

# Getting Started with Amazon Redshift

Maor Kleider, Sr. Product Manager, Amazon Redshift



# Agenda

- Introduction
- Benefits
- Use cases
- Getting started
- Q&A

# AWS Big Data Portfolio

## Collect



Amazon Kinesis  
Firehose



AWS Direct  
Connect



Amazon Kinesis  
Analytics



Amazon  
Snowball



Amazon Kinesis  
Streams

## Store



Amazon S3



Amazon Glacier



Amazon  
Dynamo DB



Amazon RDS,  
Amazon Aurora



Amazon  
CloudSearch



Amazon  
Elasticsearch

## Analyze



Amazon EMR



Amazon EC2



Amazon  
Redshift



Amazon  
Athena



Amazon  
QuickSight



Amazon Machine  
Learning

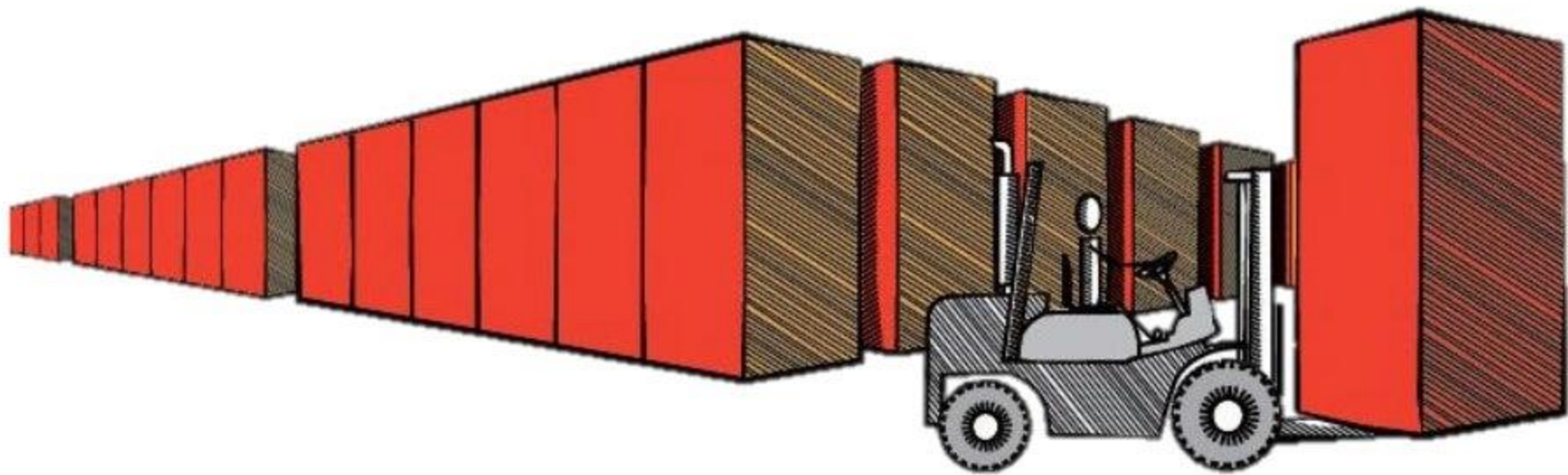


AWS Database Migration Service



AWS Glue

# Amazon Redshift



Fast, simple, petabyte-scale data warehousing for \$1,000/TB/Year

140+ features



Amazon  
Redshift

*a lot faster*  
*a lot simpler*  
*a lot cheaper*



Relational data warehouse

Massively parallel; petabyte scale

Fully managed

HDD and SSD platforms

\$1,000/TB/year; starts at \$0.25/hour

# Forrester Wave™ Enterprise Data Warehouse Q4 '15

FIGURE 2 The Forrester Wave™: Enterprise Data Warehouse, Q4 '15



The Forrester Wave™ is copyrighted by Forrester Research, Inc. Forrester and Forrester Wave™ are trademarks of Forrester Research, Inc. The Forrester Wave™ is a graphical representation of Forrester's call on a market and is plotted using a detailed spreadsheet with exposed scores, weightings, and comments. Forrester does not endorse any vendor, product, or service depicted in the Forrester Wave. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.



# Selected Amazon Redshift customers



# Use Case: Traditional Data Warehousing



Business Reporting



Advanced pipelines and queries



Secure and Compliant



Bulk Loads and Updates

**Easy Migration** – Point & Click using AWS Database Migration Service

**Secure & Compliant** – End-to-End Encryption. SOC 1/2/3, PCI-DSS, HIPAA and FedRAMP compliant

**Large Ecosystem** – Variety of cloud and on-premises BI and ETL tools



Japanese Mobile  
Phone Provider



World's Largest Children's  
Book Publisher



Powering 100 marketplaces  
in 50 countries



# Use Case: Log Analysis



Log & Machine  
IOT Data



Clickstream  
Events Data



Time-Series  
Data

**Cheap** – Analyze large volumes of data cost-effectively

**Fast** – Massively Parallel Processing (MPP) and columnar architecture for fast queries and parallel loads

**Near real-time** – Micro-batch loading and Amazon Kinesis Firehose for near-real time analytics



Interactive data analysis and  
recommendation engine



Ride analytics for pricing  
and product development



Ad prediction and  
on-demand analytics

# Use Case: Business Applications



**Multi-Tenant BI Applications**



**Back-end services**



**Analytics as a Service**

**Fully Managed** – Provisioning, backups, upgrades, security, compression all come built-in so you can focus on your business applications

**Ease of Chargeback** – Pay as you go, add clusters as needed. A few big common clusters, several data marts

**Service Oriented Architecture** – Integrated with other AWS services. Easy to plug into your pipeline



Infosys Information  
Platform (IIP)



Analytics-as-a-  
Service



Product and Consumer  
Analytics

# Amazon Redshift architecture

## Leader node

- Simple SQL endpoint
- Stores metadata
- Optimizes query plan
- Coordinates query execution

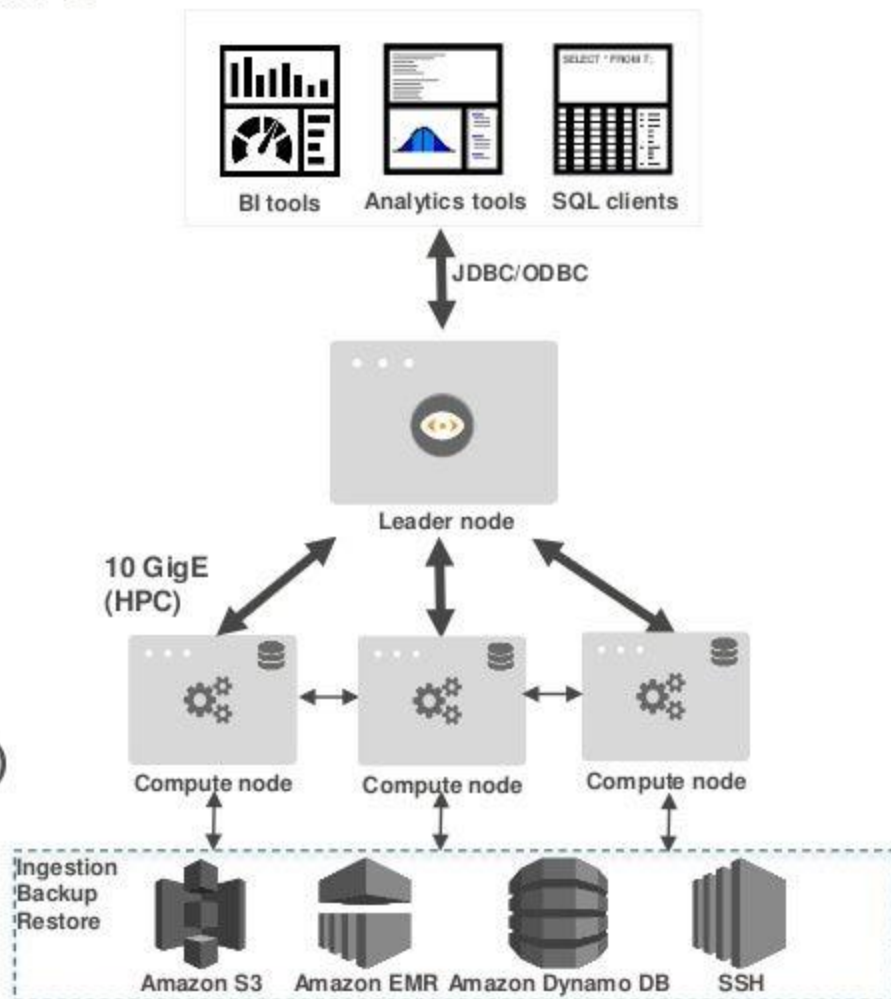
## Compute nodes

- Local columnar storage
- Parallel/distributed execution of all queries, loads, backups, restores, resizes

**Start at just \$0.25/hour, grow to 2 PB (compressed)**

DC1: SSD; scale from 160 GB to 326 TB

DS2: HDD; scale from 2 TB to 2 PB



# Benefit #1: Amazon Redshift is fast

## Dramatically less I/O

Column storage

Data compression

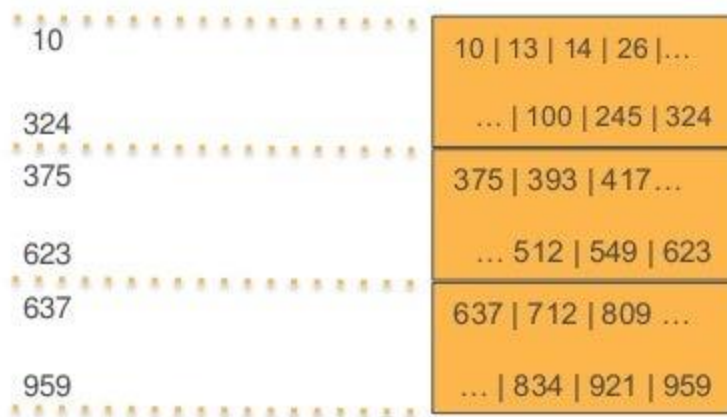
Zone maps

Direct-attached storage

Large data block sizes

```
analyze compression listing;
```

Table	Column	Encoding
listing	listid	delta
listing	sellerid	delta32k
listing	eventid	delta32k
listing	dateid	bytedict
listing	numtickets	bytedict
listing	priceperticket	delta32k
listing	totalprice	mostly32
listing	listtime	raw



# Benefit #1: Amazon Redshift is fast

## Parallel and distributed

Query

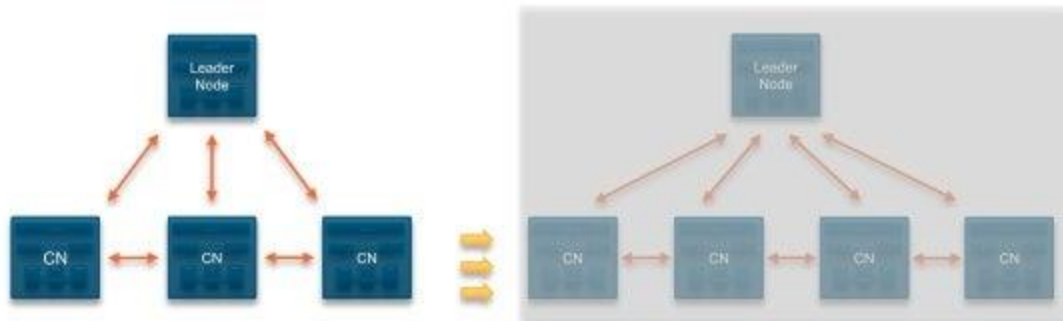
Load

Export

Backup

Restore

Resize



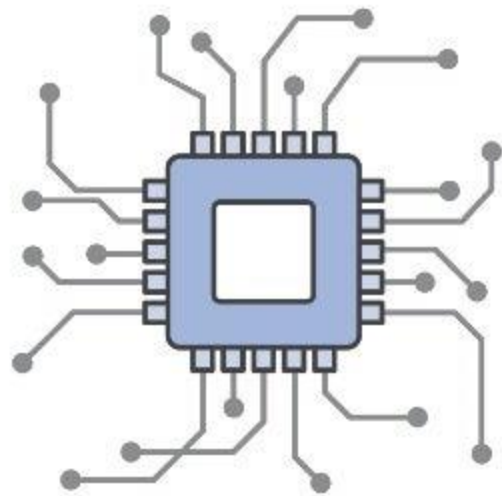
## Benefit #1: Amazon Redshift is fast

Hardware optimized for I/O intensive workloads, 4 GB/sec/node

Enhanced networking, over 1 million packets/sec/node

Choice of storage type, instance size

Regular cadence of auto-patched improvements





# Benefit #1: Amazon Redshift is fast

**REDFIN.**

"Did I mention that it's **ridiculously fast**? We're using it to provide our analysts with an alternative to Hadoop"

**boingo**

"On our previous big data warehouse system, it took around 45 minutes to run a query against a year of data, but that number went down to **just 25 seconds** using Amazon Redshift"

**4**  
CHANNEL FOUR TELEVISION

"We **regularly process multibillion row datasets and we do that in a matter of hours**. We are heading to up to 10 times more data volumes in the next couple of years, easily



Periscope  
DATA

"After investigating Redshift, Snowflake, and BigQuery, we found that Redshift offers **top-of-the-line performance at best-in-market price points**"

**Pinterest**

"...[Redshift] performance has blown away everyone here. We generally see **50-100X speedup over Hive**"

**optimum.**

"We saw a **2X performance improvement** on a wide variety of workloads. The **more complex the queries, the higher the performance improvement**"

# And has gotten faster...

## 5X Query throughput improvement over the past year

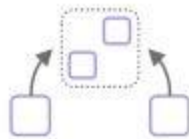
- ◆ Memory allocation (launched)
- ◆ Improved commit and I/O logic (launched)
- ◆ Queue hopping (launched)
- ◆ Query monitoring rules (coming soon)
- ◆ Power start (coming soon)
- ◆ Short query bias (coming soon)



Fast

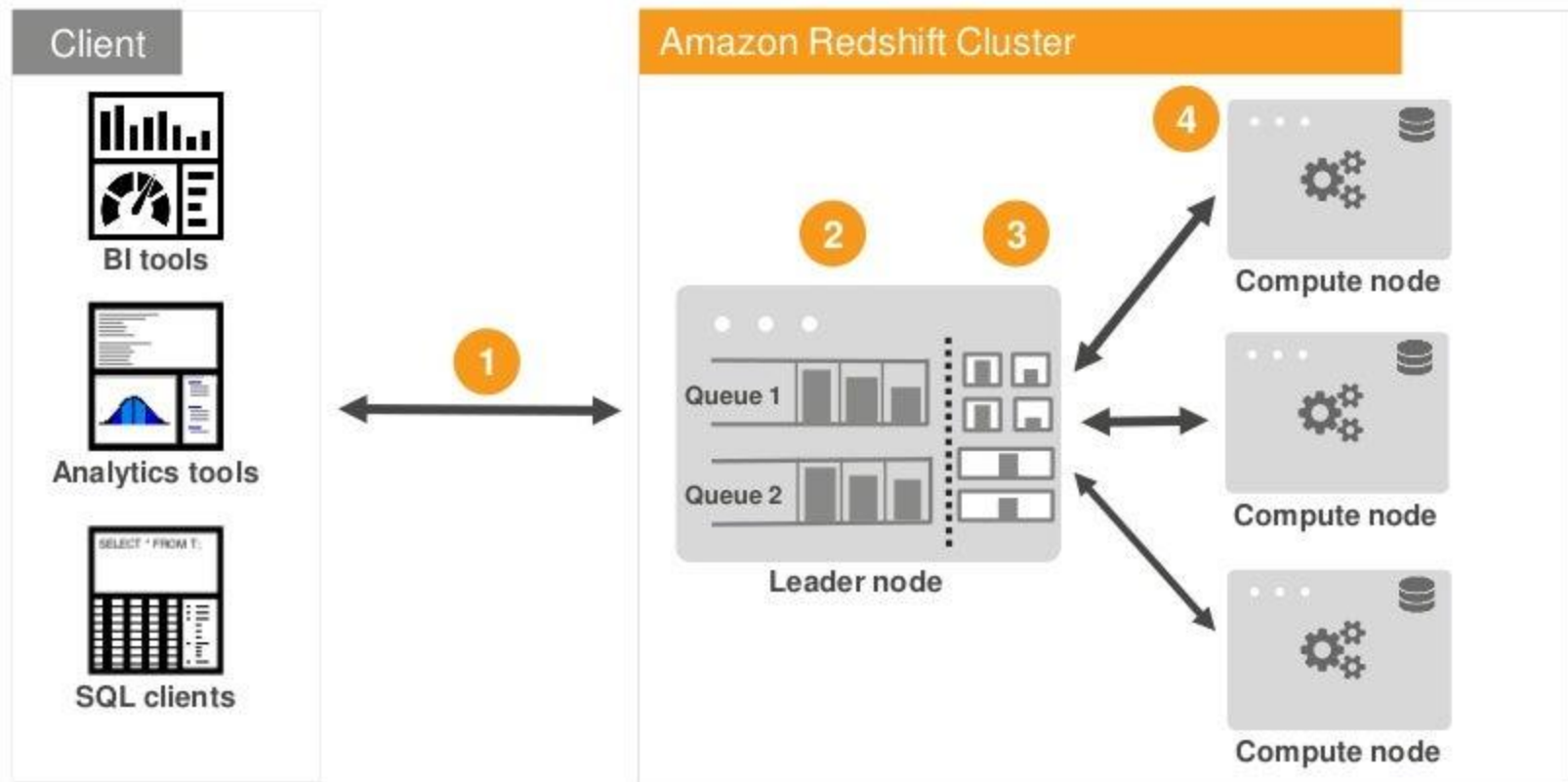
## 10X Vacuuming performance improvement

- ◆ Ensures data is sorted for efficient and fast I/O
- ◆ Reclaims space from deleted rows
- ◆ Enhanced vacuum performance leads to better system throughput

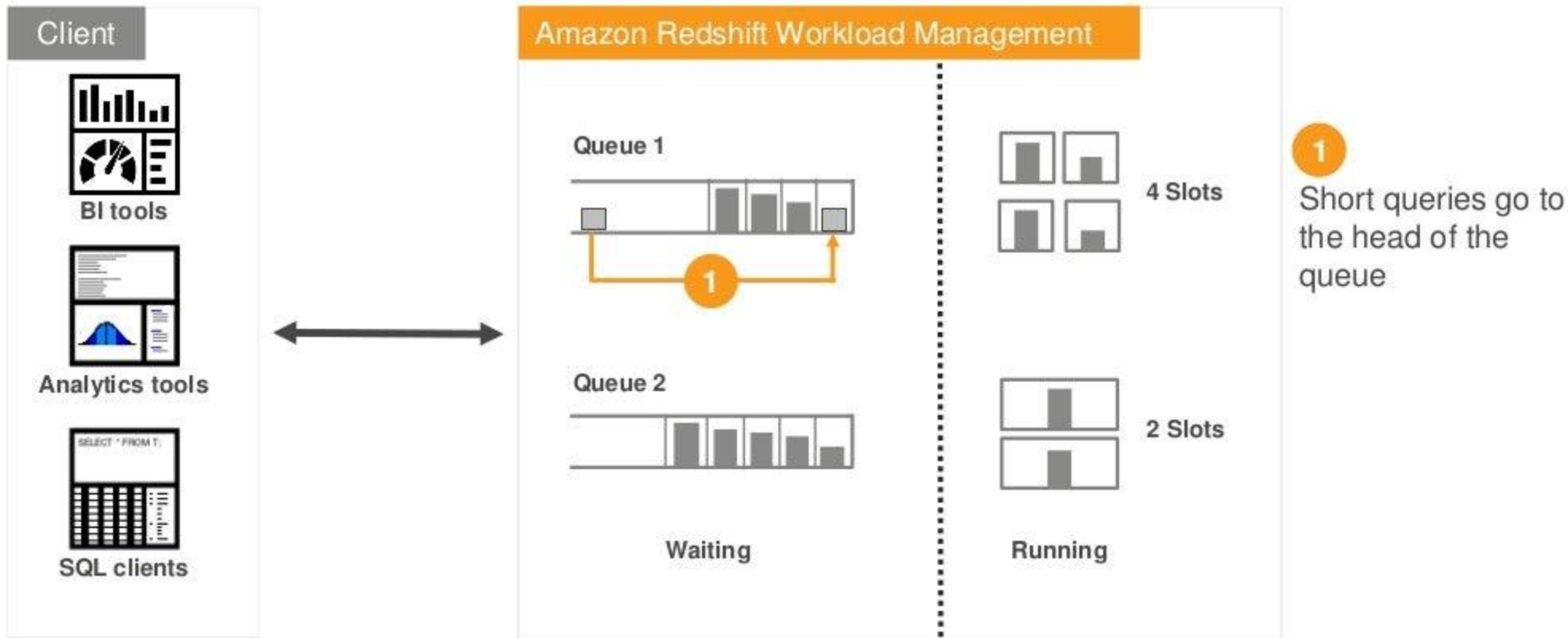


Efficient

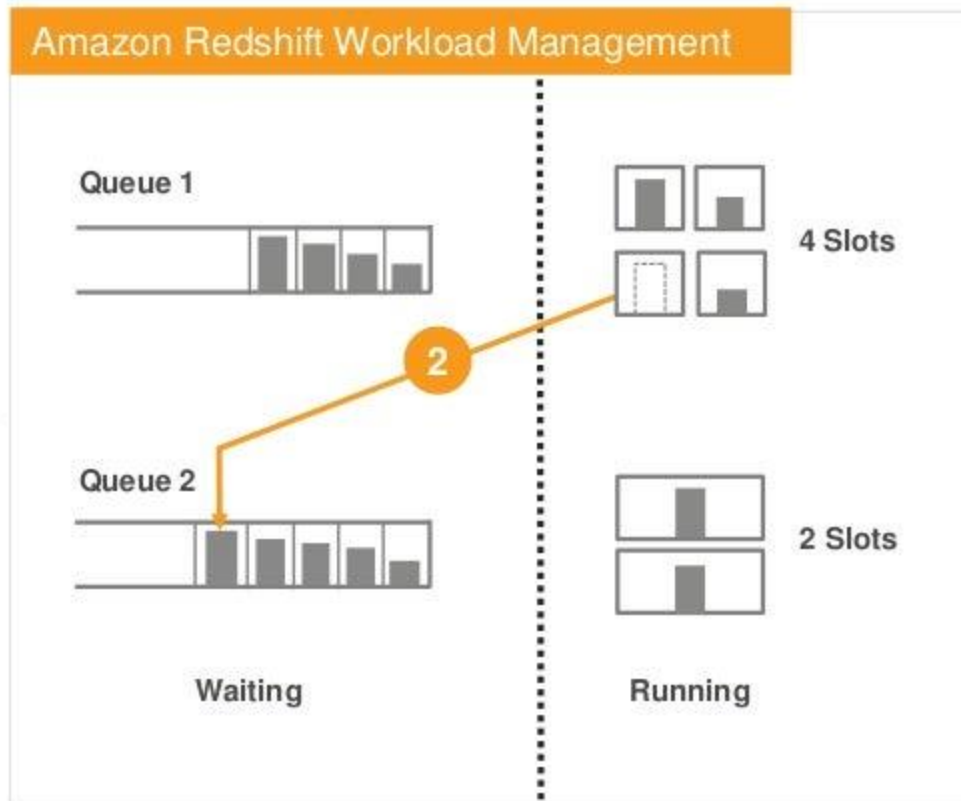
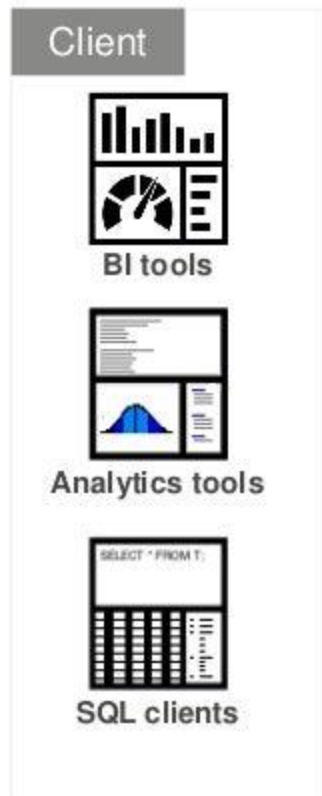
# The life of a query



# Coming soon: Short query bias



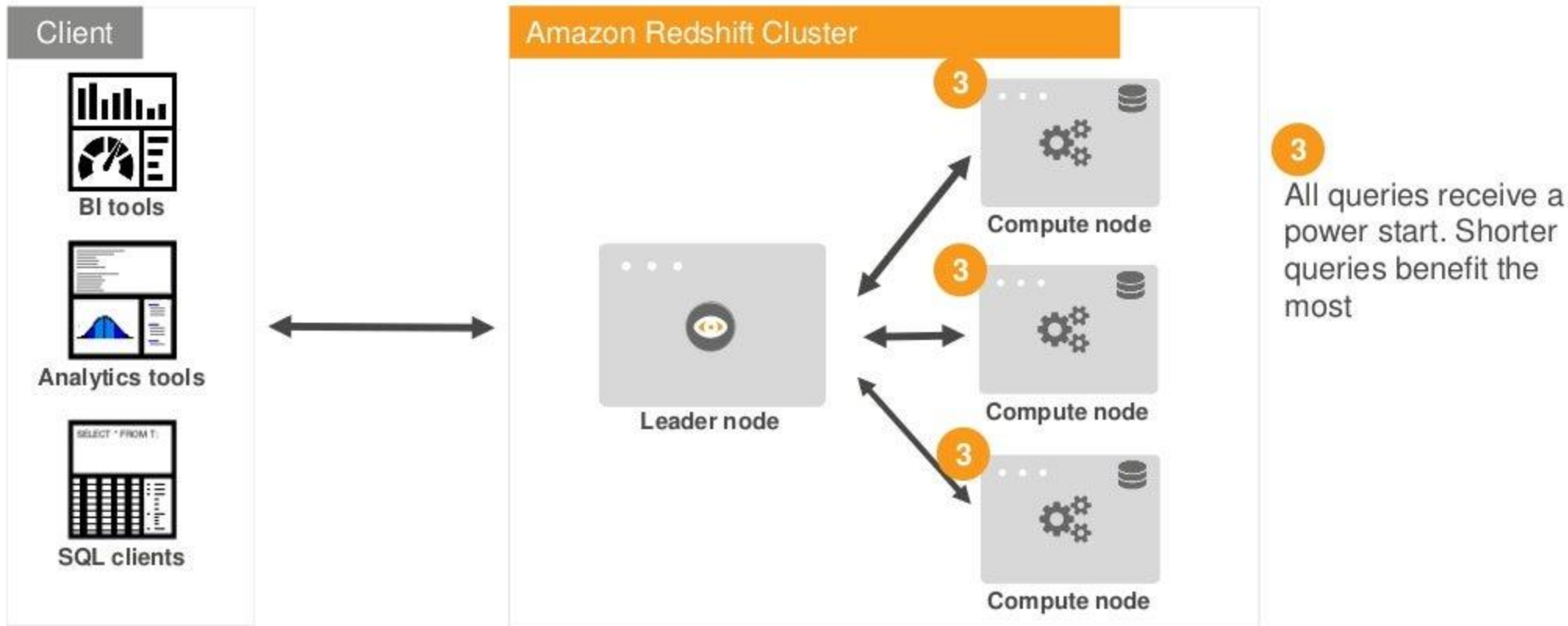
# Queue hopping



2

Release slot if a query exceeds timeout

# Coming soon: Power start





# Coming soon: Query monitoring rules

- Allows automatic handling of runaway (poorly written) queries
- Metrics with operators and values (e.g. query\_cpu\_time > 1000) create a *predicate*
- Multiple predicates can be *AND-ed* together to create a *rule*
- Multiple rules can be defined for a queue in WLM. These rules are *OR-ed* together

If { *rule* } then [*action*]

{ *rule : metric operator value* } eg: rows\_scanned > 100000

- *Metric* : cpu\_time, query\_blocks\_read, rows scanned, query execution time, cpu & io skew per slice, join\_row\_count, etc.
- *Operator* : <, >, ==
- *Value* : integer

[*action*] : hop, log, abort

# Coming soon: Query monitoring rules

**Parameters** **WLM**

## Workload Management Configuration

Workload management (WLM) is an ordered set of query queues that define how resources are allocated and how queries are routed for processing.

[Expand all queues](#)

Queue 1

Concurrency	User groups	Query groups	Match wildcards	Timeout (ms)	Memory (%)
5	<input type="text" value="Enter name"/> data_scientist	<input type="text" value="Enter name"/> fast_queries	<input type="checkbox"/>	0	

### Rules for Default queue

[Add rule](#)

Rule name	Predicate	Action	Delete									
MaxRowCount	<table border="1"><thead><tr><th>Metric name</th><th>Operator</th><th>Value</th></tr></thead><tbody><tr><td>Return row counts (rows)</td><td>&gt;</td><td>10000000</td></tr><tr><td>Query execution time (sec)</td><td>&gt;</td><td>120</td></tr></tbody></table>	Metric name	Operator	Value	Return row counts (rows)	>	10000000	Query execution time (sec)	>	120	Log	<input type="checkbox"/>
Metric name	Operator	Value										
Return row counts (rows)	>	10000000										
Query execution time (sec)	>	120										
MemoryToDisk	<table border="1"><thead><tr><th>Metric name</th><th>Operator</th><th>Value</th></tr></thead><tbody><tr><td>Memory to disk(1MB blocks)</td><td>&gt;</td><td>10000000</td></tr><tr><td>Query execution time (sec)</td><td>&gt;</td><td>120</td></tr></tbody></table>	Metric name	Operator	Value	Memory to disk(1MB blocks)	>	10000000	Query execution time (sec)	>	120	Hop	<input type="checkbox"/>
Metric name	Operator	Value										
Memory to disk(1MB blocks)	>	10000000										
Query execution time (sec)	>	120										

Default queue

Monitor and control cluster resources consumed by a query

Get notified, abort and reprioritize long-running / bad queries

Pre-defined templates for common use cases

# Coming soon: Query monitoring rules

## Common use cases:

- Protect interactive queues

```
INTERACTIVE = { "query_execution_time > 15 sec" or  
                "query_cpu_time > 1500 uSec" or  
                "query_blocks_read > 18000 blocks" } [HOP]
```

- Monitor ad-hoc queues for heavy queries

```
AD-HOC = { "query_execution_time > 120" or  
           "query_cpu_time > 3000" or  
           "query_blocks_read > 180000" or  
           "memory_to_disk > 4000000000000"} [LOG]
```

- Limit the number of rows returned to a client

```
MAXLINES = { "RETURN_ROWS > 50000" } [ABORT]
```

## Benefit #2: Amazon Redshift is inexpensive

DS2 (HDD)	Price per hour for DS2.XL single node	Effective annual price per TB compressed
On-demand	\$ 0.850	\$ 3,725
1 year reservation	\$ 0.500	\$ 2,190
3 year reservation	\$ 0.228	\$ 999

DC1 (SSD)	Price per hour for DC1.L single node	Effective annual price per TB compressed
On-demand	\$ 0.250	\$ 13,690
1 year reservation	\$ 0.161	\$ 8,795
3 year reservation	\$ 0.100	\$ 5,500

### Pricing is simple

Number of nodes x price/hour

No charge for leader node

No upfront costs

Pay as you go

# Benefit #3: Amazon Redshift is fully managed

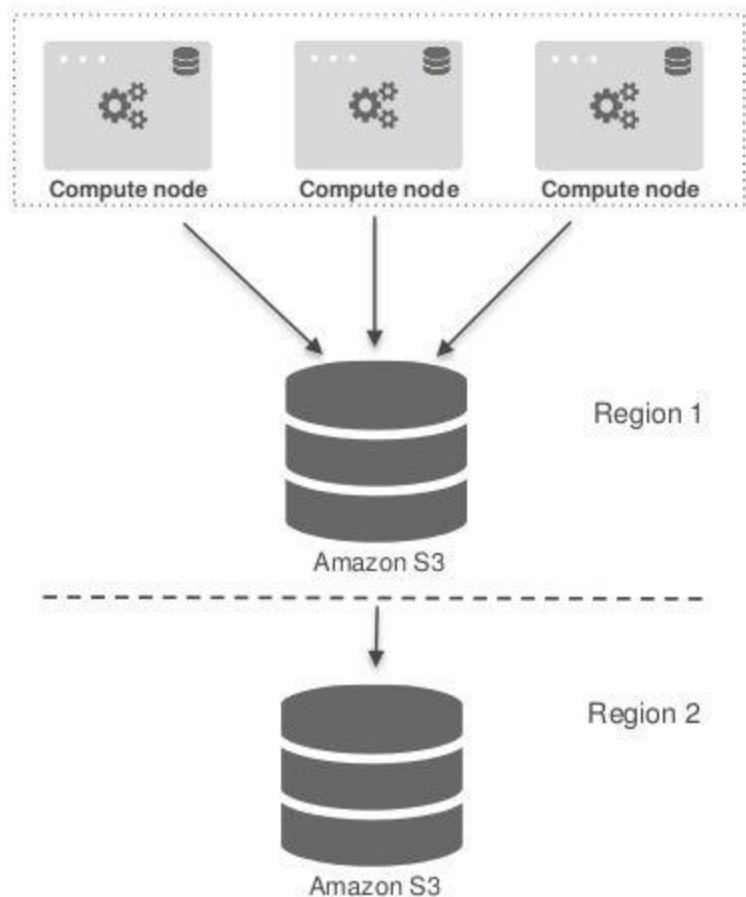
## Continuous/incremental backups

Multiple copies within cluster

Continuous and incremental backups to Amazon S3

Continuous and incremental backups across regions

Streaming restore



# Benefit #3: Amazon Redshift is fully managed

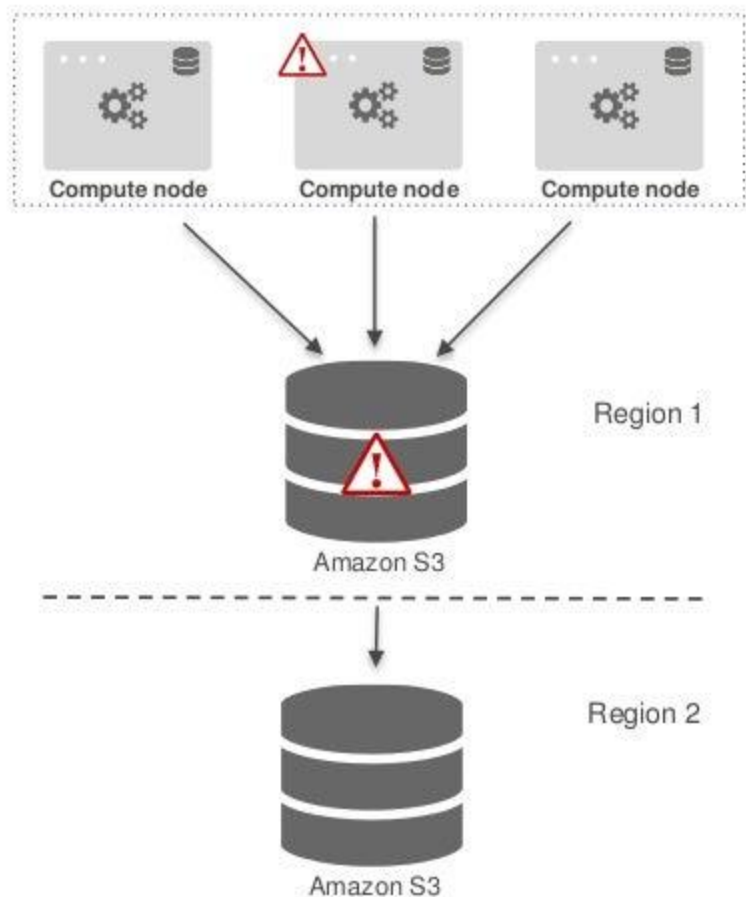
## Fault tolerance

Disk failures

Node failures

Network failures

Availability Zone/region level disasters

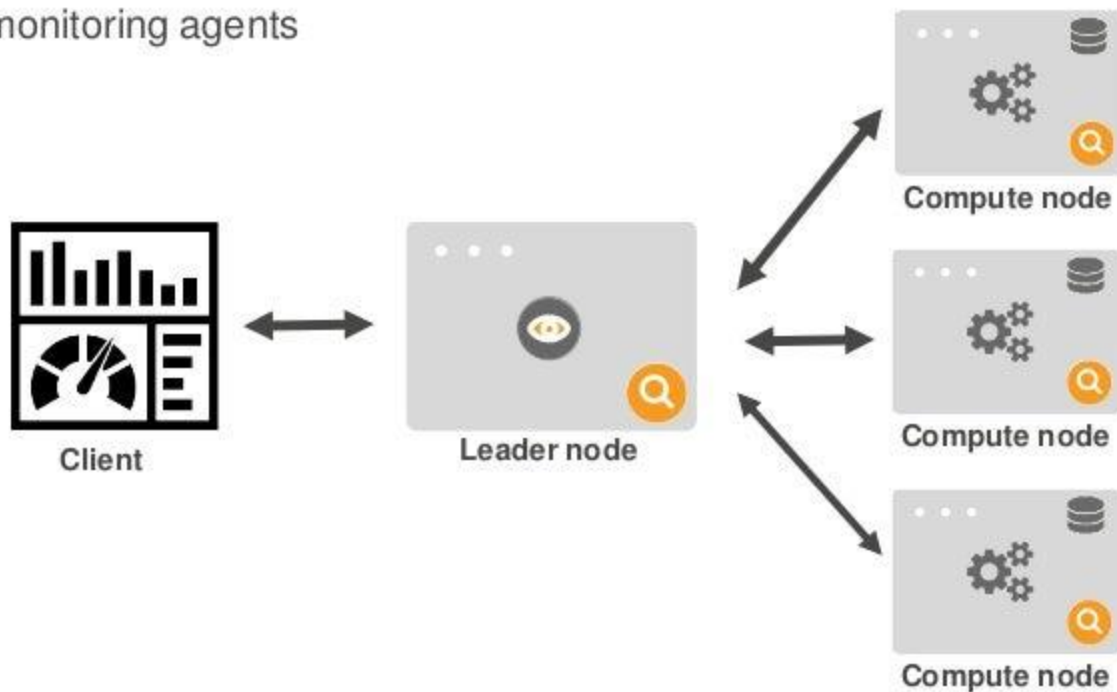




# Node fault tolerance



Data-path monitoring agents

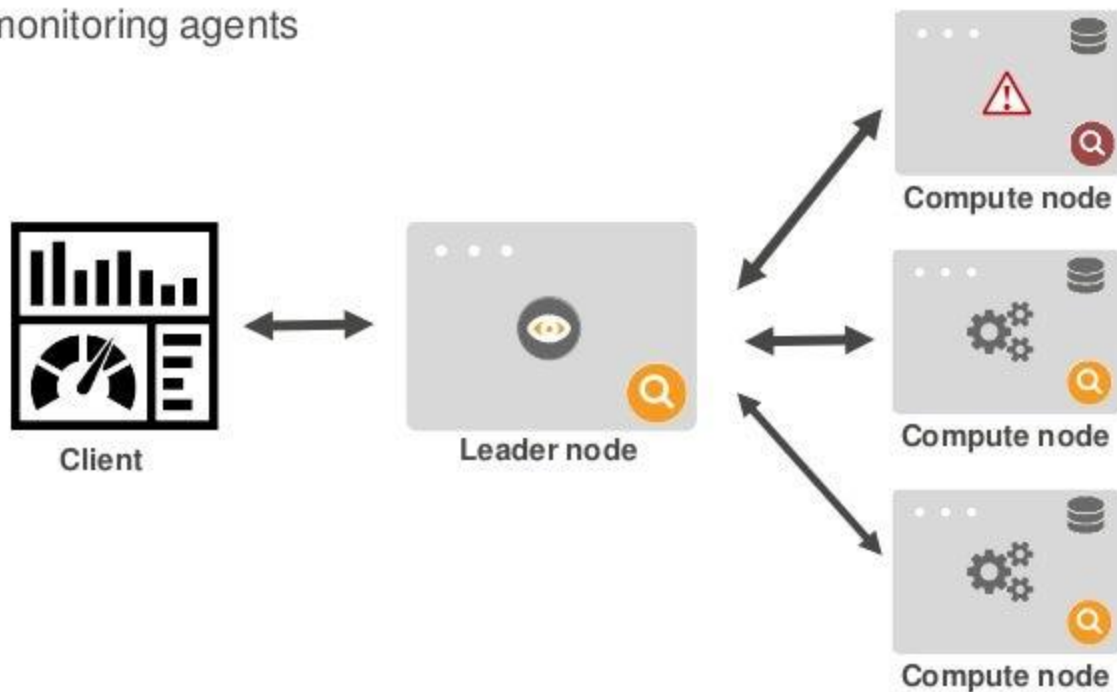


Node level monitoring  
can detect SW/HW  
issues and take action

# Node fault tolerance



Data-path monitoring agents

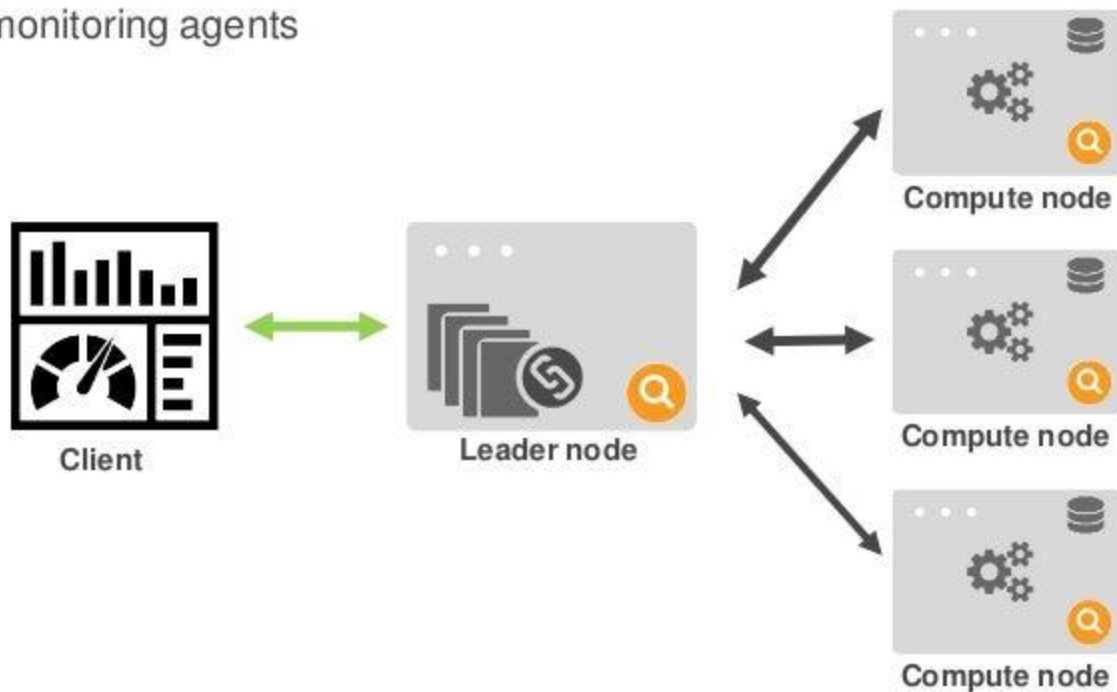


Failure is detected at one of the compute nodes

# Node fault tolerance



Data-path monitoring agents



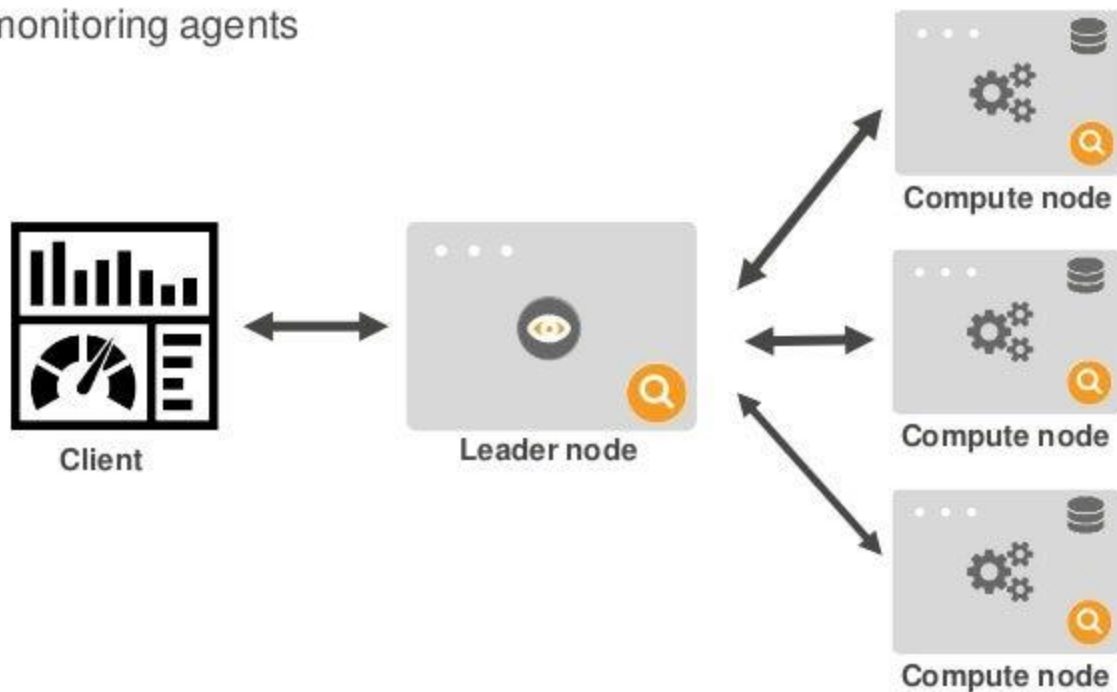
Redshift parks the connections

Next, the node is replaced

# Node fault tolerance



Data-path monitoring agents



Queries are re-submitted

# Node fault tolerance



Data-path monitoring agents



Cluster-level monitoring agents



Client



Leader node



Compute node



Compute node



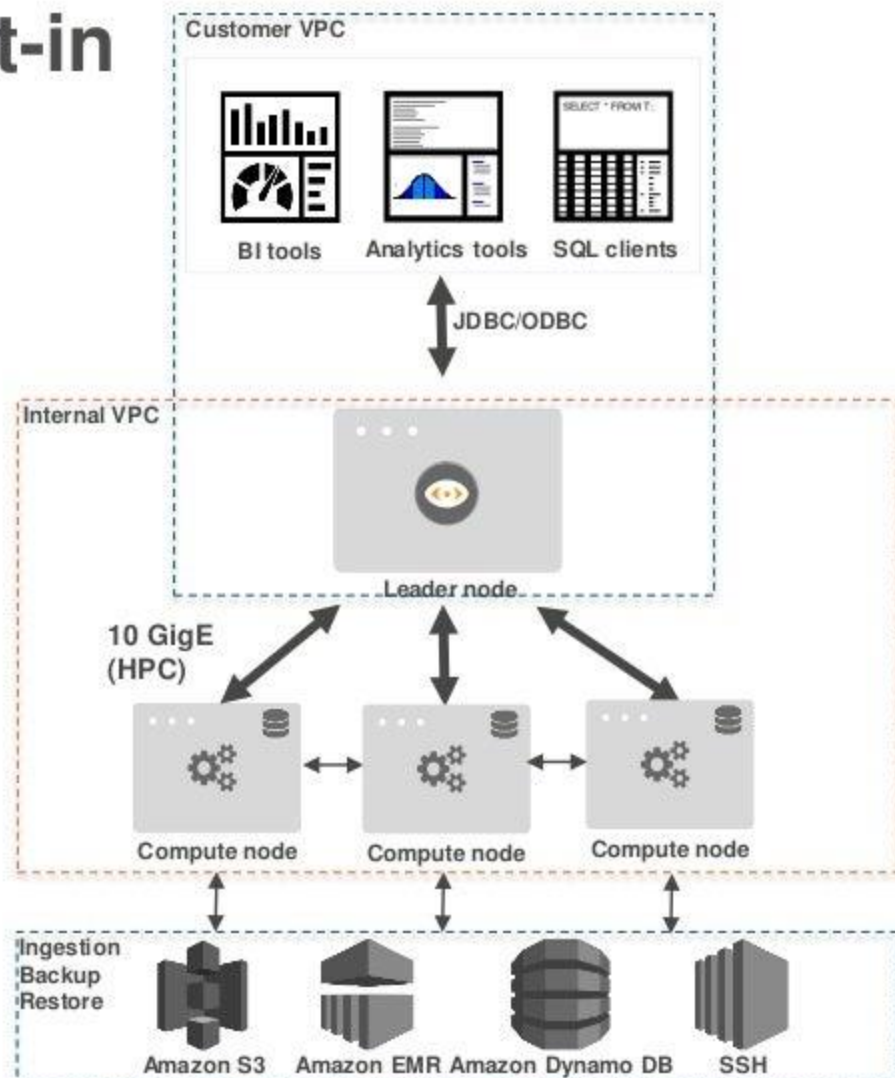
Compute node



Additional monitoring layer for the leader node and network

# Benefit #4: Security is built-in

- ◆ Load encrypted from S3
- ◆ SSL to secure data in transit
  - ◆ ECDHE perfect forward secrecy
- ◆ Amazon VPC for network isolation
- ◆ Encryption to secure data at rest
  - ◆ All blocks on disks and in S3 encrypted
  - ◆ Block key, cluster key, master key (AES-256)
  - ◆ On-premises HSM & AWS CloudHSM support
- ◆ Audit logging and AWS CloudTrail integration
- ◆ SOC 1/2/3, PCI-DSS, FedRAMP, BAA

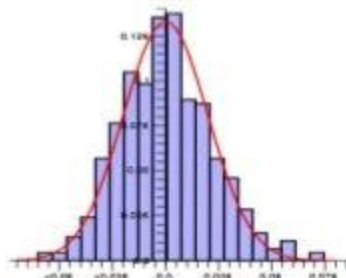




## Benefit #5: Amazon Redshift is powerful

- Approximate functions
- User defined functions
- Machine learning
- Data science

*HyperLogLog: analysis of a near-optimal cardinality algorithm*



# Benefit #6: Amazon Redshift has a large ecosystem

## Data integration



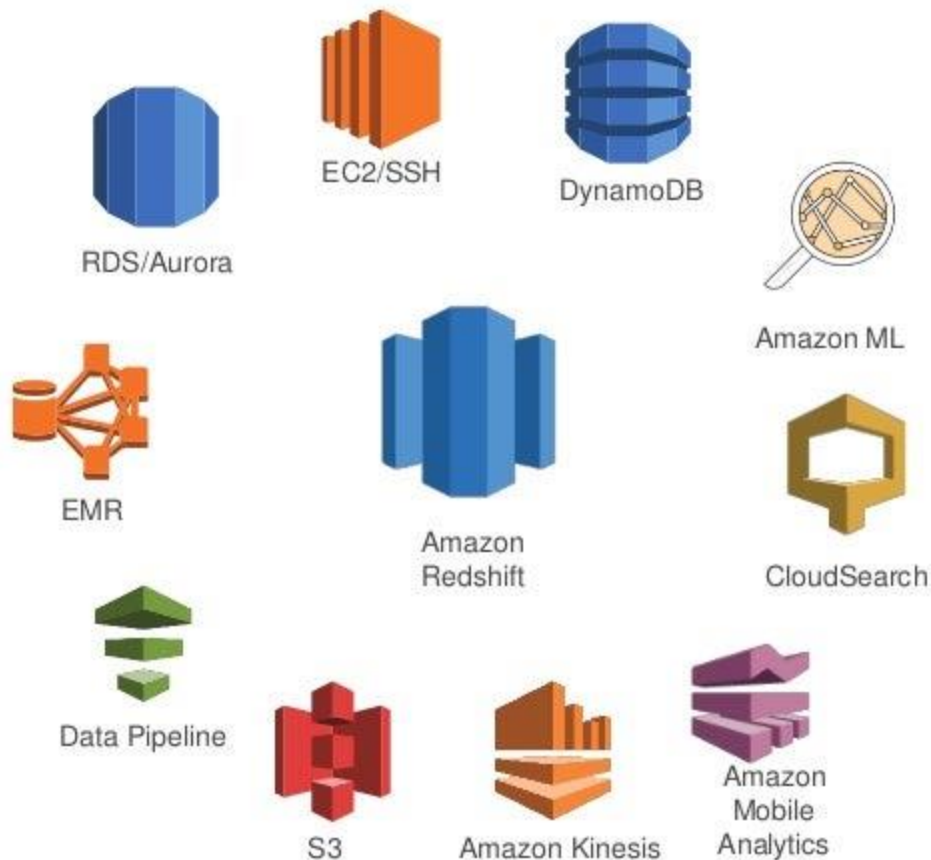
## Business intelligence



## Systems integrators



# Benefit #7: Service oriented architecture



# Use cases

# NTT Docomo: Japan's largest mobile service provider



68 million customers

Tens of TBs per day of data across a mobile network

6 PB of total data (uncompressed)

Data science for marketing operations, logistics, and so on

---

Greenplum on-premises

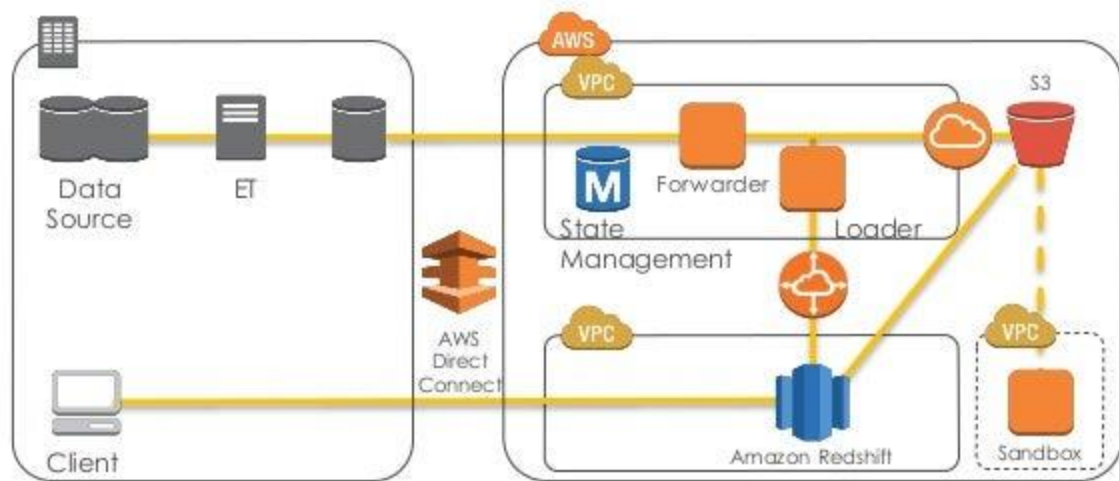
Scaling challenges

Performance issues

Need same level of security

Need for a hybrid environment

# NTT Docomo: Japan's largest mobile service provider



125 node DS2.8XL cluster  
4,500 vCPUs, 30 TB RAM  
2 PB compressed

10x faster analytic queries  
50% reduction in time for new  
BI application deployment  
Significantly less operations  
overhead



# Nasdaq: powering 100 marketplaces in 50 countries



Orders, quotes, trade executions,  
market “tick” data from 7 exchanges  
7 billion rows/day  
Analyze market share, client activity,  
surveillance, billing, and so on

---

Microsoft SQL Server on-premises

Expensive legacy DW  
(\$1.16 M/yr.)

Limited capacity (1 yr. of data  
online)

Needed lower TCO

Must satisfy multiple security  
and regulatory requirements

Similar performance

# Nasdaq: powering 100 marketplaces in 50 countries



23 node DS2.8XL cluster

828 vCPUs, 5 TB RAM

368 TB compressed

2.7 T rows, 900 B derived

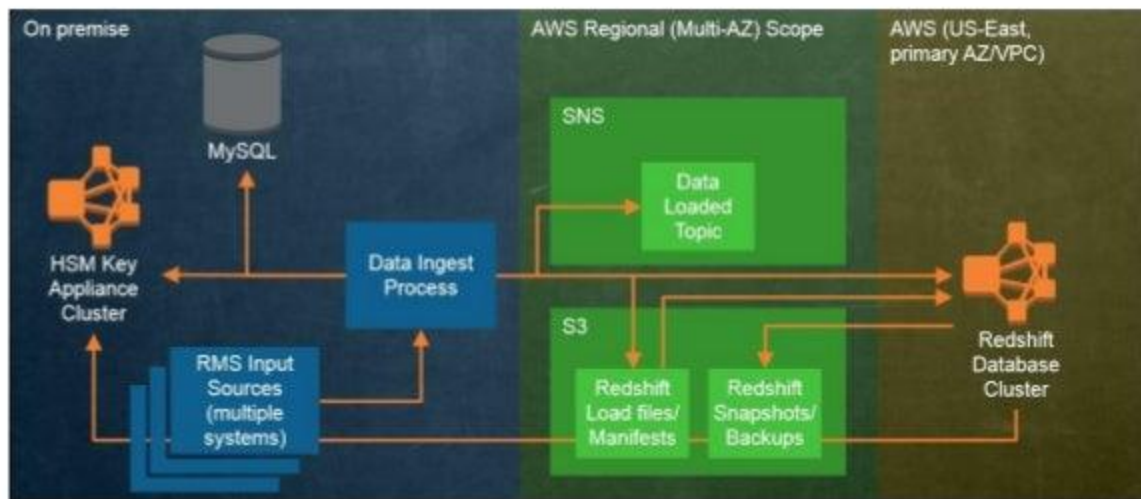
8 tables with 100 B rows

---

7 man-month migration

1/4 the cost, 2x storage, room to grow

Faster performance, very secure



# Amazon.com clickstream analytics

## Web log analysis for Amazon.com

- PBs workload, 2TB/day@67% YoY
- Largest table: 400 TB

## Understand customer behavior

## Previous solution

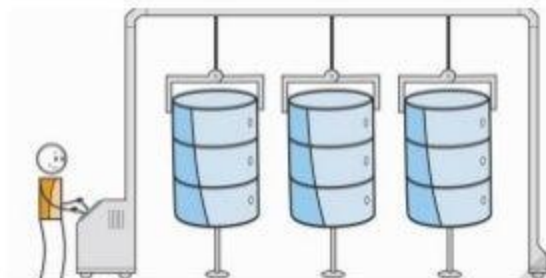
- Legacy DW (Oracle)—query across 1 week/hr
- Hadoop—query across 1 month/hr



# Results with Amazon Redshift



- Query 15 months **in 14 min**
- Load 5B rows in **10 min**
- 21B w/ 10B rows: **3 days to 2 hrs**  
(Hive → Redshift)
- Load pipeline: **90 hrs to 8 hrs**  
(Oracle → Redshift)



- 100 node DS2.8XL clusters
- Easy resizing
- Managed backups and restore
- Failure tolerance and recovery



- 20% time of one DBA
- Increased productivity

# Getting started

# Provisioning

# Enter cluster details

CLUSTER DETAILS

NODE CONFIGURATION

ADDITIONAL CONFIGURATION

REVIEW

Provide the details of your cluster. Fields marked with \* are required.

Cluster Identifier\*

demo

This is the unique key that identifies a cluster. This parameter is stored as a lowercase string. (e.g. my-dw-instance)

Database Name

myredshiftdb

Optional. A default database named dev is created for the cluster. Optionally, specify a custom database name (e.g. mydb) to create an additional database.

Database Port\*

5439

Port number on which the database accepts connections.

Master User Name\*

maor

Name of master user for your cluster. (e.g. awsuser)

Master User Password\*

\*\*\*\*\*

Password must contain 8 to 64 printable ASCII characters excluding: /, ", ', \, and @. It must contain 1 uppercase letter, 1 lowercase letter, and 1 number.

Confirm Password\*

\*\*\*\*\*

Confirm Master User Password.

Cancel

Continue



# Select node configuration

CLUSTER DETAILS

NODE CONFIGURATION

ADDITIONAL CONFIGURATION

REVIEW

Choose a number of nodes and Node Type below. Number of Compute Nodes is required for multi-node clusters.

Node Type **dc1.large**

Specifies the compute, memory, storage, and I/O capacity of the cluster's nodes.

CPU 7 EC2 Compute Units (2 virtual cores) per node

Memory 15 GiB per node

Storage 160GB SSD storage per node

I/O Performance Moderate

Cluster Type **Single Node**

Number of Compute Nodes\*

1

Single Node clusters consist of a single node which performs both leader and compute functions.

Maximum 1

Minimum 1

Cancel

Previous

Continue

# Select security settings and provision

CLUSTER DETAILS   NODE CONFIGURATION   **ADDITIONAL CONFIGURATION**   REVIEW

Provide the optional additional configuration details below.

**Cluster Parameter Group**  Parameter group to associate with this cluster.

**Encrypt Database** ☒ None ☐ KMS ☐ HSM [Learn more about database encryption](#)

Configure Networking Options:

**Choose a VPC**  The identifier of the VPC in which you want to create your cluster.

**Cluster Subnet Group**  Selected Cluster Subnet Group may limit the choice of Availability Zones.

**Publicly Accessible** ☒ Yes ☐ No Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network.

**Choose a Public IP Address** ☐ No ☒ Yes Select Yes if you want to select your own public IP address from a list of elastic IP (EIP) addresses that are already configured for your cluster's VPC. Select No if you want Amazon Redshift to provide an EIP for you instead.

**Availability Zone**  The EC2 Availability Zone that the cluster will be created in.

Optionally, associate your cluster with one or more security groups.

**VPC Security Groups**  List of VPC Security Groups to associate with this cluster.

Optionally, create a basic alarm for this cluster.

**Create CloudWatch Alarm** ☐ Yes ☒ No Create a CloudWatch alarm to monitor the disk usage of your cluster.

Optionally, associate up to 10 IAM roles with this cluster.

**Available Roles**

Cancel

Previous

Continue

# Point-and-click resize



### Resize Cluster

Choose the number of nodes and optionally a new node type for the resize operation. Note that the available node type and cluster type options may be limited by the cluster's current availability zone.

Node type  ⓘ

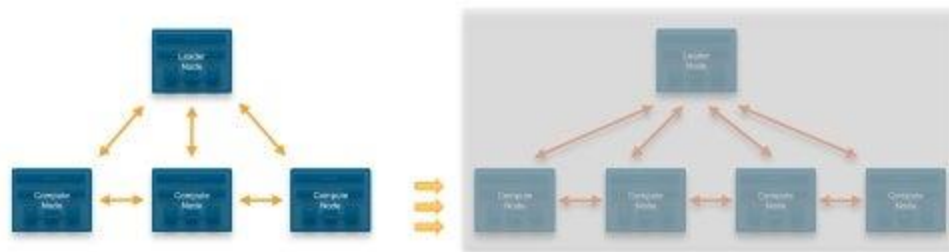
Cluster type  ⓘ

Number of nodes\*  ⓘ

Please make sure the resized cluster is large enough to hold the data that is currently on the cluster; otherwise the resize will fail.

Warning: Resizing the cluster will cause it to be restarted into read-only mode for the duration of the resize operation. All currently executing queries and database connections on the cluster will be terminated when the resize operation begins and again when it is complete. For more information, see [Resizing a cluster](#).

# Resize



- Resize while remaining online
- Provision a new cluster in the background
- Copy data in parallel from node to node
- Only charged for source cluster

# Data modeling

# Zone maps

SELECT COUNT(\*) FROM LOGS WHERE DATE = '09-JUNE-2013'

## Unsorted table



## Sorted by date



## Sort Keys

- Single column
- Compound
- Interleaved



# Single Column

[ SORTKEY ( date ) ]

Date	Region	Country
2-JUN-2015	Oceania	New Zealand
2-JUN-2015	Asia	Singapore
2-JUN-2015	Africa	Zaire
2-JUN-2015	Asia	Hong Kong
3-JUN-2015	Europe	Germany
3-JUN-2015	Asia	Korea

- Best for:
  - Queries that use 1<sup>st</sup> column (i.e. *date*) as primary filter
  - Can speed up joins and group bys
  - Quickest to VACUUM

# Compound

- Table is sorted by 1<sup>st</sup> column , then 2<sup>nd</sup> column etc.  
[ SORTKEY COMPOUND ( date, region, country) ]

Date	Region	Country
2-JUN-2015	Africa	Zaire
2-JUN-2015	Asia	Korea
2-JUN-2015	Asia	Singapore
2-JUN-2015	Europe	Germany
3-JUN-2015	Asia	Hong Kong
3-JUN-2015	Asia	Korea

- Best for:
  - Queries that use 1<sup>st</sup> column as primary filter, then other cols
  - Can speed up joins and group bys
  - Slower to VACUUM

## Interleaved

- Equal weight is given to each column.

[ SORTKEY INTERLEAVED ( date, region, country) ]

Date	Region	Country
2-JUN-2015	Africa	Zaire
3-JUN-2015	Asia	Singapore
2-JUN-2015	Asia	Korea
2-JUN-2015	Europe	Germany
3-JUN-2015	Asia	Hong Kong
2-JUN-2015	Asia	Korea

- Best for:
  - Queries that use different columns in filter
  - Queries get faster the more columns used in the filter
  - Slowest to VACUUM

- EVEN
- KEY
- ALL

# Distribution

ID	Gender	Name
101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



ID	Gender	Name
101	M	John Smith
306	F	Lisa Green



ID	Gender	Name
292	F	Jane Jones
209	M	James White



ID	Gender	Name
139	M	Peter Black
164	M	Brian Snail



ID	Gender	Name
446	M	Pat Partridge
658	F	Sarah Cyan

# Distribution

ID	Gender	Name
101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



ID	Gender	Name
101	M	John Smith
306	F	Lisa Green



ID	Gender	Name
292	F	Jane Jones
209	M	James White



ID	Gender	Name
139	M	Peter Black
164	M	Brian Snail



ID	Gender	Name
446	M	Pat Partridge
658	F	Sarah Cyan



# Distribution

ID	Gender	Name
101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



ID	Gender	Name
101	M	John Smith
139	M	Peter Black
446	M	Pat Partridge
164	M	Brian Snail
209	M	James White



ID	Gender	Name
292	F	Jane Jones
658	F	Sarah Cyan
306	F	Lisa Green



# Distribution

ID	Gender	Name
101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green



101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green

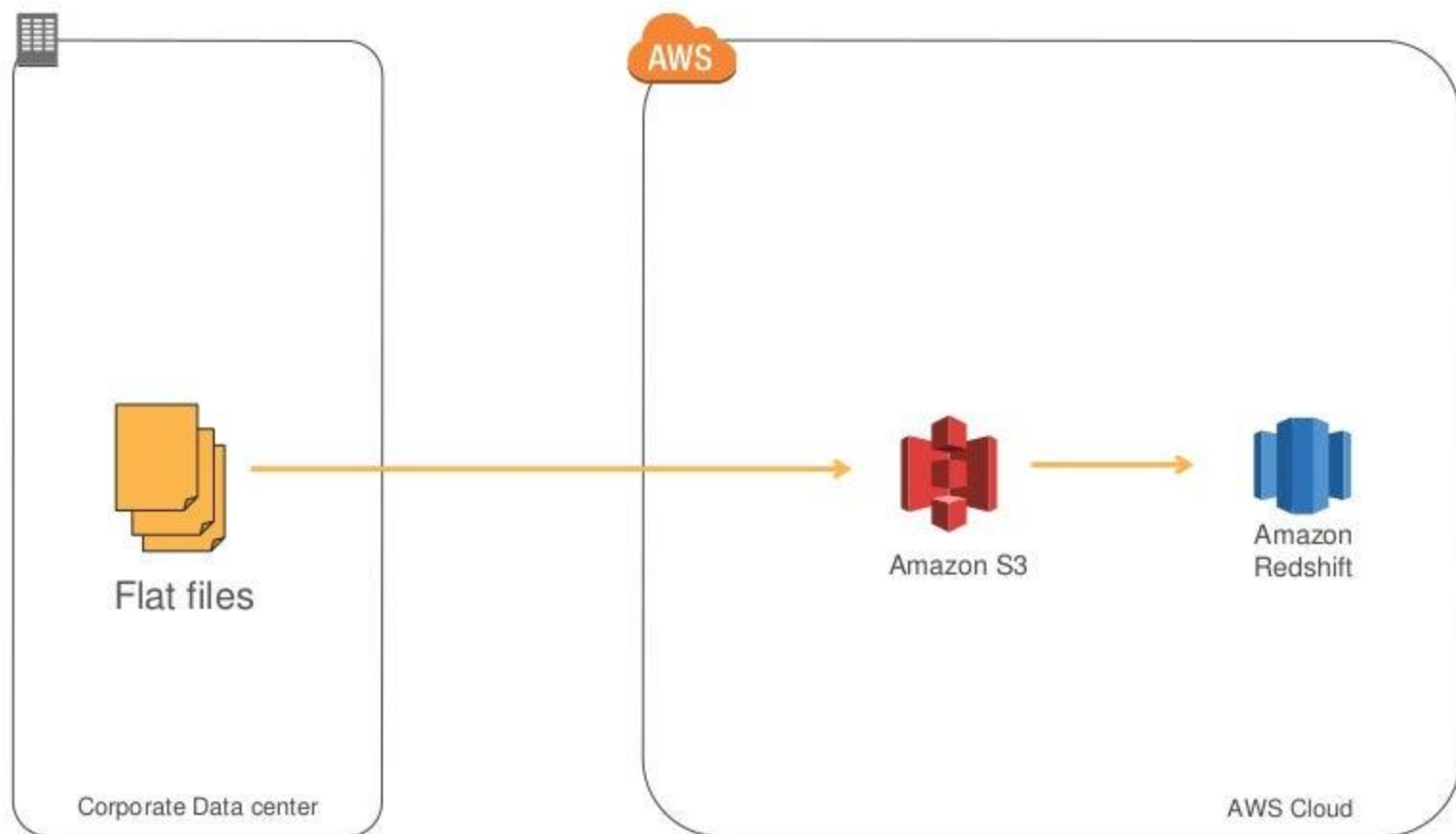


101	M	John Smith
292	F	Jane Jones
139	M	Peter Black
446	M	Pat Partridge
658	F	Sarah Cyan
164	M	Brian Snail
209	M	James White
306	F	Lisa Green

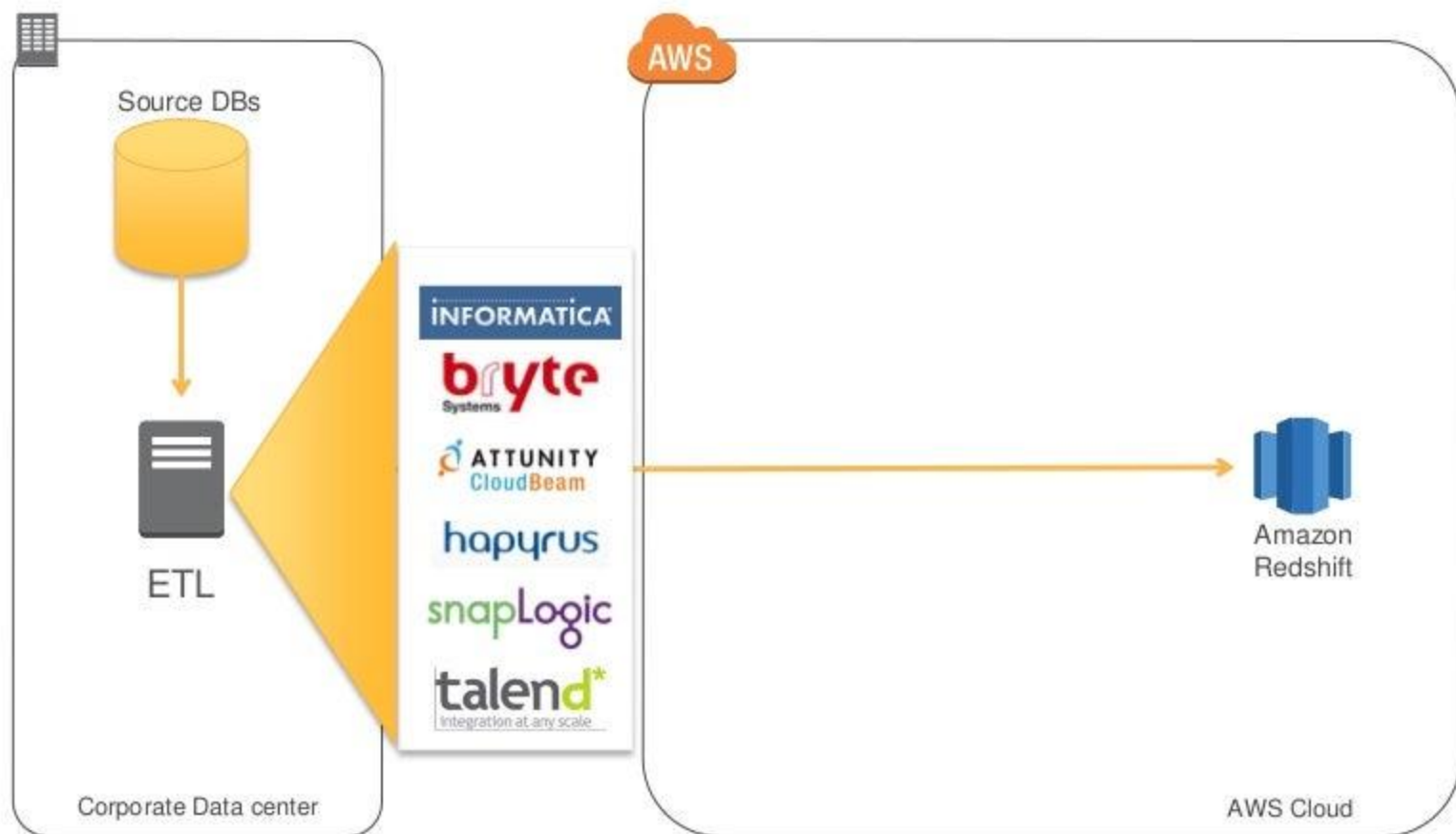
- EVEN
  - Tables with no joins or group by
- KEY
  - Large Fact tables
  - Large dimension tables
- ALL
  - Medium dimension tables (1K – 2M)
  - Small dimension tables

# Loading data

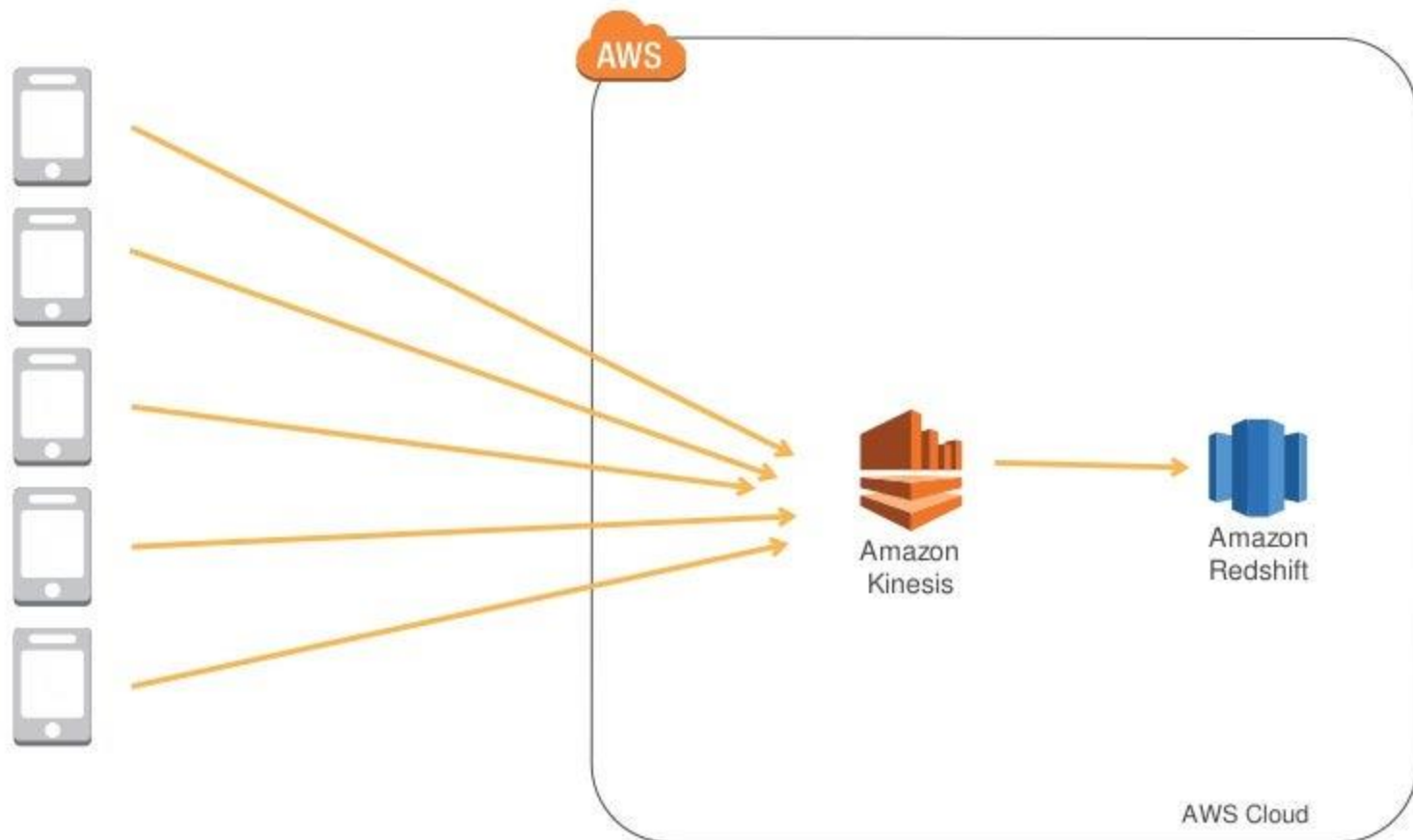
# Data loading options



# Data loading options



# Data loading options



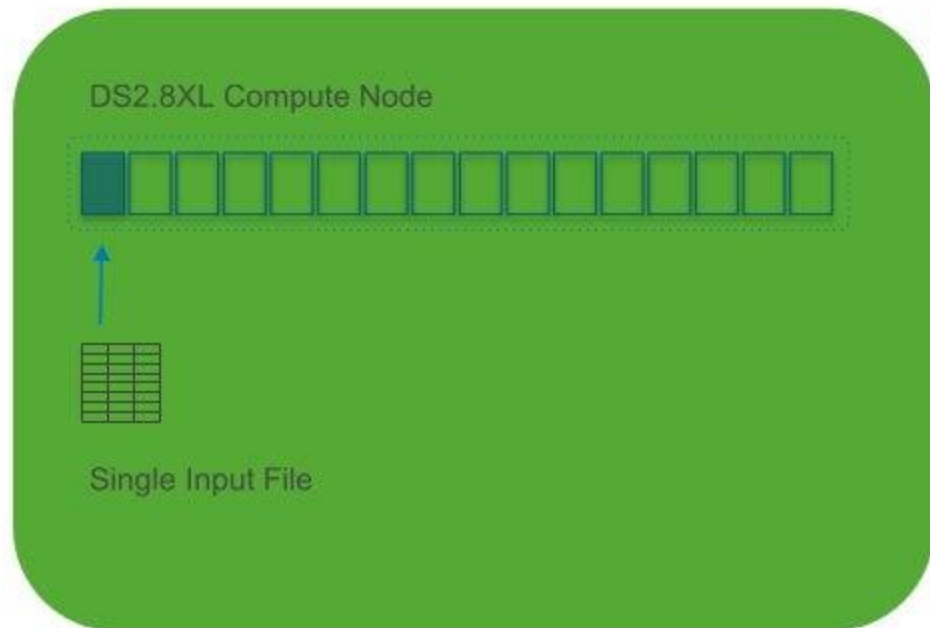
# Use multiple input files to maximize throughput

Use the COPY command

Each slice can load one file at a time

A single input file means only one slice is ingesting data

Instead of 100MB/s, you're only getting 6.25MB/s





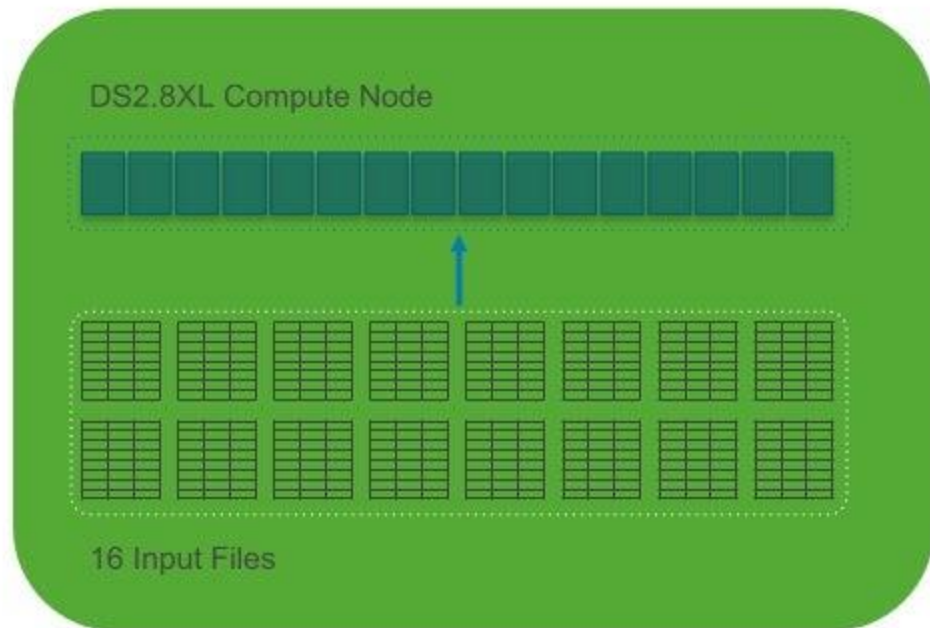
# Use multiple input files to maximize throughput

Use the COPY command

You need at least as many input files as you have slices

With 16 input files, all slices are working so you maximize throughput

Get 100MB/s per node; scale linearly as you add nodes



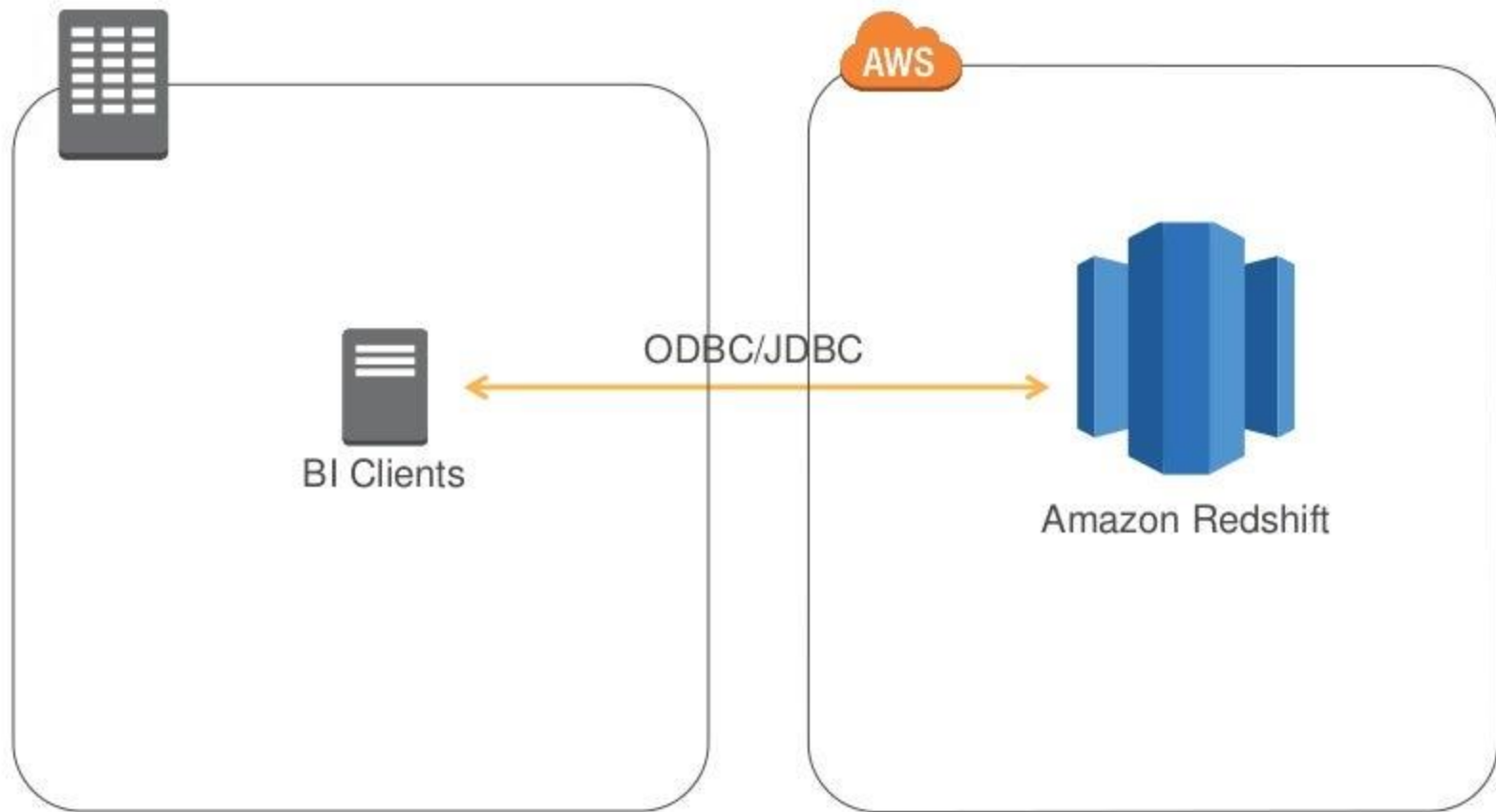
# Querying

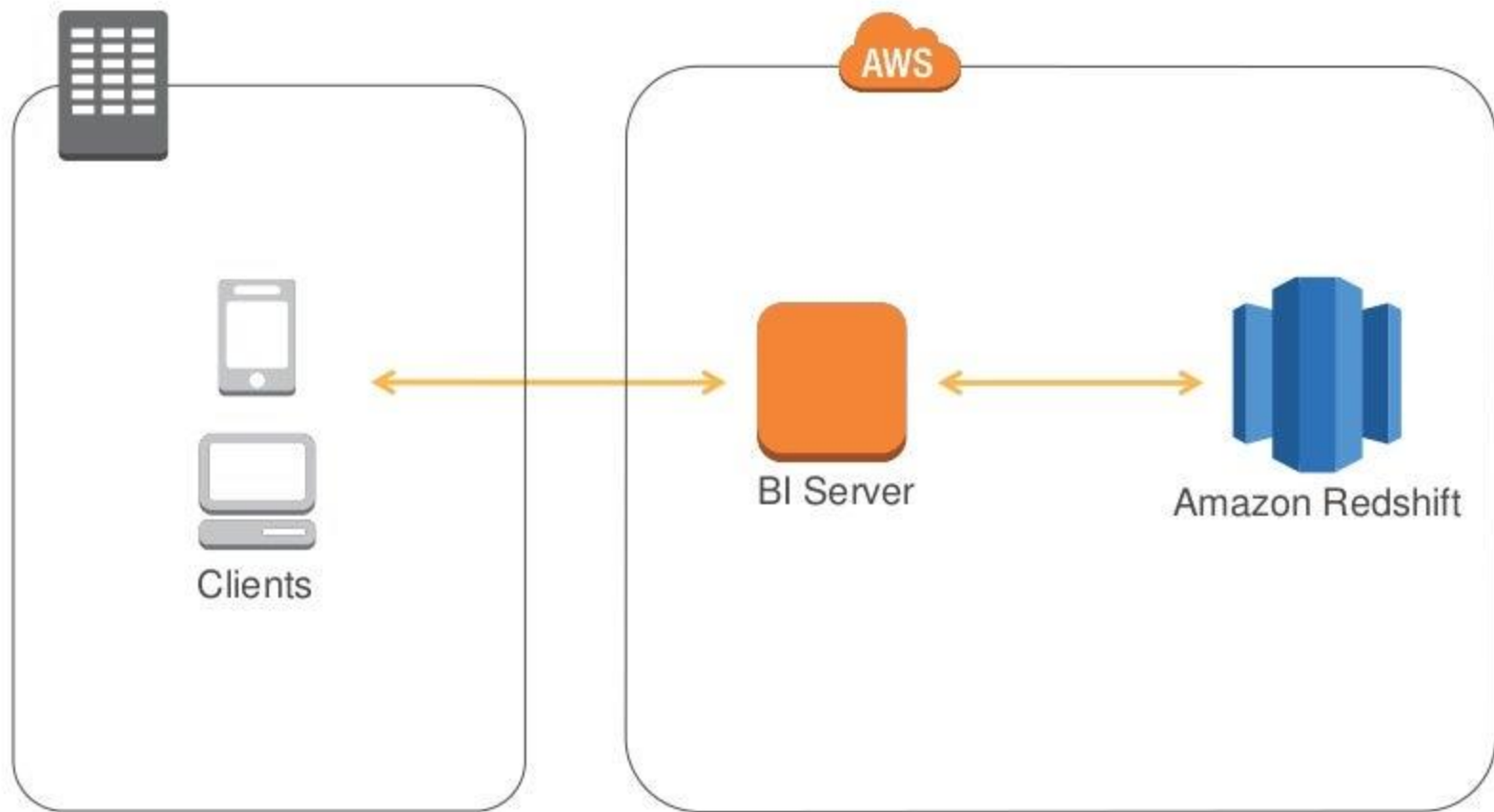
# Amazon Redshift works with your existing BI tools



JDBC/ODBC







## View explain plans

▼ Query Execution Details

Plan	Actual
------	--------

```

XN Hash Join DS_BCAST_INNER (cost=52602972108.91..68656434844.99 rows=23608 width=71)
  -> XN Seq Scan on supplier s (cost=0.00..10000.00 rows=1000000 width=22)
  -> XN Hash (cost=52602972049.89..52602972049.89 rows=23608 width=57)
    -> XN Hash Join DS_BCAST_INNER (cost=1752.83..52602972049.89 rows=23608 width=57)
      -> XN Hash Join DS_BCAST_INNER (cost=1718.72..52499650202.87 rows=70082 width=61)
        -> XN Hash (cost=31.95..31.95 rows=861 width=4)
          -> XN Hash Join DS_BCAST_INNER (cost=624.99..14000065191.80 rows=714354 width=61)
            -> XN Hash (cost=874.99..874.99 rows=87499 width=33)
              -> XN Seq Scan on lineorder lo (cost=0.00..149965.90 rows=14996590 width=17)
              -> XN Hash (cost=499.99..499.99 rows=49999 width=20)
                -> XN Seq Scan on part p (cost=0.00..499.99 rows=49999 width=20)

```

	Plan	Actual
Revenue		
Cost of sales		
Operating expenses		
Operating income		
Interest expense		
Income before taxes		
Income taxes		
Net income		

Expand Level

Collapse Level

Expand All

- Hash Join (Broadcasted Inner Table)
  - Sequential Scan on supplier s
- Hash
  - Hash Join (Broadcasted Inner Table)
  - Hash Join (Broadcasted Inner Table)
    - Hash Join (Broadcasted Inner Table)
      - Sequential Scan on lineorder lo
    - Hash
      - Sequential Scan on part p
  - Hash
    - Sequential Scan on customer c
- Hash
  - Sequential Scan on dwdate d



select

```
lo_orderkey,  
p_name,  
c_name,  
s_address,  
lo_quantity
```

from

```
lineorder lo,
part p,
supplier s,
customer c,
dwdate d
```

where

```
lo_custkey = c_custkey
and      lo_partkey = p_partkey
and      lo_suppkey = s_suppkey
and      lo_orderdate = d_datekey
and      d_sellingseason = 'Summer'
```

# Monitor query performance

Cluster: demo

Configuration

Status

Performance

Queries

Loads

Table restore

Time Range: Last Hour

Period: 5 Minutes

Statistic: Average

All Metrics

Nodes

Refresh

Queries

Hover over the queries graph or click on a query ID in the legend to inspect queries. Click and drag on any graph to zoom in.



CPU Utilization



NetworkReceiveThroughput





# Resources

## Detail Pages

- <http://aws.amazon.com/redshift>
- <https://aws.amazon.com/marketplace/redshift/>
- <https://aws.amazon.com/redshift/developer-resources/>
- [Amazon Redshift Utilities - GitHub](#)

## Best Practices

- [http://docs.aws.amazon.com/redshift/latest/dg/c\\_loading-data-best-practices.html](http://docs.aws.amazon.com/redshift/latest/dg/c_loading-data-best-practices.html)
- [http://docs.aws.amazon.com/redshift/latest/dg/c\\_designing-tables-best-practices.html](http://docs.aws.amazon.com/redshift/latest/dg/c_designing-tables-best-practices.html)
- <http://docs.aws.amazon.com/redshift/latest/dg/c-optimizing-query-performance.html>

**Thank you!**