
AWS
re:Invent

ANT404

Deep dive on best practices for Amazon Redshift

Harshida Patel

Sr. Analytics Specialist SA

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What you will learn

Amazon Redshift architecture

Amazon Redshift Advisor

Automatic table optimization

Concepts to fine-tune

Data ingestion

Workload management (WLM)

Cluster resize



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift architecture



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Cloud

Amazon Simple Workflow Service

Amazon VPC

AWS Identity and Access Management

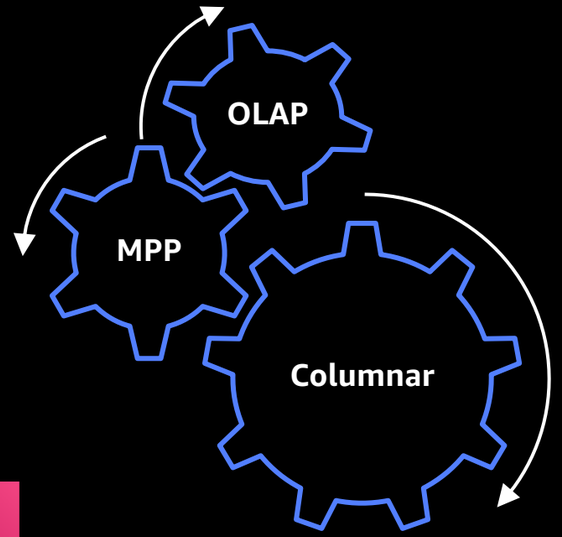
Amazon EC2

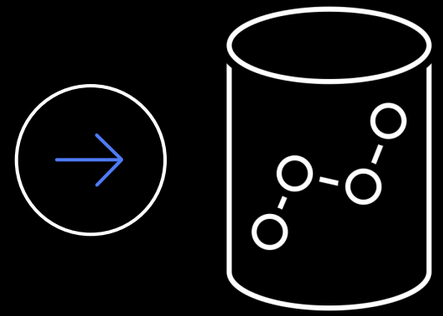
AWS Lambda

Amazon SageMaker

Amazon Redshift
seamless data lake integration







Amazon Redshift
fully managed service

Amazon Simple Storage Service

AWS Key Management Service

Amazon Route 53

Amazon CloudWatch

Amazon Redshift architecture

Massively parallel, shared nothing columnar architecture

Leader node

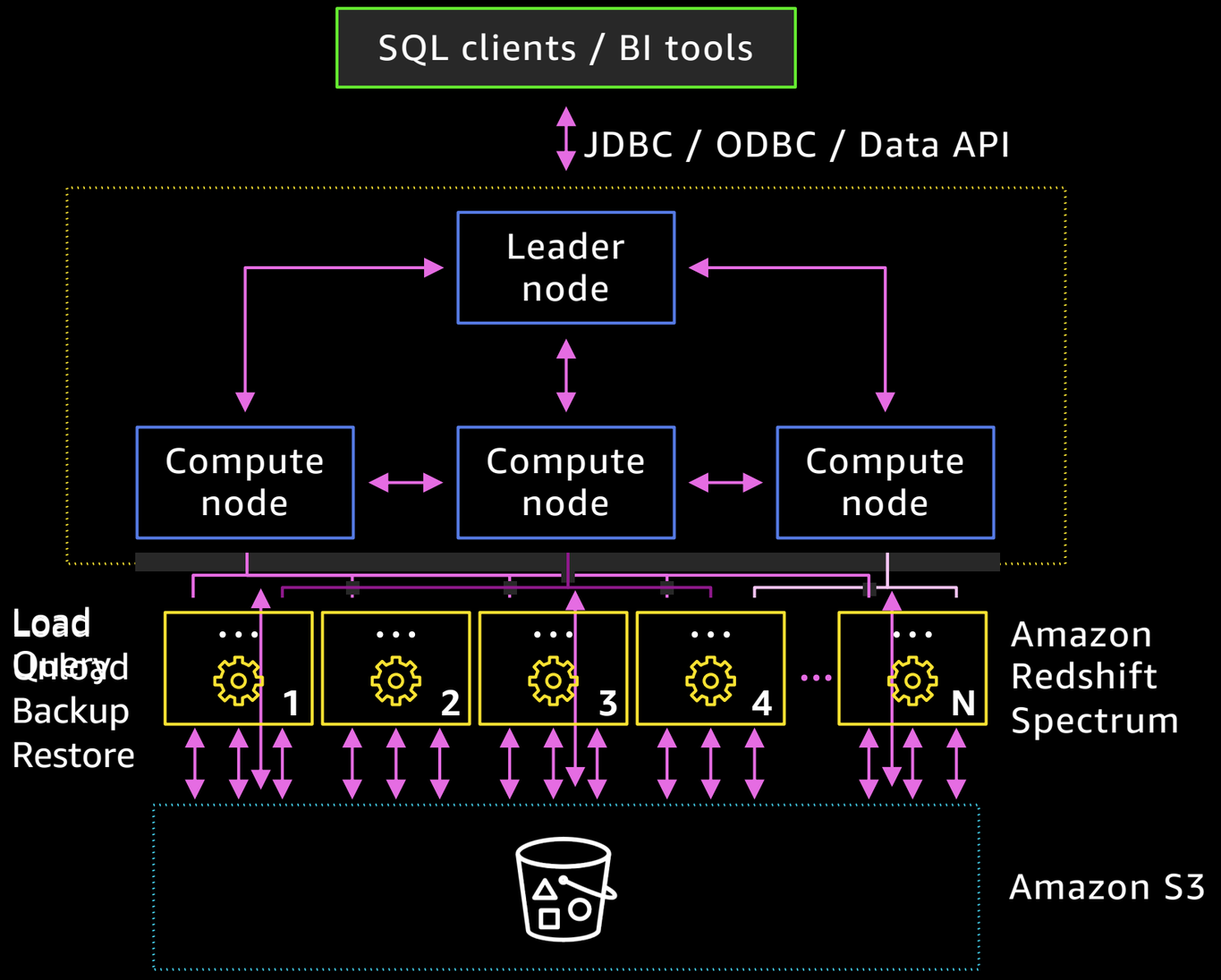
- SQL endpoint
- Stores metadata
- Coordinates parallel SQL processing

Compute nodes

- Local, columnar storage
- Executes queries in parallel
- Load, unload, backup, restore

Amazon Redshift Spectrum nodes

- Execute queries directly against Amazon Simple Storage Service (Amazon S3)



Amazon Redshift evolving architecture

Massively parallel, shared nothing columnar architecture

Leader node

Compute nodes

Amazon Redshift Spectrum nodes

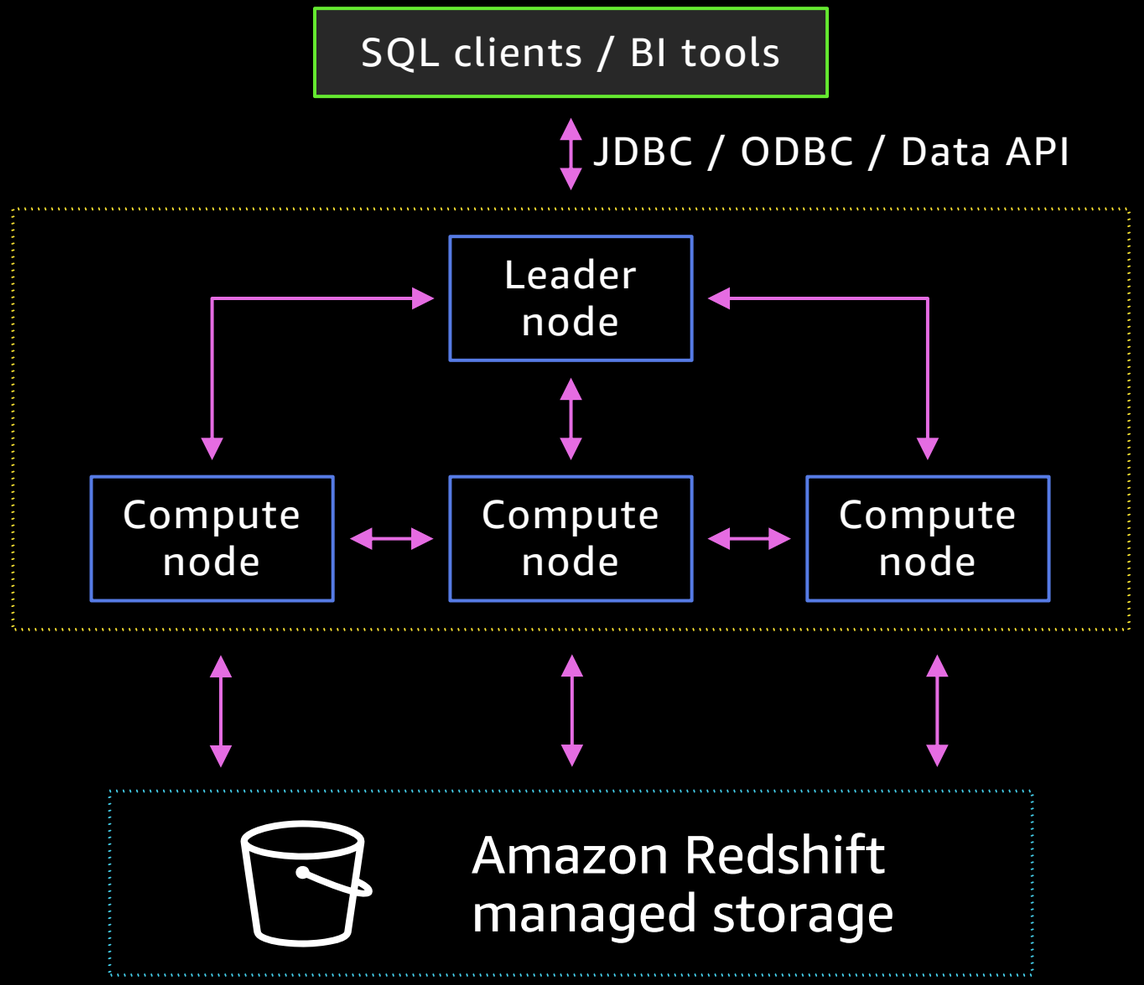
Amazon Redshift managed storage

Pay separately for storage and compute

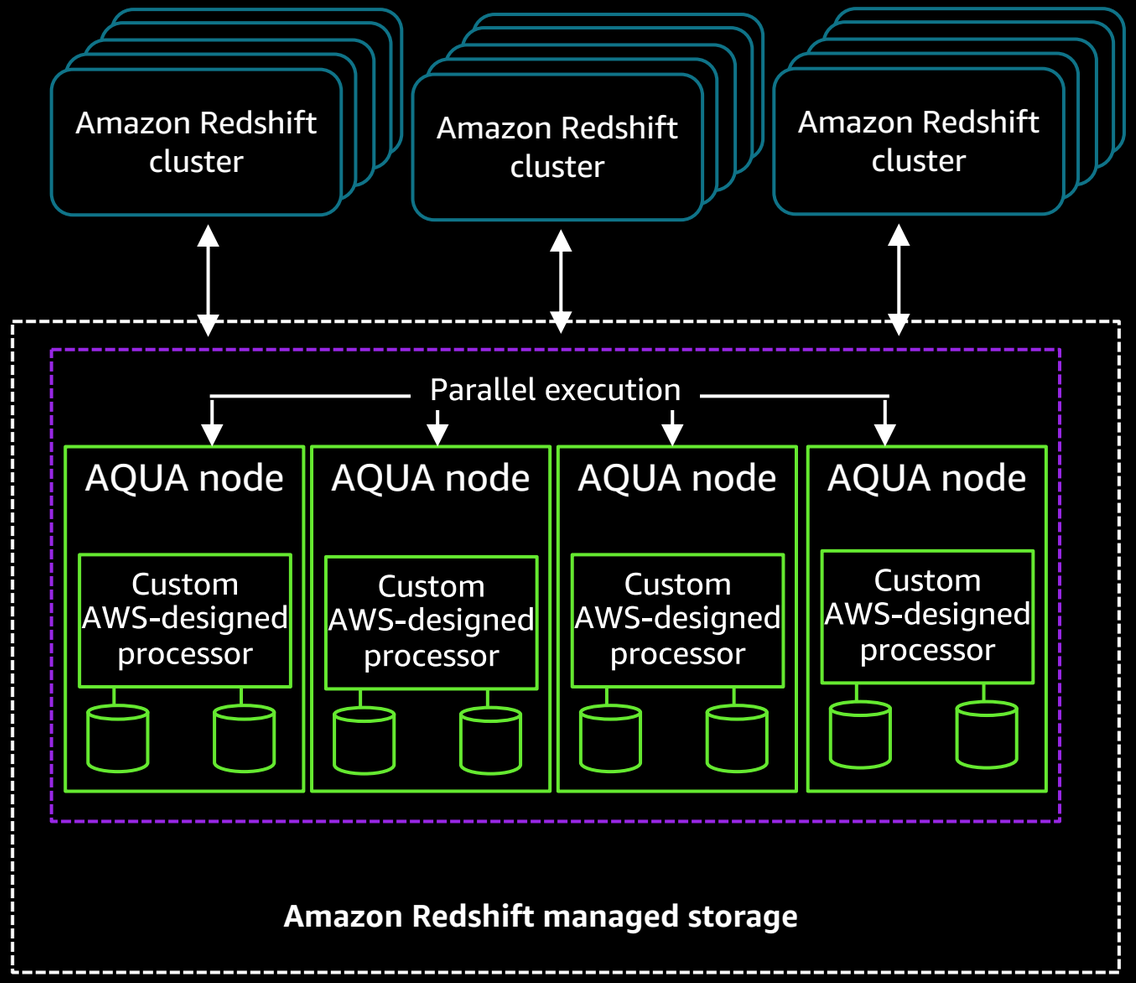
Large, high-speed SSD backed cache

Automatic scaling (up to 128 TB / instance)

Supports up to 16 PB of cluster storage



Advanced Query Accelerator (AQUA) for Amazon Redshift



New distributed and hardware-accelerated caching layer

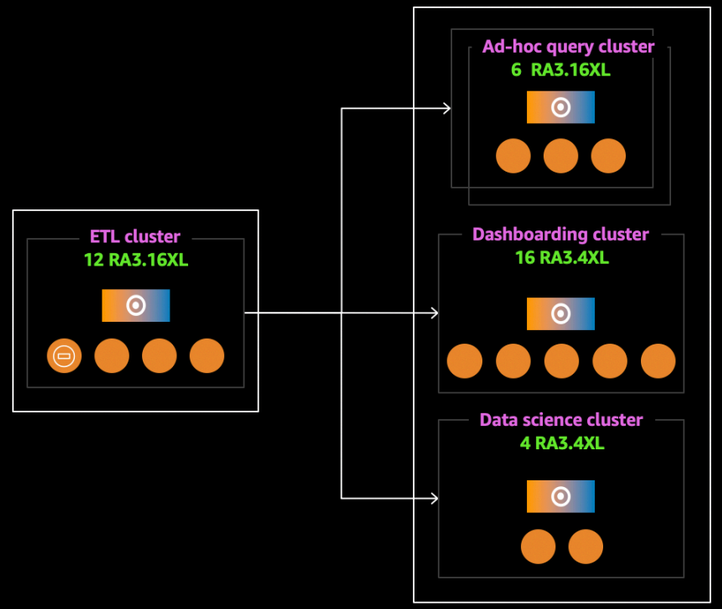
AQUA nodes with custom AWS-designed analytics processors to make operations (compression, encryption, filtering, and aggregations) faster than traditional CPUs

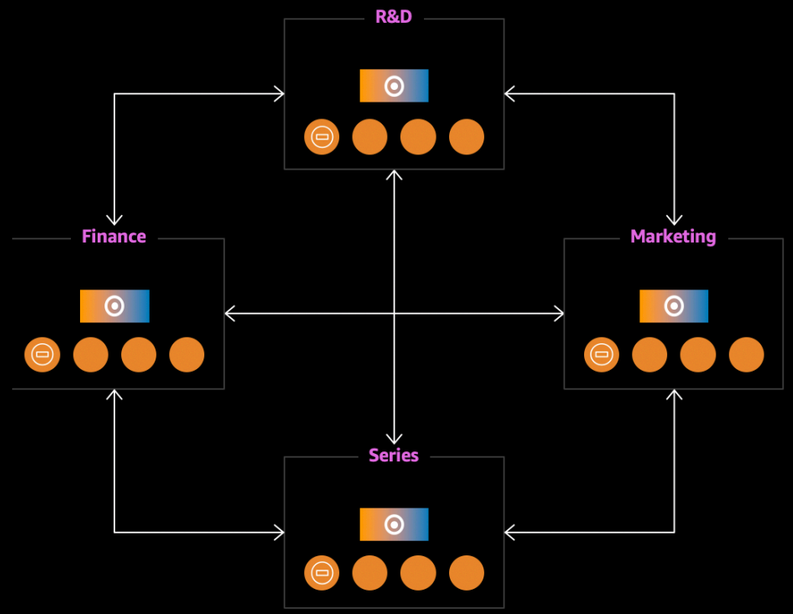
Included with the RA3 instance type at no additional cost

Data sharing

PREVIEW

A SECURE AND EASY WAY TO SHARE DATA ACROSS AMAZON REDSHIFT CLUSTERS





- Instant, granular, high-performance data access without data copies / movement
- Live and consistently updating views of data across all consumers
- Secure and governed collaboration within and across organizations and with external parties

- Workloads accessing shared data are isolated from each other
- Use cases: Cross-group collaboration and sharing, workload isolation and chargeability, data as a service
- Sharing to other AWS analytic services – coming soon

Amazon Redshift node types

RA3: Separation of compute and storage with Amazon Redshift managed storage

Solid-state disks + Amazon S3

Instance type	CPUs	Memory
RA3 XLPlus	4	32 GB
RA3 4xlarge	12	96 GB
RA3 16xlarge	48	384 GB

- RA3.4xlarge and RA3.16xlarge managed storage quota per node 128 TB
- RA3XLPlus managed storage quota per node 32 TB

Dense compute – DC2: Solid-state disks

Dense storage – DS2: Magnetic disks

Instance type	CPUs	Memory	Disk size
DC2 large	2	16 GB	160 GB
DC2 8xlarge	32	244 GB	2.56 TB
DS2 xlarge (legacy)	4	32 GB	2 TB
DS2 8xlarge (legacy)	36	244 GB	16 TB

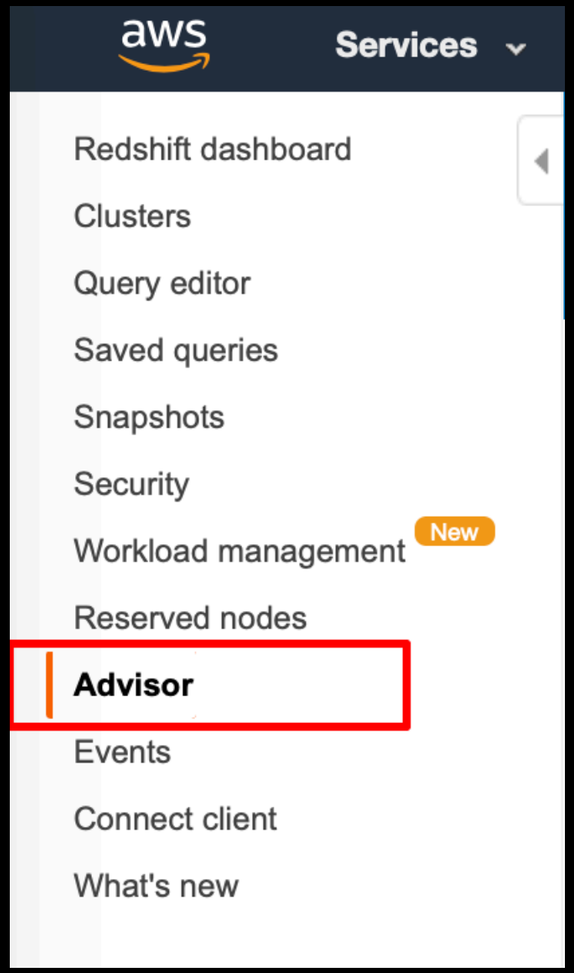
Amazon Redshift Advisor



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift Advisor



Amazon Redshift Advisor available in Amazon Redshift console

Runs daily scanning operational metadata

Observes with the lens of best practices

Provides tailored, high-impact recommendations to optimize your Amazon Redshift cluster for performance and cost savings

Types of advisor recommendations

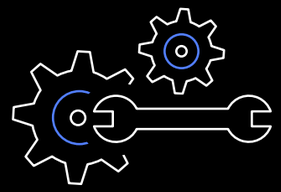


Data ingest

Speed up COPY (compression)

Skip compression analysis during COPY

Split Amazon S3 objects loaded by COPY



Query tuning

Reallocate WLM memory

Enable Short Query Acceleration (SQA)



Table optimization

Alter distribution keys on tables

Alter sort keys on tables

Column encoding



Cost savings

Delete unused clusters

Automatic table optimization

INTRODUCING AN AUTO PROPERTY FOR TABLES

Continuously scans workload patterns

Automatically adjusts sort and distkey over time to account for changes in workload

Can be enabled or disabled per table

svv_alter_table_recommendations logs the recommended changes

svl_auto_worker_action logs audit trail of changes



Distribution style



Sort keys

Concepts to fine-tune



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Terminology and concepts: Compression

Goals

Allow more data to be stored within an Amazon Redshift cluster

Improve query performance by decreasing I/O

Impact

Allows two to four times more data to be stored within the cluster

CREATE TABLE assigns default encodings, when no encoding is specified

By default, COPY automatically analyzes and compresses data on first load into an empty table

ANALYZE COMPRESSION is a built-in command that will find the optimal compression for each column on an existing table

ALTER TABLE <table-name> ALTER COLUMN <column-name> ENCODE <encode-type>

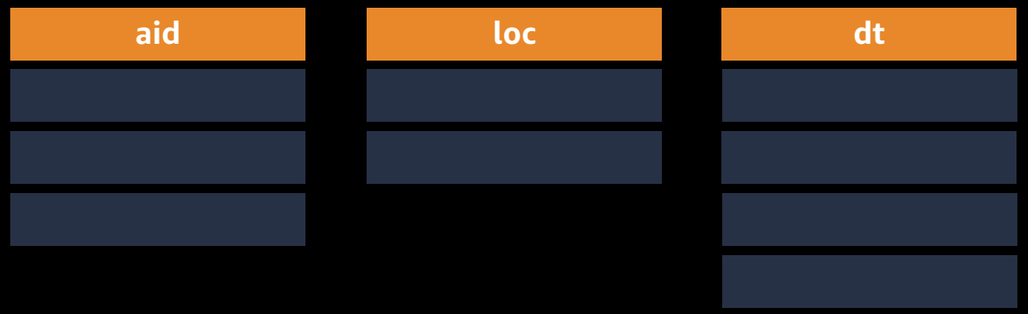
Compression: Example

```
CREATE TABLE deep_dive (
     aid  INT
    ,loc  CHAR(3)
    ,dt   DATE      );
```

aid	loc	dt
1	SFO	2020–10–20
2	JFK	2020–10–20
3	SFO	2020–04–01
4	JFK	2020–05–14

Compression: Example

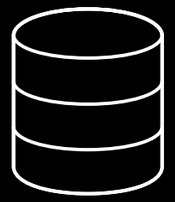
```
CREATE TABLE deep_dive (
     aid  INT        ENCODE AZ64
    ,loc  CHAR(3)    ENCODE LZO
    ,dt   DATE       ENCODE AZ64
);
```



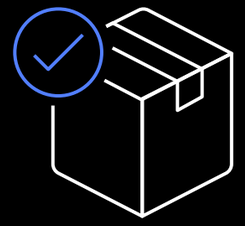
aid	loc	dt
1	SFO	2020–10–20
2	JFK	2020–10–20
3	SFO	2020–04–01
4	JFK	2020–05–14

- More efficient compression is due to storing the same data type in the columnar architecture
- Columns grow and shrink independently
- Reduces storage requirements
- Reduces I/O

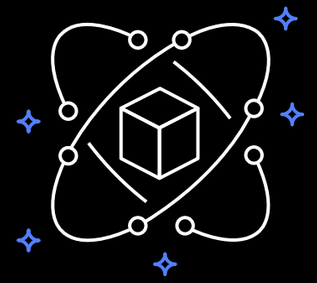
Terminology and concepts: Blocks



Column data is persisted to 1 MB immutable blocks



Blocks are individually encoded with 1 of 13 encodings



A full block can contain millions of values

Terminology and concepts: Zone maps

Goal

Eliminates unnecessary I/O

In-memory block metadata

- Contains per-block min and max values
- All blocks automatically have zone maps
- Effectively prunes blocks that cannot contain data for a given query

Terminology and concepts: Data sorting

Goal

Make queries run faster by increasing the effectiveness of zone maps and reducing I/O

Impact

Enables range-restricted scans to prune blocks by leveraging zone maps

Achieved with the table property
SORTKEY defined on one or more columns

Optimal sort key is dependent on:

- Query patterns
- Business requirements
- Data profile

Zone maps and sorting: Example

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2020';
```

Unsorted table



MIN: 01-JUNE-2020
MAX: 20-JUNE-2020



MIN: 08-JUNE-2020
MAX: 30-JUNE-2020



MIN: 12-JUNE-2020
MAX: 20-JUNE-2020



MIN: 02-JUNE-2020
MAX: 25-JUNE-2020

Sorted by date



MIN: 01-JUNE-2020
MAX: 06-JUNE-2020



MIN: 07-JUNE-2020
MAX: 12-JUNE-2020



MIN: 13-JUNE-2020
MAX: 21-JUNE-2020



MIN: 21-JUNE-2020
MAX: 30-JUNE-2020

Data distribution

Distribution style is a table property that dictates how that table's data is distributed throughout the cluster

KEY: Value is hashed; same value goes to same location (slice)

ALL: Full table data goes to the first slice of every node

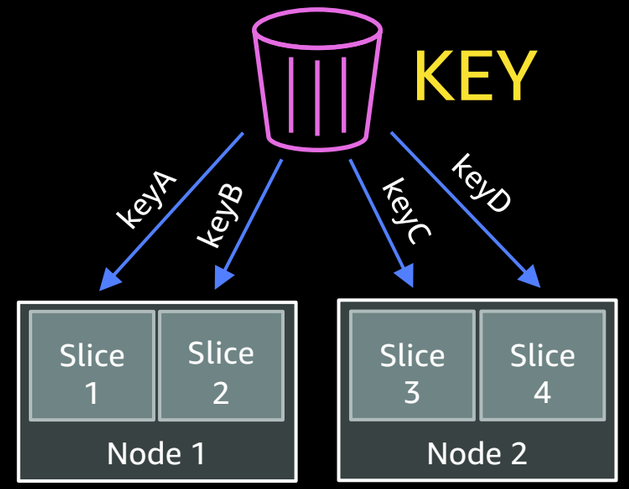
EVEN: Round robin

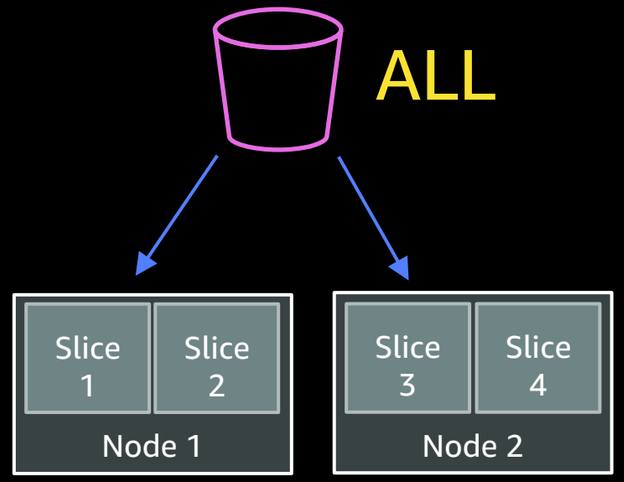
AUTO: Automatically manages diststyle

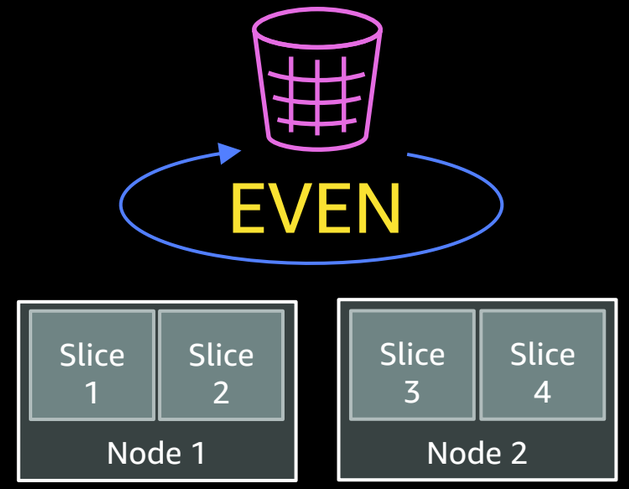
Goals

Distribute data evenly for parallel processing

Minimize data movement during query processing

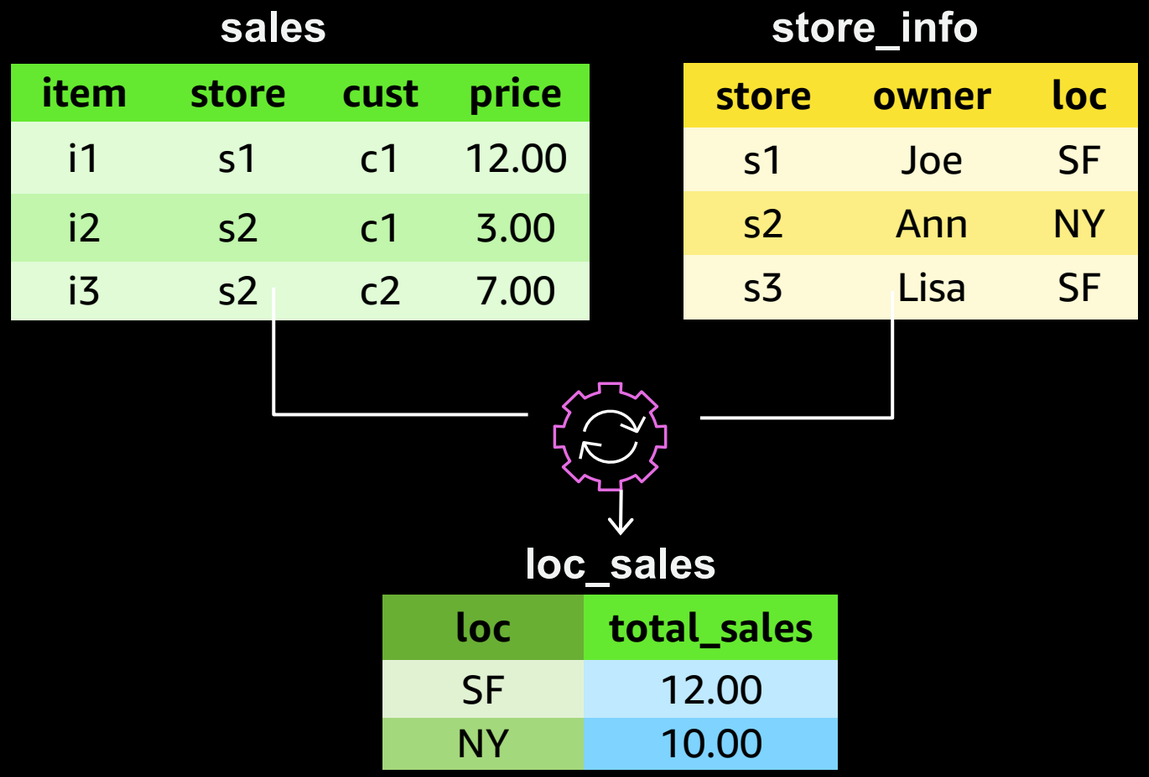






Terminology and concepts: Materialized views

- Created from a single or multiple base tables
- **Projections** – Select subset of columns
- **Filters** - Select subset of rows
- **INNER JOINs** from multiple base tables
- **Aggregates, GROUPBY, HAVING**
  - Summarize column values from matching rows
  - Associative aggregate functions
- Precomputed results **persisted** in the materialized view
- Base table changes incrementally refreshed
- Can use different distribution style and sort key than base tables



Best practices: Table design summary

Use the default auto table optimization

Materialized views to physicalize complex queries

If you choose to fine-tune:

- Add compression to columns
  - Use AZ64 where possible, ZSTD / LZO for most (VAR)CHAR columns
- Add sort keys on the columns that are frequently filtered on
- Co-locate large tables using DISTSTYLE KEY using high cardinality column
- Avoid distribution keys on temporal columns

Data ingestion



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

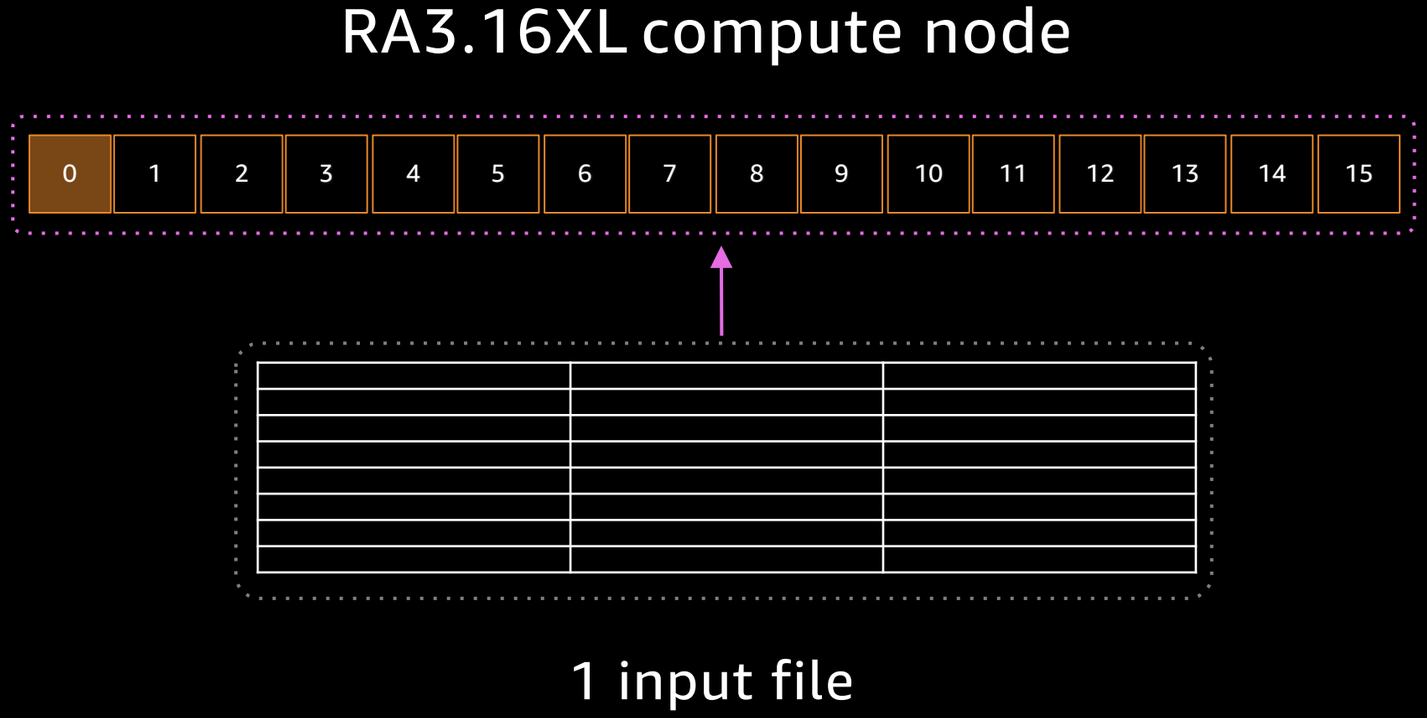
Data ingestion: COPY statement

Ingestion throughput

Each slice's query processors can load one file at a time:

- Streaming decompression
- Parse
- Distribute
- Write

Realizing only partial node usage as 6.25% of slices are active

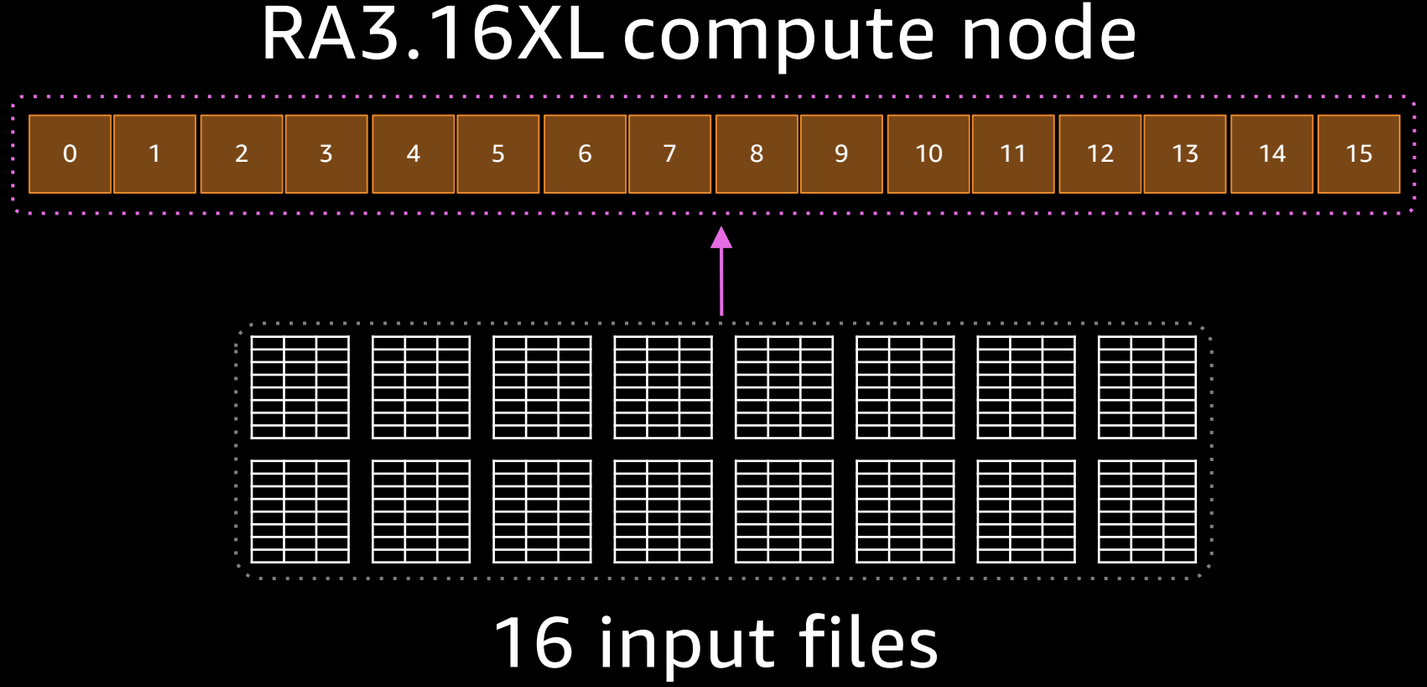


Data ingestion: COPY statement

Number of input files should be a multiple of the number of slices

Splitting the single file into 16 input files, all slices are working to maximize ingestion performance

COPY continues to scale linearly as you add nodes



Recommendation is to use delimited files – 1 MB to 1 GB after gzip compression

Data ingestion: Amazon Redshift Spectrum

Use INSERT INTO SELECT from external Amazon S3 tables

- Aggregate incoming data
- Select subset of columns and/or rows
- Manipulate incoming column data with SQL
- Load data in alternative file formats: Amazon ION, Grok, Parquet, ORC, RCFile, and Sequence

Best practices

- Save cluster resources for querying and reporting rather than on ELT
- Filtering / aggregating incoming data can improve performance over COPY

Design considerations

- Repeated reads against Amazon S3 are not transactional
- $5/TB of (compressed) data scanned

Design considerations: Data ingestion

Designed for large writes

- Batch processing system, optimized for processing massive amounts of data
- 1 MB size plus immutable blocks means that we clone blocks on write so as not to introduce fragmentation
- Small write (~1–10 rows) has similar cost to a larger write (~100K rows)

UPDATE and DELETE

- Immutable blocks means that we only logically delete rows on UPDATE or DELETE
- (AUTO) VACUUM to remove ghost rows from table

(AUTO) VACUUM

The VACUUM process runs either manually or automatically in the background

Goals

- VACUUM will remove rows that are marked as deleted
- VACUUM will globally sort tables
  - For tables with a sort key, ingestion operations will locally sort new data and write it into the unsorted region

Best practices

- VACUUM should be run only as necessary
- For the majority of workloads, AUTO VACUUM DELETE will reclaim space and AUTO TABLE SORT will sort the needed portions of the table
- In cases where you know your workload – VACUUM can be run manually
- Use VACUUM BOOST at off-peak times (blocks deletes), which is as quick as Deep Copy

(AUTO) ANALYZE

The ANALYZE process collects table statistics for optimal query planning

In the vast majority of cases, AUTO ANALYZE automatically handles statistics gathering

Best practices

ANALYZE can be run periodically after ingestion on just the columns that WHERE predicates are filtered on

Utility to manually run VACUUM and ANALYZE on all the tables in the cluster: https://bit.ly/34ZR3PP

Workload management



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workload management

Allows for the separation of different query workloads

Goal

Prioritize important queries

Throttle / abort less important queries

Terminology and concepts: WLM attributes

SQA

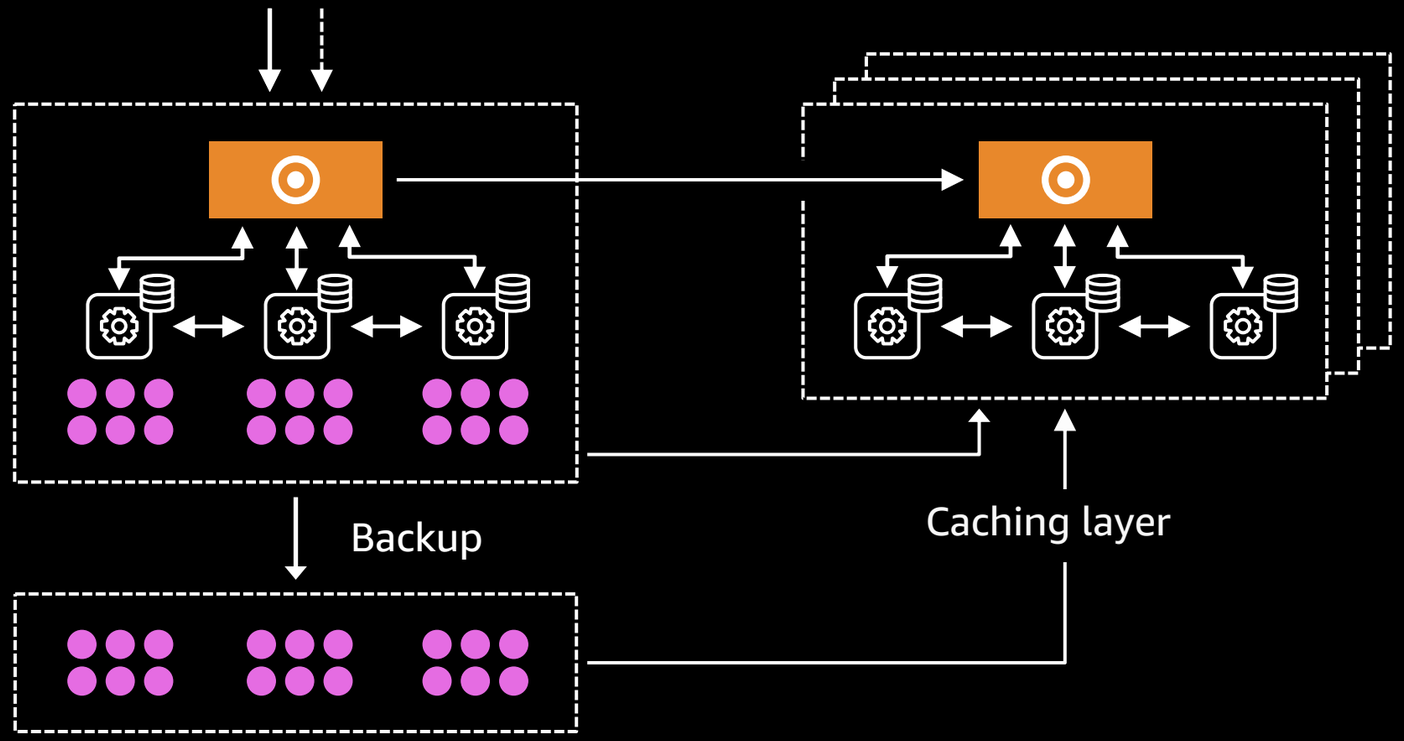
Automatically detect short-running queries and runs them within the short query queue, if queuing occurs

Concurrency scaling

When queues are full, queries are routed to transient Amazon Redshift clusters

Concurrency scaling

Amazon Redshift automatically adds transient clusters, in seconds, to serve sudden spike in concurrent requests with consistently fast performance



For every 24 hours that your main cluster is in use, you accrue a one-hour credit for Concurrency Scaling

This means that Concurrency Scaling is free for > 97% of customers

How it works:

1. All queries go to the leader node; user only sees less wait for queries
2. When queries in designated WLM queue begin queuing, Amazon Redshift automatically routes them to the new clusters, enabling Concurrency Scaling automatically
3. Amazon Redshift automatically spins up a new cluster, processes waiting queries, and automatically shuts down the Concurrency Scaling cluster

Query monitoring rules (QMR)

Extension of WLM

Allow the automatic handling of runaway (poorly written) queries

Rules applied to a WLM queue allow queries to be:

- LOGGED
- ABORTED
- HOPPED

Goals

- Protect against wasteful use of the cluster
- Log resource-intensive queries

Automatic WLM

Allows for prioritization of different query workload

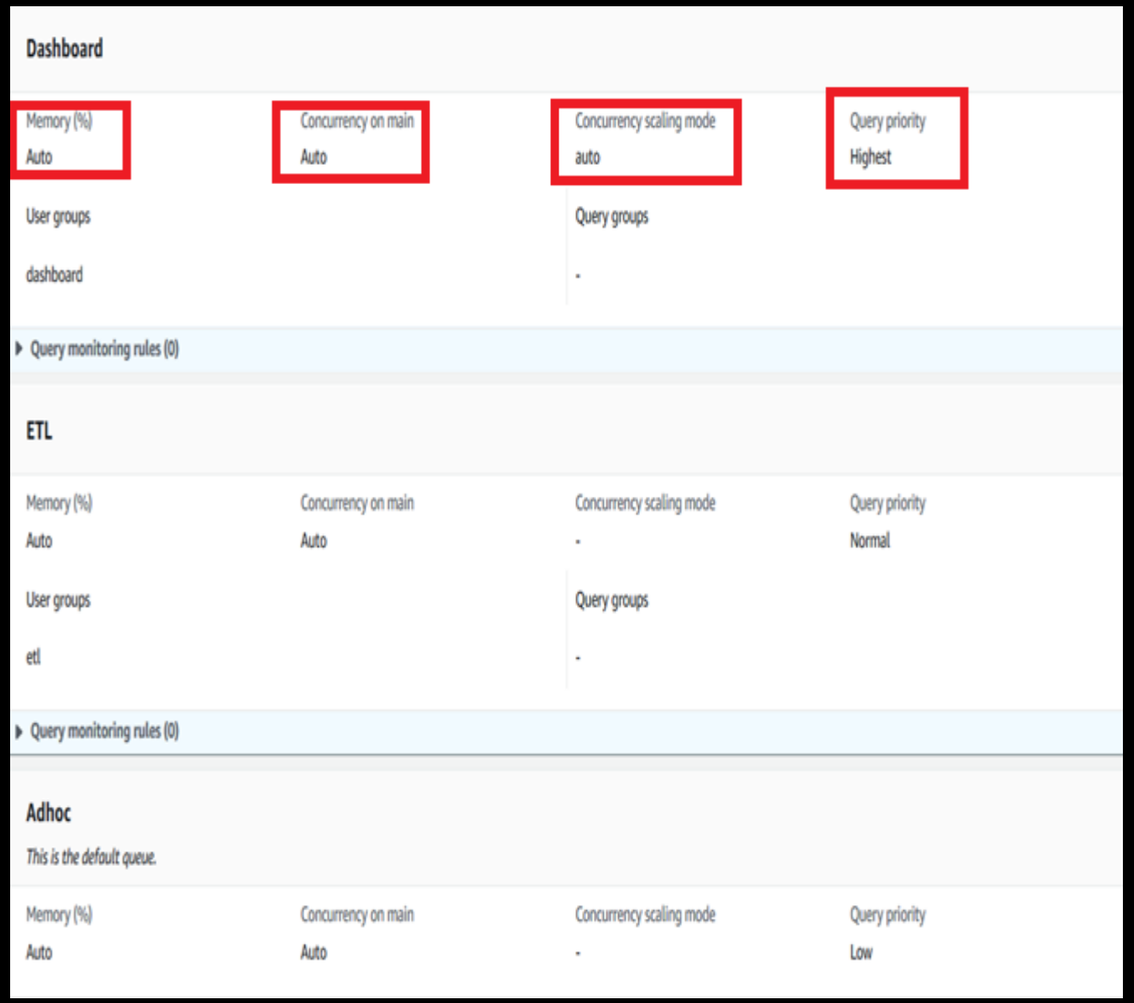
Goals

- Simplify WLM

Automatically controls concurrent number of executing queries

Automatically divides cluster memory

SQA enabled by default



WLM: Example manual WLM

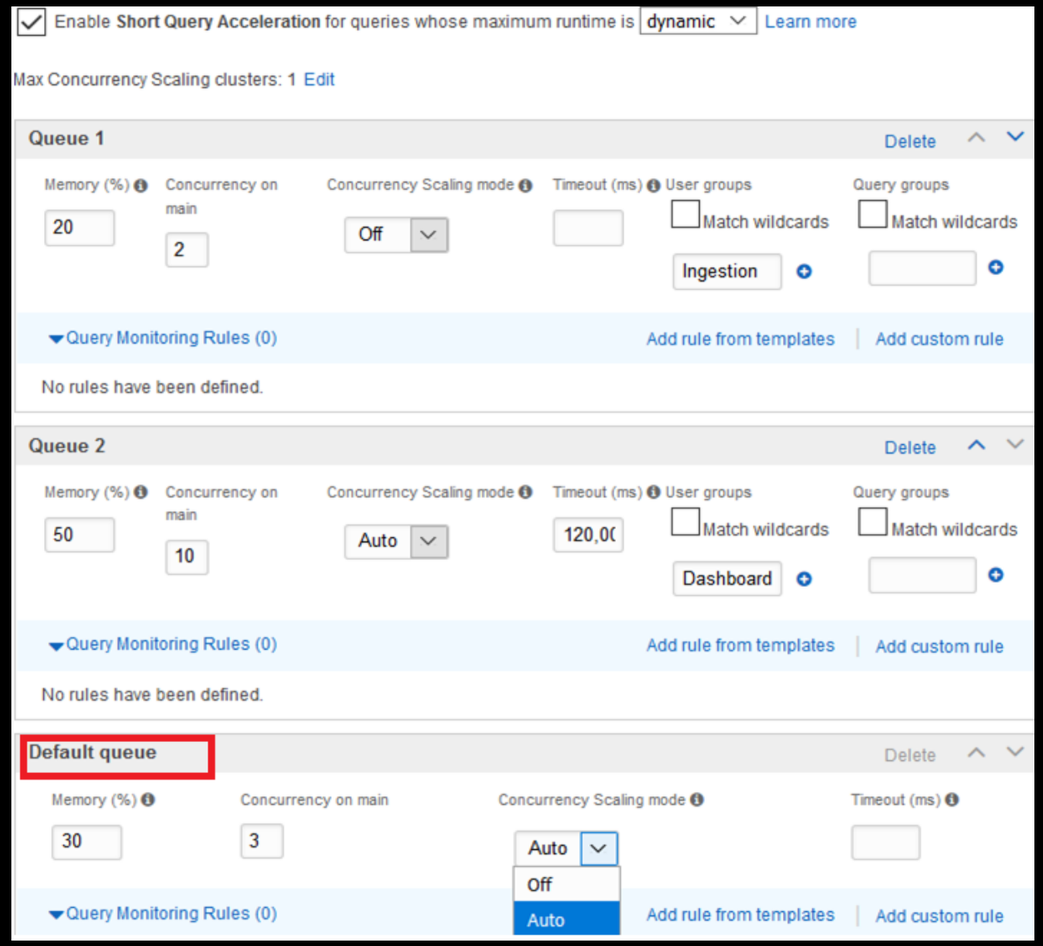
Enable SQA

Hidden superuser queue can be used by admins, manually switched into:

```
SET query_group TO
'superuser'
```

The superuser queue has a single slot, the equivalent of 5–7% memory allocation, and no timeout

Create a queue for each workload type:



Best practices: WLM and QMR

Use Auto WLM if your workload is highly unpredictable, or you are using the old default WLM

Use QMR rather than WLM to set query timeouts

Use QMR to log long-running queries

Use manual WLM if you understand your workload patterns or require throttling certain types of queries depending on the time of day

Use WLM to limit ingestion / ELT concurrency to two to three

To maximize query throughput, use WLM to throttle the number of concurrent queries to 15 or less

Save the superuser queue for administration tasks and canceling queries

Cluster resize



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resizing Amazon Redshift

Classic resize

- Data is transferred from old cluster to new cluster (within hours)
- Change node types

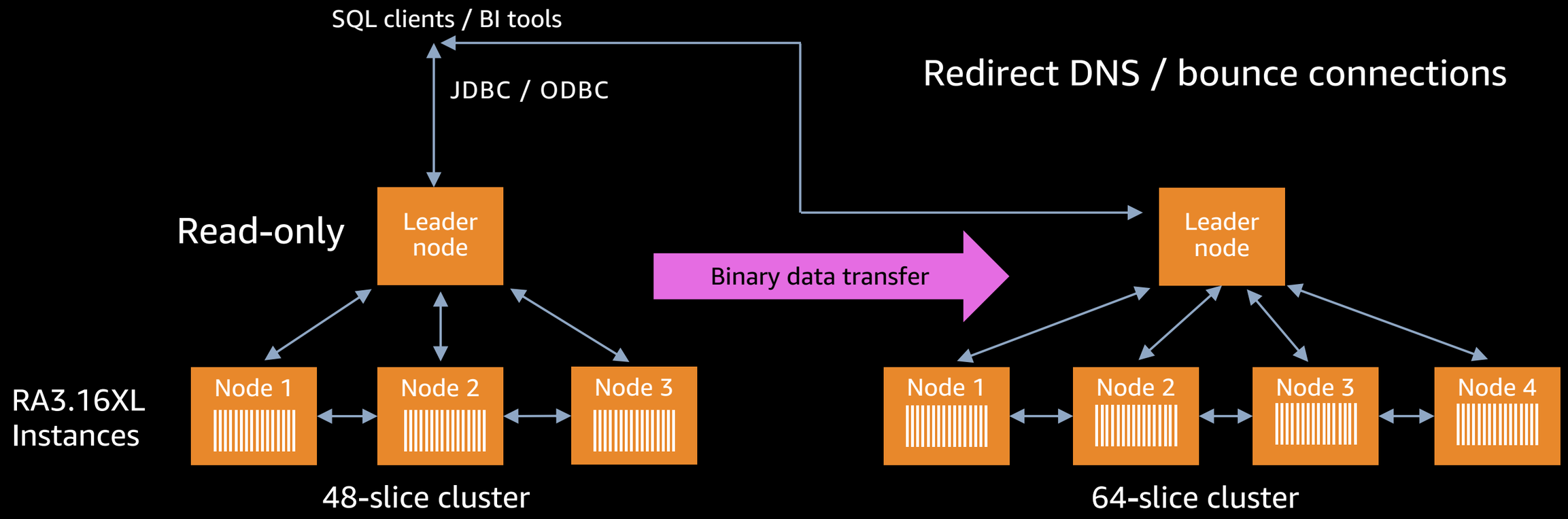
Elastic resize

- Nodes are added / removed to / from existing cluster (within minutes)
- Change node types

Cross-instance restore

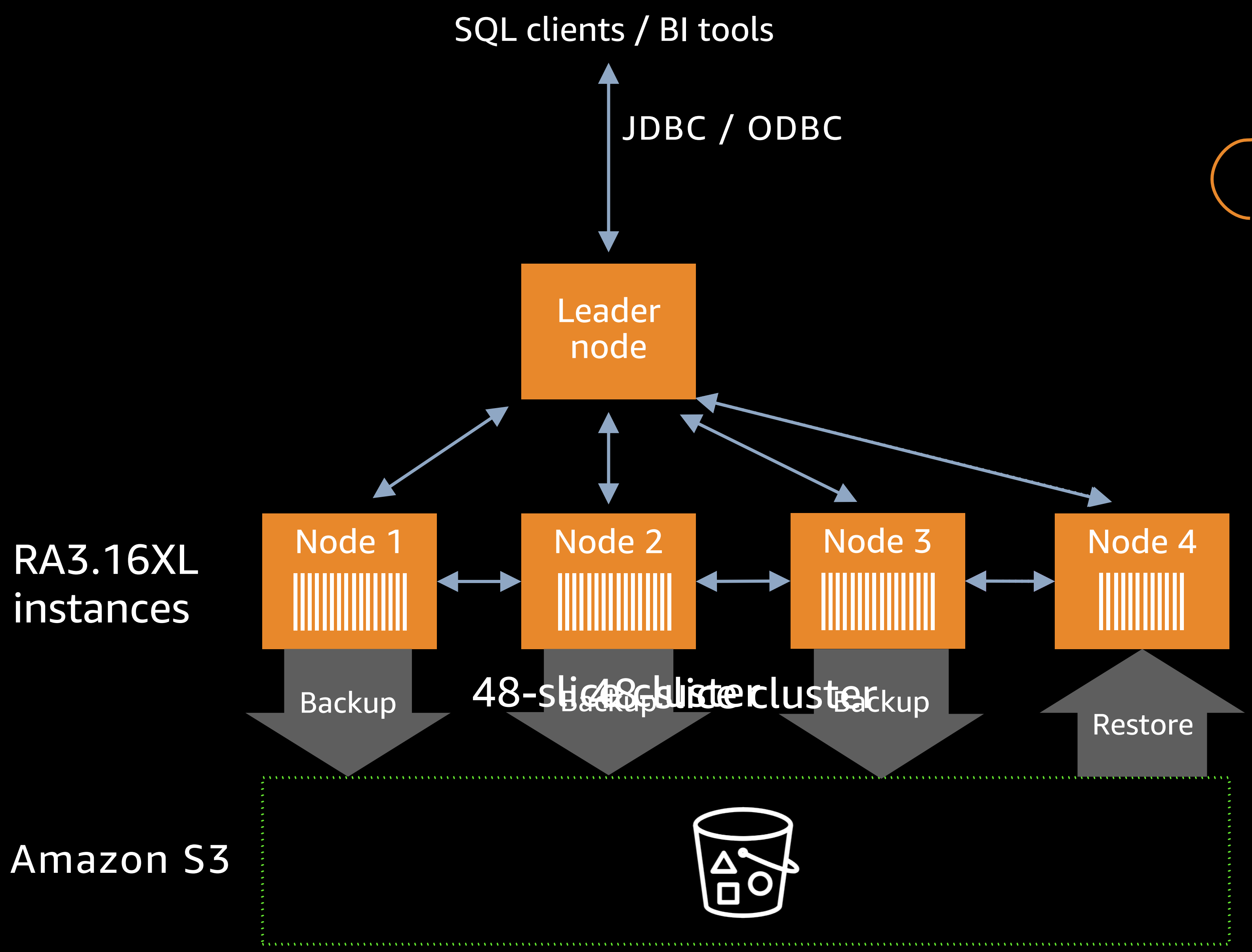
- Customers can now restore to new node types and counts

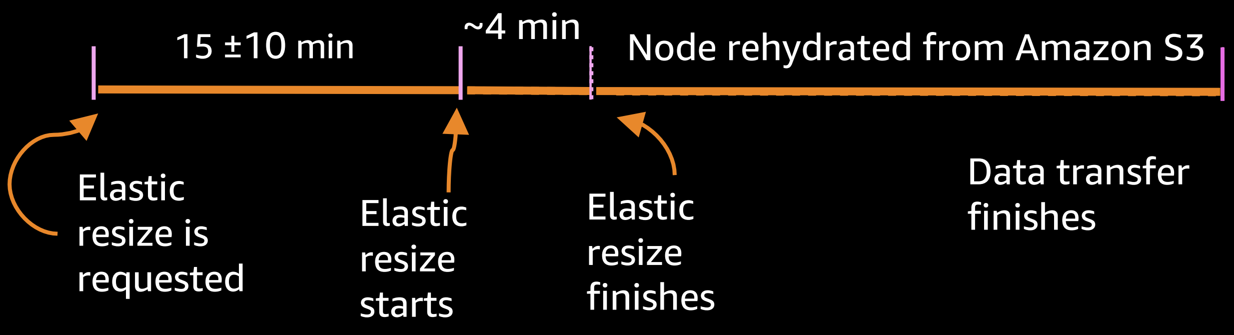
Classic resize



- Source cluster is placed into read-only mode during resize
- All data is copied and redistributed on the target cluster
- Allows for changing node types

Elastic resize





- Elastic resize starts; we take an automatic snapshot to Amazon S3 and provision the new node(s)
- In-flight queries/connections are on hold
- Cluster is fully available for reads and writes
- Some queries within transactions maybe rollback

When to use elastic vs. classic resize

	Elastic resize	Classic resize
Scale up and down for workload spikes		
Incrementally add / remove storage		
Change cluster instance type		
If elastic resize is not an option because of sizing limits		
Limited availability during resize	< 5 minutes (parked connections)	1–24 hours (read-only)

Additional resources



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS big data blog: Amazon Redshift

Amazon Redshift Engineering's Advanced Table Design Playbook

https://aws.amazon.com/blogs/big-data/amazon-redshift-engineerings-advanced-table-design-playbook-preamble-prerequisites-and-prioritization/

Zach Christopherson

Top 10 Performance Tuning Techniques for Amazon Redshift

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-techniques-for-amazon-redshift/

Matt Scaer, Manish Vazirani, and Tarun Chaudhary

Best Practices for Amazon Redshift Spectrum

https://aws.amazon.com/blogs/big-data/10-best-practices-for-amazon-redshift-spectrum/

Po Hong and Peter Dalton

Next steps

Already using Amazon Redshift?

Ask your account team for a free Amazon Redshift optimization session

Learn more about what's new

https://aws.amazon.com/redshift/whats-new/

Getting started with Amazon Redshift?

Ask your account team for a 10-minute demo

Learn about starting a proof of concept

https://docs.aws.amazon.com/redshift/proof-of-concept-playbook.html

Or request help from AWS

https://pages.awscloud.com/redshift-proof-of-concept-request

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.