Best Practices on Scaling Amazon Redshift

Tony Gibbs, Sr. Data Warehouse Solutions Architect
Himanshu Raja, Sr. Product Manager

November 15, 2018

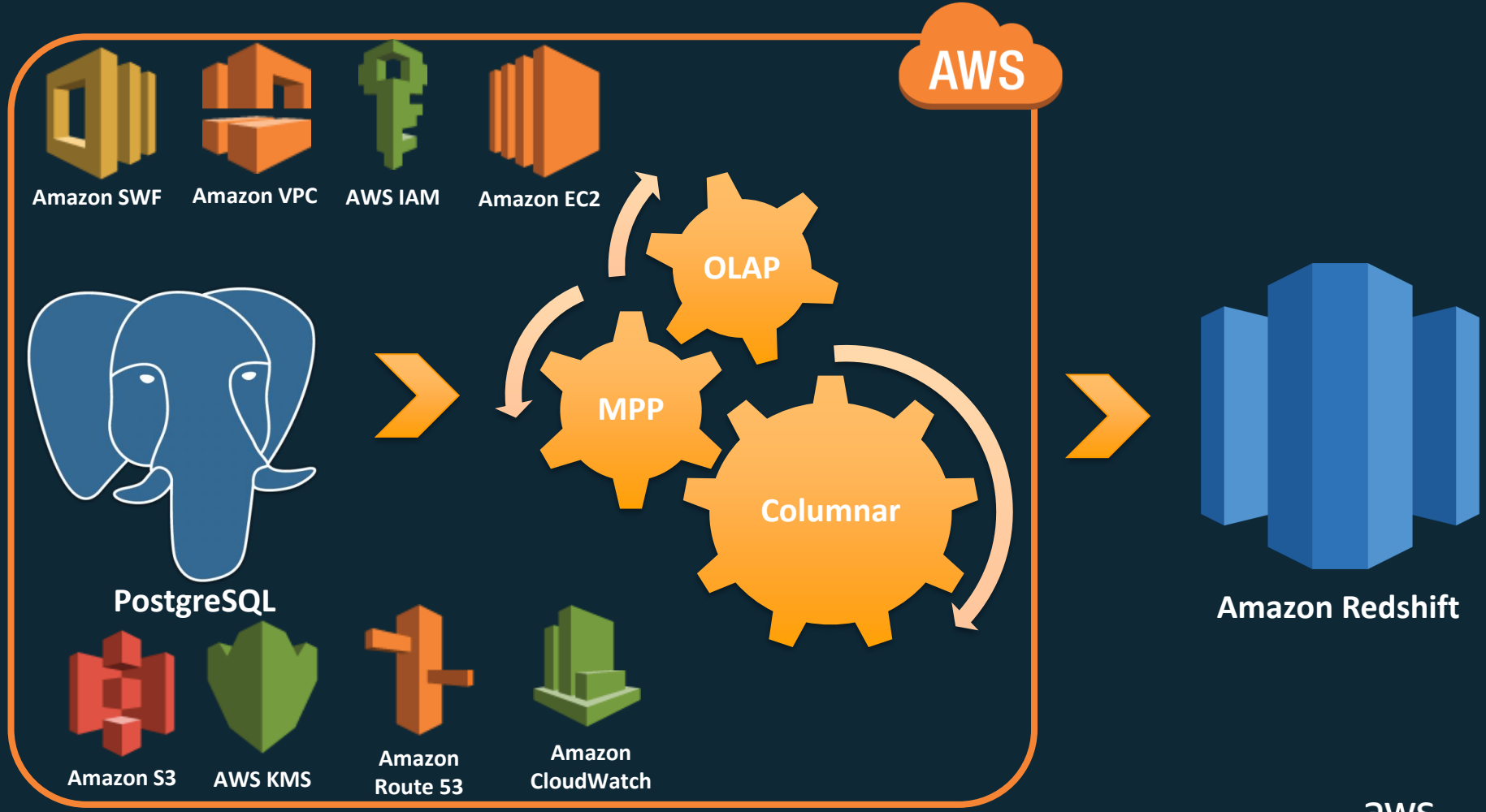
aws

Amazon Redshift Best Practices Overview

- History and development
- Architecture and Concepts
- Scaling Amazon Redshift
    - Introducing the new Elastic Resize
- Best Practices - COPY
- Open Q&A

aws

History and Development



February 2013

> 130 Significant Patches

> 180 Significant Features

November 2018

Architecture and Concepts

Amazon Redshift Architecture

Massively parallel, shared nothing columnar architecture

Leader node

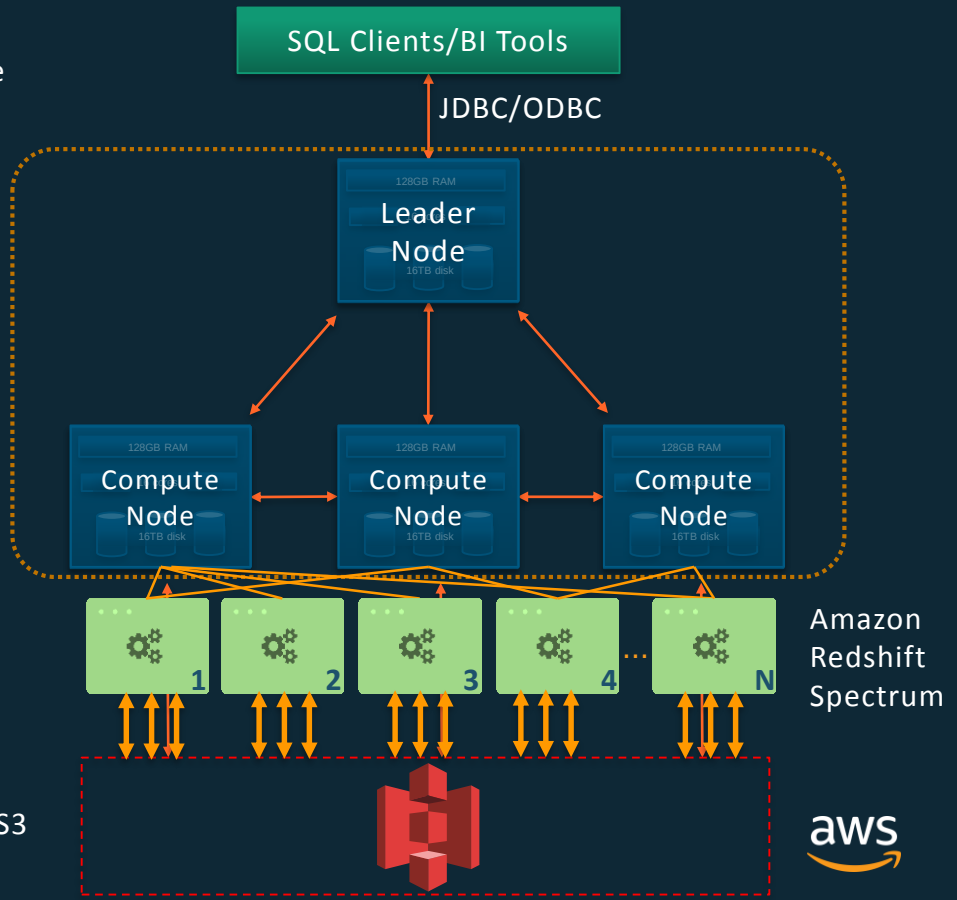
- SQL endpoint
- Stores metadata
- Coordinates parallel SQL processing

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)



Terminology and Concepts: Columnar

Amazon Redshift uses a columnar architecture for storing data on disk

Goal: reduce I/O for analytics queries

Physically store data on disk by column rather than row

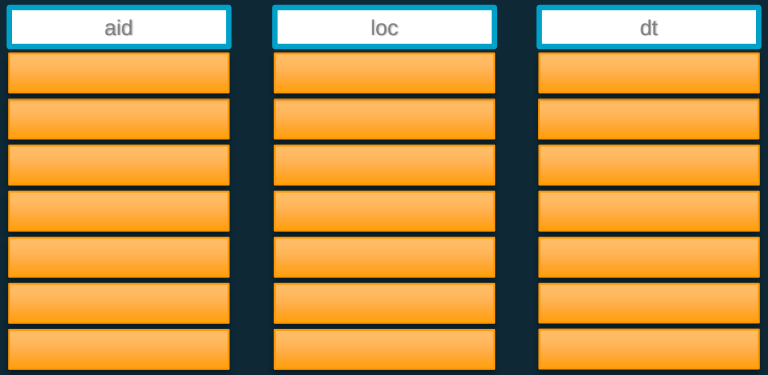
Only read the column data that is required

aws

Columnar Architecture: Example

```
CREATE TABLE deep_dive (
     aid  INT       --audience_id
    ,loc  CHAR(3)   --location
    ,dt   DATE      --date
);
```

aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14




```
SELECT min(dt) FROM deep_dive;
```

Row-based storage

- Need to read everything
- Unnecessary I/O

aws

Columnar Architecture: Example

```
CREATE TABLE deep_dive (
     aid  INT       --audience_id
    ,loc  CHAR(3)   --location
    ,dt   DATE      --date
);
```

aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14


```
SELECT min(dt) FROM deep_dive;
```

Column-based storage

- Only scan blocks for relevant column

aws

Terminology and Concepts: Slice

A *slice* can be thought of like a virtual compute node

- Unit of data partitioning
- Parallel query processing

Facts about slices:

- Each compute node has either 2, 16
- Table rows are distributed to slices
- A slice processes only its own data

aws

Terminology and Concepts: Nodes

Dense Compute—DC2

- Solid state disks

Dense Storage—DS2

- Magnetic disks

Instance Type	Disk Type	Size	Memory	CPUs	Slices
DC2 large	NVMe SSD	160 GB	16 GB	2	2
DC2 8xlarge	NVMe SSD	2.56 TB	244 GB	32	16
DS2 xlarge	Magnetic	2 TB	32 GB	4	2
DS2 8xlarge	Magnetic	16 TB	244 GB	36	16

Scaling Amazon Redshift

How can you scale Redshift today?

Resize: data is transferred from old cluster to new cluster

- Change cluster size
- Change node types

Redshift Spectrum

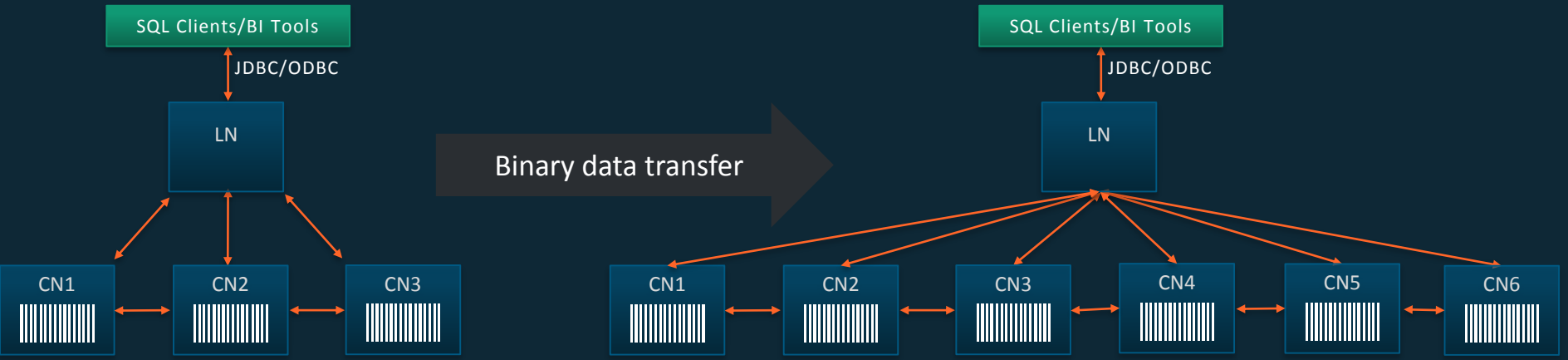
- Query data directly from S3 with scalable compute layer

Restore from snapshot: spin up a parallel cluster

- Ad-hoc data analysis

aws

Classic resize enables scaling up and down



- Complete data is moved and re-distributed
- Cluster is in read-only mode during resize

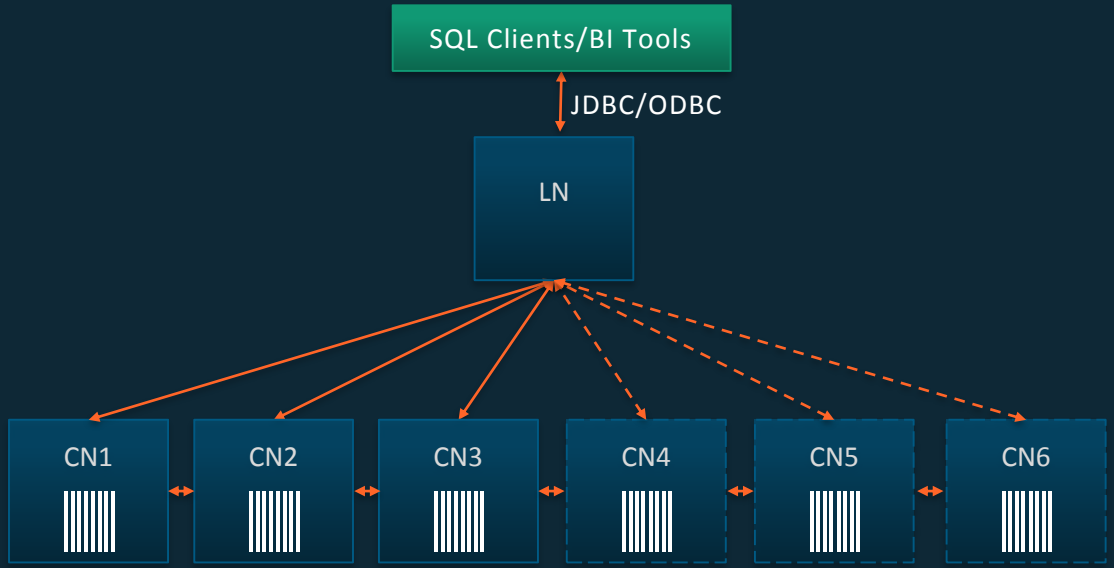
Introducing Elastic Resize

With elastic resize you can now add or subtract nodes in minutes

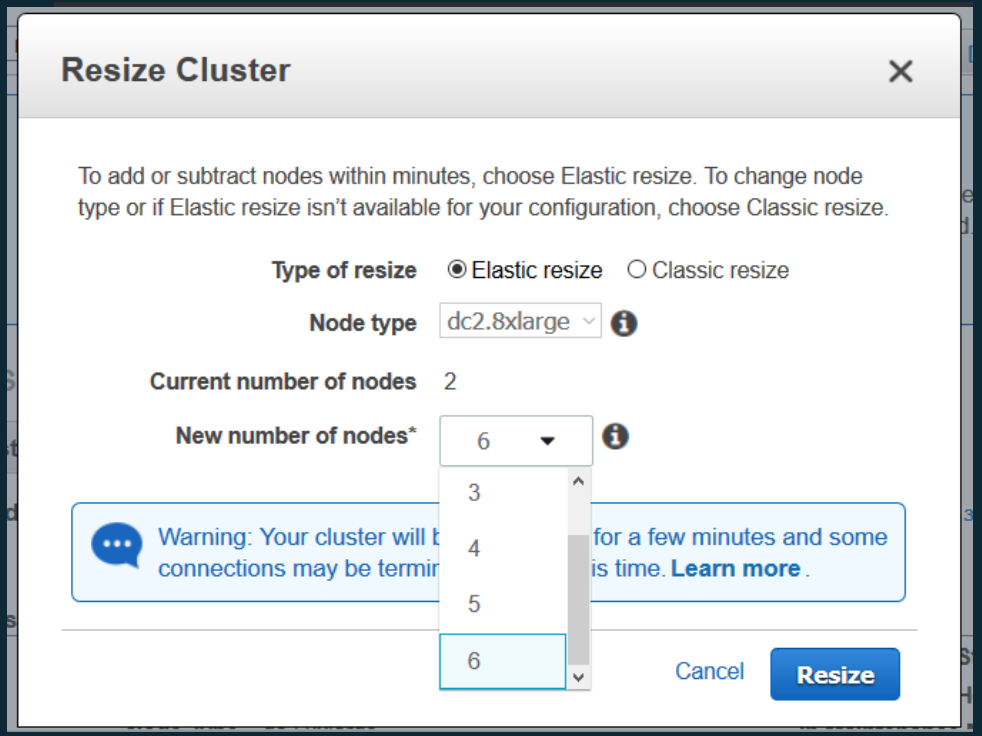
- ✓ Scale compute and storage on-demand
- ✓ Faster query processing
- ✓ Finish large ETL jobs faster
- ✓ Save on-demand cost during off-peak hours

aws

Elastic resize enables scaling up and down in minutes



Use console or CLI to do elastic resize

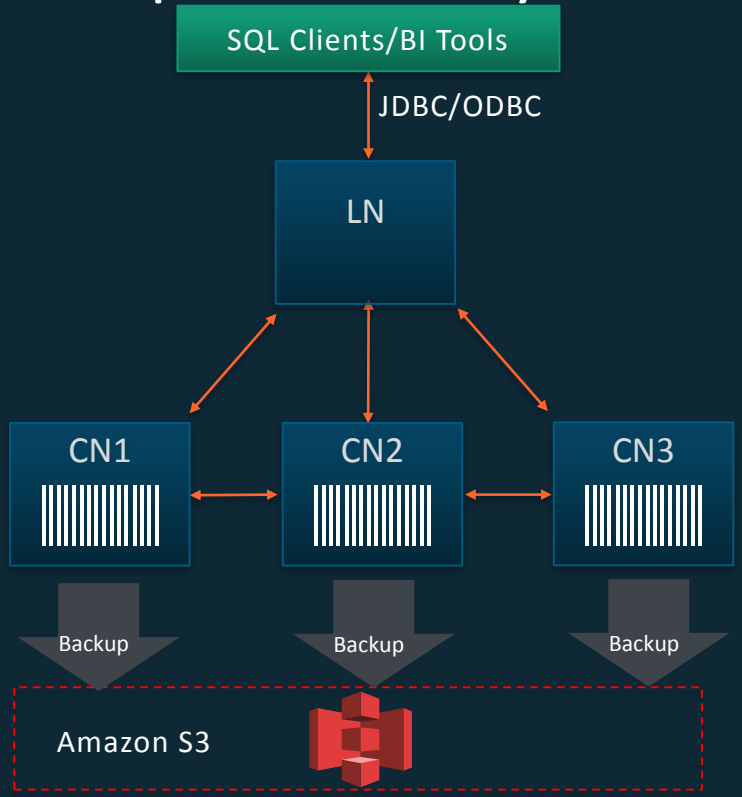


resize-cluster

resize-cluster
--cluster-identifier <value>
[--cluster-type <value>]
[--node-type <value>]
--number-of-nodes <value>
[--classic | --no-classic]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]

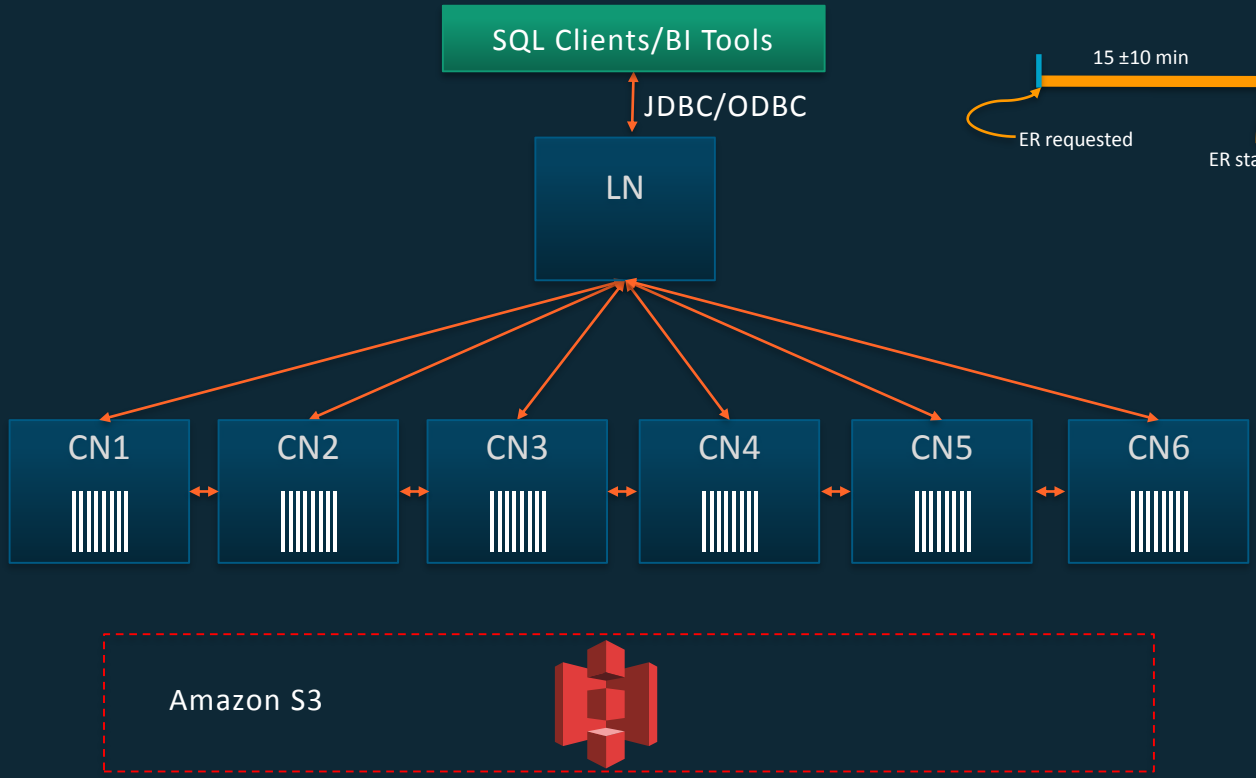
aws

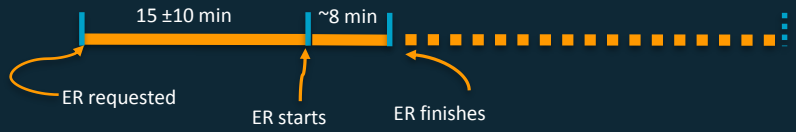
Elastic resize begins after an up to date snapshot is synced to Amazon S3



- At the start of elastic resize, we take an automatic snapshot to Amazon S3
- Cluster is fully available for read and writes

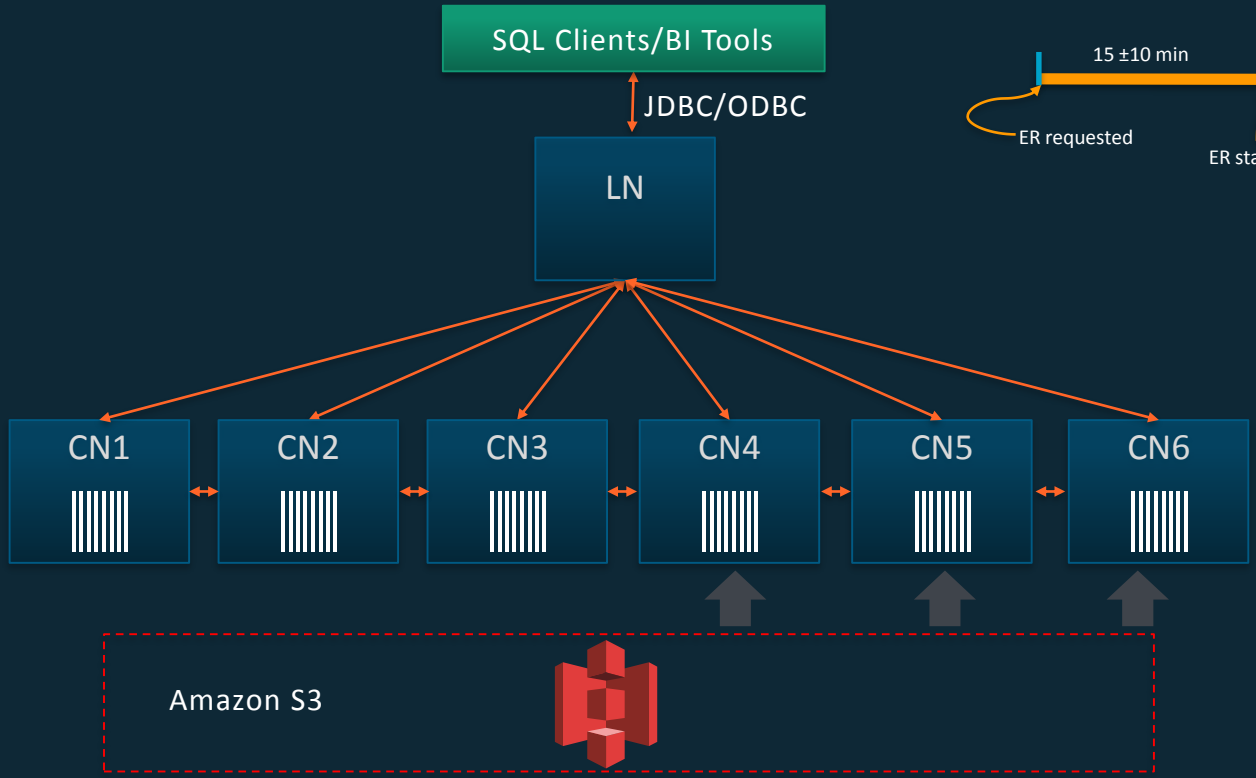
Elastic resize finishes in minutes

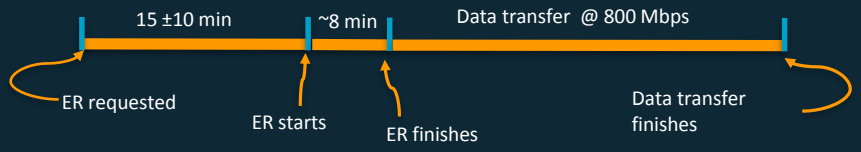




- During resize, new nodes are added or removed
- Slices are shuffled
- All queries are put on hold
- Some queries may get terminated

Data transfer is prioritized based on workload

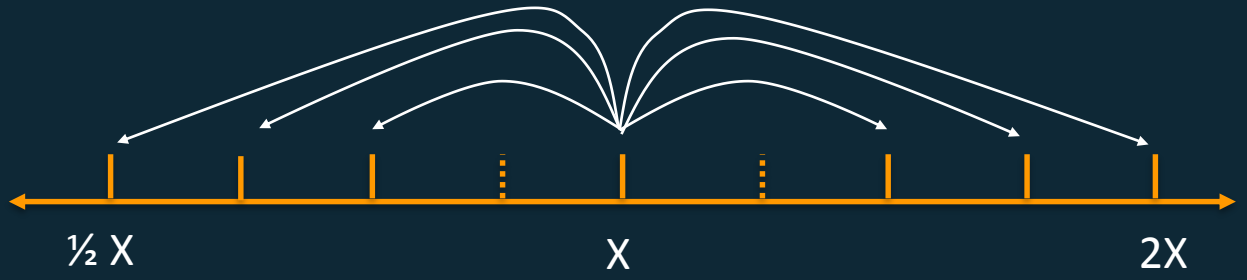




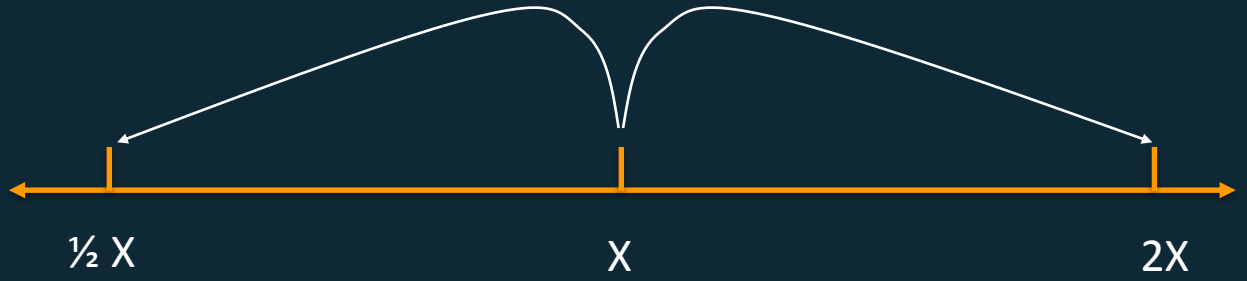
- Cluster is fully available; data transfer continues in the background
- Hot blocks are moved first

Elastic resize options depend on the node type

DS2.8XL
DC2.8XL



DS2.XL
DC2.L



aws

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 (SSD <-> HDD)		✓
If elastic resize is not an option		✓
Limited availability during resize	5-10 minutes (parked connections)	1-24 hours (read-only)

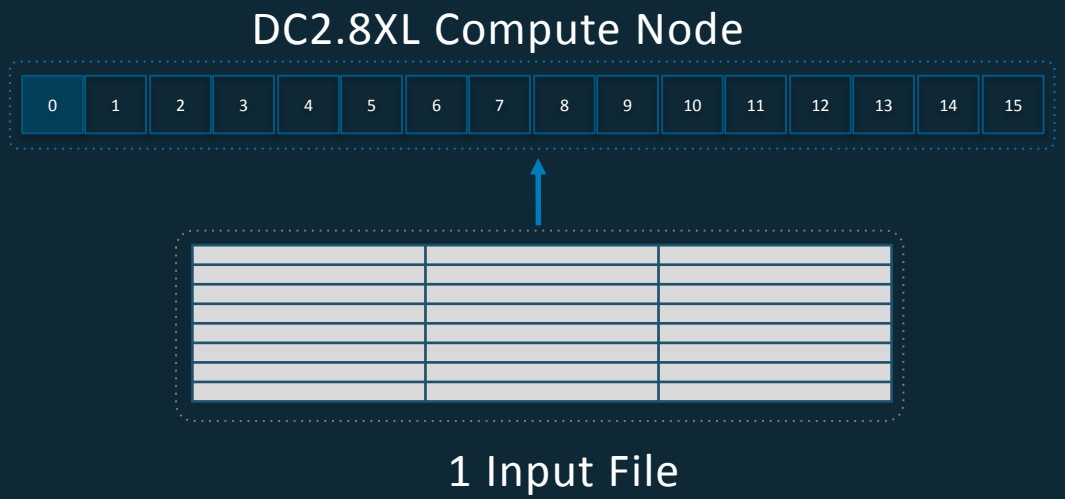
Best Practices - COPY

Best Practices: 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

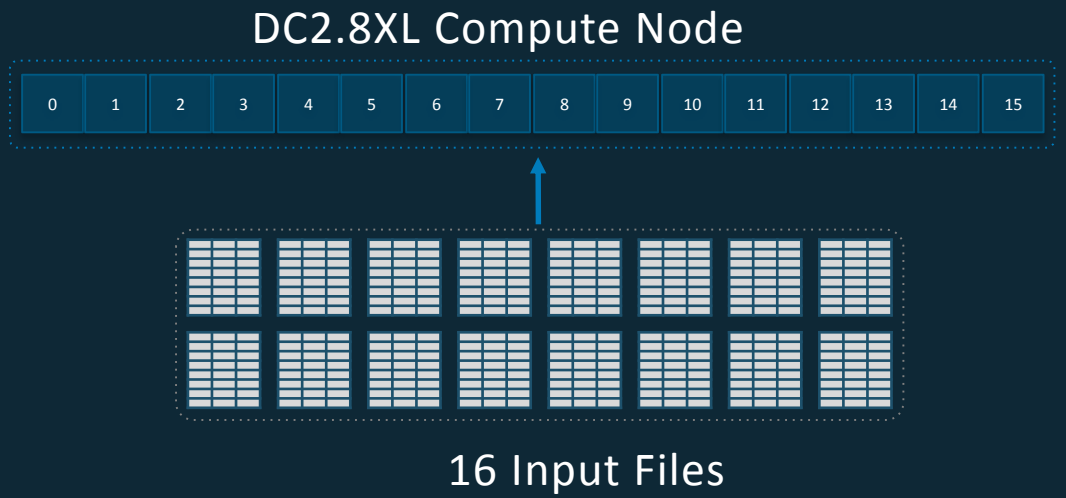


Best Practices : COPY Statement

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

aws

Best Practices: COPY Statement

Delimited files are recommend

- Pick a simple delimiter '|' or ',' or tabs
- Pick a simple NULL character (\N)
- Use double quotes and an escape character (' \ ') for varchars
- UTF-8 varchar columns take four bytes per char

Split files into a number that is a multiple of the total number of slices in the Amazon Redshift cluster

```
SELECT count(slice) from stv_slices;
```

Files sizes should be 1 MB–1 GB after gzip compression

aws

Additional Resources

Documentation

Resizing Clusters in Amazon Redshift
https://docs.aws.amazon.com/redshift/latest/mgmt/rs-resize-tutorial.html

Elastic Resize
https://docs.aws.amazon.com/redshift/latest/mgmt/rs-tutorial-elastic-resize.html

Classic Resize
https://docs.aws.amazon.com/redshift/latest/mgmt/rs-tutorial-classic-resize.html

CLI
https://docs.aws.amazon.com/cli/latest/reference/redshift/resize-cluster.html

API
https://docs.aws.amazon.com/redshift/latest/APIReference/API_ResizeCluster.html

Thank You!

Tony Gibbs
Himanshu Raja