



AWS re:Invent

ANT 413 - R

How to tame unpredictable analytics workloads with Amazon Redshift

Raj Sett

Database Engineer II

Amazon Redshift

Amazon Web Services

What will you learn?

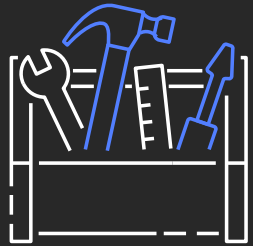
After this session, you will know how to

- Bring a new Amazon Redshift cluster up from a snapshot
- Configure automatic WLM with query priorities
- Alter query priorities on the fly for
 - A single query
 - The whole cluster
- Enable concurrency scaling to handle workload spikes

Demo: bring up a cluster

Automated cluster maintenance

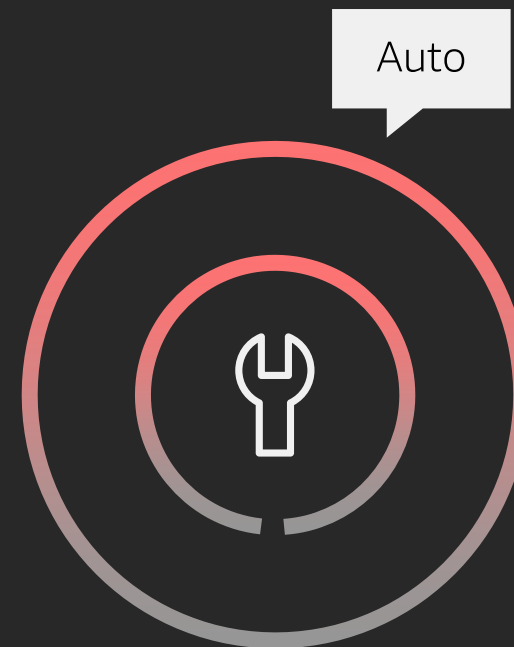
New



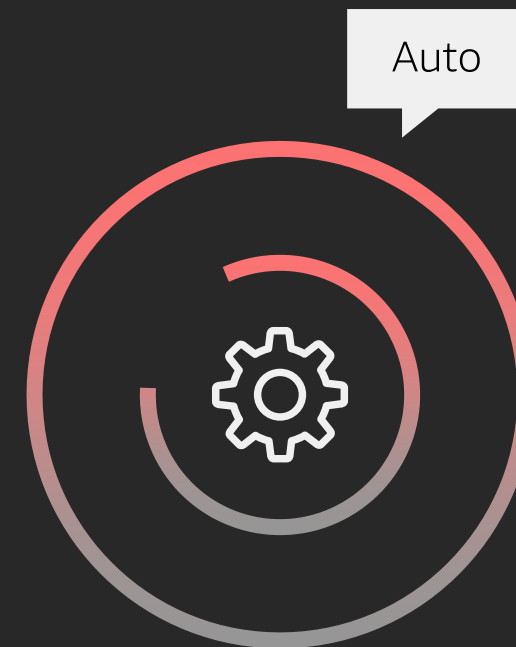
Maintenance processes like vacuum & analyze will automatically run in the background

Moving toward zero maintenance

Amazon Redshift automatically adjusts the WLM concurrency setting to deliver optimal throughput



Analyze



WLM
concurrency
setting



Vacuum

Automatic WLM: spend resources where needed

Jun '19

Auto WLM – Dynamic concurrency

- ML to decide *how many queries to process in parallel* to optimize **throughput and query run times**
- Minimum **waste** of compute resources

Sept '19

Auto WLM – Query priorities

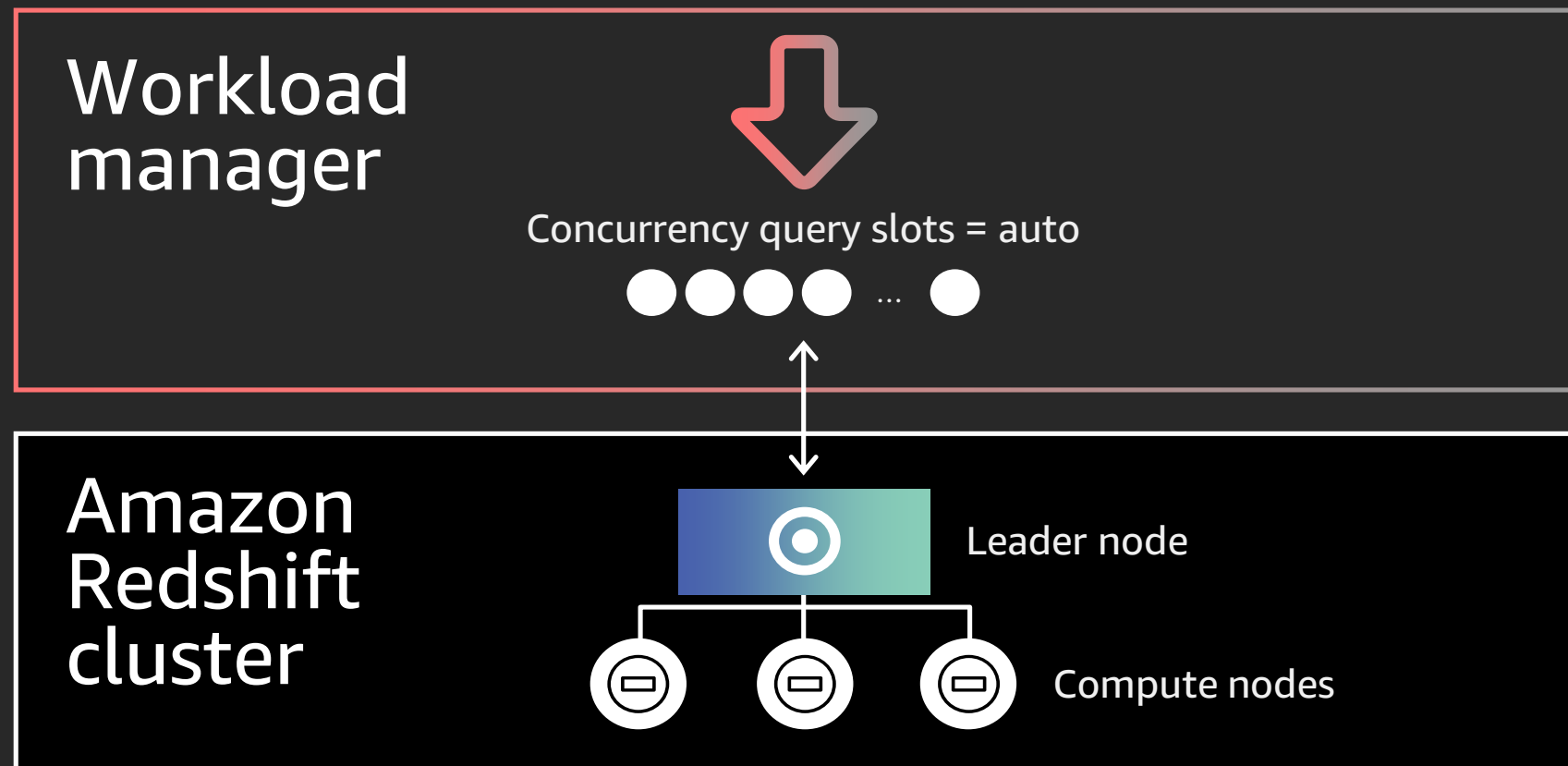
- Amazon Redshift manages query traffic and executes based **on your priority**
- Predictable **SLAs**
- Scale compute resources based on **priority**



Demo: configure auto WLM

Auto WLM – Dynamic concurrency

With dynamic concurrency, Amazon Redshift determines the number of parallel queries **automatically and dynamically to deliver optimal throughput** for changing workloads



Amazon Redshift monitors the query traffic and automatically determines the number of parallel slots to **optimize memory allocation to queries**

Amazon Redshift's WLM allows you to input the number of parallel queries that run on your cluster

What are query priorities?

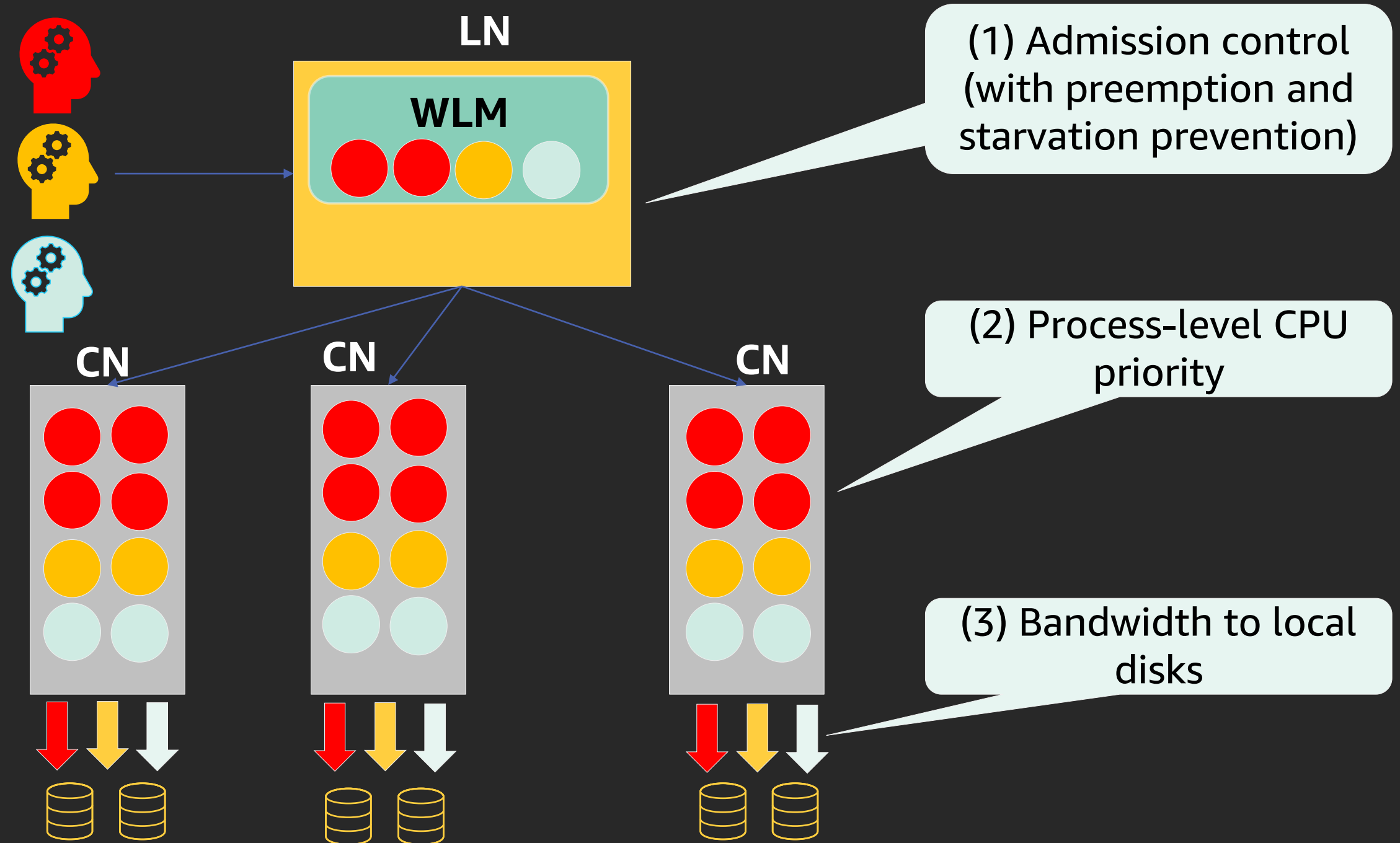
- Gives customers ability to **categorize workloads based on business needs** and assign priorities—e.g., ETL, daily reporting, dashboards, data science team
- Amazon Redshift manages assignment of resources to queries based on priorities
- **Orders of magnitude simpler than manual WLM**
- No static partitioning of memory between workloads

The screenshot displays the Amazon Redshift Query Monitoring console with three queues configured:

- Queue 1:** Priority is set to **Highest**, Memory (%) is **Auto**, Concurrency on main is **Auto**, Concurrency Scaling mode is **Auto**, User groups include **dashboard**, and Query groups are empty.
- Queue 2:** Priority is set to **Normal**, Memory (%) is **Auto**, Concurrency on main is **Auto**, Concurrency Scaling mode is **Auto**, User groups include **reporting**, and Query groups are empty.
- Default queue:** Priority is set to **Low**, Memory (%) is **Auto**, Concurrency on main is **Auto**, Concurrency Scaling mode is **Off**, and both User groups and Query groups are empty.

Each queue configuration includes a **Delete** button, a **Query Monitoring Rules (0)** section, and links to **Add rule from templates** and **Add custom rule**.

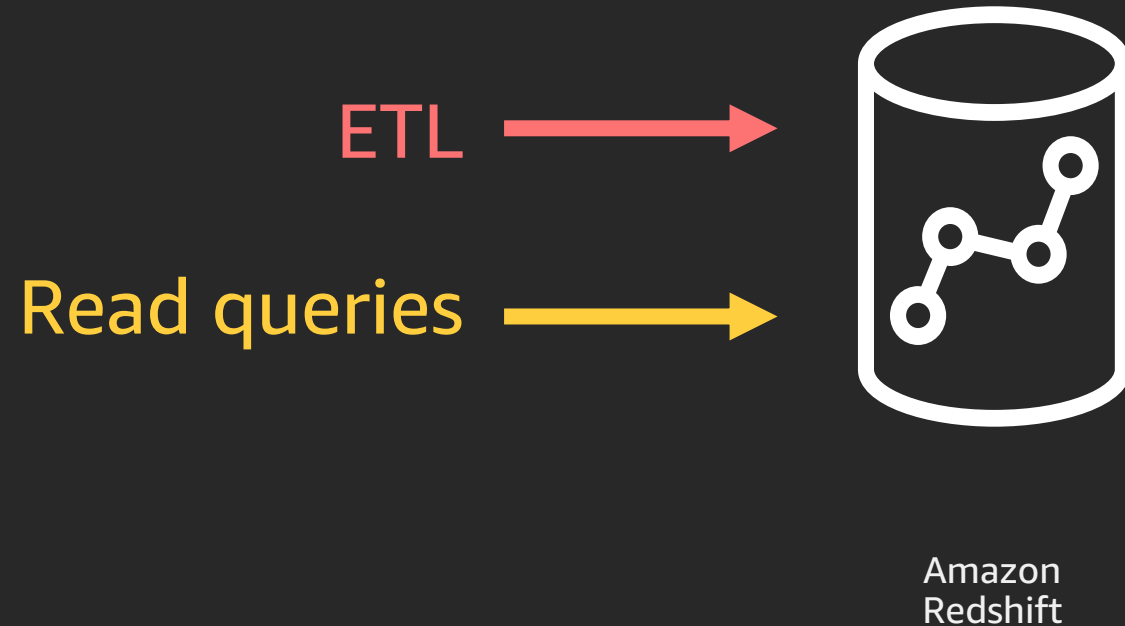
Query priorities – Under the hood



Auto WLM – Query priorities

Scenario 1 – ETL

- ETL workload executing concurrently with other read queries



Scenario 2 – Reporting

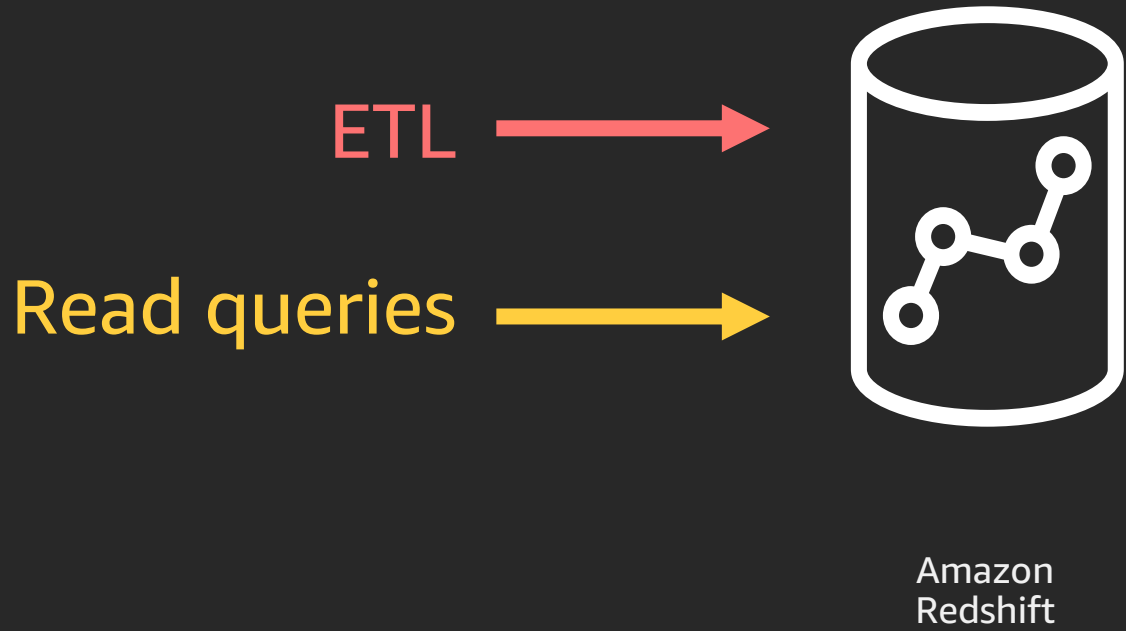
- Dashboarding, reporting & data science running concurrently



Auto WLM – Query priorities

ETL

- ETL workload executing concurrently with other read queries



No priorities

ETL	35 QpH
Read queries	245 QpH

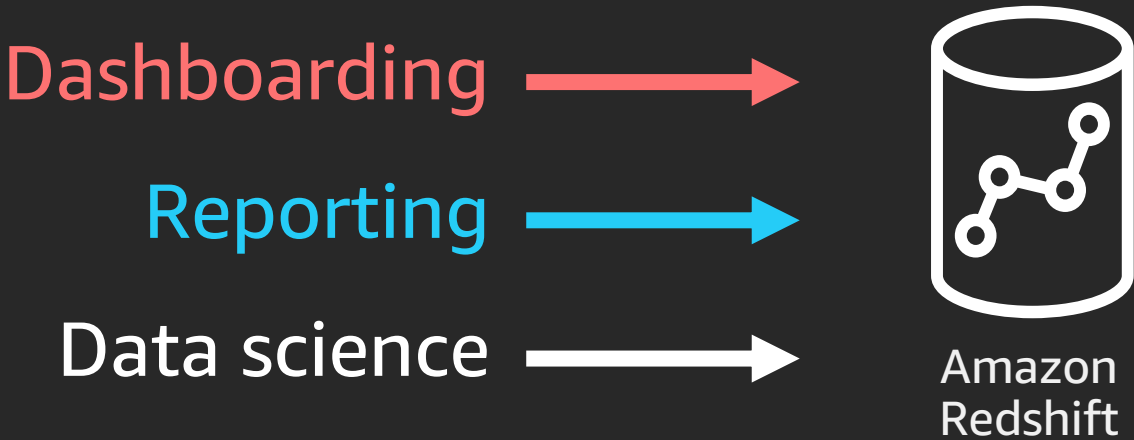
With priorities

ETL	117 QpH
Read queries	114 QpH

Auto WLM – Query priorities

Reporting

Dashboarding, reporting
& data science running
concurrently



**Using TPC-H 3T scale, 3 node dc2.8xl, 9 concurrency,
on Amazon Redshift version 1.0.10480*

No priorities

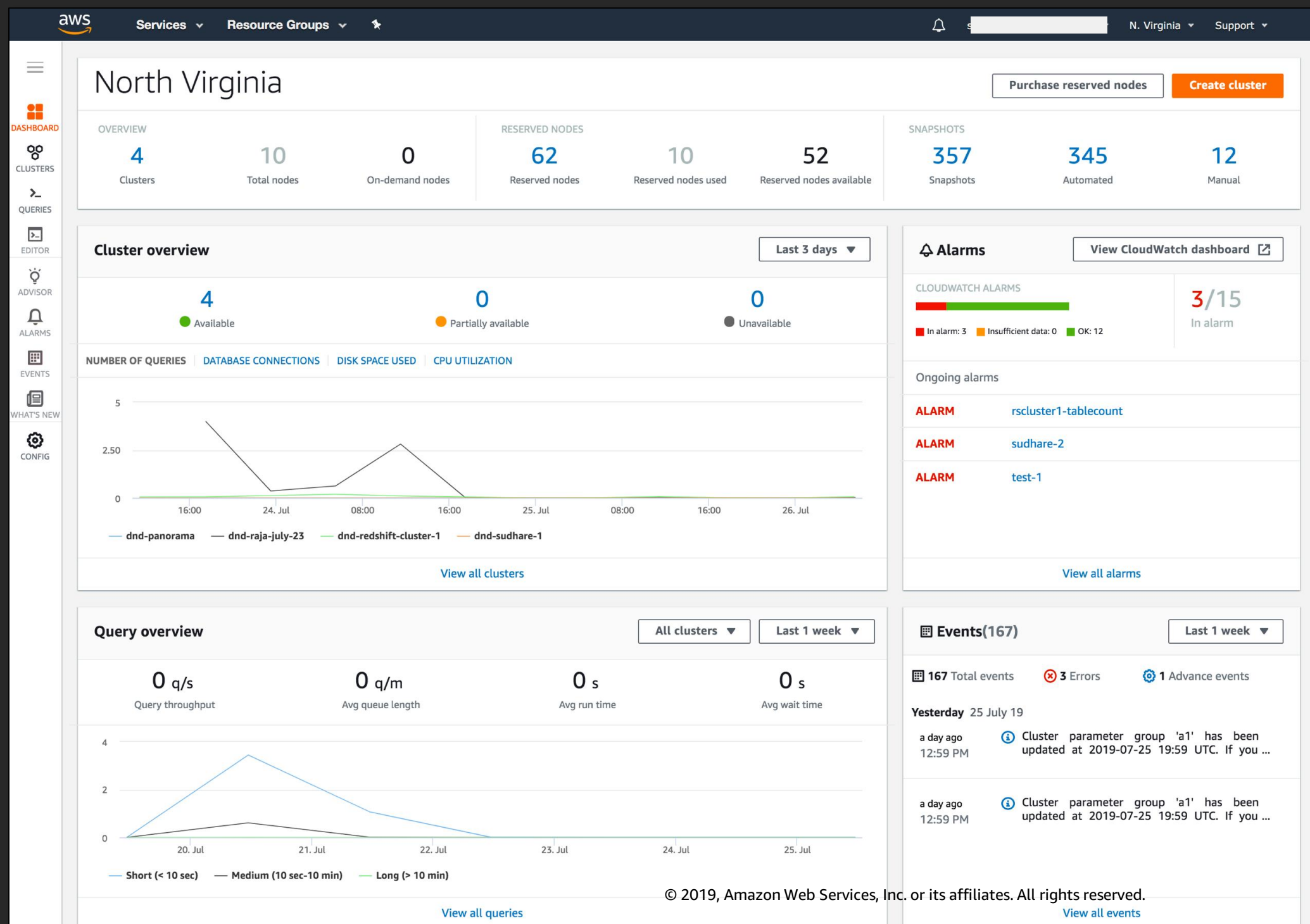
Dashboarding (<10 secs)	230 QpH
Reporting (10–59 secs)	113 QpH
Data science (>60 secs)	70 QpH

With priorities

Dashboarding (highest)	436 QpH
Reporting (high)	160 QpH
Data science (lowest)	56 QpH

Demo: configure query priorities

New monitoring dashboard



Consolidated view of all resources, activity, alarms & events

Easy navigation to specific areas in the console

Feature-rich query editor on the console

The screenshot displays the AWS Redshift console interface. On the left, the 'Data objects' sidebar lists various database objects like 'applicable_roles', 'check_constraints', etc. The main area is divided into three tabs: 'Query editor', 'Query results', and 'Table details'. The 'Query editor' tab shows a SQL query: `SELECT top 10 spectrumuseast1_internal... FROM spectrumuseast1_internal... WHERE spectrumuseast1_internal... AND spectrumuseast1_internal... GROUP BY spectrumuseast1_internal... ORDER BY 1 DESC;`. The 'Query results' tab shows a line chart titled 'Query 2318' with 'eventid' on the X-axis and 'sum' on the Y-axis. The 'Query history' tab shows a table of query results with columns 'eventid' and 'sum'.

Query editor

Select schema: information_schema

Filter tables and views

40 tables

applicable_roles, check_constraints, column_domain_usage, column_privileges, column_udt_usage, columns, constraint_column_usage, constraint_table_usage, data_type_privileges, domain_constraints, domain_udt_usage, domains, element_types, enabled_roles, information_schema_catalog..., key_column_usage, parameters, referential_constraints, role_column_grants, role_routine_grants, role_table_grants, role_usage_grants, routine_privileges, routines, schemata

Run Save

Query results

Query 2318

Finished, started on July 26, 2019 at 11:43:14

ELAPSED TIME: 00 m 01 s

Chart settings: Line, X axis: eventid, Y axis: sum

Execution Data Visualize

Query history

Query 2318

Finished, started on July 26, 2019 at 11:43:14

ELAPSED TIME: 00 m 01 s

Rows returned (10)

eventid

8798, 8797, 8796, 8795, 8794, 8793, 8792, 8791, 8790, 8789

Table details

Query 2318

Finished, started on July 26, 2019 at 11:43:14

ELAPSED TIME: 00 m 01 s

Execution timeline: Merge, Network, Sort, HashAggregate

Cost: 1000000116744.37, 1000000116744.37, 1000000116744.37, 116744.15, 0.00%

Completed

From July 23, 2019 at 07:02:43 PM To July 23, 2019 at 07:02:50 PM

Statistics

Rows returned: 259,015

Total data scanned: 3.20 MB

Query execution

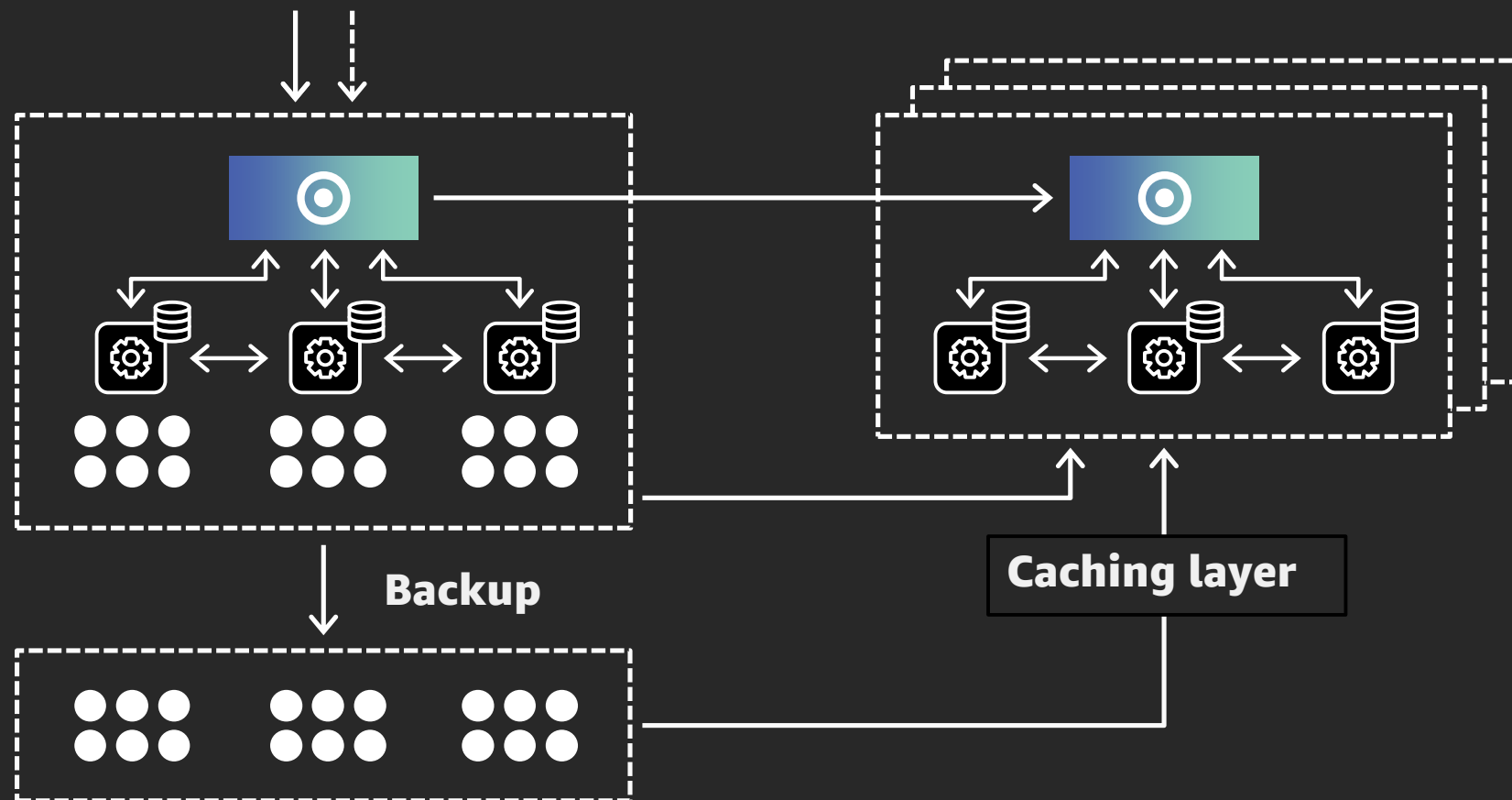
- Run one, run selected queries
- Full-screen writing mode
- Multiple query tabs
- Auto query formatting
- Insert saved snippets
- Query assist

Results

- Visual query plan
- Auto result visualizations
- Previewing data, schema

Concurrency scaling

Amazon Redshift automatically adds transient clusters, in seconds, to serve sudden spike in concurrent requests with consistently fast performance. No hydration required.



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:

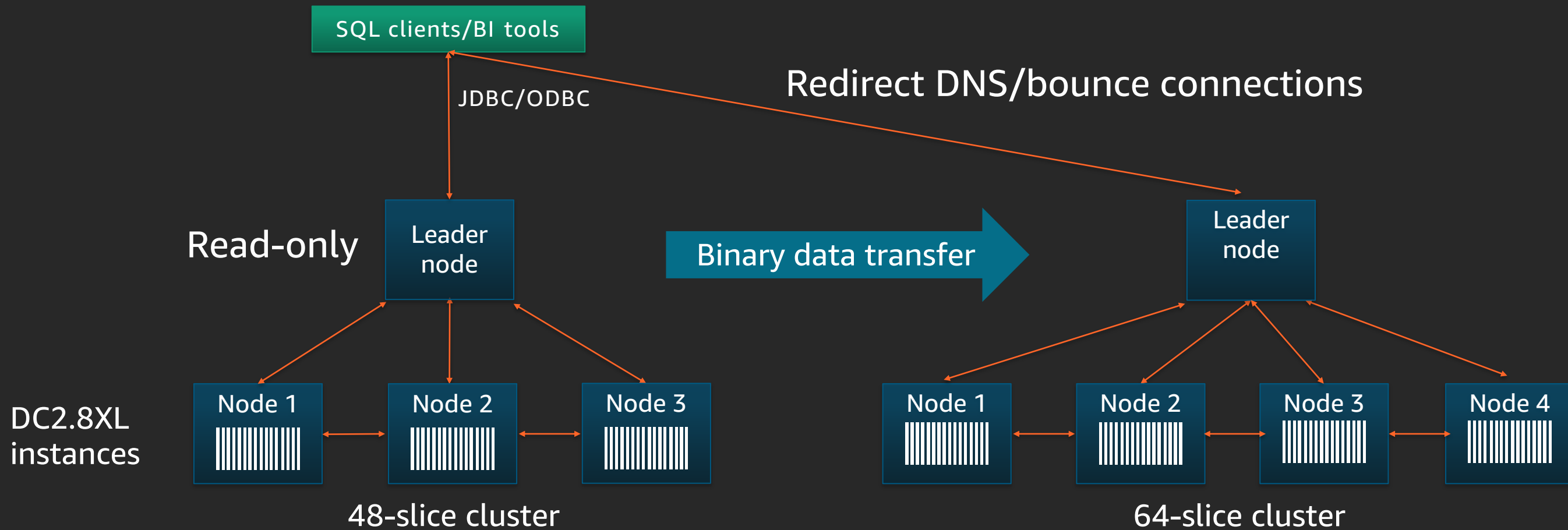
- All queries go to the leader node; user only sees less wait for queries
- When queries in designated WLM queue begin queuing, Amazon Redshift automatically routes them to the new clusters, enabling concurrency scaling automatically
- Amazon Redshift automatically spins up a new cluster, processes waiting queries & automatically shuts down the concurrency scaling cluster

Demo: enable concurrency scaling

Resizing Amazon Redshift

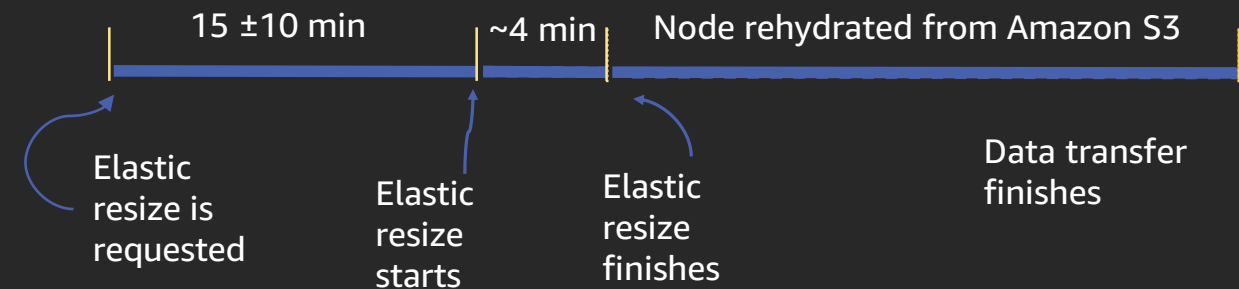
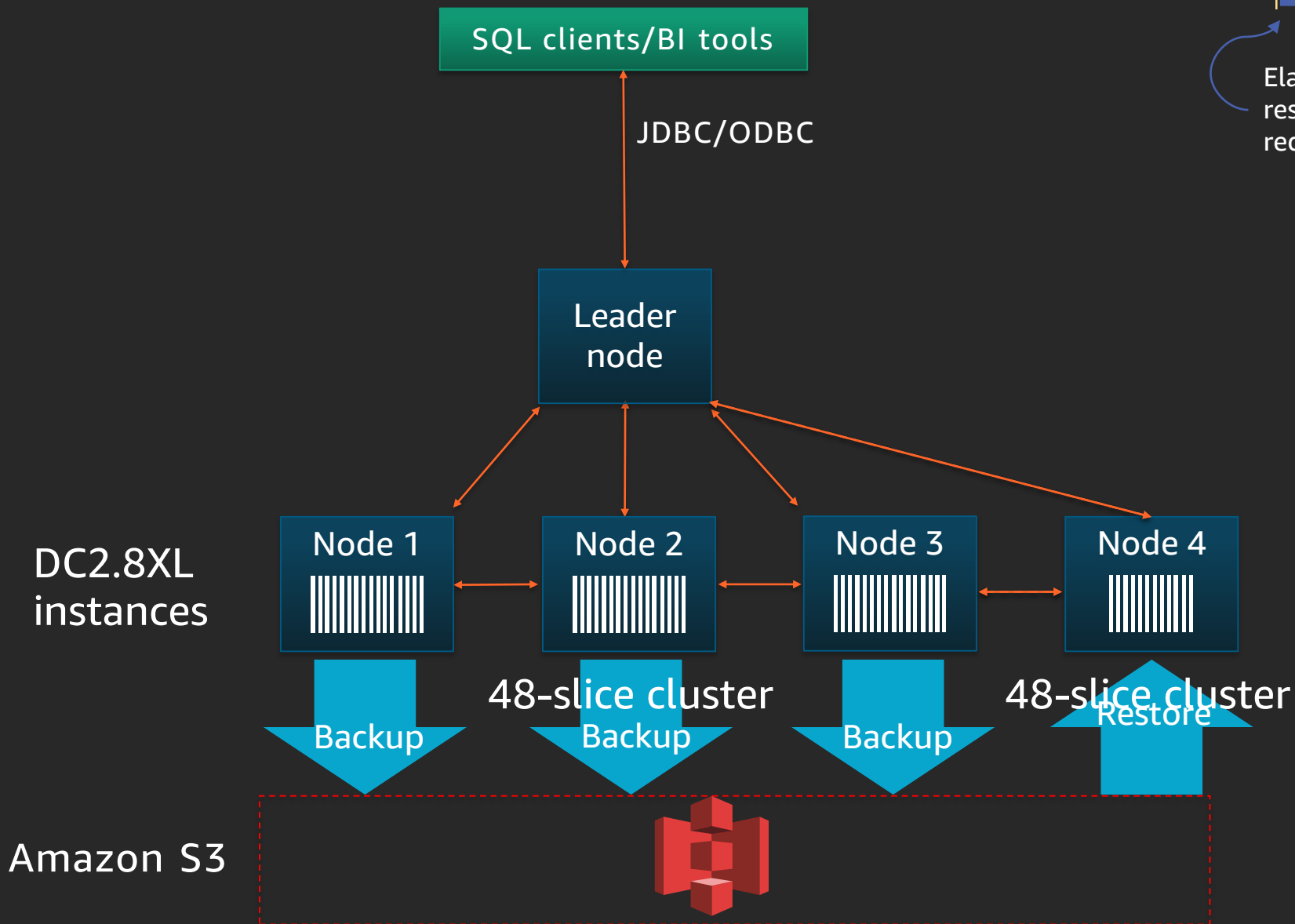
- Classic resize
 - Data is transferred from old cluster to new cluster
 - Cluster is read only while the resize is in progress
- Elastic resize
 - Nodes are quickly added/removed on existing cluster
- Cross-instance restore
 - Customers can now restore to new node types and counts

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



- At the start of elastic resize, we take an automatic snapshot to Amazon Simple Storage Service (Amazon S3) and provision the new node(s)
- Cluster is fully available for read and writes
- Slices are redistributed to/from nodes
- Inflight queries/connections are put on hold
- Some queries within transactions may be rollback
- Cluster is fully available; data transfer continues in the background
- Hot blocks are moved first

Q&A

AWS Labs on GitHub – Amazon Redshift

<https://github.com/aws-labs/amazon-redshift-utils>

<https://github.com/aws-labs/amazon-redshift-monitoring>

<https://github.com/aws-labs/amazon-redshift-udfs>

Admin scripts

Collection of scripts for running diagnostics on your cluster

Admin views

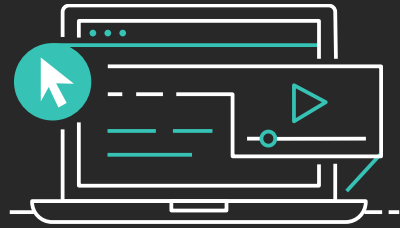
Collection of views for managing your cluster, generating schema DDL & so on

SimpleReplay utility

Utility that can replay Amazon Redshift audit logs; useful for testing new Amazon Redshift settings

Learn big data with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate data analytics skills



New free digital course, Data Analytics Fundamentals, introduces Amazon S3, Amazon Kinesis, Amazon EMR, AWS Glue, and Amazon Redshift



Classroom offerings, including Big Data on AWS, feature AWS expert instructors and hands-on labs



Validate expertise with the **AWS Certified Big Data - Specialty** exam or the new **AWS Certified Data Analytics - Specialty** beta exam

Visit aws.amazon.com/training/paths-specialty/

Thank you!

Raj Sett

rajsett@amazon.com



Please complete the session
survey in the mobile app.