AWS Redshift

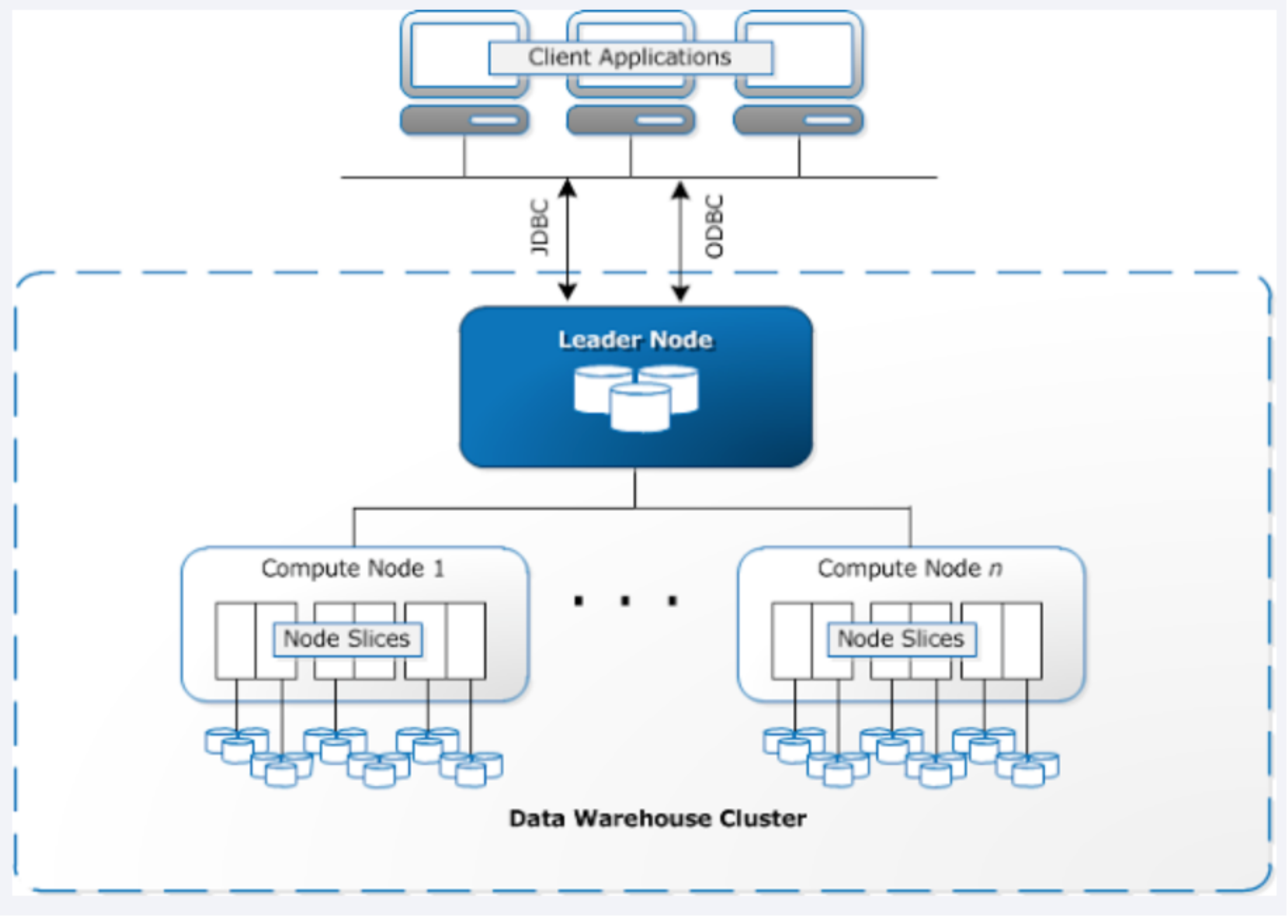
- **Amazon Redshift is a fully managed, fast, and powerful, petabyte-scale data warehouse service.**
- Redshift is an OLAP data warehouse solution based on PostgreSQL.
- Redshift automatically helps
  - set up, operate, and scale a data warehouse, from provisioning the infrastructure capacity.
  - patches and backs up the data warehouse, storing the backups for a user-defined retention period.
  - monitors the nodes and drives to help recovery from failures.
  - significantly lowers the cost of a data warehouse, but also makes it easy to analyze large amounts of data very quickly
  - provide fast querying capabilities over structured and semi-structured data using familiar SQL-based clients and business intelligence (BI) tools using standard ODBC and JDBC connections.
  - uses replication and continuous backups to enhance availability and improve data durability and can automatically recover from node and component failures.
  - scale up or down with a few clicks in the AWS Management Console or with a single API call
  - distributes & parallelize queries across multiple physical resources
  - supports VPC, SSL, AES-256 encryption, and Hardware Security Modules (HSMs) to protect the data in transit and at rest.

Redshift Performance

- **Massively Parallel Processing (MPP)**
  - automatically distributes data and query load across all nodes.
  - makes it easy to add nodes to the data warehouse and enables fast query performance as the data warehouse grows.
- **Columnar Data Storage**
  - organizes the data by column, as column-based systems are ideal for data warehousing and analytics, where queries often involve aggregates performed over large data sets
  - columnar data is stored sequentially on the storage media, and require far fewer I/Os, greatly improving query performance
- **Advance Compression**
  - Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on a disk.

    - employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores.
    - doesn’t require indexes or materialized views and so uses less space than traditional relational database systems.
    - automatically samples the data and selects the most appropriate compression scheme, when the data is loaded into an empty table
- **Result Caching**
    - Redshift caches the results of certain types of queries in memory on the leader node.
    - When a user submits a query, Redshift checks the results cache for a valid, cached copy of the query results. If a match is found in the result cache, Redshift uses the cached results and doesn’t run the query.
    - Result caching is transparent to the user.
- **Complied Code**
    - Leader node distributes fully optimized compiled code across all of the nodes of a cluster. Compiling the query decreases the overhead associated with an interpreter and therefore increases the runtime speed, especially for complex queries.

Redshift Architecture



- **Clusters**
  - Core infrastructure component of a Redshift data warehouse
  - Cluster is composed of one or more compute nodes.
  - If a cluster is provisioned with two or more compute nodes, an additional leader node coordinates the compute nodes and handles external communication.
  - Client applications interact directly only with the leader node.
  - Compute nodes are transparent to external applications.
- **Leader node**
  - Leader node manages communications with client programs and all communication with compute nodes.
  - It parses and develops execution plans to carry out database operations
  - Based on the execution plan, the leader node compiles code, distributes the compiled code to the compute nodes, and assigns a portion of the data to each compute node.
  - Leader node distributes SQL statements to the compute nodes only when a query references tables that are stored on the compute nodes. All other queries run exclusively on the leader node.
- **Compute nodes**

  - Leader node compiles code for individual elements of the execution plan and assigns the code to individual compute nodes.
  - Compute nodes execute the compiled code and send intermediate results back to the leader node for final aggregation.
  - Each compute node has its own dedicated CPU, memory, and attached disk storage, which is determined by the node type.
  - As the workload grows, the compute and storage capacity of a cluster can be increased by increasing the number of nodes, upgrading the node type, or both.
- **Node slices**
  - A compute node is partitioned into slices.
  - Each slice is allocated a portion of the node's memory and disk space, where it processes a portion of the workload assigned to the node.

Redshift Serverless

- Redshift Serverless is a serverless option of Redshift that makes it more efficient to run and scale analytics in seconds without the need to set up and manage data warehouse infrastructure.
- Redshift Serverless automatically provisions and intelligently scales data warehouse capacity to deliver high performance for demanding and unpredictable workloads.

Redshift Availability & Durability

- Redshift replicates the data within the data warehouse cluster and continuously backs up the data to S3 (11 9's durability).
- Redshift mirrors each drive's data to other nodes within the cluster.
- Redshift will automatically detect and replace a failed drive or node.