

Amazon redshift

fully managed petta-byte scale data warehouse

What is Redshift?

- Fully-managed, petabyte scale data warehouse service
- 10X better performance than other DW's
 - Via machine learning, massively parallel query execution, columnar storage
- Designed for OLAP, not OLTP
- Cost effective
- SQL, ODBC, JDBC interfaces
- Scale up or down on demand
- Built-in replication & backups
- Monitoring via CloudWatch / CloudTrail



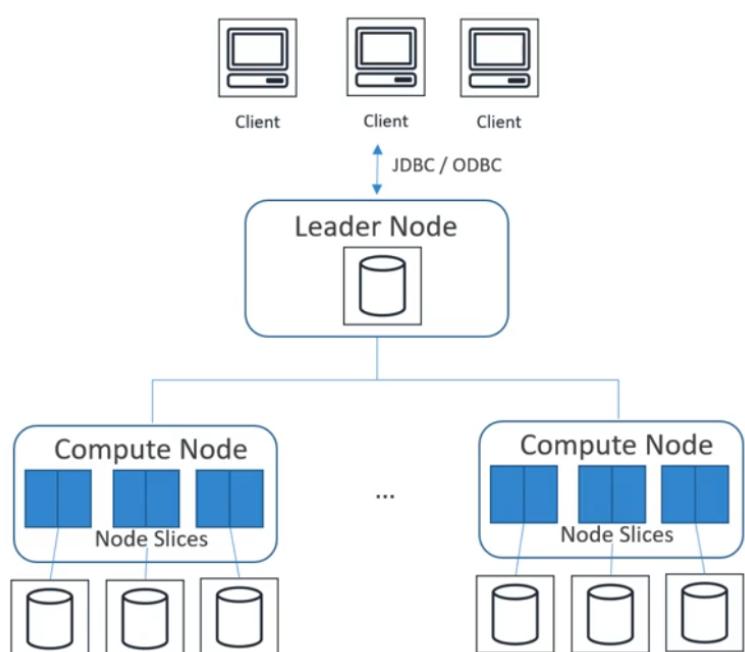
**Amazon
Redshift**

special designed for online process olap not oltp (oltp is row based storage)

Redshift Use-Cases

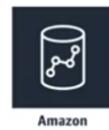
- Accelerate analytics workloads
- Unified data warehouse & data lake
- Data warehouse modernization
- Analyze global sales data
- Store historical stock trade data
- Analyze ad impressions & clicks
- Aggregate gaming data
- Analyze social trends

Redshift architecture



Redshift Spectrum

- Query exabytes of unstructured data in S3 without loading
- Limitless concurrency
- Horizontal scaling
- Separate storage & compute resources
- Wide variety of data formats
- Support of Gzip and Snappy compression



Amazon
Redshift



AWS Glue



Amazon S3

Redshift Performance

- Massively Parallel Processing (MPP)
- Columnar Data Storage
- Column Compression

Redshift Durability

- Replication within cluster
- Backup to S3
 - Asynchronously replicated to another region
- Automated snapshots
- Failed drives / nodes automatically replaced
- However – limited to a single availability zone (AZ)

you need to have two nodes to replication one node failed redshift will detect the other node and run on the second node

easy way to back up the files auto snap shots will present on the redshift you need to enable the automated snapshots

most frequently added data how ever added first

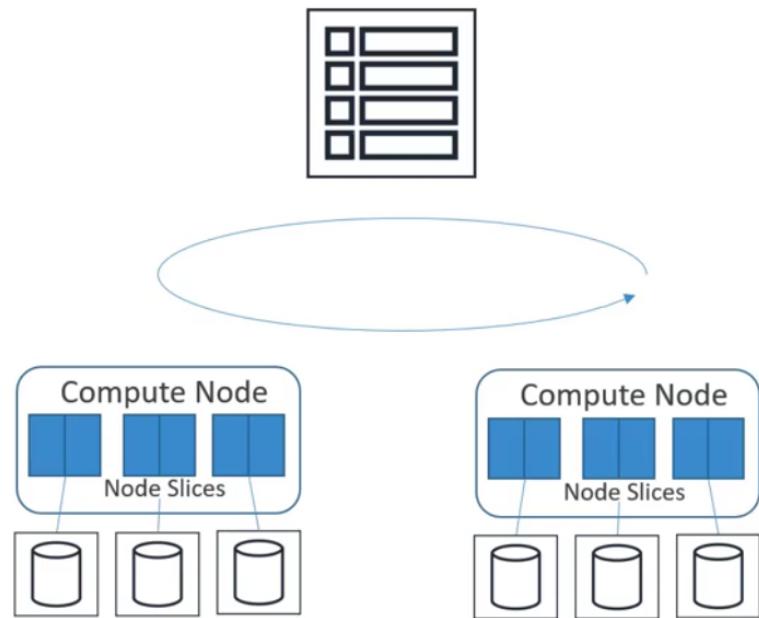
Scaling Redshift

- Vertical and horizontal scaling on demand
- During scaling:
 - A new cluster is created while your old one remains available for reads
 - CNAME is flipped to new cluster (a few minutes of downtime)
 - Data moved in parallel to new compute nodes

Redshift Distribution Styles

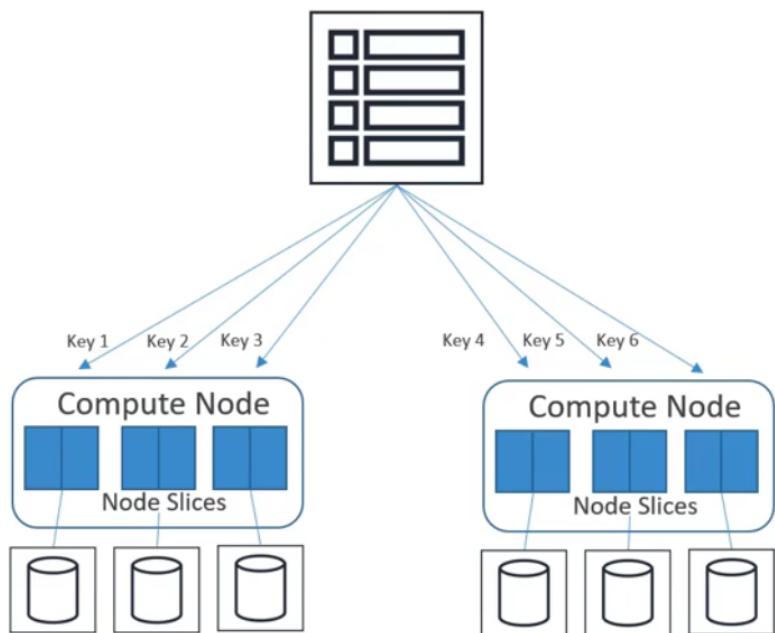
- AUTO
 - Redshift figures it out based on size of data
- EVEN
 - Rows distributed across slices in round-robin
- KEY
 - Rows distributed based on one column
- ALL
 - Entire table is copied to every node

EVEN distribution



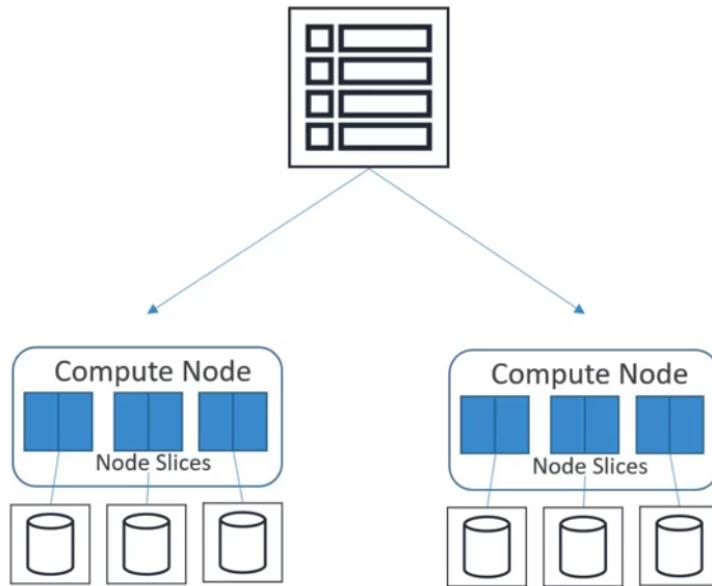
evenly spread the data

KEY distribution



key distribution matching the keys and distribute the data

ALL distribution



all not fast update it will slow and small tables to all distribution pg class distribution

Redshift Sort Keys

- Rows are stored on disk in sorted order based on the column you designate as a sort key
- Like an index
- Makes for fast range queries
- Choosing a sort key
 - Recency? Filtering? Joins?
- Single vs. Compound vs Interleaved sort keys



you need to set the time stamp for the sort key recent used data

Sort Keys: Single Column

| Date | Genre | Movie |
|-----------|-----------|--------------------------------------|
| 3/18/2019 | Comedy | Monty Python and the Holy Grail |
| 3/18/2019 | Adventure | Indiana Jones and the Temple of Doom |
| 3/18/2019 | Drama | Interstellar |
| 3/18/2019 | Drama | The Dark Knight |
| 3/19/2019 | Fantasy | The Lord of the Rings |
| 3/19/2019 | Drama | 12 Angry Men |
| 3/19/2019 | Adventure | Inception |

you dont set the short key it will take compound key

Sort Keys: Interleaved

| Date | Genre | Movie | Date | Genre | Movie |
|-----------|-----------|--------------------------------------|-----------|-----------|--------------------------------------|
| 3/18/2019 | Adventure | Indiana Jones and the Temple of Doom | 3/19/2019 | Drama | 12 Angry Men |
| 3/18/2019 | Comedy | Monty Python and the Holy Grail | 3/19/2019 | Adventure | Inception |
| 3/18/2019 | Drama | Interstellar | 3/18/2019 | Adventure | Indiana Jones and the Temple of Doom |
| 3/18/2019 | Drama | The Dark Knight | 3/18/2019 | Drama | Interstellar |
| 3/19/2019 | Adventure | Inception | 3/18/2019 | Comedy | Monty Python and the Holy Grail |
| 3/19/2019 | Drama | 12 Angry Men | 3/18/2019 | Drama | The Dark Knight |
| 3/19/2019 | Fantasy | The Lord of the Rings | 3/19/2019 | Fantasy | The Lord of the Rings |

| Date | Genre | Movie |
|-----------|-----------|--------------------------------------|
| 3/18/2019 | Adventure | Indiana Jones and the Temple of Doom |
| 3/19/2019 | Adventure | Inception |
| 3/18/2019 | Comedy | Monty Python and the Holy Grail |
| 3/18/2019 | Drama | Interstellar |
| 3/18/2019 | Drama | The Dark Knight |
| 3/19/2019 | Drama | 12 Angry Men |
| 3/19/2019 | Fantasy | The Lord of the Rings |

interleaved short key is used to set the short key multi query for multi tables

Importing / Exporting data

- COPY command
 - Parallelized; efficient
 - From S3, EMR, DynamoDB, remote hosts
 - S3 requires a manifest file and IAM role
- UNLOAD command
 - Unload from a table into files in S3
- Enhanced VPC routing

COPY command: More depth

- Use COPY to load large amounts of data from outside of Redshift
- If your data is already in Redshift in another table,
 - Use INSERT INTO ...SELECT
 - Or CREATE TABLE AS
- COPY can decrypt data as it is loaded from S3
 - Hardware-accelerated SSL used to keep it fast
- Gzip, Izop, and bzip2 compression supported to speed it up further
- Automatic compression option
 - Analyzes data being loaded and figures out optimal compression scheme for storing it
- Special case: narrow tables (lots of rows, few columns)
 - Load with a single COPY transaction if possible
 - Otherwise hidden metadata columns consume too much space



Redshift copy grants for cross-region snapshot copies

- Let's say you have a KMS-encrypted Redshift cluster and a snapshot of it
- You want to copy that snapshot to another region for backup
- In the destination AWS region:
 - Create a KMS key if you don't have one already
 - Specify a unique name for your snapshot copy grant
 - Specify the KMS key ID for which you're creating the copy grant
- In the source AWS region:
 - Enable copying of snapshots to the copy grant you just created

DBLINK

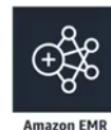
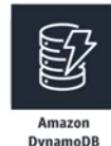
- Connect Redshift to PostgreSQL (possibly in RDS)
- Good way to copy and sync data between PostgreSQL and Redshift



```
CREATE EXTENSION postgres_fdw;
CREATE EXTENSION dblink;
CREATE SERVER foreign_server
    FOREIGN DATA WRAPPER postgres_fdw
    OPTIONS (host '<amazon_redshift_ip>', port '<port>', dbname '<database_name>', sslmode
'require');
CREATE USER MAPPING FOR <rds_postgresql_username>
    SERVER foreign_server
    OPTIONS (user '<amazon_redshift_username>', password '<password>');
```

Integration with other services

- S3
- DynamoDB
- EMR / EC2
- Data Pipeline
- Database Migration Service



s3 parallel processing export data from s3 to redshift

dynamodb using the copy command to set the data from s3

emr ec2 using ssh

data pipeline automated the data processing move the data

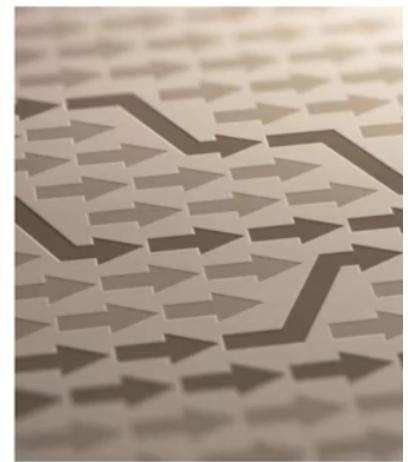
Concurrency Scaling

- Automatically adds cluster capacity to handle increase in concurrent **read** queries
- Support virtually unlimited concurrent users & queries
- WLM queues manage which queries are sent to the concurrency scaling cluster

concurrency means how many query the data at ones

Automatic Workload Management

- Creates up to 8 queues
- Default 5 queues with even memory allocation
- Large queries (ie big hash joins) -> concurrency lowered
- Small queries (ie inserts, scans, aggregations) -> concurrency raised
- Configuring query queues
 - Priority
 - Concurrency scaling mode
 - User groups
 - Query groups
 - Query monitoring rules



Manual Workload Management

- One default queue with concurrency level of 5 (5 queries at once)
- Superuser queue with concurrency level 1
- Define up to 8 queues, up to concurrency level 50
 - Each can have defined concurrency scaling mode, concurrency level, user groups, query groups, memory, timeout, query monitoring rules
 - Can also enable query queue hopping
 - Timed out queries "hop" to next queue to try again

Short Query Acceleration (SQA)

- Prioritize short-running queries over longer-running ones
- Short queries run in a dedicated space, won't wait in queue behind long queries
- Can be used in place of WLM queues for short queries
- Works with:
 - CREATE TABLE AS (CTAS)
 - Read-only queries (SELECT statements)
- Uses machine learning to predict a query's execution time
- Can configure how many seconds is "short"

VACUUM command

- Recovers space from deleted rows
- VACUUM FULL
- VACUUM DELETE ONLY
- VACUUM SORT ONLY
- VACUUM REINDEX

Redshift anti-patterns

- Small data sets
 - Use RDS instead
- OLTP
 - Use RDS or DynamoDB instead
- Unstructured data
 - ETL first with EMR etc

- ETL first with ETLR etc.

- BLOB data

- Store references to large binary files in S3, not the files themselves.

Resizing Redshift Clusters

- Elastic resize

- Quickly add or remove nodes of same type
 - (It *can* change node types, but not without dropping connections – it creates a whole new cluster)
- Cluster is down for a few minutes
- Tries to keep connections open across the downtime
- Limited to doubling or halving for some dc2 and ra3 node types.

- Classic resize

- Change node type and/or number of nodes
- Cluster is read-only for hours to days

- Snapshot, restore, resize

- Used to keep cluster available during a classic resize
- Copy cluster, resize new cluster

New Redshift features for 2020

- RA3 nodes with managed storage
 - Enable independent scaling of compute and storage
- Redshift data lake export
 - Unload Redshift query to S3 in Apache Parquet format
 - Parquet is 2x faster to unload and consumes up to 6X less storage
 - Compatible with Redshift Spectrum, Athena, EMR, SageMaker
 - Automatically partitioned

Newer Redshift features

- RA3 nodes with managed storage
 - Enable independent scaling of compute and storage
 - SSD-based
- Redshift data lake export
 - Unload Redshift query to S3 in Apache Parquet format
 - Parquet is 2x faster to unload and consumes up to 6X less storage
 - Compatible with Redshift Spectrum, Athena, EMR, SageMaker
 - Automatically partitioned
- Spatial data types
 - GEOMETRY, GEOGRAPHY
- Cross-Region Data Sharing
 - Share live data across Redshift clusters without copying
 - Requires new RA3 node type
 - Secure, across regions and across accounts

AQUA

- Advanced Query Accelerator
- Available on ra3.4xl, ra3.16xl
- Pushes reduction and aggregation queries closer to the data
- Up to 10X faster, no extra cost, no code changes.
- Also benefits from high-bandwidth connection to S3
- All you have to do is turn it on in your cluster configuration (when using the supported node types)

Redshift Cluster

AQUA

S3

Redshift security concerns

- Using a Hardware Security Module (HSM)
 - Must use a client and server certificate to configure a trusted connection between Redshift and the HSM
 - If migrating an unencrypted cluster to an HSM-encrypted cluster, you must create the new encrypted cluster and then move data to it.
- Defining access privileges for user or group
 - Use the GRANT or REVOKE commands in SQL
 - Example: grant select on table foo to bob;

Redshift Serverless

- Automatic scaling and provisioning for your workload
- Optimizes costs & performance
 - Pay only when in use
- Uses ML to maintain performance across variable & sporadic workloads
- Easy spinup of development and test environments
- Easy ad-hoc business analysis
- You get back a serverless endpoint, JDBC/ODBC connection, or just query via the console's query editor.

The screenshot shows the 'Get started with Amazon Redshift Serverless (Preview)' wizard. It includes sections for 'Serverless credit' (with a link to learn more), 'Choose starter base configuration' (checkbox), 'Configuration' (with 'Use default settings' and 'Customize settings' options), 'Database name and password' (with a 'Database name' input field containing 'dev'), and 'Admin user credentials'.

Redshift Serverless: Getting Started

- Need an IAM role with this policy
- Define your
 - Database name
 - Admin user credentials
 - VPC
 - Encryption settings
 - AWS-owned KMS by default
 - Audit logging
- Can manage snapshots & recovery points after creation

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "redshift-serverless:*",  
      "Resource": "*"  
    }  
  ]  
}
```

Resource Scaling in Redshift

- Capacity measured in Redshift Processing Units (RPU's)
- You pay for RPU-hours (per second) plus storage
- Base RPU's
 - You can adjust base capacity
 - Defaults to AUTO
 - But you can adjust from 32-512 RPU's to improve query performance
- Max RPU's
 - Can set a usage limit to control costs
 - Or, increase it to improve throughput

**redshift vs redshfit server less **

Redshift Serverless

- Does everything Redshift can, except:
 - Redshift Spectrum
 - Parameter Groups
 - Workload Management
 - AWS Partner integration
 - Maintenance windows / version tracks
- No public endpoints (yet)
 - Must access within a VPC

Redshift Serverless: Monitoring

- Monitoring views
 - SYS_QUERY_HISTORY
 - SYS_LOAD_HISTORY
 - SYS_SERVERLESS_USAGE
 - ...and many more
- CloudWatch logs
 - Connection & user logs enabled by default
 - Optional user activity log data
 - Under /aws/redshift/serverless/
- CloudWatch metrics
 - QueriesCompletedPerSecond, QueryDuration, QueriesRunning, etc.
 - Dimensions: DatabaseName, latency (short/medium/long), QueryType, stage

