out benefits.

Database Storage

When data gets loaded into Snowflake tables, the data is stored in the compressed and columnar format in the most optimized way. Snowflake uses Amazon Web Services S3 (Simple Storage Service) cloud storage for the same purpose.

Snowflake manages almost all the admin and management aspects of how this data is stored in S3 — the size of a file, its structure, columnar compression, metadata definition of data storage. The data objects stored in S3 is not visible to customers. They can only be accessed through SQL query operations.

Query Processing

All the query part is performed in the processing layer. Queries are processed using virtual warehouses. Virtual warehouse acts as an independent cluster allocated with separate workload as per our requirements. It uses AWS EC2 for achieving this purpose. It is the most prime feature of Snowflake as compared to Redshift which lacks such mechanism.

Cloud Services

It is a layer for a collection of all the managed services that coordinate tasks across Snowflake architecture. Snowflake supports multiple ways of connecting to most of the services.

A web-based user interface and command line clients (e.g. SnowSQL) both are capable of managing and using Snowflake.

Primary/Foreign Keys Constraints

Redshift

Primary keys and foreign key constraints are just for information. They are not mandatory in Redshift. However, primary keys and foreign keys are used to design an effective query plan by the query engine. Hence, it is a good practice to declare them. The query planner uses these relationships but it assumes that all keys in Amazon Redshift tables are valid as loaded. So, we need to show extra care with integrity constraints. If the application allows invalid keys, few queries could return spiked results. Amazon Redshift enforces NOT NULL column constraints. Data distribution, workload management of queries, data partition, configuring nodes, clusters, table sorting, and S3 are some unique feature to store and access the data in the most efficient way.

Snowflake

Snowflake also supports defining constraints but does not enforce them, except for NOT NULL as in the case of Redshift. Snowflake supports constraints on permanent, transient, and temporary tables. Constraints can be defined on any number of columns of any data types. For Snowflake Time Travel (data recovery), when previous versions of a table are recovered, the current version of the constraints on the table is used because the history of metadata is not stored on Snowflake. It is a zero management data warehousing service as data distribution, workloads, configuring nodes, backups, and most of the tasks related to managing and storing data are either managed by Snowflake or are a matter of few clicks. Snowflake focuses on analyzing the data more rather than managing them. We can create many virtual warehouses and configure them as per need. It is very cost effective and easy to create.

Performance Differences

Redshift

Redshift has various ways to get high-performance parallel queries. Experts believe that it results in a speedup of 8 times on long-running queries over PostgreSQL, MySQL, and SQL Server.

- **Workload management:** Database admins can control query queues where queries can be provided more priority over any other ETL jobs as per our requirement.
- **Data compression:** Individual columns in Redshift are stored separately. We can define the compression type while creating the table. It also helps in high throughput while transferring data across the cluster.
- **Query optimizer:** It is intelligently designed for massively parallel processing as per the trend of modern data warehousing service.

Snowflake

Snowflake is a relational columnar-store cluster warehousing solution similar to Redshift supporting MPP.

- **Virtual computation warehouses:** Snowflake provides the capability of creating virtual warehouses for each of your independent tasks. For example, a reporting query can hit a virtual warehouse 1 and KPI query can hit virtual warehouse 2 as they are independently querying the data the performance remains the same. Also, ETL jobs can run on slower and less expensive warehouse and business related queries can run on high-performance warehouse during business hours. You can easily scale up, down, or pause compute power. Also, you only pay when you query.
- **Data retention:** Snowflake has a time travel feature which can help you easily revisit the historical data anytime between the last ninety days. Redshift can be configured for auto backup in S3.

- **Automatic tuning efforts**: Snowflake self-tunes the performance of the system as you use it. It even takes care of scaling and resizing as per demand. A very little hands-on admin approach is required as it manages optimization related tasks on its own. Thus, you barely need a database admin to perform the mentioned tasks. In Redshift, a database admin is required.

Note: For a cluster that runs 24 hours a day Redshift is the best option. Whereas for reporting queries and when ETL is only done when required then Snowflake is a better option as you are only charged when you query the data warehouse.

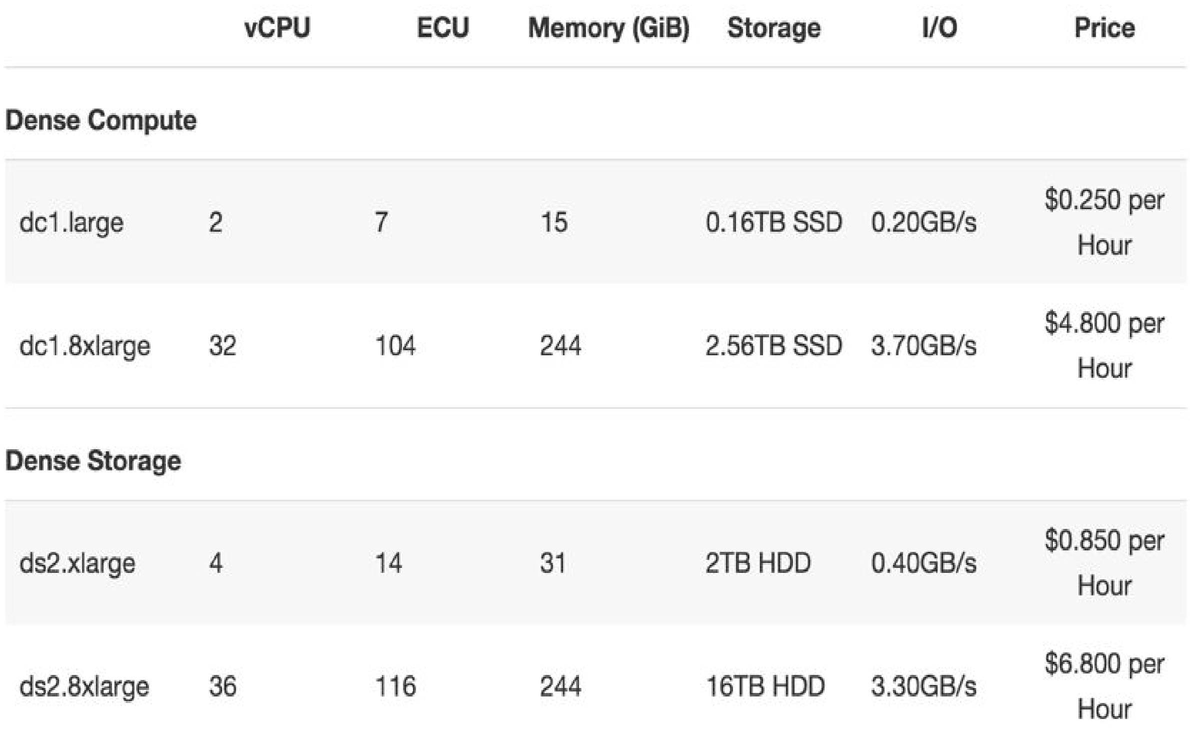
Pricing Models

Redshift

Redshift operates on two kinds of pricing model as mentioned below:

- On-demand Pricing - Pay at an hourly rate.
- Reserved Instance Pricing - 1 or 3-year contract and it is 75% cheaper than the on-demand model.

Redshift charges are based on the number of hours and number of nodes. The pricing starts at $0.25 per-hour for 160GB data. Redshift lets you choose the hardware specifications as per your requirement. It helps you find how much storage and throughput you get from the money invested.

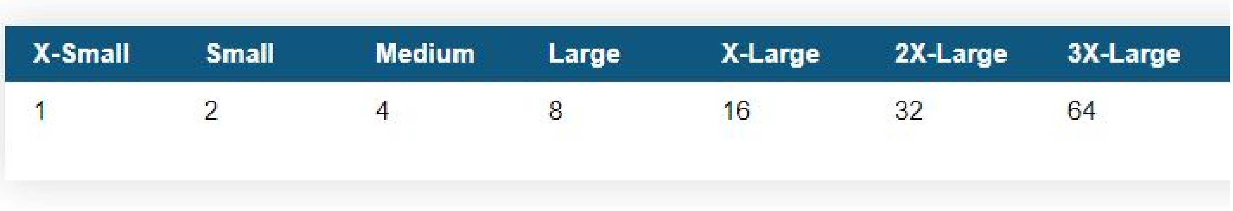


	vCPU	ECU	Memory (GiB)	Storage	I/O	Price
Dense Compute						
dc1.large	2	7	15	0.16TB SSD	0.20GB/s	$0.250 per Hour
dc1.8xlarge	32	104	244	2.56TB SSD	3.70GB/s	$4.800 per Hour
Dense Storage						
ds2.xlarge	4	14	31	2TB HDD	0.40GB/s	$0.850 per Hour
ds2.8xlarge	36	116	244	16TB HDD	3.30GB/s	$6.800 per Hour

Source: [AWS Redshift Pricing]

Snowflake

Snowflake pricing largely depends upon the usage pattern. It charges an hourly rate for each of the virtual warehouses created. Data storage is decoupled so it is charged separately as $0.20 per TB per month. It offers 7 different types of the warehouse. The X-small is the smallest which is charged at $2 per hour. Snowflake offers dynamic pricing model which means clusters will shut down when not in use and automatically start when in use. They also can be resized on the fly depending upon the workload thus saves more money.



X-Small	Small	Medium	Large	X-Large	2X-Large	3X-Large
1	2	4	8	16	32	64

Source: Snowflake Manual

Choosing the Right Cluster

Selecting the right cluster depends on your usage patterns. If the cluster is up and running 24 hours a day (due to ETL or reporting) Redshift is the better option. If the ETL runs ones in a week and only querying of data is required as per demand Snowflake is the option.

Scalability

Redshift

Let’s consider if you are trying to load 1TB of data for the below instances. Data load speeds are proportional to the number of nodes defined in the cluster as shown by the findings below:

- A single node XL instance will take close to 16 hours.
- A multi-node XL instance of two nodes will take close to 9 hours.
- A multi-node 8XL instance of two nodes will take close to 1.5 hours.

Querying the data

A query will run faster when there is a number of nodes but the performance does not rise linearly. Redshift clusters are optimized for multiple node clusters supporting MPP in the best possible way.

Resizing

Redshift offers to resize, closing, launching of the cluster by a simple API call or by few mouse clicks from AWS console. The clusters can be upscaled or vice versa with few minutes of downtime.

Snowflake

As Snowflake is easy to use and accessible on almost any scale for all the users and applications deployed on the cloud. It manages storage, compute, and metadata separately. Billions of rows of data can be queried by concurrent users sitting anywhere. Storage and compute can be scaled up or down independently and the metadata service will automatically scale up and down as per the requirement.

In the environment, shutting down database operations for overnight is not required as Snowflake does this of its own. We can create independent clusters on the fly and assign it to the users based on priority and requirement. Thus you can have different users, different compute capacity but all pointing to the same data lake.

Unique Features

Redshift

- **Automatic Columnar Compression:** It provides better performance at lower costs.
- **Elastic MapReduce Access**: If you have data stored in EMR Data then it can be copied from an Elastic MapReduce cluster to a Redshift cluster.
- **Concurrency:** You can configure a maximum of 50 simultaneous queries across all user queues (Workload management).We can increase the concurrency to get better query performance for some long running queries.
- **Max Result Set Size:** The cursor counts and result set sizes can be configured.However, read the documentation carefully before proceeding with this step.
- **Resizing Indicator:** You can monitor the progress of cluster resizing task in the AWS Redshift console.

Snowflake

- **Full SQL database:** It supports DDL, DML, analytical functions, transactions, and complex joins.
- **Variety of data**: Snowflake ingests almost all kinds of data, either from traditional sources or machine-generated sources without tradeoffs. Snowflake supports both structured and semi-structured data like JSON and Avro.
- **No management:** Snowflake is a data warehouse as a service running in the cloud and thus there is no infrastructure to manage or knobs to turn. Snowflake automatically handles infrastructure requirement, optimization of queries or tables, data distribution, availability, and data security.
- **Performance:** Snowflake processes reports and KPIs at very high speed because of the columnar database engine.
- **Broad ecosystem:** Snowflake integrates with almost all kind of tools near to its ecosystem like Hevo, Redshift, BigQuery. The different custom connectors include ODBC, JDBC, Javascript, Python, Spark, R, and Node.js.

How to Perform ETL to Redshift and Snowflake?

AWS Redshift and Snowflake are high performing databases. However, migrating data from sources to Amazon Redshift and Snowflake involves multiple complex stages and can be a cumbersome experience.

If you want to load any data easily into Redshift and Snowflake without any hassle, you can try out Hevo. Hevo automates the flow of data from various sources to Amazon Redshift and Snowflake in real time and at zero data loss. In addition to migrating data, you can also build aggregates and joins on Redshift and Snowflake to create materialized views that enable faster query processing.