🚦 Optimizing Query Performance in Amazon Redshift by Resolving Data Skew

🎯 Project Objective:

To simulate and resolve uneven load distribution across nodes in an Amazon Redshift provisioned cluster by optimizing table distribution styles.

🧩 Problem Scenario:

A corporation using a Redshift provisioned cluster (4-node `ra3.4xlarge`) with `DISTSTYLE EVEN` observed **poor query performance during peak loads**. Monitoring revealed that one node consistently experienced **high CPU utilization**, while others were **underutilized**. This indicated **data skew** and poor slice-level balance.

The goal: **Redistribute data uniformly** across all nodes using appropriate distribution styles.

🔬 Phase 1: Problem Simulation

1️⃣ Cluster Setup

- ✅ Created a Redshift provisioned cluster with **4 ra3.4xlarge nodes**

2️⃣ Dataset Selection

- ✅ Used NYC Yellow Taxi trip data from **Kaggle**
- ✅ Selected a small sample dataset (<50MB) to simulate skew behavior

3️⃣ S3 Upload

- ✅ Uploaded CSV to:

  `s3://nyc-taxi-dataset-bucket/yellow_tripdata_2015-01.csv`

4️⃣ Redshift Table Creation

- ✅ Created table `trips_even` using `DISTSTYLE EVEN`:

```
CREATE TABLE trips_even (
  VendorID INT,
  tpep_pickup_datetime TIMESTAMP,
  tpep_dropoff_datetime TIMESTAMP,
```

```
  passenger_count INT,
  trip_distance FLOAT,
  pickup_longitude FLOAT,
  pickup_latitude FLOAT,
  RateCodeID INT,
  store_and_fwd_flag VARCHAR(1),
  dropoff_longitude FLOAT,
  dropoff_latitude FLOAT,
  payment_type INT,
  fare_amount FLOAT,
  extra FLOAT,
  mta_tax FLOAT,
  tip_amount FLOAT,
  tolls_amount FLOAT,
  improvement_surcharge FLOAT,
  total_amount FLOAT
);
```

5 Data Load to Redshift

- ✅ Attached IAM Role: `AmazonS3ReadOnlyAccess`
- ✅ Loaded data using COPY:

```
COPY trips_even
FROM 's3://nyc-taxi-dataset-bucket/yellow_tripdata_2015-01.csv'
IAM_ROLE 'arn:aws:iam::535002864770:role/redshift_cluster_role'
CSV
IGNOREHEADER 1
TIMEFORMAT 'auto';
```

6 Query Workload Simulation

- ✅ Ran analytical aggregations and self-joins:

```
SELECT payment_type, COUNT(*), AVG(total_amount), SUM(tip_amount)
FROM trips_even
GROUP BY payment_type;

SELECT t1.payment_type, AVG(t1.total_amount + t2.total_amount)
FROM trips_even t1
JOIN trips_even t2
  ON t1.VendorID = t2.VendorID
WHERE t1.passenger_count = t2.passenger_count
GROUP BY t1.payment_type;
```

7 Monitoring

- ✅ Monitored using CloudWatch and Query Editor v2:
  - Query execution time
  - CPU utilization by node
  - Slice usage via:

```sql
SELECT slice, COUNT(*)
FROM stv_blocklist
GROUP BY slice;
```

🧠 Observation

- ⚠️ Uneven row distribution across slices
- 🔴 One slice (node) had **disproportionate number of blocks**
- 🔺 CloudWatch showed that one node had **spikes in CPU usage**
- ✅ `DISTSTYLE EVEN` resulted in **data skew** under this dataset pattern

🔧 Next Phase: Optimization Using DISTSTYLE KEY (Planned)

🎯 Plan:

1. Create new table `trips_key` with `DISTSTYLE KEY` using high-cardinality column `VendorID`
2. Reload same dataset into `trips_key`
3. Rerun same query workloads
4. Compare slice-level distribution and CPU balance across nodes

---

📌 Conclusion (Pending Completion of Phase 2)

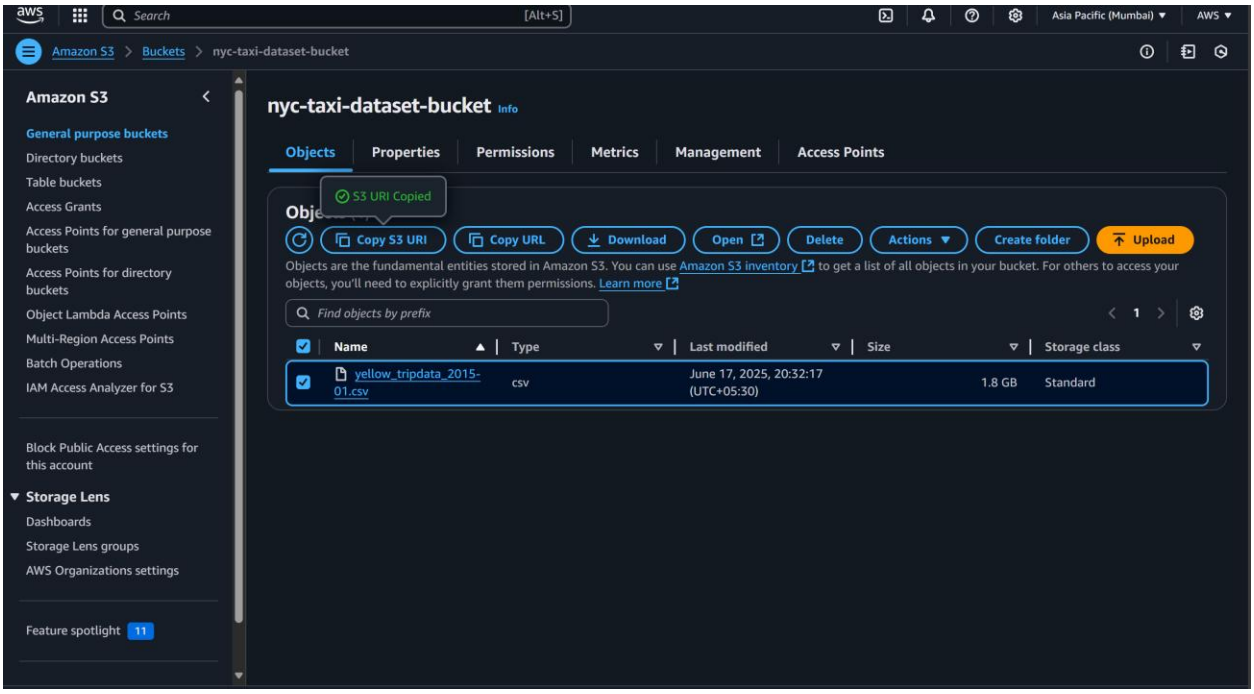
This simulation demonstrates that **DISTSTYLE EVEN is not always optimal**, especially when dealing with skewed keys. **Using a distribution key** aligned with query filters or joins is expected to improve both **execution time** and **node-level CPU utilization**.

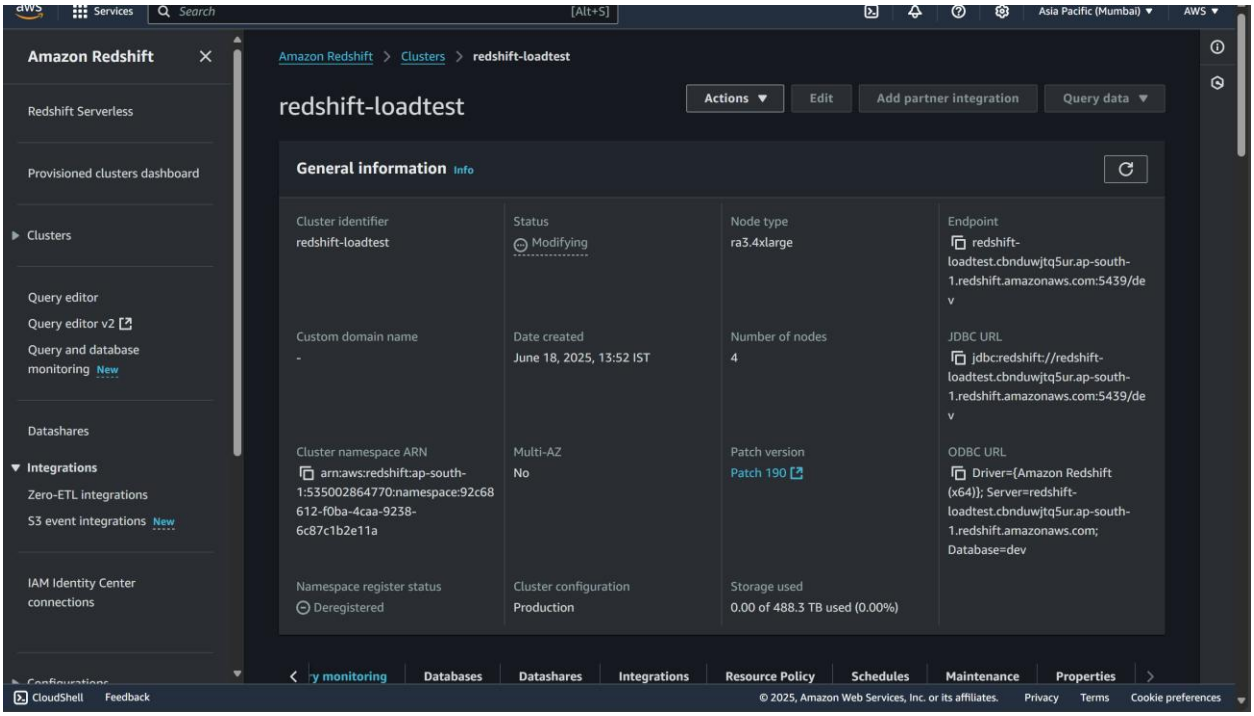
🛠️ Tools Used:

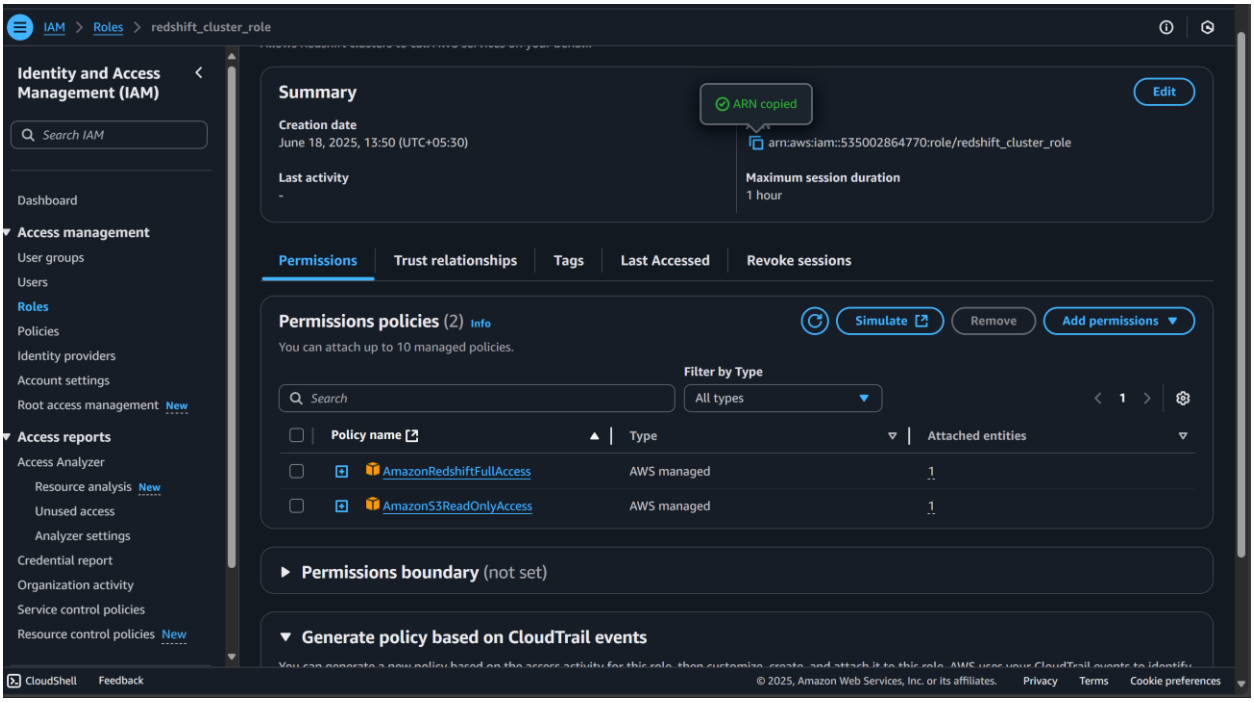
- Amazon Redshift (Provisioned Cluster)
- Amazon S3
- AWS CloudWatch
- Redshift Query Editor v2
- Kaggle NYC Taxi Dataset

📂 Workspace Output (Paste Your Results Below)

📝 Paste screenshots, metrics, SQL outputs, and graphs here:


Search
[Alt+S]
Asia Pacific (Mumbai)
AWS
Amazon S3 > Buckets > nyc-taxi-dataset-bucket
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points for general purpose buckets
Access Points for directory buckets
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
nyc-taxi-dataset-bucket Info
Objects
Properties
Permissions
Metrics
Management
Access Points
S3 URI Copied
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
yellow_tripdata_2015-01.csv
csv
June 17, 2025, 20:32:17 (UTC+05:30)
1.8 GB
Standard


Services
Search
[Alt+S]
Asia Pacific (Mumbai)
AWS
Amazon Redshift
Redshift Serverless
Provisioned clusters dashboard
Clusters
Query editor
Query editor v2
Query and database monitoring New
Datashares
Integrations
Zero-ETL integrations
S3 event integrations New
IAM Identity Center connections
Amazon Redshift > Clusters > redshift-loadtest
redshift-loadtest
Actions
Edit
Add partner integration
Query data
General information Info
Cluster identifier
redshift-loadtest
Status
Modifying
Node type
ra3.4xlarge
Endpoint
redshift-loadtest.cbnduwjtq5ur.ap-south-1.redshift.amazonaws.com:5439/dev
Custom domain name
-
Date created
June 18, 2025, 13:52 IST
Number of nodes
4
JDBC URL
jdbc:redshift://redshift-loadtest.cbnduwjtq5ur.ap-south-1.redshift.amazonaws.com:5439/dev
Cluster namespace ARN
arn:aws:redshift:ap-south-1:535002864770:namespace:92c68612-f0ba-4caa-9238-6c87c1b2e11a
Multi-AZ
No
Patch version
Patch 190
ODBC URL
Driver={Amazon Redshift (x64)}; Server=redshift-loadtest.cbnduwjtq5ur.ap-south-1.redshift.amazonaws.com; Database=dev
Namespace register status
Deregistered
Cluster configuration
Production
Storage used
0.00 of 488.3 TB used (0.00%)
Databases
Datashares
Integrations
Resource Policy
Schedules
Maintenance
Properties
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


IAM > Roles > redshift_cluster_role
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Resource control policies New
Summary
ARN copied
Edit
Creation date
June 18, 2025, 13:50 (UTC+05:30)
arn:aws:iam::535002864770:role/redshift_cluster_role
Last activity
-
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (2) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
Search
All types
1
Policy name
Type
Attached entities
AmazonRedshiftFullAccess
AWS managed
1
AmazonS3ReadOnlyAccess
AWS managed
1
Permissions boundary (not set)
Generate policy based on CloudTrail events
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences