Amazon Redshift Warehouse on Cloud Part 1

RAJAS WALAVALKAR

Associate Solution Architect - Quantiphi
&
Community Builder

7th MAY
3:00 pm

Live stream on AWS UG INDIA

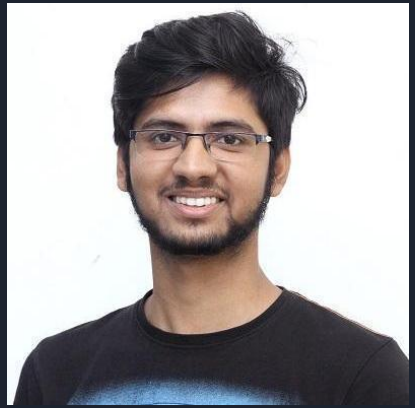
AWS
User Groups
Mumbai

@awsugmum

Introduction

Rajas Walavalkar - Associate Solution Architect at Quantiphi Analytics & AWS Community Builder

Expertised in AWS Data Analytics Services, Big Data Tools, Data Lake, Warehouse and Business Intelligence & Dashboarding tools



Linkedin - https://in.linkedin.com/in/rajas-walavalkar-4ab21b15b

Medium - rajaswalavalkar.medium.com

Agenda

1. Redshift Overview
    a. Introduction
    b. High level Redshift Architecture
    c. Compute Node Types and their Use cases

2. Hands on
    a. Redshift Create Cluster & Tables
    b. Redshift - Load Data using COPY Command
    c. Query Data on Redshift - Save it in a table - CTAS query
    d. Unload the data from the table into an S3 bucket

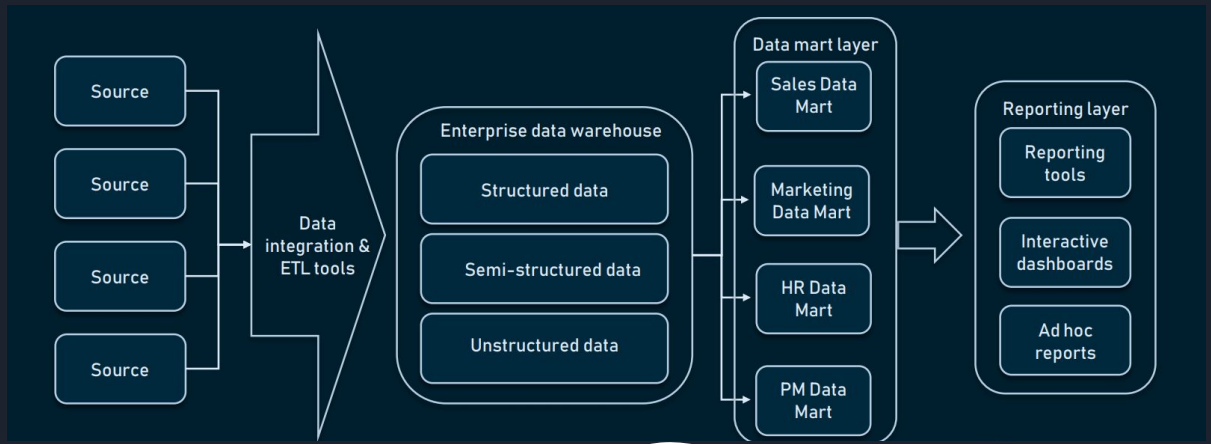
2. Redshift Materialized views
    a. Materialized Overview & Use case
    b. Hands-on - Create materialized view, Query the View, Refresh the view

1. Redshift Overview

What is Amazon Redshift?

Amazon Redshift is an enterprise wide **secured**, **fully managed** & **scalable** Data Warehouse solution that AWS provides

Application of Warehouse



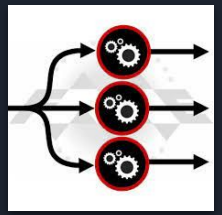
Features

- Columnar Data storage

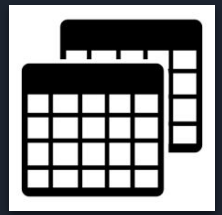
- Concurrency scaling

- Automatic Caching

- MPP (Massive Parallel Processing)



- Materialized Views

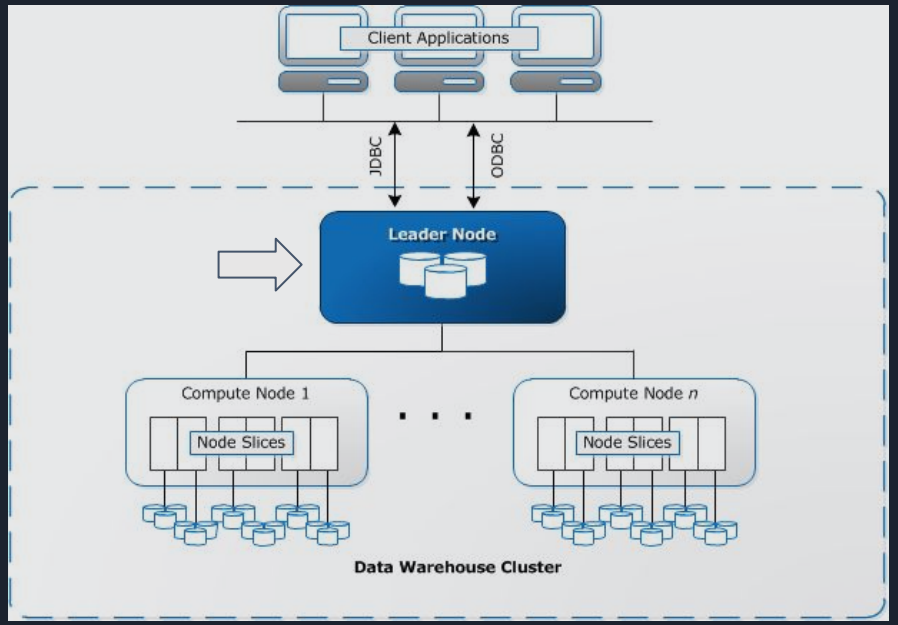


- Data Encryption - at Rest & in-transit

Redshift Internal Architecture

- **Cluster -** A cluster is composed of one or more compute nodes and a leader node

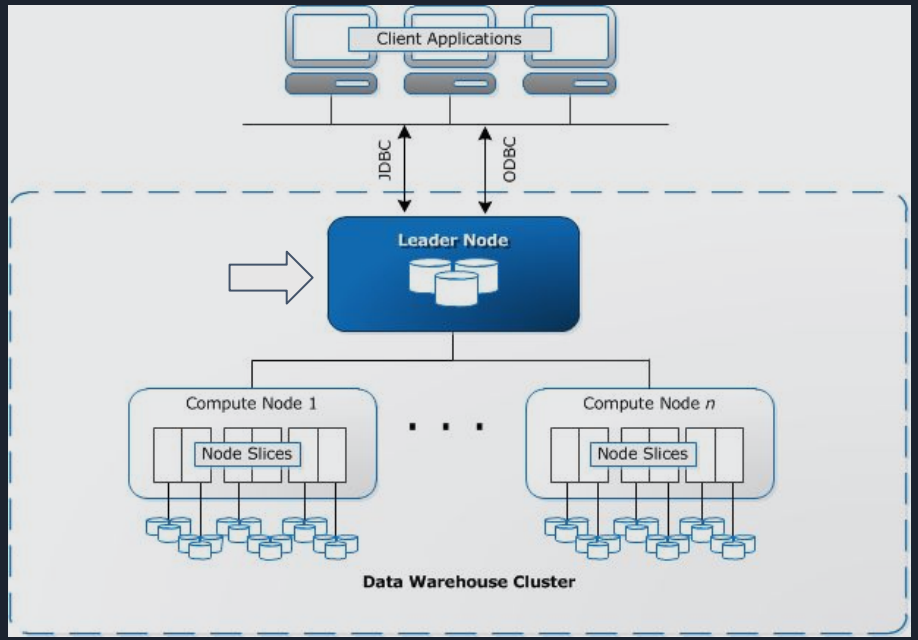
- **Leader Node -**
  - The leader node manages communications with client programs and compute nodes
  - It stores all the table statistics and the location of the data partitions spread across the compute nodes



Redshift Internal Architecture

Leader Node -

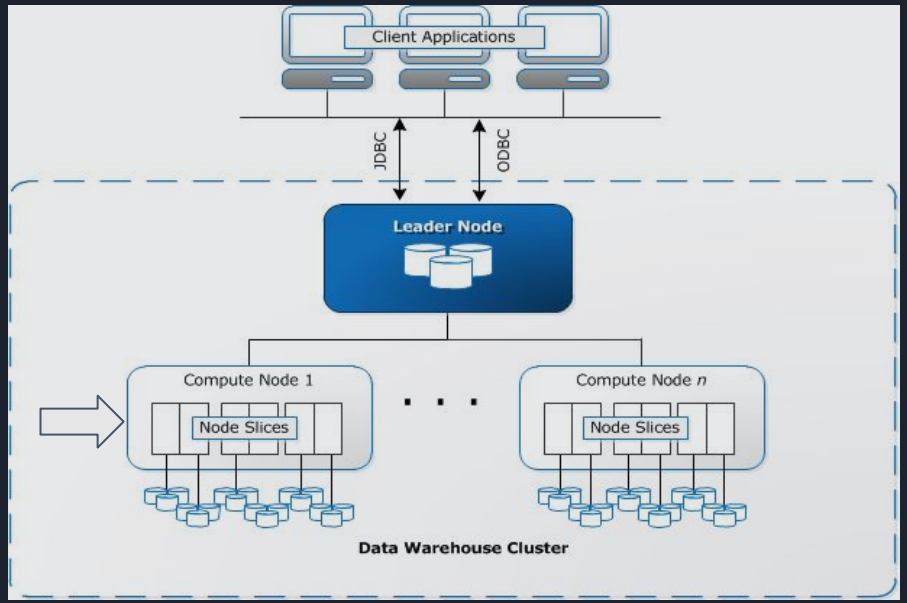
- It created Query Execution plan and assigns tasks to compute nodes as per the plan
- It maps all the outputs of the compute nodes and accordingly gives the final result back to the client
- For a multi-node cluster Amazon makes sure that the Leader node is Fault tolerant



Redshift Internal Architecture

Compute Node -

- The compute nodes runs the assigned tasks and sends the intermediate results back to the leader node
- Each compute node has its own dedicated CPU, memory, and attached disk storage
- As the workload grows, we can increase the compute capacity and storage capacity of a cluster by increasing the number of nodes



Redshift Compute Node Types

1. **Dense Storage (DS2) Type**

   These Compute nodes are used when you have huge data volume storage requirement (HDD Disks) and less Compute requirement. ***AWS recommends to upgrade the dense storage compute nodes to RA3 Types***

Dense storage node types

Node size	vCPU	RAM (GiB)	Default slices per node	Storage per node	Node range	Total capacity
ds2.xlarge	4	31	2	2 TB HDD	1–32	64 TB
ds2.8xlarge	36	244	16	16 TB HDD	2–128	2 PB

Redshift Compute Node Types

2. **Dense Compute (DC2) Type**

   DC2 nodes types provide a greater compute than storage power. These nodes store your data locally (SSD Disks) for high performance, and as the data size grows, you can add more compute nodes to increase the storage capacity of the cluster

Dense compute node types

Node size	vCPU	RAM (GiB)	Default slices per node	Storage per node	Node range	Total capacity
dc2.large	2	15	2	160 GB NVMe-SSD	1–32	5.12 TB
dc2.8xlarge	32	244	16	2.56 TB NVMe-SSD	2–128	326 TB
dc1.large[1]	2	15	2	160 GB SSD	1–32	5.12 TB
dc1.8xlarge[1]	32	244	32	2.56 TB SSD	2–128	326 TB

Redshift Compute Node Types

3. **RA3 Node Type**

   RA3 node type are used when your Compute requirements are high as well as your storage requirements are also huge. RA3 uses 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

RA3 node types

Node size	vCPU	RAM (GiB)	Default slices per node	Managed storage quota per node	Node range with create cluster	Total managed storage capacity
ra3.xlplus	4	32	2	32 TB[1,5]	1–16[2]	1024 TB[2,4]
ra3.4xlarge	12	96	4	128 TB[1]	2–32[3]	8192 TB[3,4]
ra3.16xlarge	48	384	16	128 TB[1]	2–128	16,384 TB[4]

2. Let’s do some Hands-on

Prerequisites

1. You need to have admin access to the AWS account to follow the steps mentioned below

2. Download the dataset from the Kaggle form here. For this you will have to sign-in into your Kaggle account.

3. Follow the section of Create IAM role as a part of Pre-requisites

4. Follow the section of create an S3 bucket and uploading CSV data files which will be required while creating tables on Redshift

Further…

1. Create an IAM role for Redshift Cluster

2. Create a S3 bucket for uploading the CSV data files

3. Create the Redshift Cluster

4. reate Tables & Load the data
    a. Using Create Table Queries and COPY Commands
    b. Using a Visual Editor 2

5. Try Some Interesting and Cool Stuff…
    a. Use CTAS Query
    b. Use UNLOAD Command
    c. Create & Refresh Materialized views

Materialized Views

Compute Once and Query Multiple Times…

A materialized view is a database object which contains a precomputed result set, based on an SQL query over one or more base tables

Features

Pre computes the KPIs

Supports Incremental Refresh

Refresh views as per your need

Materialized views on materialized views

Use Case for Materialized Views

Compute Once and Query Multiple Times…

Events : This is a Event dimension which stores attributes of an event

Sales : This is a Sales fact table having all the required KPIs of sales

Solution:

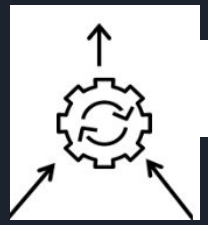
Create a tickets_mv materialized view which will pre-calculate the *total_sales* by different events

What if the base tables have new records inserted? - Refresh your materialized view then

Materialized view

tickets_mv

eventname	total_sales
Dolly Parton	10.00
Gotterburg	48.00



"What were the total sales by event?"

events

eventid	catid	eventname
1	8	Gotterburg
18	8	Gotterburg
5311	9	Dolly Parton

sales

salesid	eventid	cust	price
1	1	c1	12.00
2	18	c1	36.00
3	5311	c2	10.00

Thank you!

AWS
User Groups