
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ANT345

Amazon Redshift: 10 years of innovation in integration, data sharing & more

Neeraja Rentachintala

Senior Manager, Product Management
Amazon Redshift

Ippokratis Pandis

VP/Distinguished Engineer
Amazon Redshift

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift

TENS OF THOUSANDS OF CUSTOMERS USE AMAZON REDSHIFT AND PROCESS EXABYTES OF DATA PER DAY



Easy, secure & reliable



Focus on getting from data to insights in seconds without worrying about infrastructure

Analyze all your data



Get insights running real-time and predictive analytics on complex, scaled data across your operational databases, data lake, and data warehouse

Best price performance at any scale



Gain up to 5x better price performance than other cloud data warehouses, and dynamically scale to improve query speed for complex and critical workloads

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift enables high performance analytics at scale across all data for all users

Core data warehouse pillars

- Security & Availability
- Performance
- Compute and storage elasticity
- Serverless
- Autonomics



Amazon Redshift

AWS service integrations

- Analytics for all data – data warehouse, data lake, streaming & operational databases
- Analytics for all users - BI, machine learning & advanced analytics

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

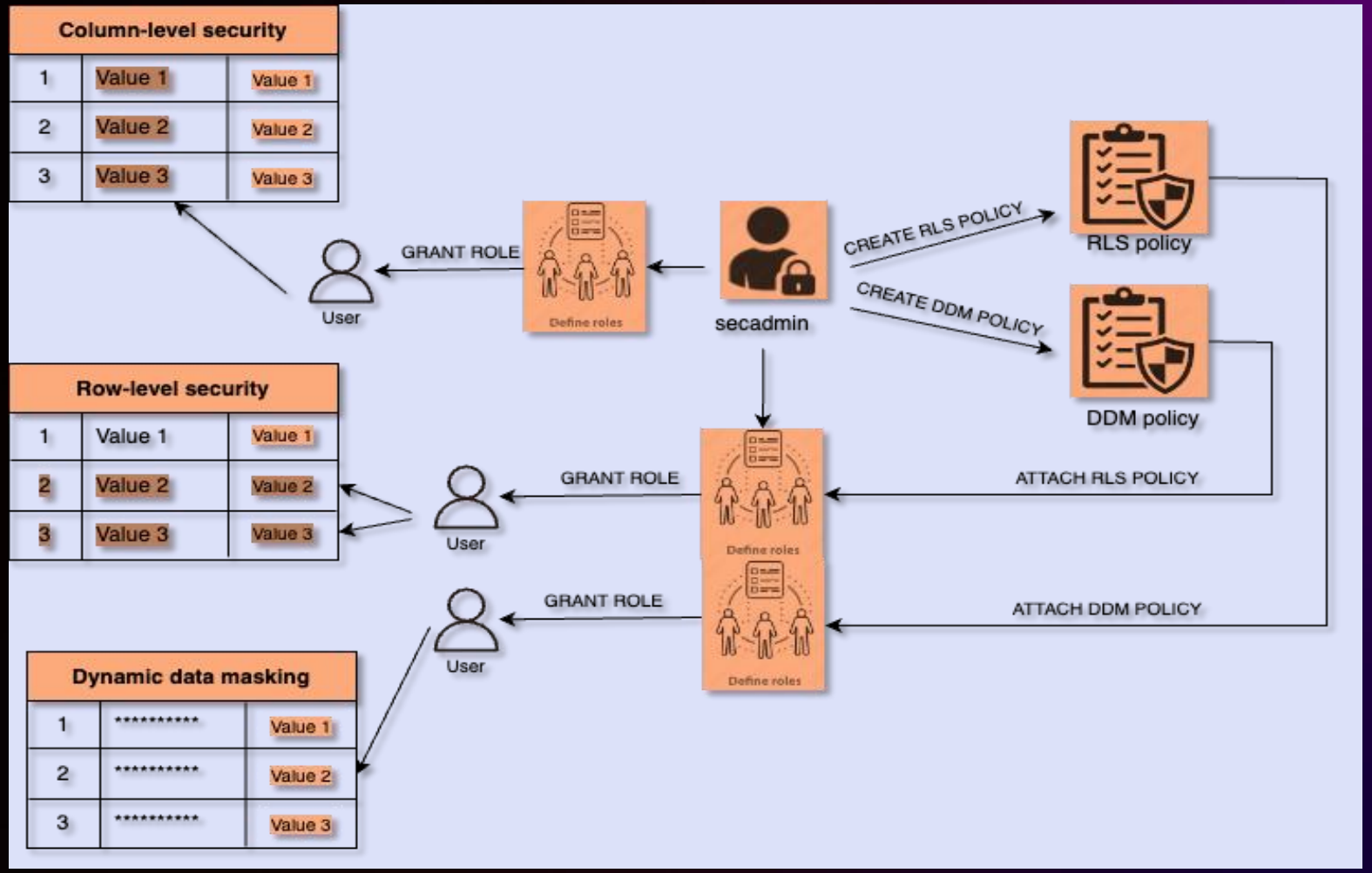
The Amazon Redshift pillars

1. Security and Availability
2. Performance
3. Storage and Compute Elasticity
4. Autonomics
5. Serverless

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

Industry-leading security and access control, out-of-the box at no additional cost

- AWS security standards and compliance certifications
- Granular authorization at row and column level
- Amazon Virtual Private Cloud (VPC) support
- Identity management and federation
- Sensitive data protection with dynamic data masking
- Auditing and compliance

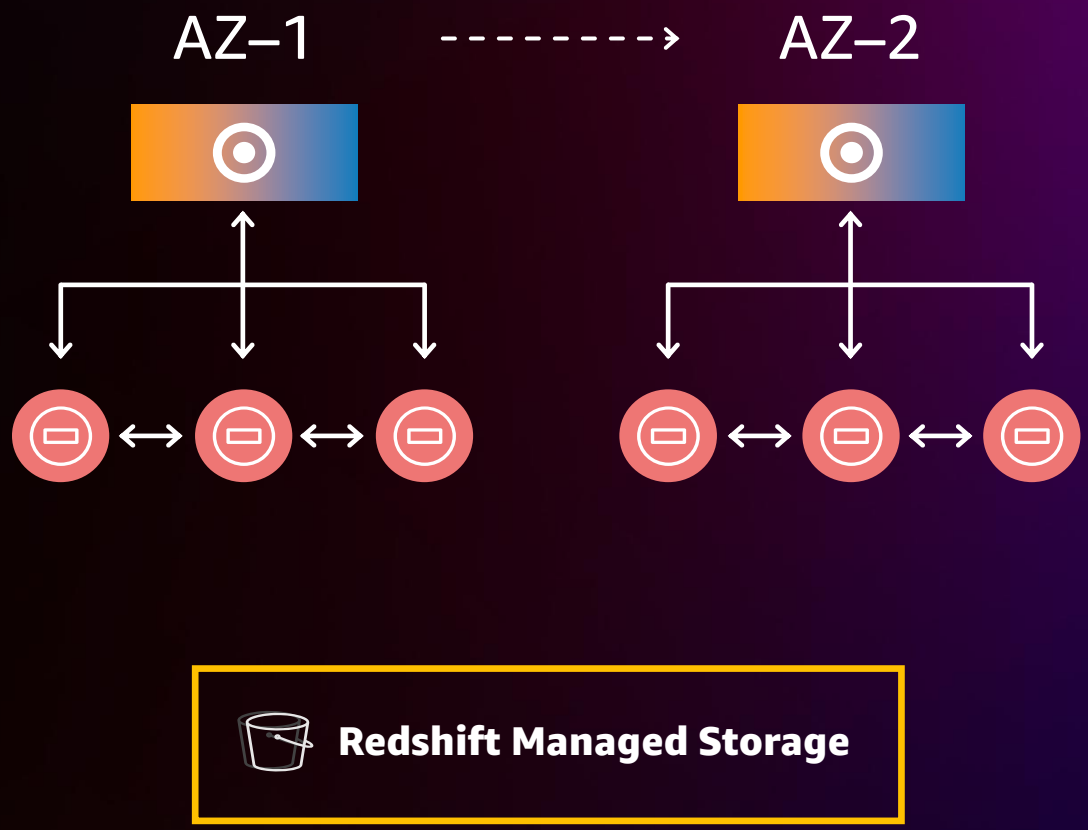


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cross-AZ cluster recovery

RELOCATE YOUR CLUSTER IN ANOTHER AZ IN RESPONSE TO FAILURE

- Avoid resource constraints in the current AZ during resume and resize
- Relocate with zero data loss (RPO = 0)
- No need to restore from a snapshot
- Cluster is created in another AZ on-demand, so cost of a standby replica cluster is avoided



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Amazon Redshift Multi-AZ (Preview)



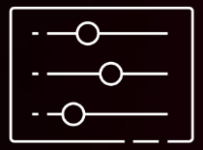
Data Sources

Highly scalable, high performance business critical applications



Amazon Redshift

Cloud Data Warehouse



Enable Multi-AZ for maximum availability



Low RTO during AZ failures



Zero RPO w/ Redshift Managed Storage



No application changes required



Active-Active workload processing

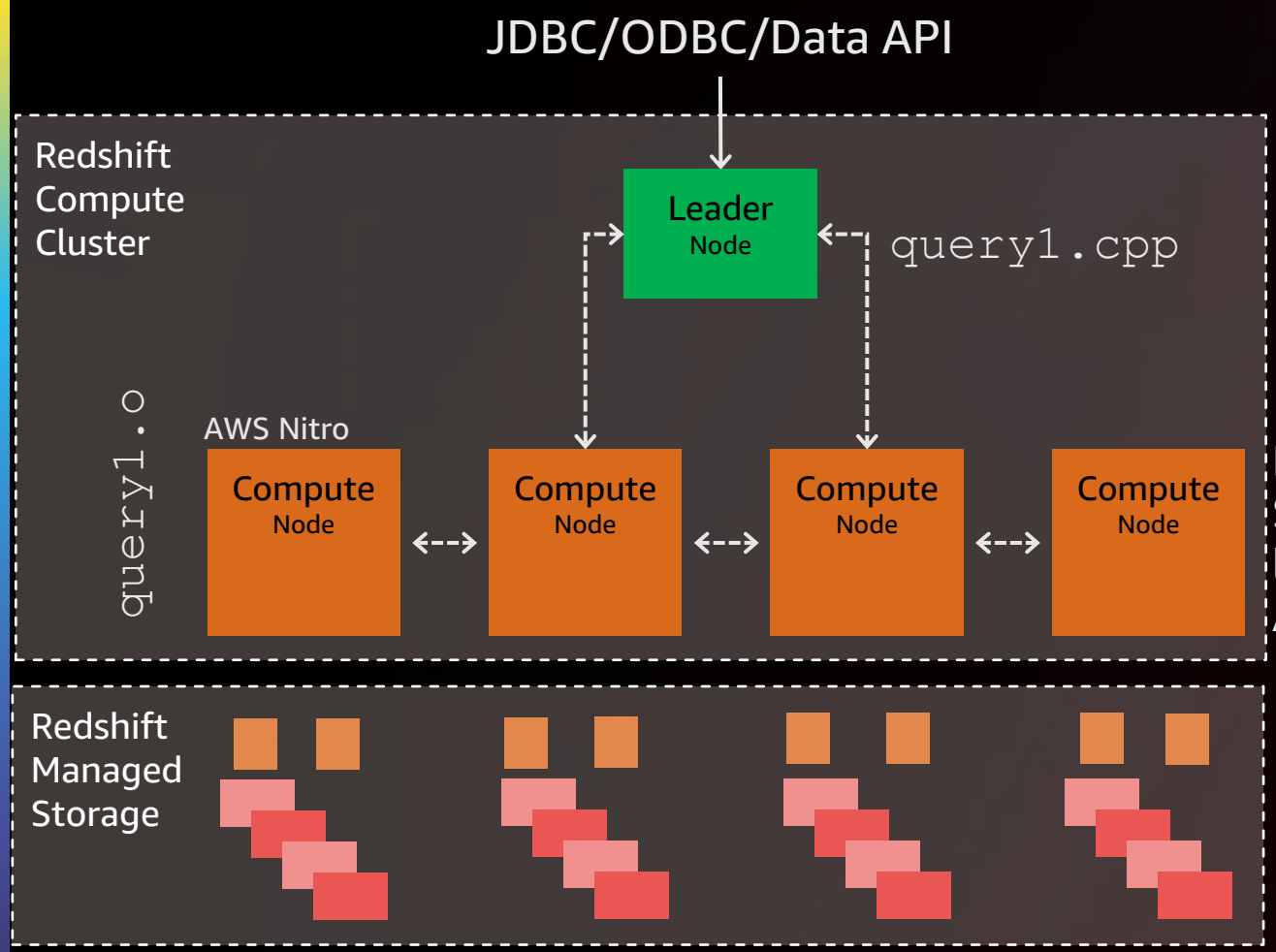
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

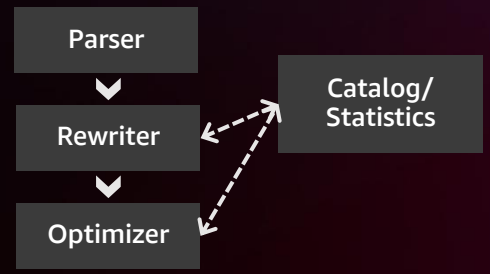
The Amazon Redshift pillars

1. Security and Availability
2. Performance
3. Storage and Compute Elasticity
4. Autonomics
5. Serverless

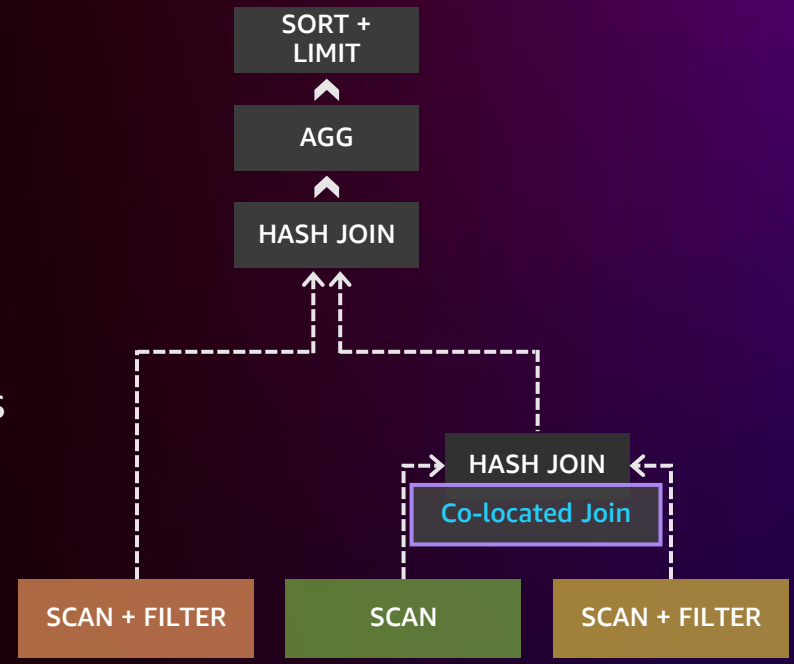
aws
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Executing a query in Amazon Redshift





Min/max pruning
SIMD scans from local-attached SSDs
AZ64 encoding



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Code generation, Prefetching, Vectorization,

••• **SELECT sum(R.val) FROM R, S WHERE R.key = S.key AND R.val < 50**

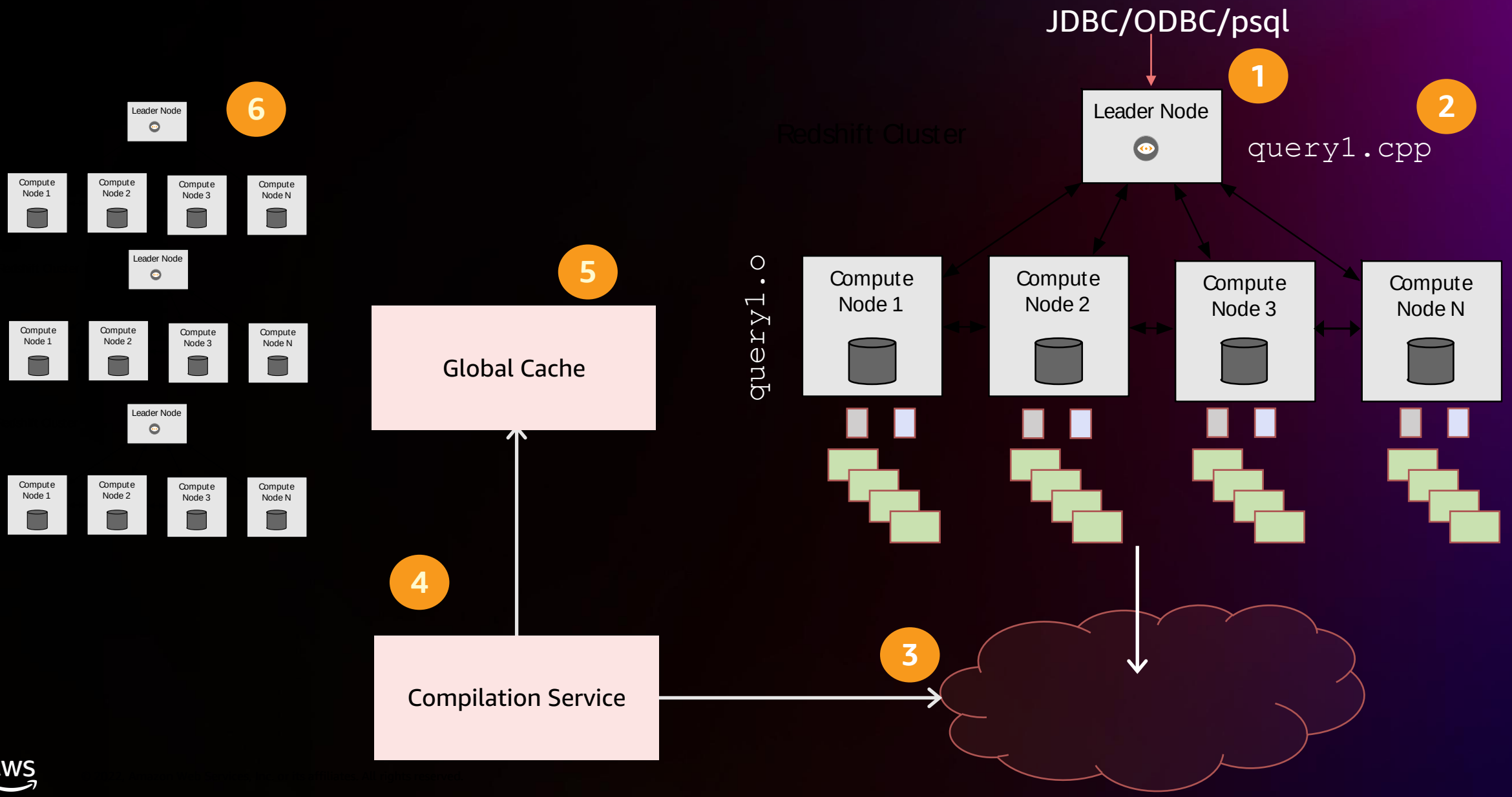
```
// Loop over the tuples of R.
while (scan_step->has_next()) {
  // Get next value for R.key.
  auto field1 = fetcher1.get_next();
  // Get next value for R.val.
  auto field2 = fetcher2.get_next();
  // Apply predicate R.val < 50.
  if (field2 < 50) {
    // Hash R.key and probe the hash table.
    size_t h1 = hash(field1) & (ht1_sz - 1);
    for (auto* p1 = ht1[h1];
         p1 != nullptr; p1 = p1->next) {
      // Evaluate the join condition R.key = S.key.
      if (field1 == p1->field1) sum1 += field2;
    }
  }
}
```

```
// Hash R.key and probe the hash table.
size_t h1 = hash(field1) & (ht1_sz - 1);
// Prefetch the current tuple.
prefetch(ht1[h1]);
// Push the current tuple in the buffer.
prefetchbuffer1.push(field1, field2);
// Pop a prefetched earlier tuple.
h1, field1, field2 =
    prefetchbuffer1.pop();
// Probe the earlier tuple.
for (auto* p1 = ht1[h1];
     p1 != nullptr; p1 = p1->next) {
   ...
}
```

+ AVX2 Vectorization

+ Min/Max Pruning, Late Materialization, …

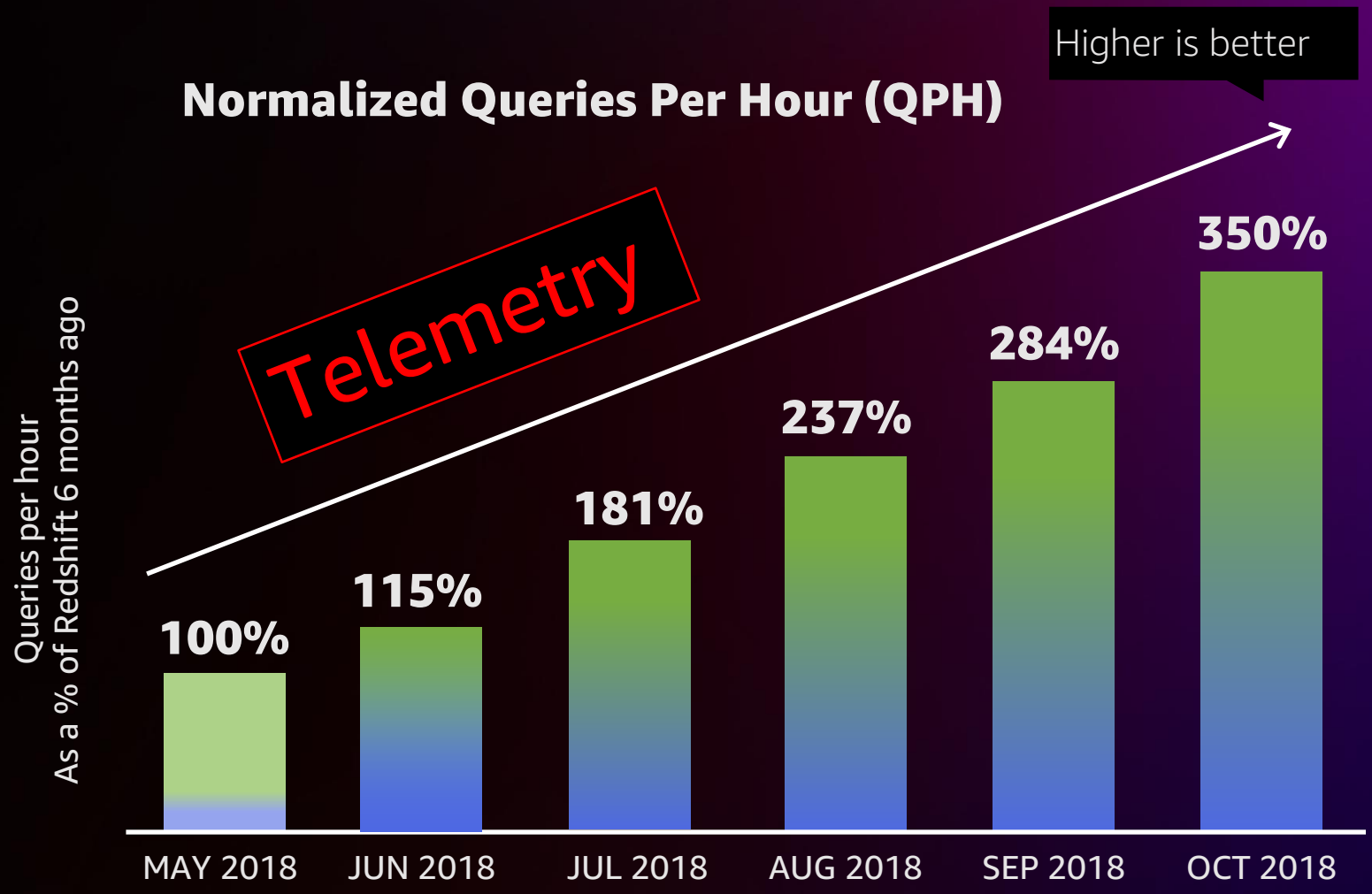
Compilation-as-a-Service



Continuous telemetry and benchmarking

>3x faster between May'18 and Oct'18

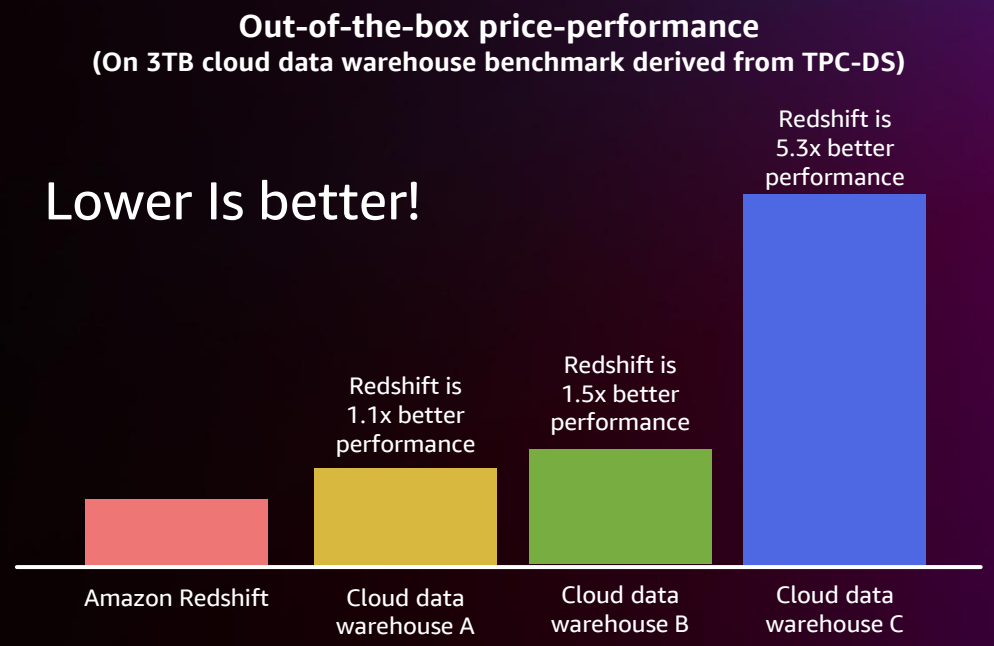
- Short query acceleration
- Query result caching
- Compiled code cache
- Query rewrites for selective joins
- Hash join memory utilization
- Faster string manipulation
- Cache line prefetching for HashJoins and HashAggregations
- Late materialization

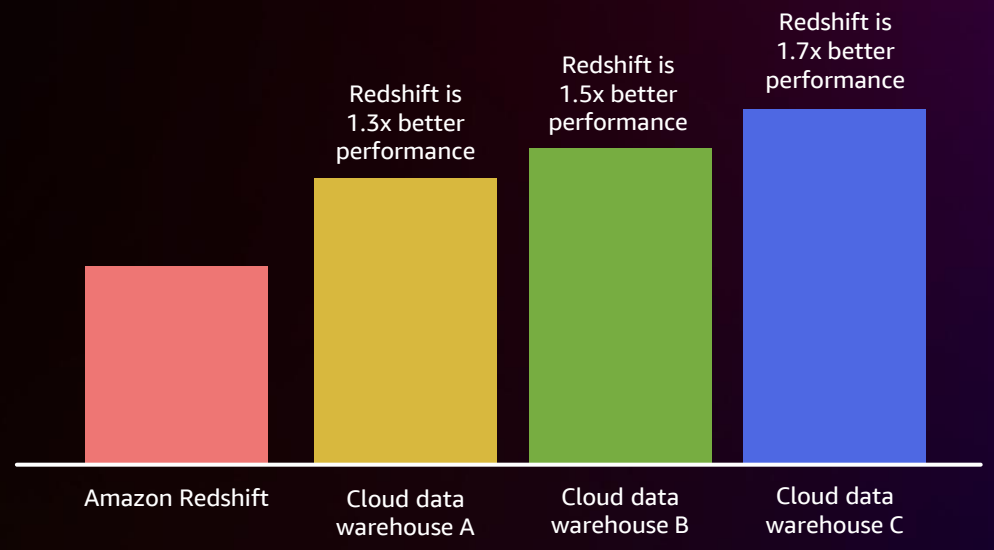


* Cloud DW benchmark is based on TPC-DS (v2.10) with no query modifications done

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up to 5X better price performance



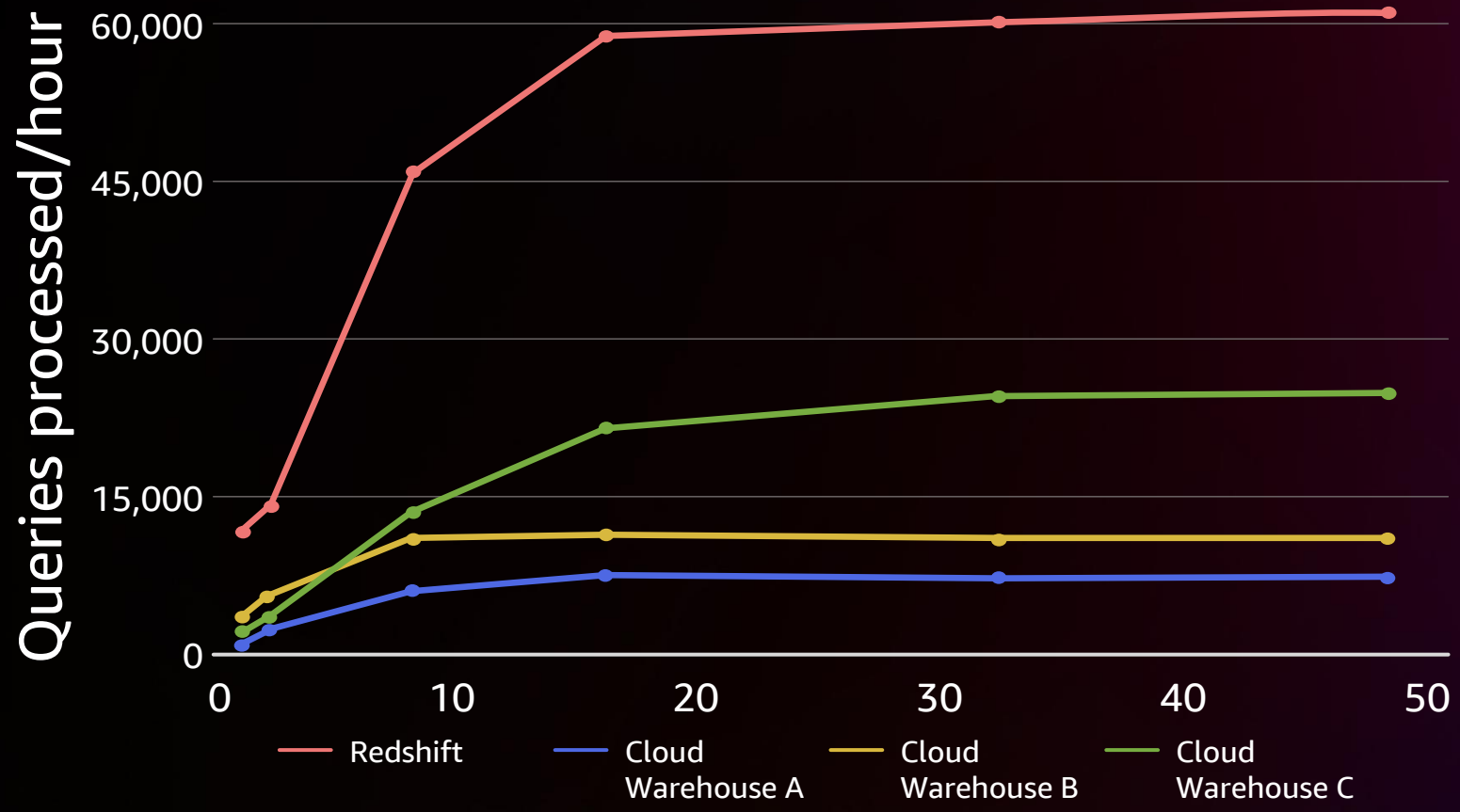


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up to 7X better price-performance for BI dashboards

Cloud DW 10GB benchmark*

Query throughput for short queries



Higher Is better!

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Amazon Redshift pillars

1. Security and Availability
2. Performance
3. Storage and Compute Elasticity
4. Autonomics
5. Serverless

aws

Amazon Redshift Managed Storage

PETABYTE SCALE HIGH PERFORMANCE STORAGE



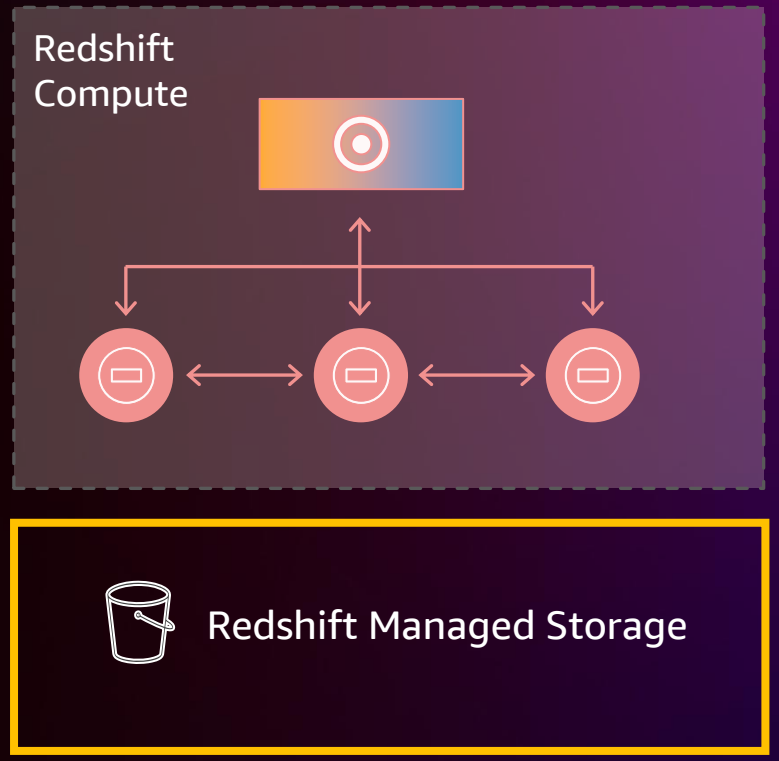
Managed storage



Large high-speed cache



High-bandwidth networking



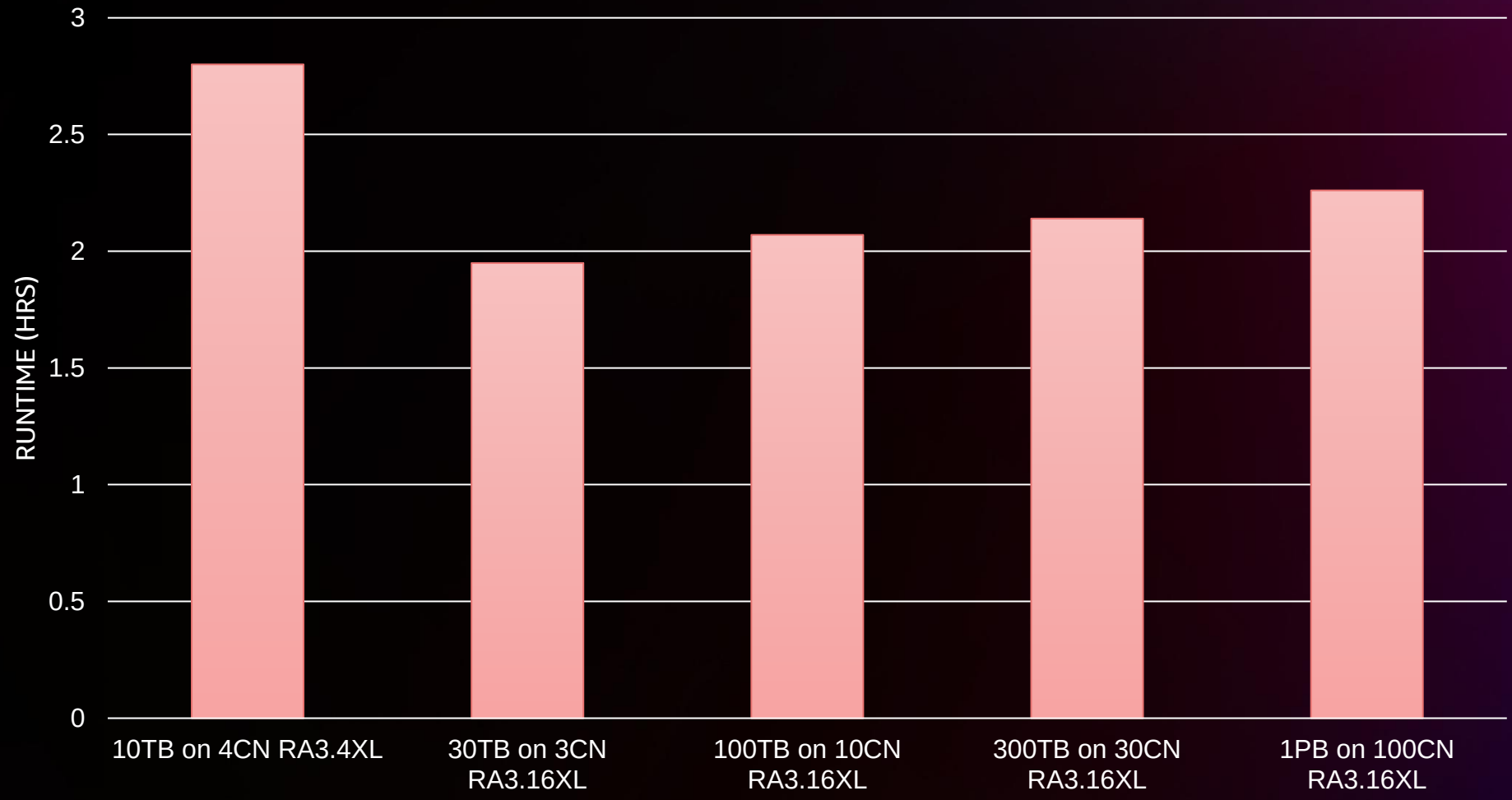
Allows scaling compute and storage independently

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Linear scalability to the PB scale

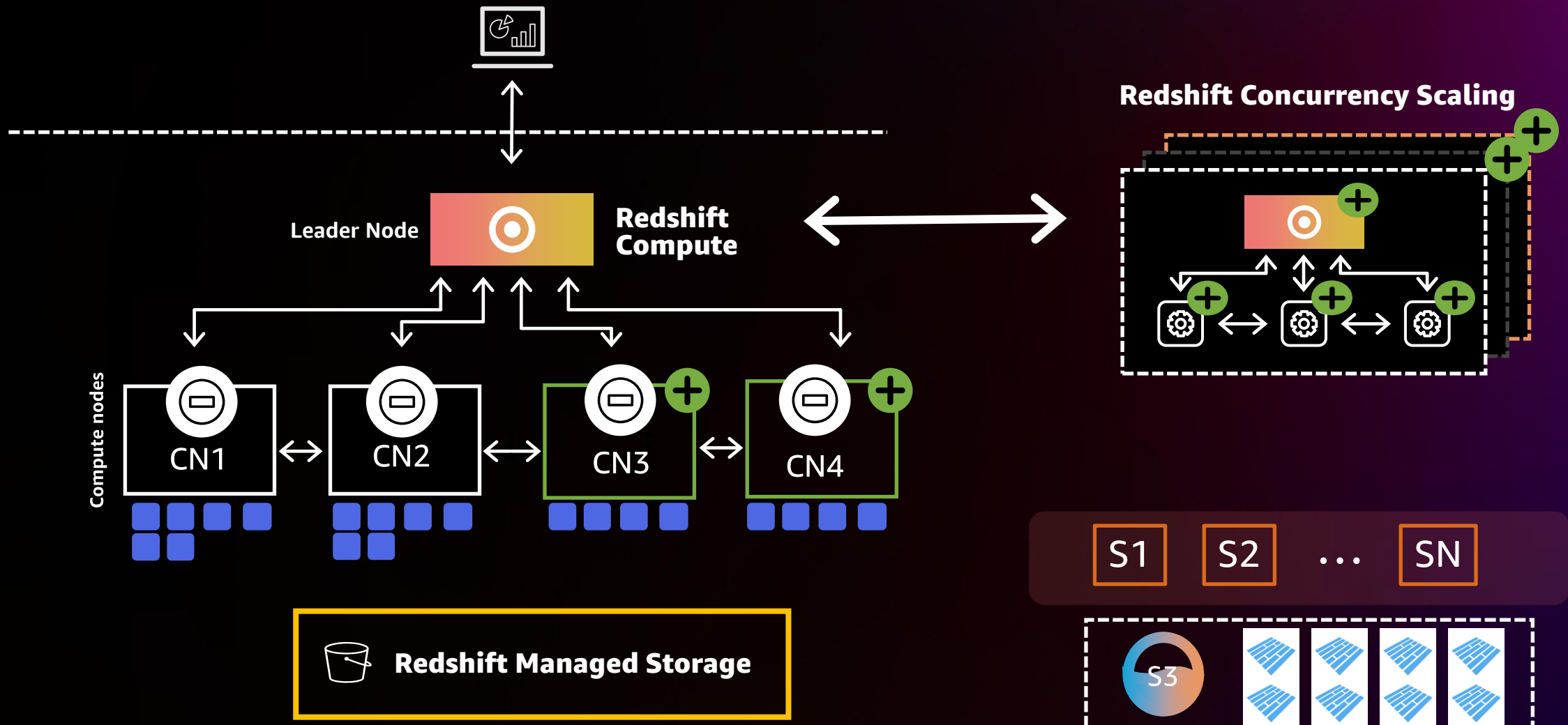
Cloud DW benchmark*



* Cloud DW benchmark is based on TPC-DS (v2.13) with no query modifications done

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

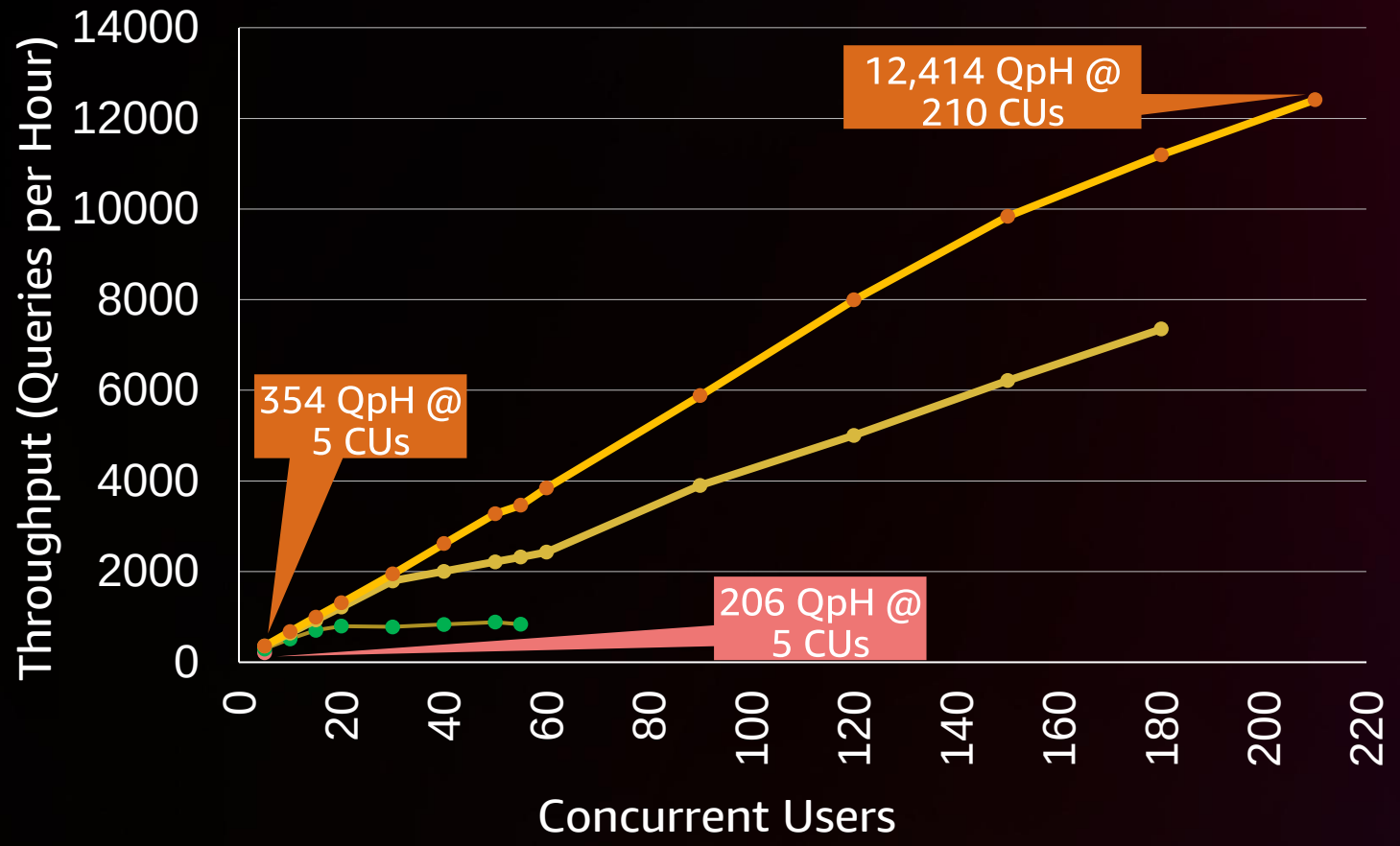
Compute Elasticity



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Compute elasticity improvements

>60x improvement in concurrency

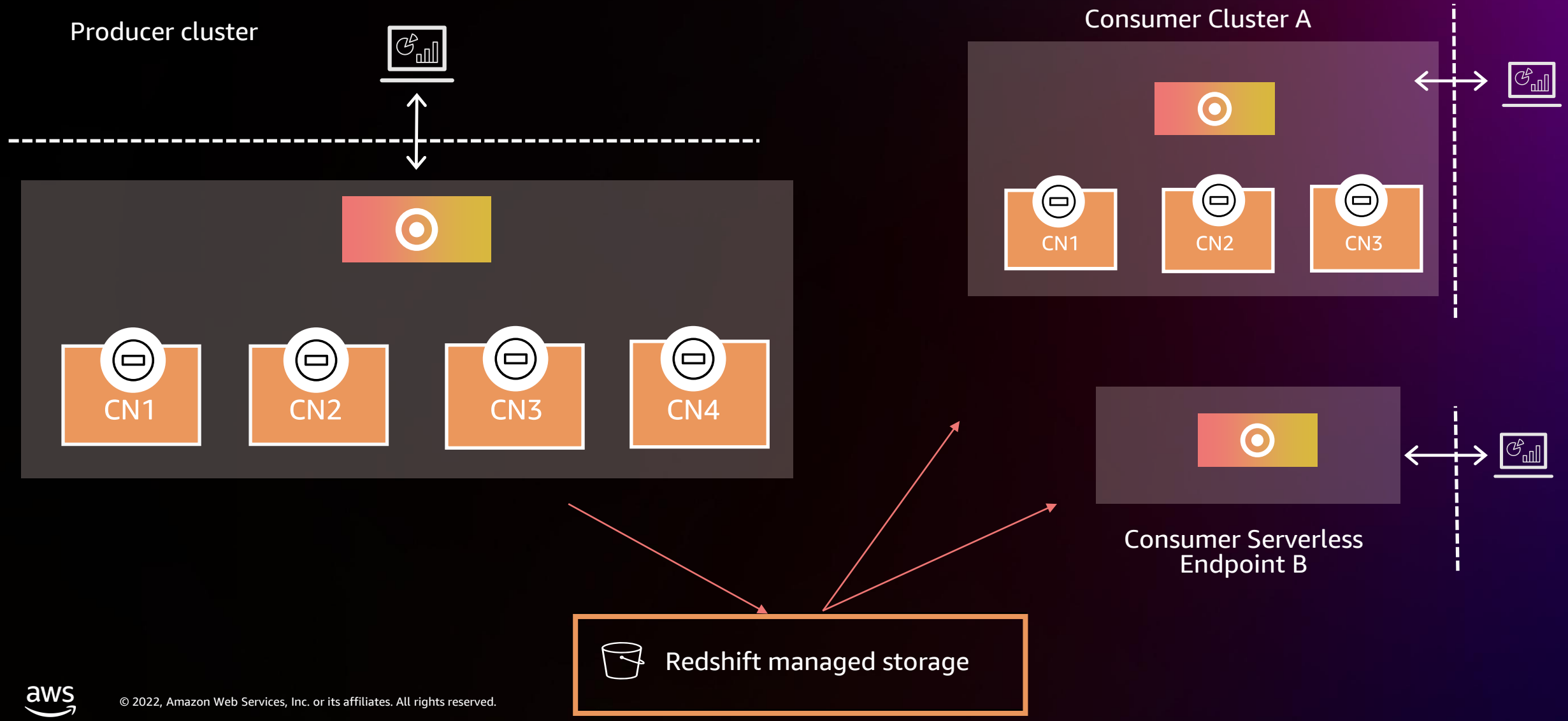


Cloud DW 3TB benchmark*

* Cloud DW benchmark is based on TPC-DS (v2.13) with no query modifications done

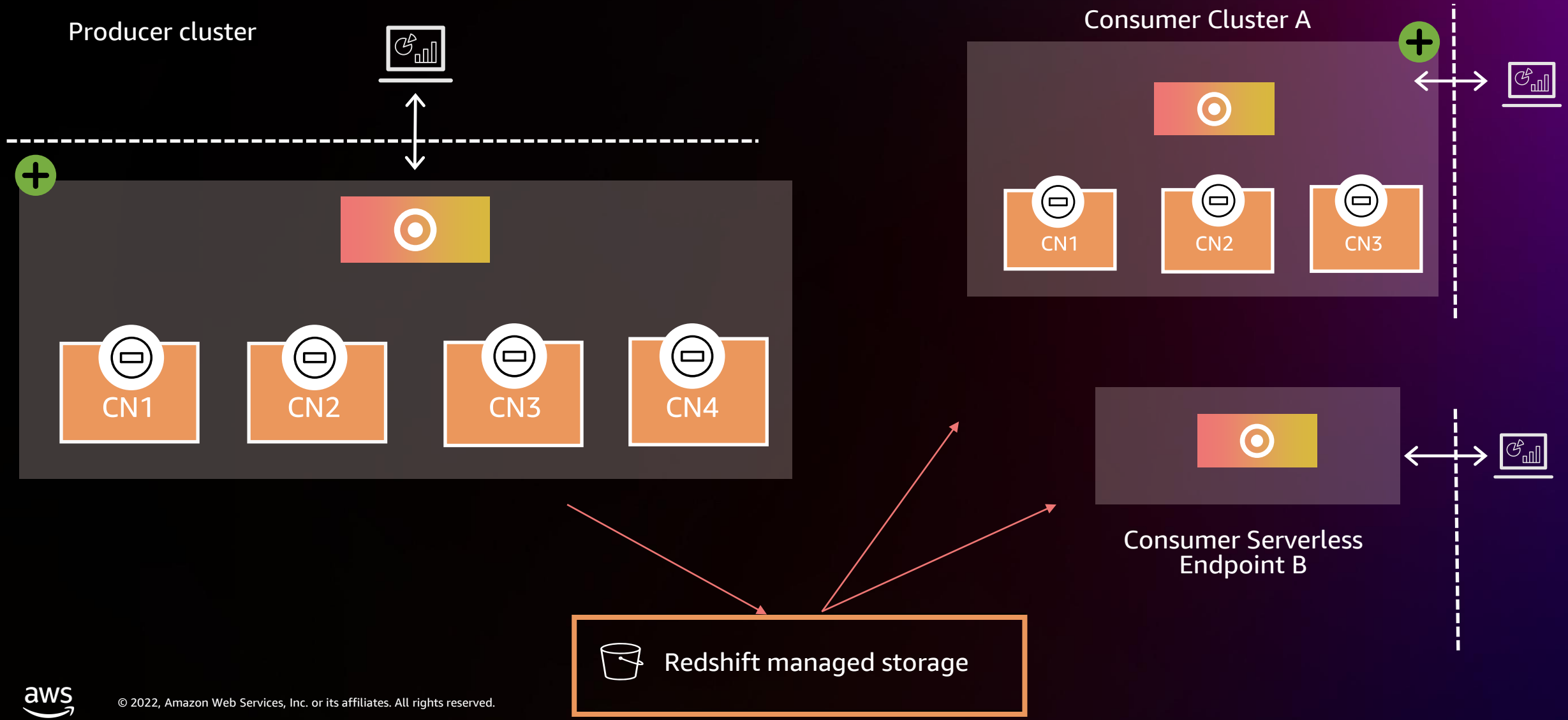
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data sharing



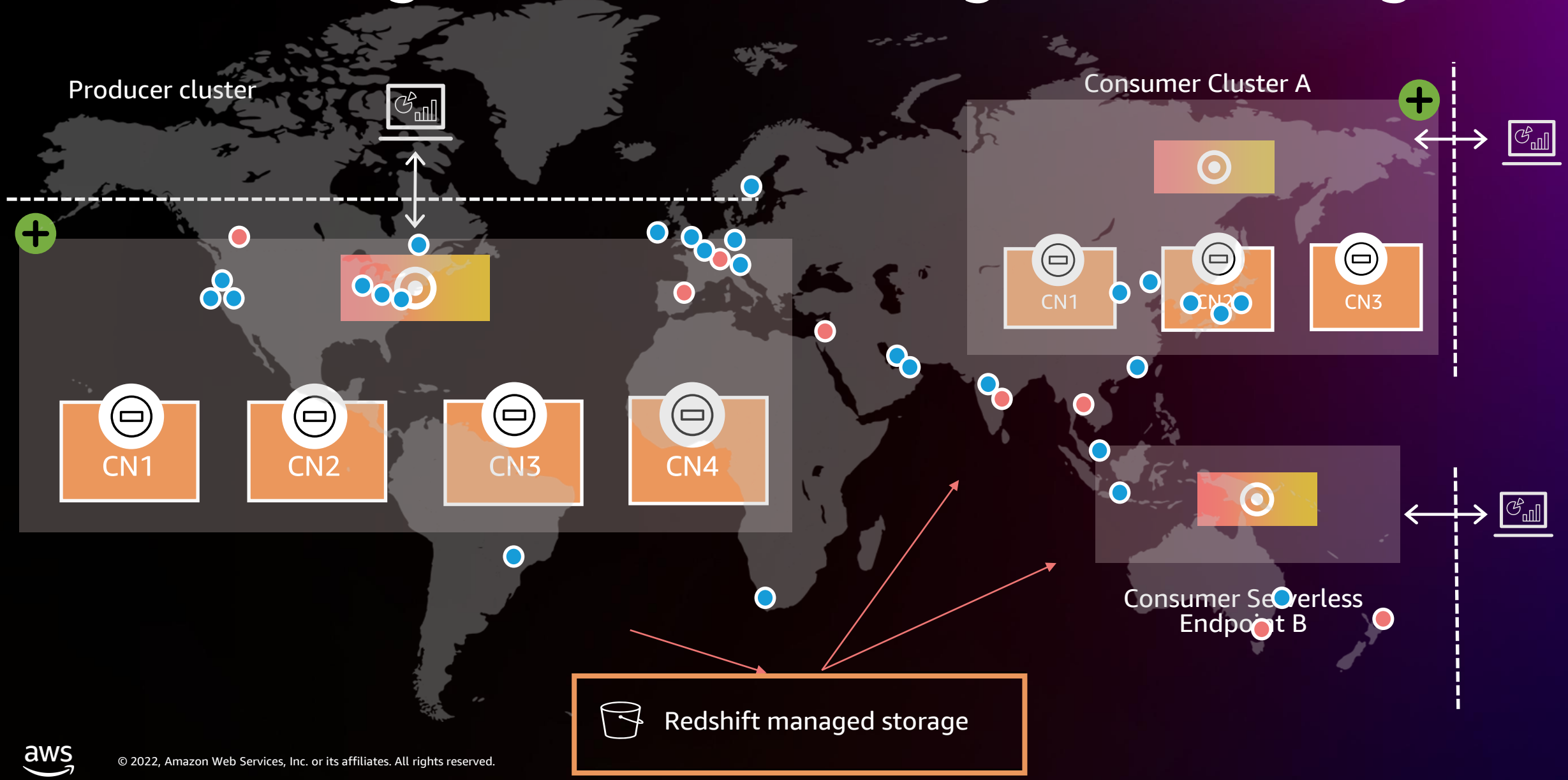
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data sharing with auto-scaling



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data sharing with auto-scaling, across the globe



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Amazon Redshift pillars

1. Security and Availability
2. Performance
3. Storage and Compute Elasticity
4. Autonomics
5. Serverless

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

Auto Maintenance Operations

ML-BASED OPTIMIZATIONS TO GET STARTED EASILY AND GET YOUR BEST PERFORMANCE QUICKLY

Automatic Table Optimization

- Self-tuning
- Optimize the physical design without DBA intervention
- Apply sort and distribution keys

Auto-Analyze

- Start with the tables that are more outdated and used more

Auto-Vacuum {Delete, Full}

- Priority-based and incremental
- Start with the most frequently used tables
- Perform vacuum if many rows have been added or deleted

Auto-Materialized View Refresh

- Refresh the MVs that are more likely to be used next

…And keep on adding more optimizations


Automatic Analyze


Automatic Vacuum Delete


Distribution/Sort Key Advisors


Automatic Table Distribution Keys


Automatic Table Sort Keys


Automatic Table Column Encoding


Auto Workload Manager

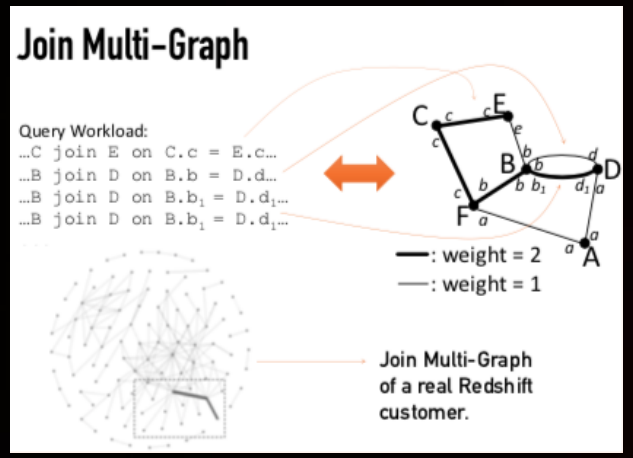

Automatic Table Sort


Auto refresh & rewrite of MVs

Redshift Advisor: Dist-Key Recommendation

How does it work?

1. Monitors query workload and builds a graph to represent joins.
2. Uses combinatorial optimization techniques to solve a novel graph theoretical problem *optimally.*
3. Optimizes data distribution to minimize network communication among CNs.





Fast and Effective Distribution-Key Recommendation for Amazon Redshift

Panos Parchas
Amazon Web Services
Berlin, Germany
parchp@amazon.com

Yonatan Naamad
Amazon Devices
Sunnyvale, CA, USA
ynaamad@amazon.com

Peter Van Bouwel
Amazon Web Services
Dublin, Ireland
pbbouwel@amazon.com

Christos Faloutsos
Carnegie Mellon University & Amazon
faloutso@amazon.com

Michalis Petropoulos
Amazon Web Services
Palo Alto, CA, USA
mpetropo@amazon.com

ABSTRACT

How should we split data among the nodes of a distributed data warehouse in order to boost performance for a forecasted workload? In this paper, we study the effect of different data partitioning schemes on the overall network cost of pairwise joins. We describe a generally-applicable data distribution framework initially designed for Amazon Redshift, a fully-managed petabyte-scale data warehouse in the cloud. To formalize the problem, we first introduce the *Join Multi-Graph*, a concise graph-theoretic representation of the workload history of a cluster. We then formulate the *"Distribution-Key Recommendation"* problem – a novel combinatorial problem on the *Join Multi-Graph*– and relate it to problems studied in other subfields of computer science. Our theoretical analysis proves that *"Distribution-Key Recommendation"* is **NP**-complete and is hard to approximate efficiently. Thus, we propose BAW, a hybrid approach that combines heuristic and exact algorithms to find a good data distribution scheme. Our extensive experimental evaluation on real and synthetic data showcases the efficacy of our method into recommending optimal (or close to optimal) distribution keys, which improve the cluster performance by reducing network cost up to **32x** in some real workloads.

PVLDB Reference Format:
Panos Parchas, Yonatan Naamad, Peter Van Bouwel, Christos Faloutsos, Michalis Petropoulos. Fast and Effective Distribution-Key Recommendation for Amazon Redshift. *PVLDB*, 13(11): xxxx-yyyy, 2020.
DOI: https://doi.org/10.14778/3407790.3407834

chines, how should we distribute the data to reduce network cost and ensure fast query execution?

Amazon Redshift [12, 2, 3] is a massively parallel processing *MPP* database system, meaning that both the storage and processing of a cluster is distributed among several machines (*compute nodes*). In such systems, data is typically distributed row-wise, so that each row appears (in its entirety) at one compute node, but distinct rows from the same table may reside on different machines. Like most commercial data warehouse systems, Redshift supports 3 distinct ways of distributing the rows of each table:

- *"Even"* (= uniform/random), which distributes rows among the compute nodes in a a round-robin fashion;
- *"All"* (= replicated), which makes full copies of a database table on each of the machines;
- *"Dist-Key"* (=fully distributed = hashed), which hashes the rows of a table on the values of a specific attribute known as *distribution key* (DK).

In this work, we focus on the latter approach. Our primary observation is that, if both tables participating in a join are distributed on the joining attributes, then that join enjoys great performance benefits due to minimized network communication cost. In this case, we say that the two tables are *collocated* with respect to the given join. The goal of this paper is to minimize the network cost of a given query workload by carefully deciding which (if any) attribute should be used to hash-distribute each database table.

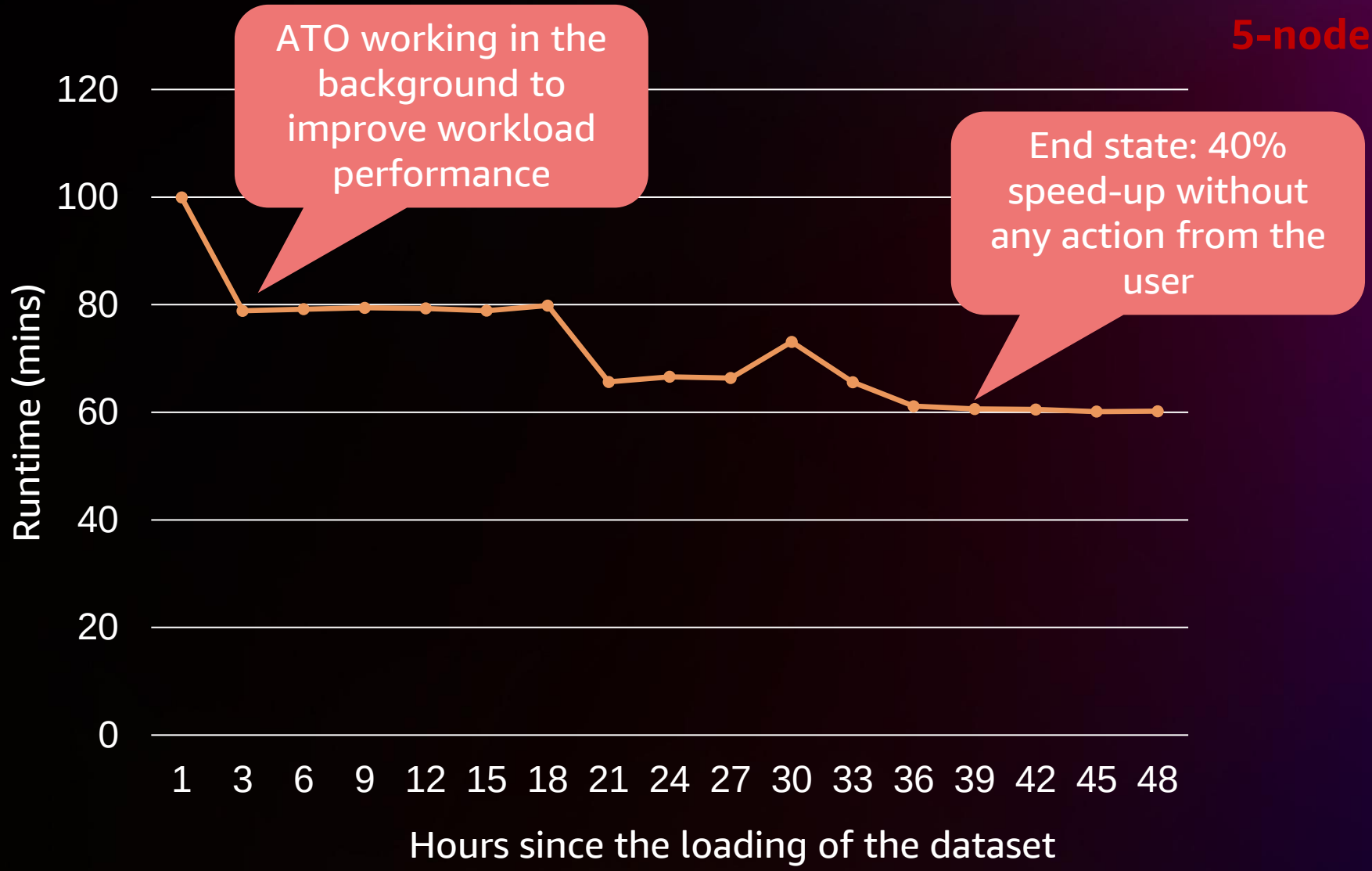
Figure 1 illustrates collocation through an example. Consider the schema of Figure 1(a) and the following query Q1:

P. Parchas, Y. Naamad, P. Van Bouwel, C. Faloutsos, and M. Petropoulos. "Fast and Effective Distribution-Key Recommendation for Amazon Redshift." In PVLDB, 2020.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automatic Table Optimization

Cloud DW derived from TPC-H 30TB 5-node RA3.16XL



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automated Materialized views

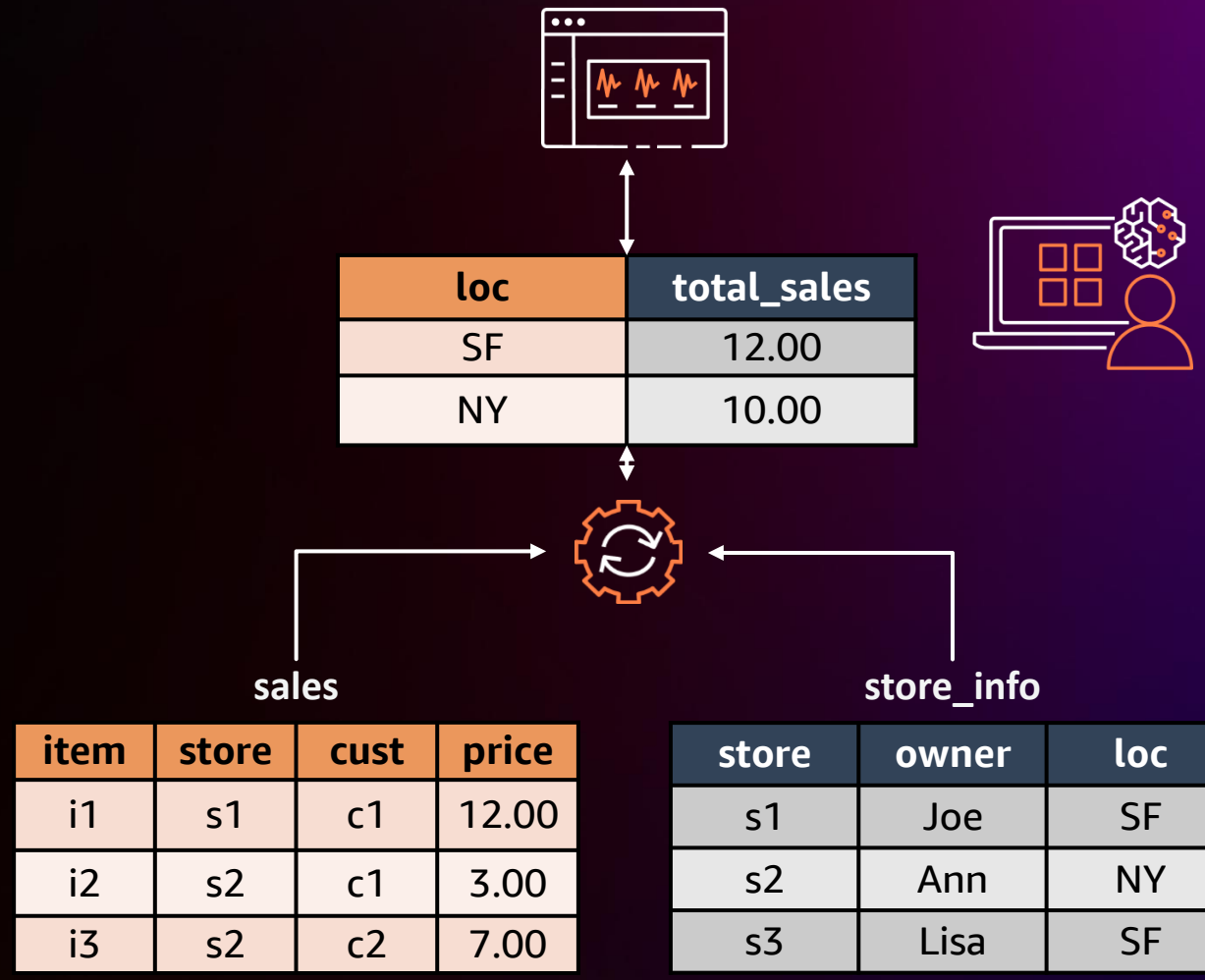
SPEED UP QUERY PERFORMANCE BY ORDERS OF MAGNITUDE WITH PRECOMPUTED RESULTS

Accelerate predictable workloads like BI, reports, and dashboards with precomputed results

MVs can be based on one or more Redshift tables or external tables (Spectrum, Federated Query)

Incremental updates, auto refresh, auto query rewrite

Automated MVs: Automatically create or delete MVs with incremental refresh

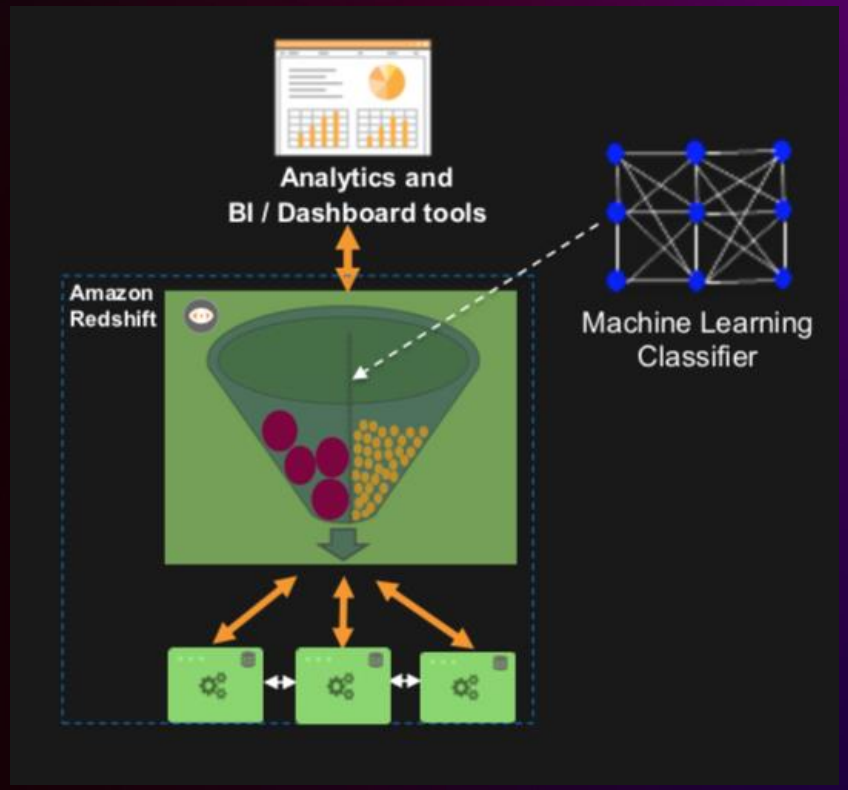


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automated workload management

Short Query Acceleration

- Short queries do not get stuck behind long running ones
- Higher throughput, less variability
- Customized on the running workload, at runtime.



Query Predictor

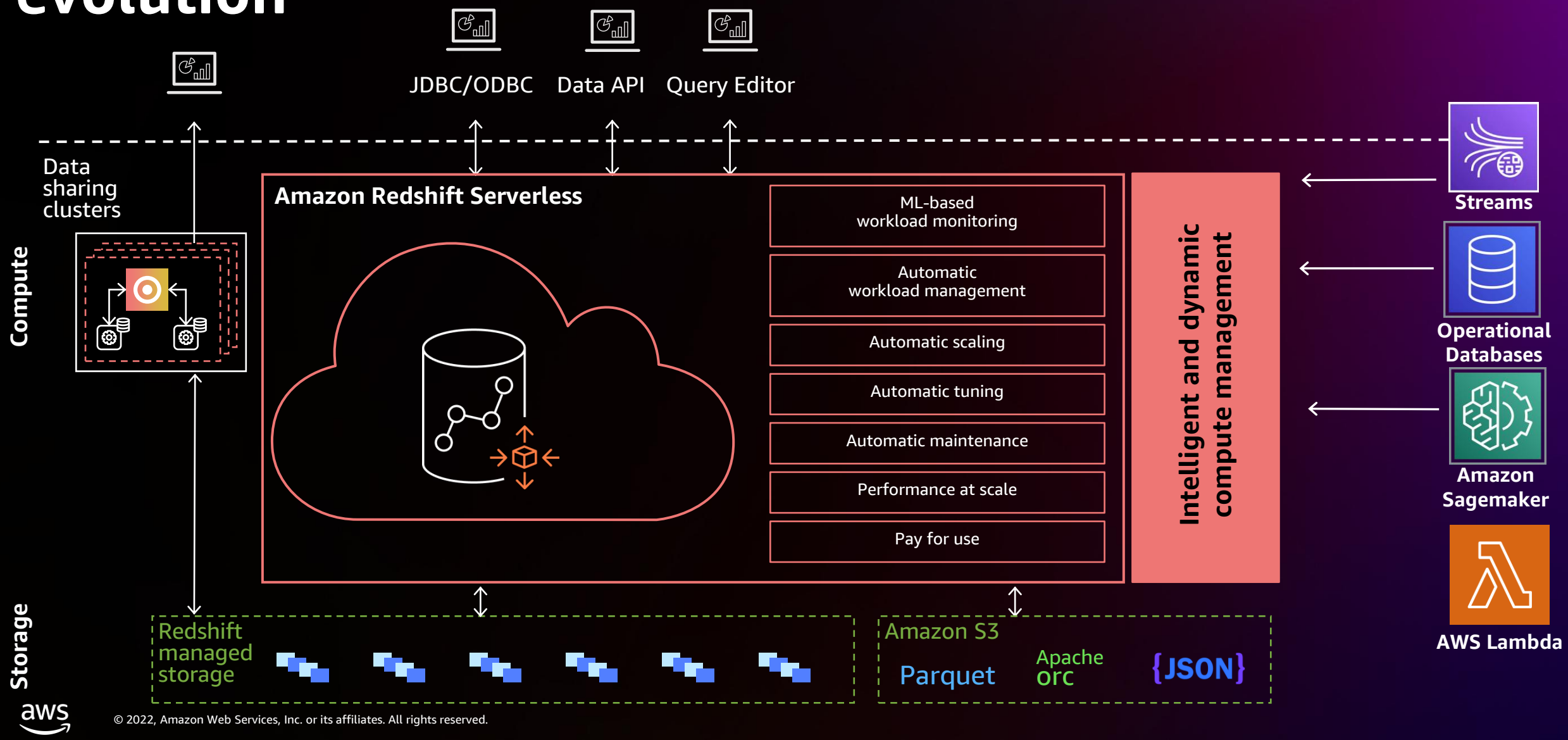
- Predict with confidence the resource needs of each query
- Determine the number of queries to process in parallel to optimize throughout, performance and resource utilization
- Used for scheduling decisions

The Amazon Redshift pillars

1. Security and Availability
2. Performance
3. Storage and Compute Elasticity
4. Autonomics
5. Serverless

aws

Intelligent and dynamic compute management evolution



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift enables high performance analytics at scale across all data for all users

Core data warehouse pillars

- Security & Availability
- Performance
- Compute and storage elasticity
- Serverless
- Autonomics



Amazon Redshift

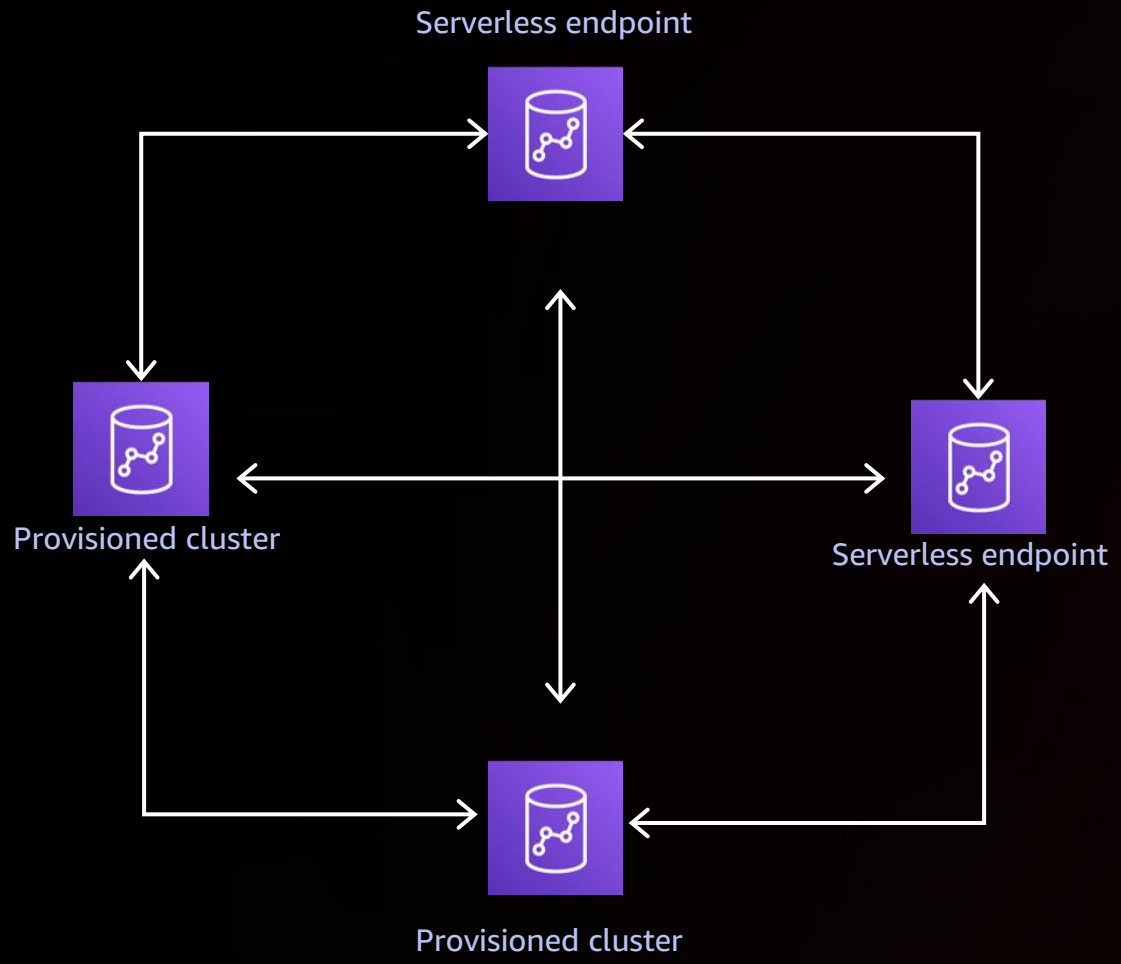
AWS service integrations

- Analytics on all data – data warehouse, data lake, streaming & operational databases
- Analytics for all users - BI, machine learning & advanced analytics

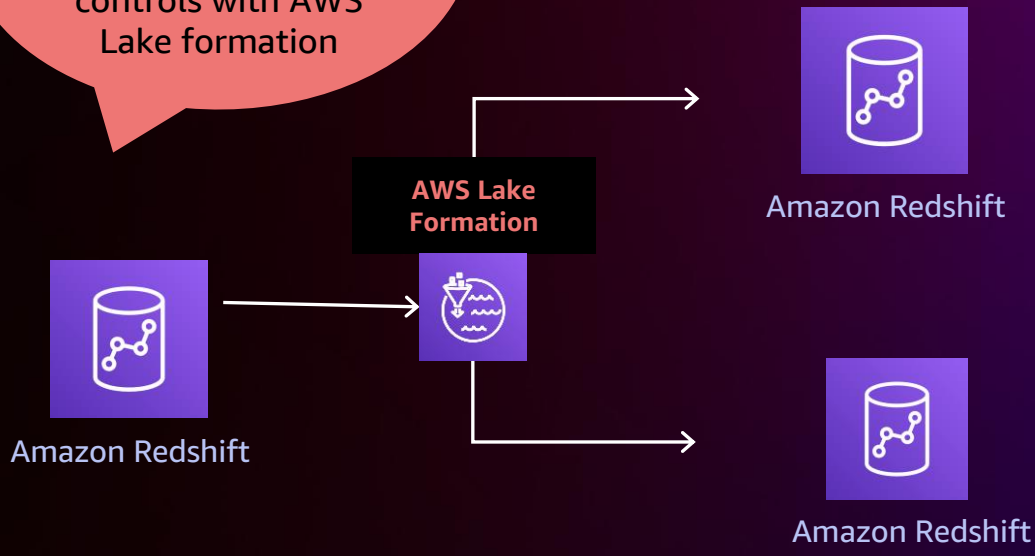
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Flexible Data mesh architectures to seamlessly share data within and across organizations



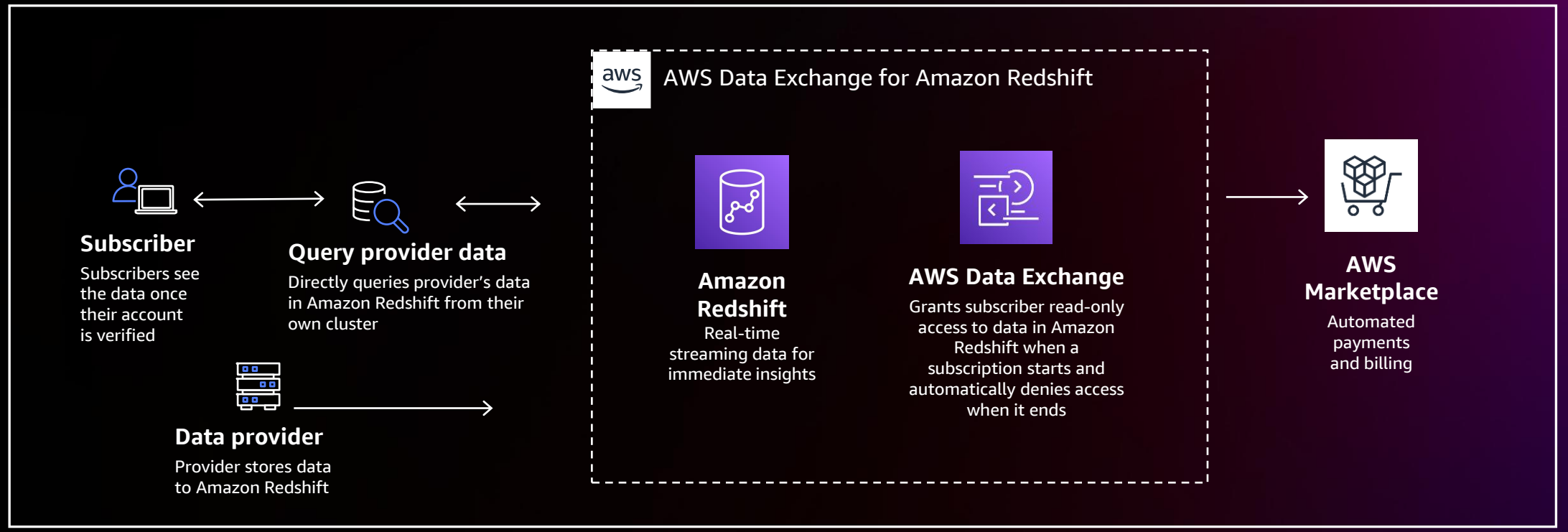
NEW! Amazon Redshift Data sharing now supports centralized access controls with AWS Lake formation



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Share and collaborate on third-party data

EXTEND YOUR DATA MESH BEYOND YOUR ORGANIZATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Analytics on all data

New!



Auto-copy from Amazon S3

GA



Real-time streaming ingestion

Streams

New!



Zero-ETL integration

Operational databases



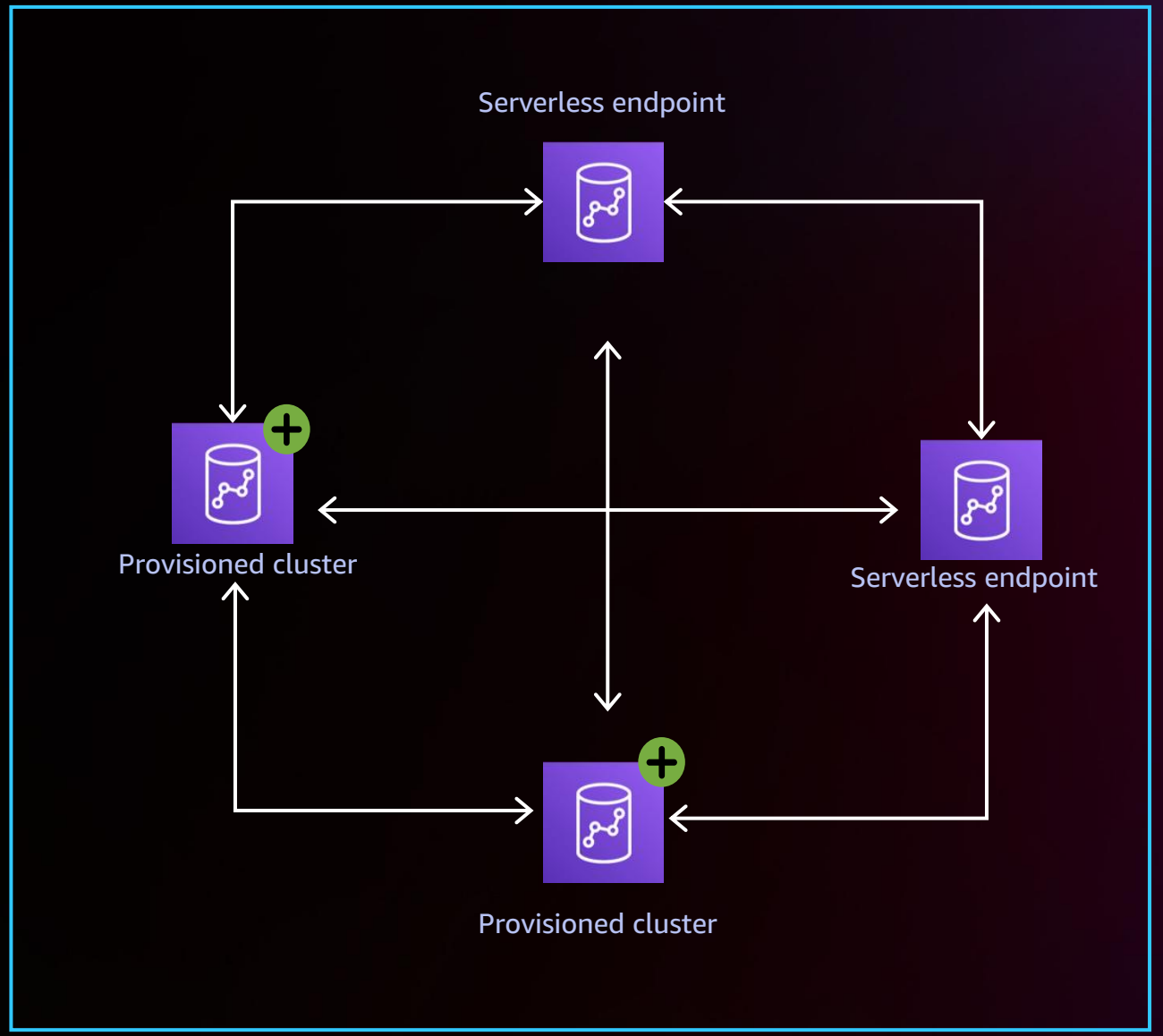
Federated query

Operational databases



Amazon Data Exchange integration

Third-party data





Analytics for all users

JDBC/ODBC

Data API

Query Editor



New!

Integration for Apache Spark



Amazon EMR



Amazon Glue

Redshift ML



Amazon Sage Maker

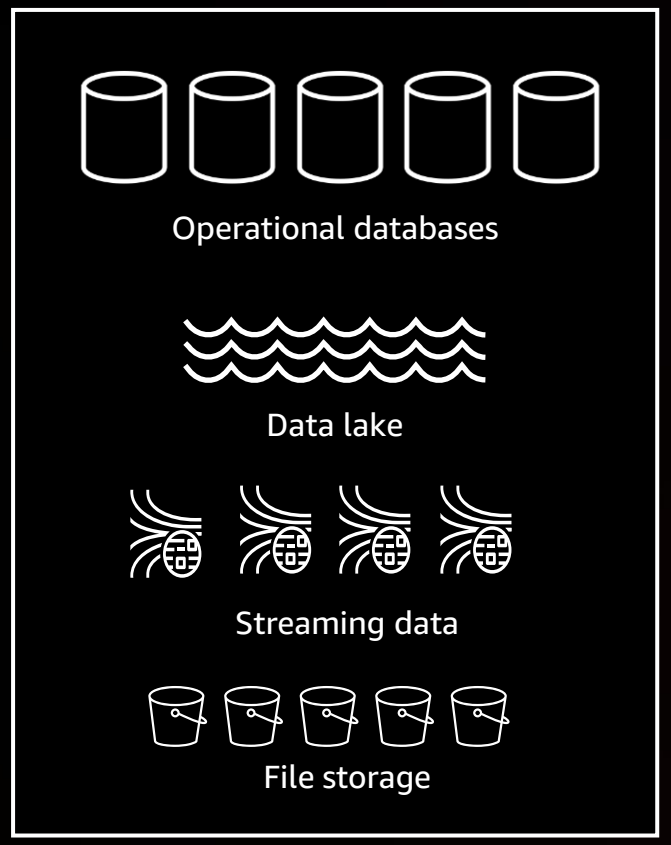
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ingesting data into analytics systems is complex

MULTIPLE SOURCE SYSTEMS REQUIRE SEPARATE & COMPLEX MANUAL DATA PIPELINES

Data Sources



Manual data pipelines



X **Expensive and cumbersome** to build and maintain ETL jobs

X **Complex reconstruction** of the data especially with schema changes

X **Incomplete, inconsistent, and stale views** of data, limiting insights

Analytics

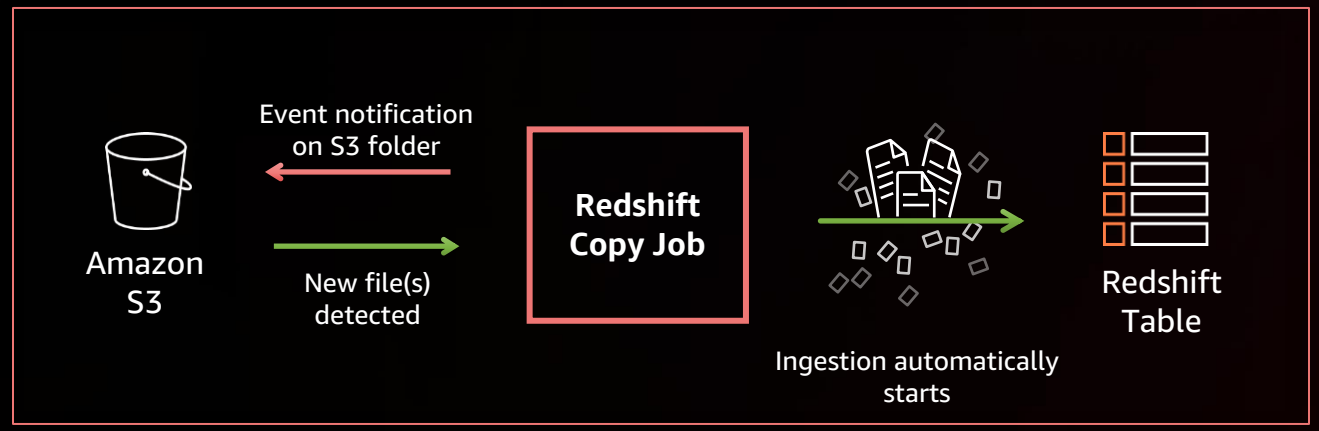


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift Auto-Copy from Amazon S3

Simplified & automated file ingestion
from Amazon S3 into Redshift



Simple, low code data ingestion

Avoid re-ingestion and manual tracking of loaded files

Easily convert your existing COPY statements into automatic ingestion jobs

Automatic ingestion of new data from Amazon S3 based on user defined configurations

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ingesting data continuously from Amazon S3

1. Specify file source and target table



Amazon S3

Create an Amazon S3 bucket / folder and provide permissions to access the files in that folder.

For example : s3://test_autocopy/



Redshift Table

Create your target Amazon Redshift table that you need to automatically populate with new files.

```
CREATE TABLE lineitem
(l_orderkey int8 encode mostly32,
 l_partkey int8 encode mostly32,
l_suppkey int4 NOT NULL,
l_linenumber int4 encode delta,
l_quantity NUMERIC(12,2) encode bytedict);
```

2. Create a copy job to start auto-copy

The COPY command in Amazon Redshift now supports JOB parameters to create, run and manage copy jobs.

```
COPY lineitem
FROM s3://test_autocopy/1
[COPY PARAMETERS]
JOB CREATE auto_ingest_lineitem1;

COPY lineitem
FROM s3://test_autocopy/2
[COPY PARAMETERS]
JOB CREATE auto_ingest_lineitem2;
```

You can append JOB CREATE <job_name> parameters to your existing copy statements to convert them to jobs

3. Manage your copy jobs

```
COPY JOB LIST
-- Lists all copy jobs in your database

COPY JOB SHOW <job_name>
-- Returns the DDL for the copy job

COPY JOB DROP <job_name>
-- Drops the copy job

COPY JOB ALTER <job_name>
AUTO ON | OFF
-- Sets auto copy to on or off.

COPY JOB RUN <job_name>
-- Manually run a copy job when auto copy is set to off
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

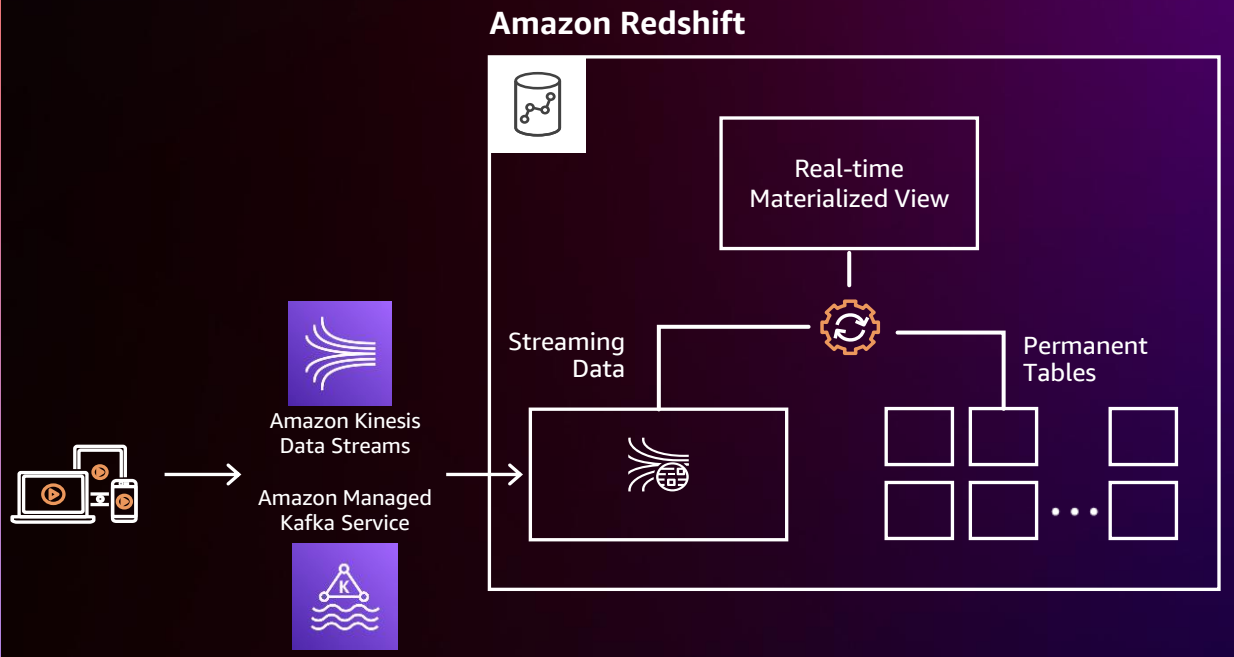
Amazon Redshift Real-time Streaming Ingestion

NATIVE INTEGRATION WITH STREAMING ENGINES FOR FAST AND EASY DATA INGESTION

High throughput and low latency data ingestion

Easily ingest from Amazon Kinesis Data Streams (KDS) streams or Amazon Managed Streaming for Apache Kafka (MSK) topics without complex data pipelines

Define a schema or ingest semi-structured data with SUPER & VARBYTE data types for ad-hoc queries

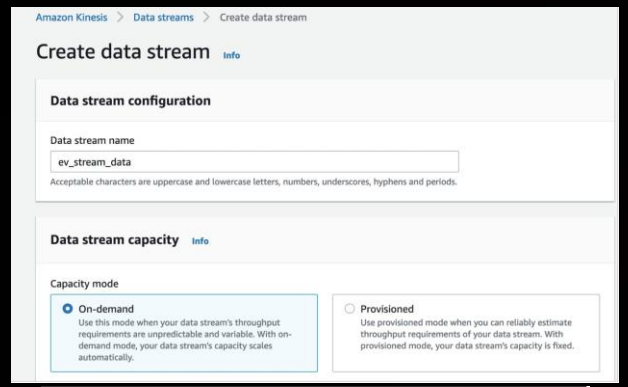


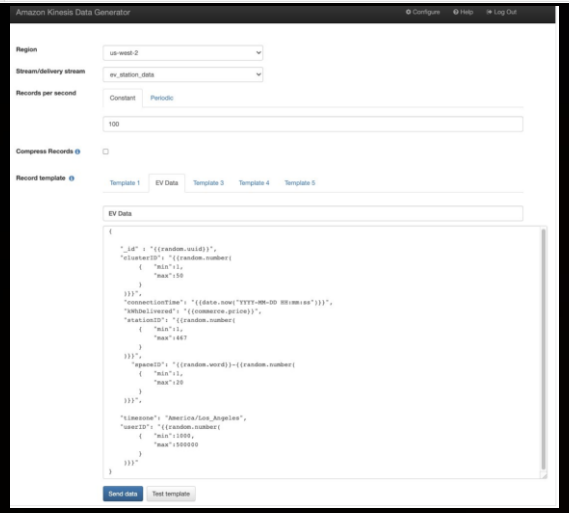
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ingesting streaming data into Amazon Redshift

1. Create stream in Amazon KDS or Amazon MSK and send data





2. Create a materialized view and start loading streaming data into Redshift

```
CREATE EXTERNAL SCHEMA evdata FROM KINESIS IAM_ROLE
'arn:aws:iam::0123456789:role/redshift-streaming-role
```

```
CREATE MATERIALIZED VIEW ev_station_data AS SELECT
approximatearrivaltimestamp,
partitionkey,
shardid,
sequencenumber,
json_extract_path_text(from_varbyte(data, 'utf-
8'),'connectionTime')::varchar(20) as connectionTime,
       json_extract_path_text(from_varbyte(data, 'utf-
8'),'kWhDelivered')::DECIMAL(10,2) as kWhDelivered,
      json_extract_path_text(from_varbyte(data, 'utf-
8'),'userID')::varchar(30) as userID
FROM evdata."ev_stream_data";
```

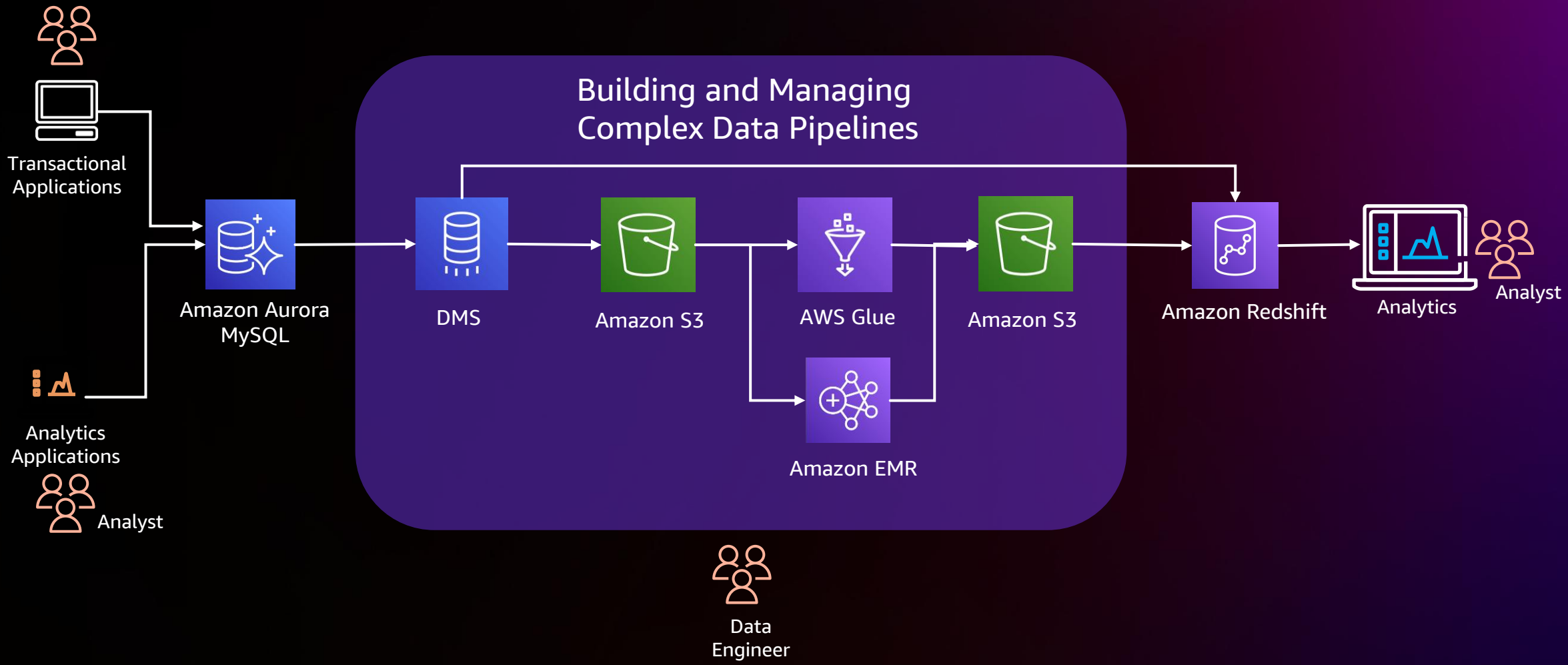
3. Query streaming data

```
SELECT
TO_TIMESTAMP(connectionTime, 'YYYY-MM-DD HH24:MI:SS') AS
connectiontime ,
SUM(WkhDelivered) AS Energy_Consumed ,
COUNT(DISTINCT userID) AS #Users
FROM ev_station_data
GROUP BY to_timestamp(connectionTime, 'YYYY-MM-DD
HH24:MI:SS')
ORDER BY 1
DESC;
```

connectiontime	energy_consumed	#users
2022-02-27 23:52:07+00	72870	131
2022-02-27 23:52:06+00	510892	998
2022-02-27 23:52:05+00	461994	934

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

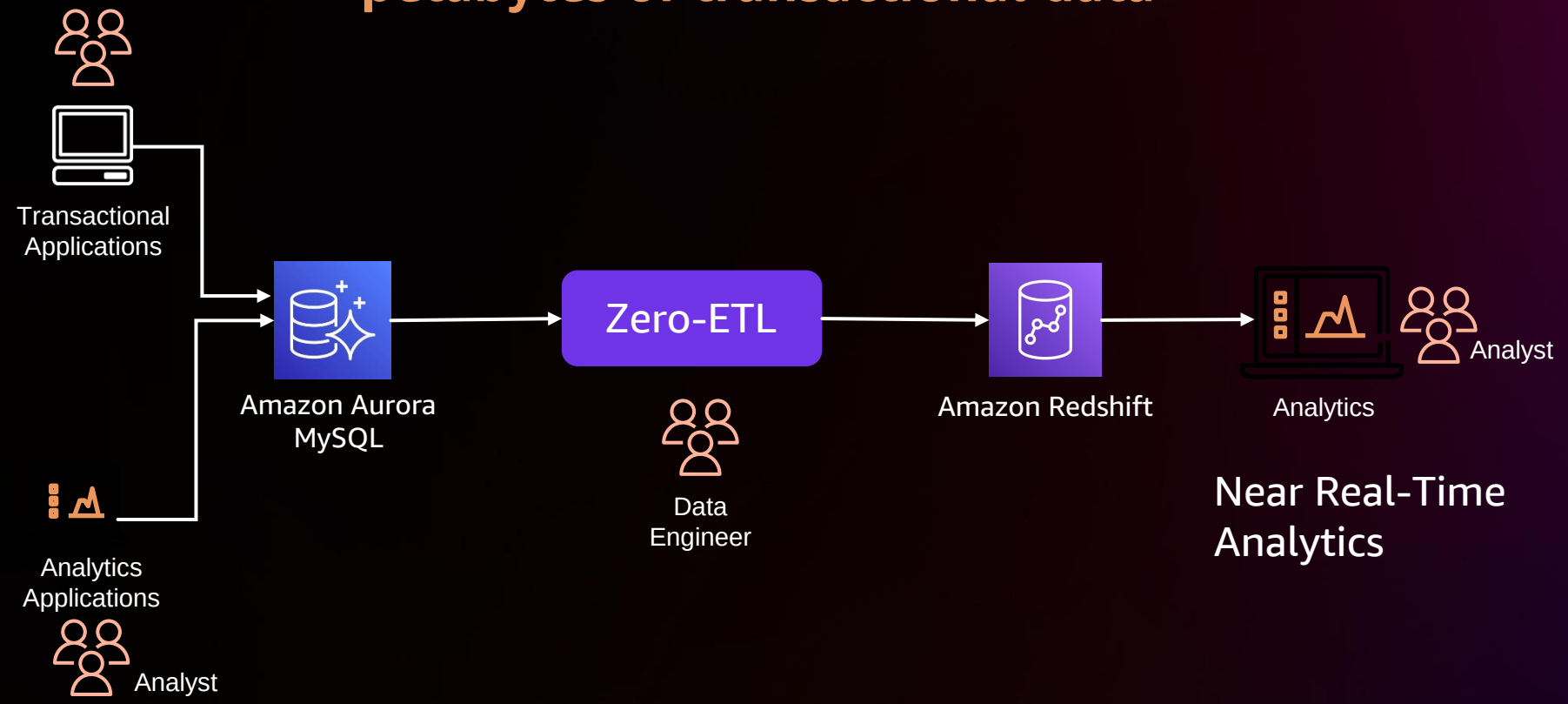
Transaction Analytics At Scale



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Zero-ETL to Amazon Redshift

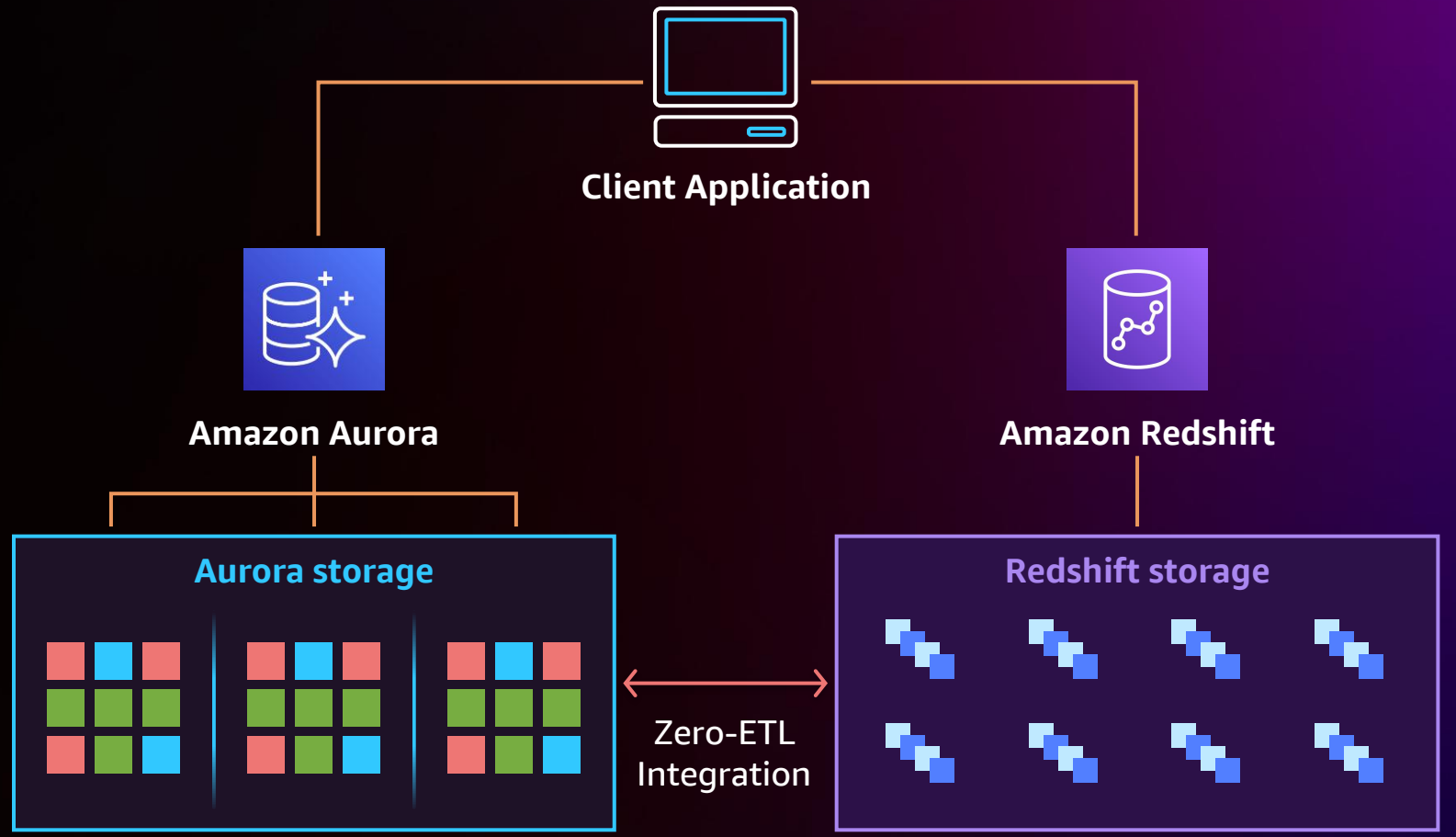
An **easy** and **secure** way to enable **near real-time analytics** on **petabytes of transactional data**



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Designed for simplicity and performance

- Easy creation of Redshift integration target
- Automated data seeding and continuous replication at storage layer
- Consolidate data from multiple Aurora database clusters
- Run analytics along side ingestion
- Monitor progress, lag, and performance from Redshift system tables and CloudWatch

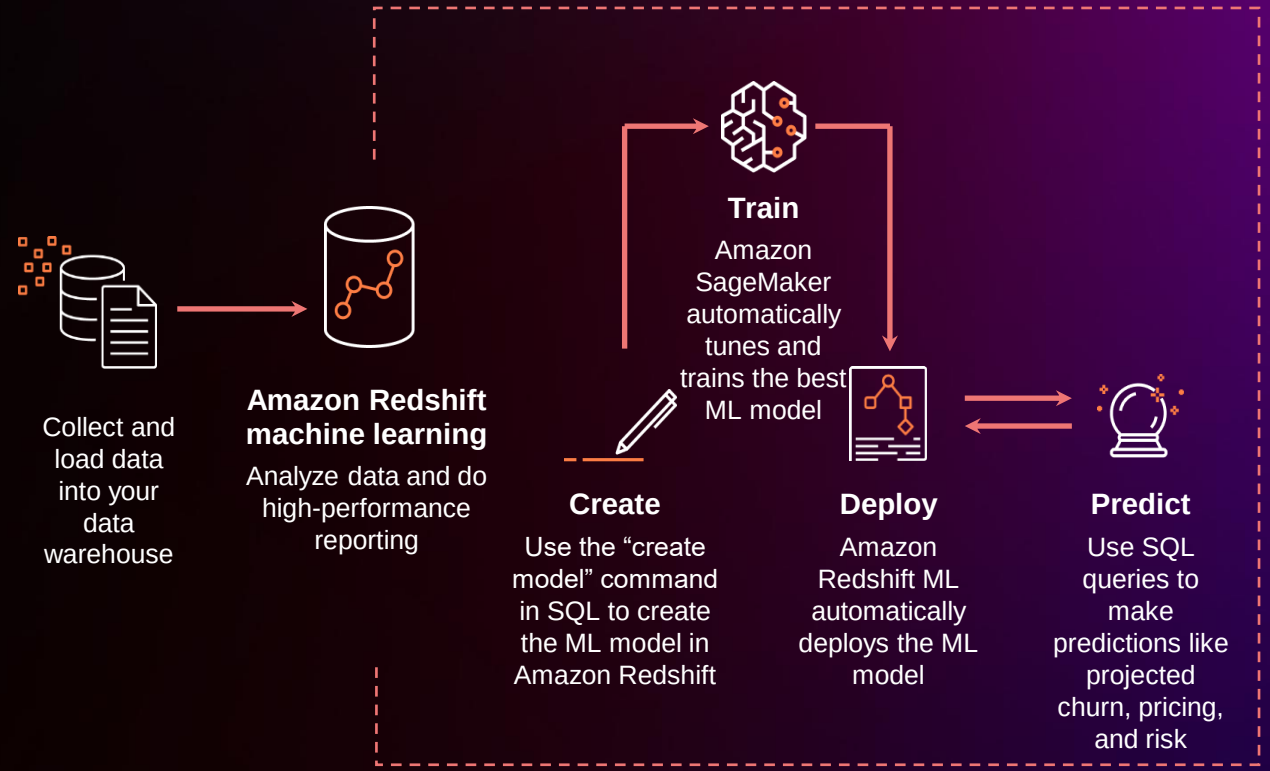


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift ML

EASILY CREATE AND TRAIN ML MODELS USING SQL QUERIES WITH AMAZON SAGEMAKER

- Train and create ML models using SQL
- Automatic pre-processing, creation, training, deployment, and inferencing of models
- SageMaker models for in-database or remote inference
- Supervised and unsupervised trainings
- Run an inference as invoking a user-defined function as part of SQL statements



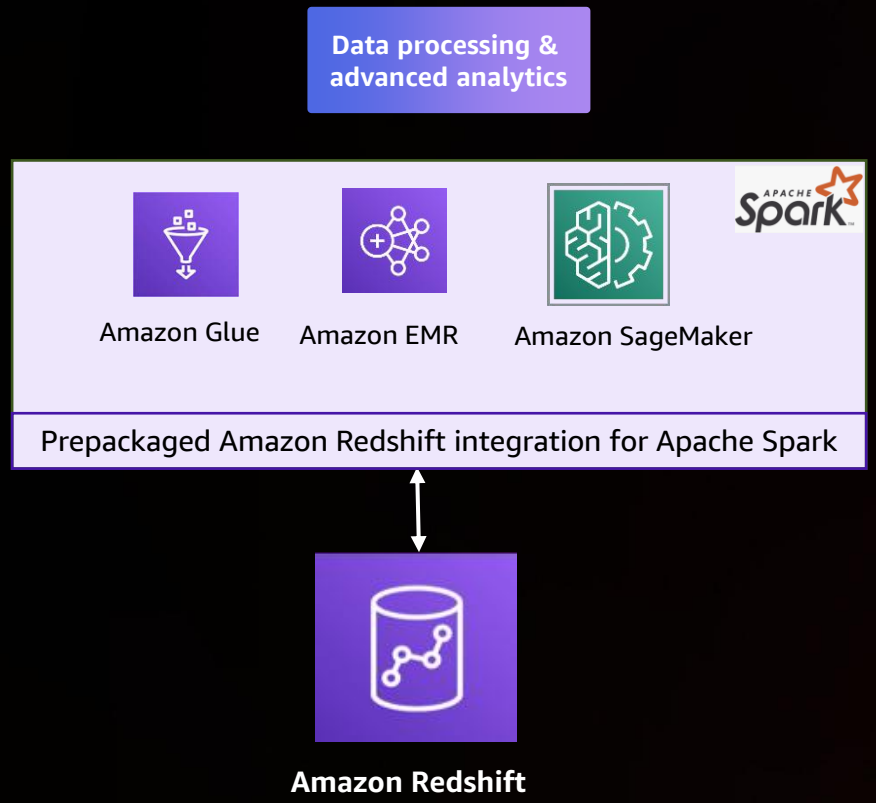

```
CREATE MODEL customer_churn
FROM (
SELECT c.age, c.zip, c.monthly_spend, c.monthly_cases, c.active FROM customer_info_table c)
TARGET c.active
FUNCTION predict_customer_churn

SELECT n.id, n.firstName, n.lastName,
predict_customer_churn(n.age,c.zip,..)
AS activity_prediction
FROM new_customers n
WHERE n.marital_status = 'single'
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Redshift integration for Apache Spark

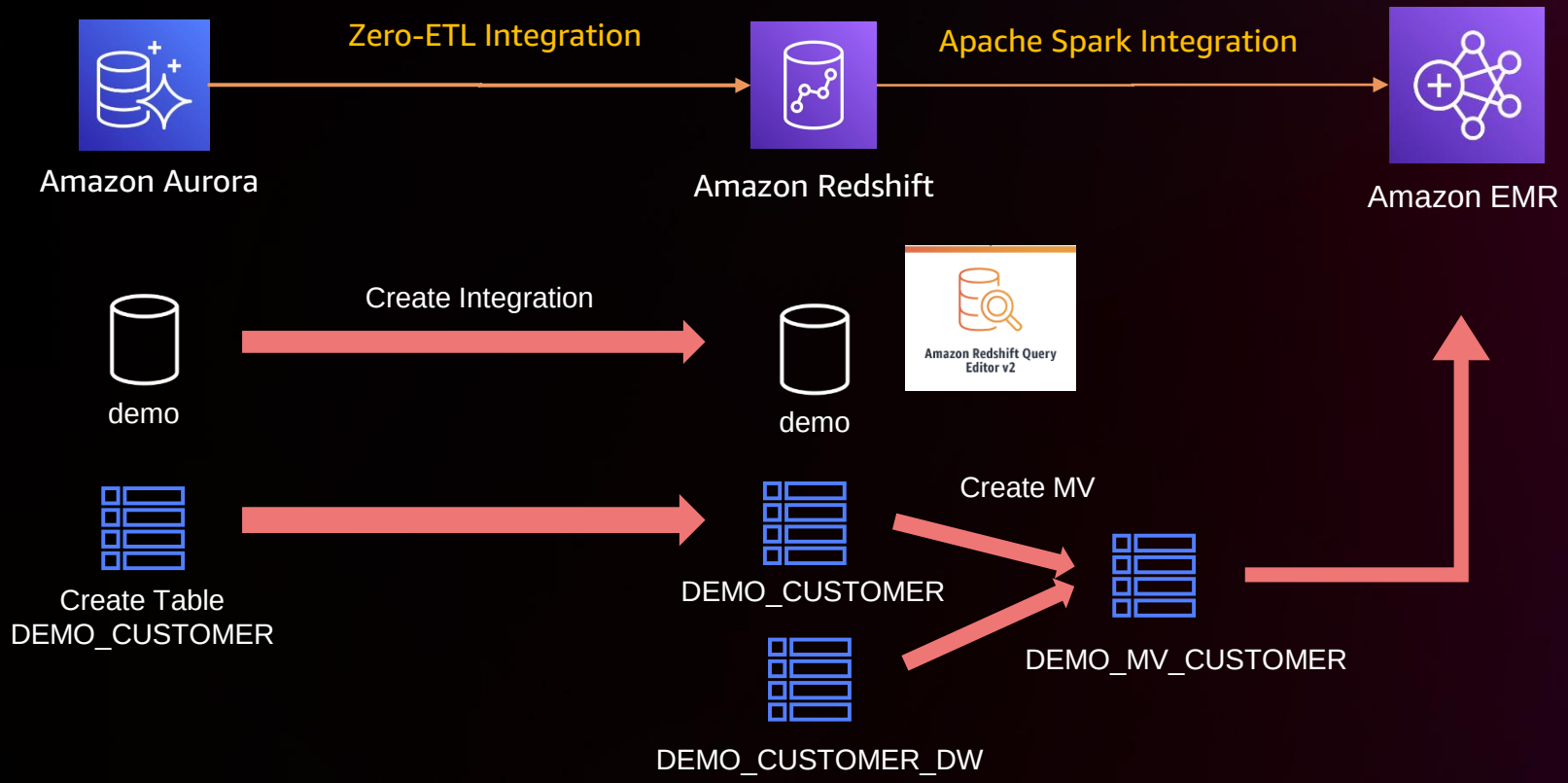
Simplify and speed up Apache Spark applications accessing Amazon Redshift data from AWS analytics services



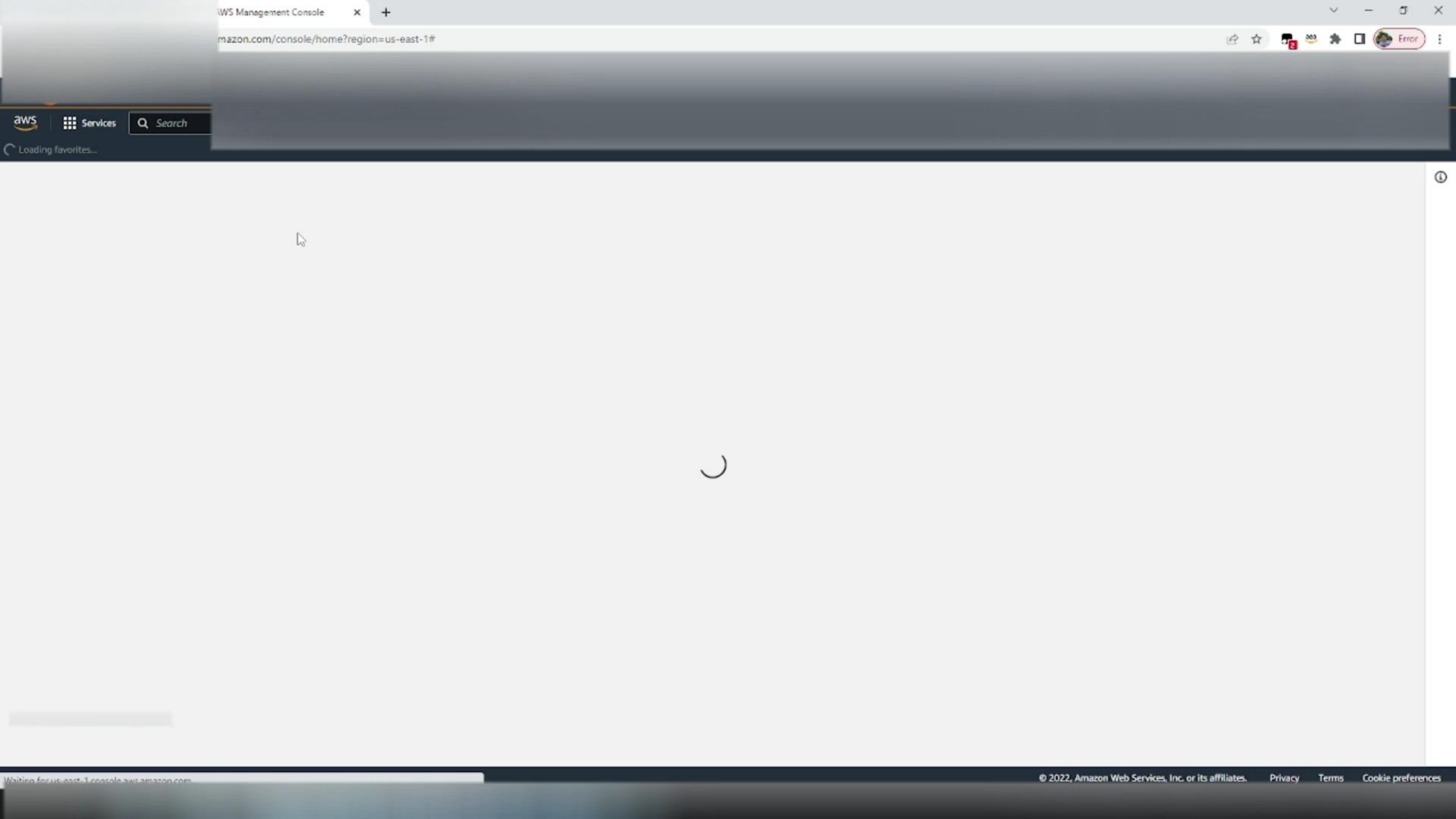
- Build Apache Spark applications using Java, Python, Scala, with easy access to data in your data warehouse
- Get started quickly with a fully packaged and certified integration that does not require manual setup and maintenance
- Advanced pushdown optimizations to speed up query processing - accelerates 3TB out of the box cloud data warehousing benchmark queries derived from TPC-DS by 10x
- Improved security with IAM-based credentials

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s see a quick demo



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.


WS Management Console
mazon.com/console/home?region=us-east-1#
Error
aws
Services
Search
Loading favorites...
Writing for us-east-1.console.aws.amazon.com...
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Amazon Redshift: 10 years of innovation



- Tens of thousands of customers process exabytes of data daily with Amazon Redshift
- Industry-leading security and access control, out-of-the box at no additional cost
- Continuous focus on performance and scalability
- Ability to elastically support 10s of PBs of data and 1000s of users
- Autonomics that make Redshift easy to use
- Serverless experience with intelligent compute management
- Tight integration with the broad AWS environment
- Data Mesh with Amazon Redshift Provisioned and Serverless

And it is still Day 1

Get Started today

New to Amazon Redshift?

- ✓ IDC study on Business value of Amazon Redshift
- ✓ Explore demos, customer stories, and the latest features on aws.amazon.com/redshift
- ✓ Get trained
- ✓ Ask your account team for a 10-minute demo

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/

Thinking of migration?

AWS teams and programs can help you with

Building your business case, creating migration plan

Deciding on budget, scope, etc.

Conduct a proof of concept or architect your data strategy

Get Help

aws.amazon.com/redshift

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Ippokratis Pandis
ippo@amazon.com

Neeraja Rentachintala
neerajre@amazon.com



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.