# Amazon RedShift

Fully managed, petabyte – scaled data warehouse, AWS distributed

* 10 X better than other data warehouses
  + Via machine learning, massively powerful parallel execution
  + Columnar storage
* Designed for OLAP not for OLTP
* Cost effective
* SQL, ODBC, JDBC interfaces
* Scales up or down on demand
* Built in replication and backups
* Monitoring via cloud watch / cloud trail

**Use Cases**

* if you want your analytics to be fast
* want to unify data warehouse and data lake

## Red shift Spectrum

* query exabyte of unstructured data in S3 without loading
* limitless of concurrency
* horizontal scaling
* wide verity of data formats support
* Gzip and Snappy compression support
* it uses MPP that’s why its fast
* Columnar Data Storage **(This type of store not good for OLTP)**
* Column compression

**Redshift Durability**

* Data stored in 3 places
* Snapshots to s3 for disaster recovery
* Notes failed node and replaced node
* When node failure, most accessed data will be replicated
* They are limited to single availability zone
* 3 clusters multi-AZ announced

Redshift Distribution Styles

* **Auto** 
  + if not specified AWS does it , based on size
* **EVEN** 
  + rows distributed across sliced in round robin
* **Key** 
  + Rows distributed based on one column
* **All**
  + Entire table is copied

### Importing and Exporting

* Copy command
  + Paralyzed ; efficient
  + From s3 to, EMR , Dynamo DB, Remote, Hostos
  + S3 requires manifest file and IAM role
* UNLOAD
  + Get data out of redshift table into S3
* Enhanced VPC routing
  + Enhanced VPC Routing forces all COPY and UNLOAD traffic (i.e., data going in and out of Redshift) to go through your Amazon VPC (Virtual Private Cloud), rather than through the public internet.
* Auto copy data to S3
* Amazon Aurora 0 ETL integration
  + Aurora database to Redshit table
* Redshift Streaming ingestion
  + Kenisis database to MSK

Copy Command :

* Copy command can decrypt data as it is loaded from S3
* Automation compression option
* If you have a narrow table lot of row and but very few columns use a single command
* Copy is designed to parallel things

Red Shift copy grants for cross-region snapshot copies

* The copy-grants option is used when copying a **manual snapshot** of a Redshift cluster across **regions or accounts** to preserve **database-level GRANT permissions** (e.g., GRANT SELECT, GRANT USAGE).
* **Why it's needed:**

By default, **GRANTs are not retained**, only data and schema are.  
Without copy-grants, you'd need to **manually reassign user access**, which is risky and time-consuming — especially in **disaster recovery** or **multi-region** setups.

* In the destination of AWS :
  + **kms-key-id**: Defines the **KMS encryption key** to use in the **destination region**.
  + **target-snapshot-identifier**: Provides a **unique name** for the copied snapshot.
  + **--copy-grants**: Optional flag to **preserve user access permissions** (GRANTs) during the copy.

Reshift Integration with other services

Redshift World load manager (WLM)

* Amazon Redshift Workload Management (WLM) controls how system resources like memory and concurrency are allocated to different query types. It helps prioritize important workloads (e.g., BI dashboards) while isolating heavy ones (e.g., ETL jobs). You can use **manual WLM** to configure queues, memory, and timeouts, or **auto WLM** to let Redshift manage resources based on query priority. This ensures better performance, fairness, and control over long-running queries.

Concurrency Scaling

* Handle concurrent workloads
* Work with WLM
* Automatically add clusters to handle increase concurrent read queries

Automatic workload management

* Default 5 queues

Short query acceleration (option to accelerate WLMs)

* Idea is to prioritize short queries or that take short times
* Short query should not wait after long queries in queue
* Create table and Read only queries
* Can configure what time is short
* Uses machine learning algorithms to predict query time

**Vacuum Command**

* Recover space from deleted storage
* Vacuum full , vacuum delete (gives sorted space)
* Vacuum index , vacuum sort index

Redshift anti patterns

* Small datasets use RDS instead
* Not good for OLTP (use RDS)
* Unstructured data
* BLOB Data

Resize Redshift

* Elastic resize
  + Add or remove nodes of same type
  + Cluster is down for few minutes
  + Double or half only
* Classic resize
  + Take hours or days. only read only state
* Snap , restore , resize
  + Copy cluster, resize cluster

Newer redshift features

* RA3 nodes with managed storage
  + Enables independent scaling of compute and storage
  + SSD- based
* Redshift Data lake export
  + Export data to data lake s3 in parquet format
  + That data is compatible with athena , sagemaker ,
  + Automatically partitioned
* Spatial data types
* Cross region data sharing

Redshift security concerns

* HSM issue with redshift
  + Must use client and server certificates to configure a trusted connection between redshift and HSM
* Defining access privileges for user or group of user
  + USER GRANT and REVOKE commands in SQL

**# Redshift Serverless**

* Auto scaling and provisioning for your workload
* Optimizes costs and performance
  + Pay only what you are using only
* Uses ML to maintain performance across variable and sporadic workloads

Easy spinup of development and test environment

**Redshift scaling how does it work**

* Capacity is measure in RPU
* You pay for RPU hours (per second)
* Base RPU’s : you can adjust base capacity
* MAX RPUs
  + Can set usage limit

**Redshift vs redshift serverless**

* parameter group are not supported , workload management , version track not supported
* No public access point

Redshift serverless : monitoring

* History of query , usage of cluster etc.
* Publish data to cloud watch
  + Connection and user logs
  + Optional user activity
* Cloud watch matrix
  + Query duration , queries running etc
  + Dimensions : Database name , latency stage etc

**Redshift materialized Views**

* **Materialized Views** in Amazon Redshift store the **results of a query physically**, unlike standard views which compute results on-the-fly. This improves **query performance**, especially for complex joins and aggregations
* Does not take sub queries
* Is fast as they are precomputed result with out accessing base table
* Best for repetitive queries

**Redshift data sharing**

* Securely share live data for read only
* Why
  + Workload isolation
  + Cross group collaboration
  + Sharing data in dev/test/prod
* What can you share
  + Table , views, UDF
* Producer/ consumer as it is only read share
* Must use RA3 nodes , must be encrypted

Redshift lambda UDF (user defined functionality )

* Use custom function in AWS
* Lambda inside sql queries

**Amazon Redshift Federated Queries – Quick Overview**

* **Federated Queries** let Amazon Redshift **query data in external databases** (like RDS or Aurora PostgreSQL/MySQL) **without moving the data**.
* must establish connectivity between your redshift cluster and RDS/ Aurora
  + put them in the same VPC subnet
  + Or use vpc peering
* Credentials must be in AWS Secrets manager

Redshift views

| **Type** | **Prefix** | **Use Case** |
| --- | --- | --- |
| **System Tables** | STL\_ | Low-level logs (queries, scans, loads, errors) |
| **System Views** | SVL\_ | Views on STL\_ tables (easier to use) |
| **System Views (Stats)** | SVV\_ | Current state of system: users, tables, roles |
| **System Catalog** | PG\_ | PostgreSQL-style catalog tables |

Redshift Data API

* Secure HTTP endpoint for sql statements to redshift clusters
  + Provisioned or serverless
  + Individual or batch queries
* Asynchronous
* Does not require managing connections
  + No drivers needed
* Password not sent via api
* Query duration 24 hours, active query at one time 500, result size 100 mb
* Packet for query data is 64 kb