

Databases and Analytics on AWS

Praful Kava

Solution Architect, Database and Analytics, AWS

pkava@amazon.com

Rajeev Thottathil

Solution Architect, Database and Analytics, AWS

thottr@amazon.com

Agenda

- AWS Data and Analytics Strategy
- Purpose-Built Database strategy : Right tool for the right job
 - RDS
 - Aurora
 - DynamoDB
 - ElastiCache
 - Timestream DB
 - Quantum Ledger DB (QLDB)
- Building Data Lake on AWS : S3, Lake Formation and Glue

Our strategy & our beliefs

1. There is going to be an explosion in data.
2. Cloud will enable a different architecture.
3. One size does not fit all—databases should be purpose-built.

2010

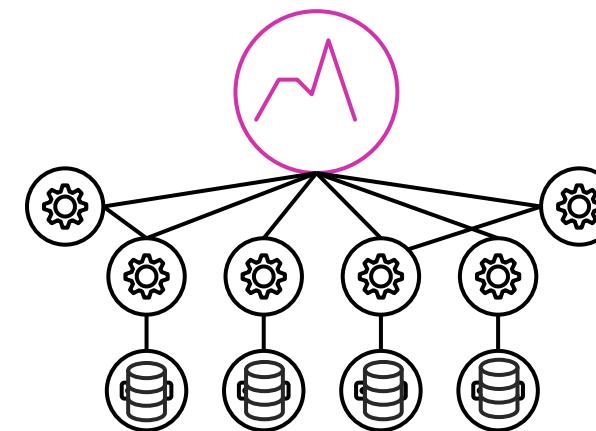
Trends that impact the way you think about data

Explosion of data



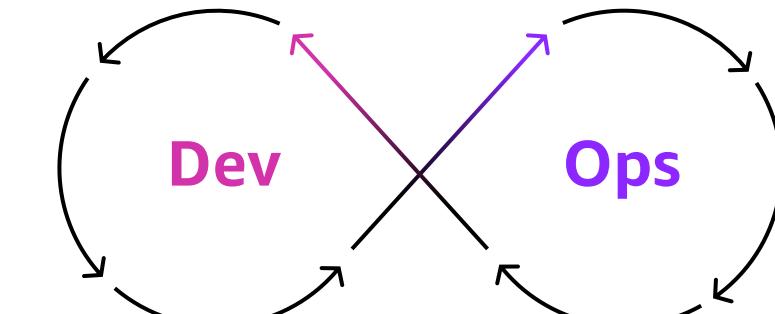
Data grows 10x every 5 years driven by network connected smart devices

Micro-services changes data and analytics requirements



Micro-services architecture decreases need for one-size fits all databases and increases need for real-time monitoring and analytics

Rapid rate of change driven by DevOps



Transition from IT to DevOps increases rate of change

Our portfolio

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning



QuickSight



SageMaker

Analytics



Redshift
Data warehousing



EMR
Hadoop + Spark



Athena
Interactive analytics



Elasticsearch Service
Operational Analytics



Kinesis Data Analytics
Real time

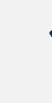
Data Lake



S3/Glacier



Lake Formation NEW
Data Lakes



Glue
ETL & Data Catalog

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Our portfolio

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning



QuickSight



SageMaker

Analytics



Redshift
Data warehousing



EMR
Hadoop + Spark



Athena
Interactive analytics



Elasticsearch Service
Operational Analytics



Kinesis Data Analytics
Real time

Databases



Aurora
MySQL, PostgreSQL



DynamoDB
Key value, Document



RDS
MySQL, PostgreSQL, MariaDB,
Oracle, SQL Server



ElastiCache
Redis, Memcached



RDS on VMware



Neptune
Graph



QLDB
Ledger Database



Timestream
Time Series

Data Lake



S3/Glacier



Lake Formation
Data Lakes



Glue
ETL & Data Catalog

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Our portfolio

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning



QuickSight



SageMaker

Analytics



Redshift

Data warehousing



EMR

Hadoop + Spark



Athena

Interactive analytics



Elasticsearch Service

Operational Analytics



Kinesis Data Analytics

Real time

Databases



Aurora

MySQL, PostgreSQL



DynamoDB

Key value, Document



RDS

MySQL, PostgreSQL, MariaDB,
Oracle, SQL Server



ElastiCache

Redis, Memcached



RDS on VMware



Neptune

Graph

Blockchain



Managed
Blockchain



Blockchain
Templates

Data Lake



S3/Glacier



Lake Formation

Data Lakes



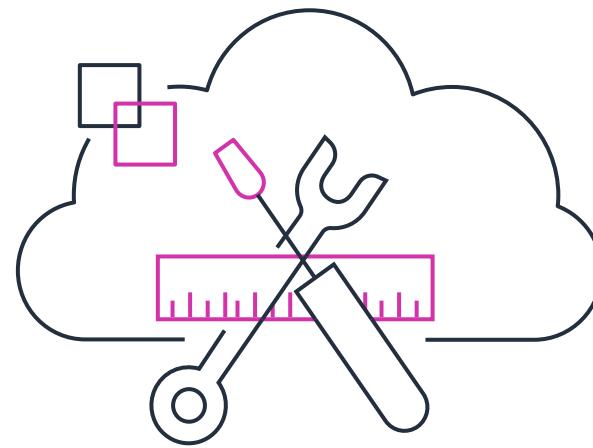
Glue

ETL & Data Catalog

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Three type of projects



Quickly build new
apps in the cloud

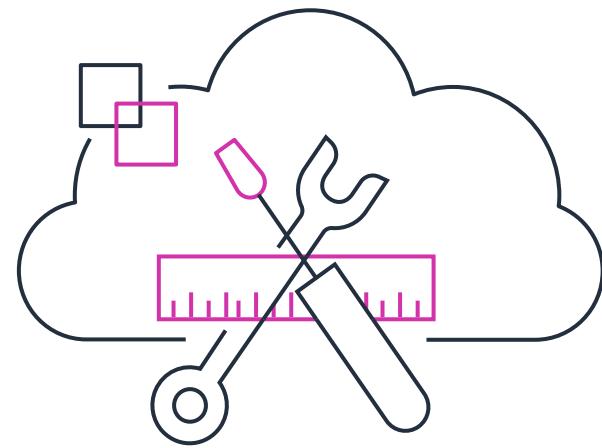


“Lift and shift” existing
apps to the cloud



Gain new
insights

Customers tell us: they have three type of projects



Quickly build new
apps in the cloud

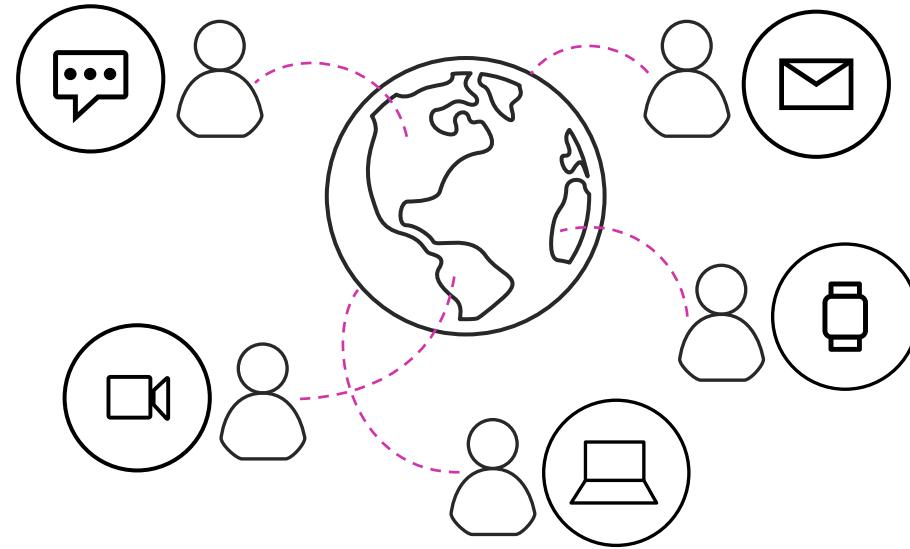


“Lift and shift” existing
apps to the cloud

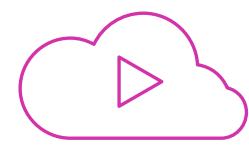


Gain new
insights

Modern apps create new requirements



Ride hailing



Media streaming



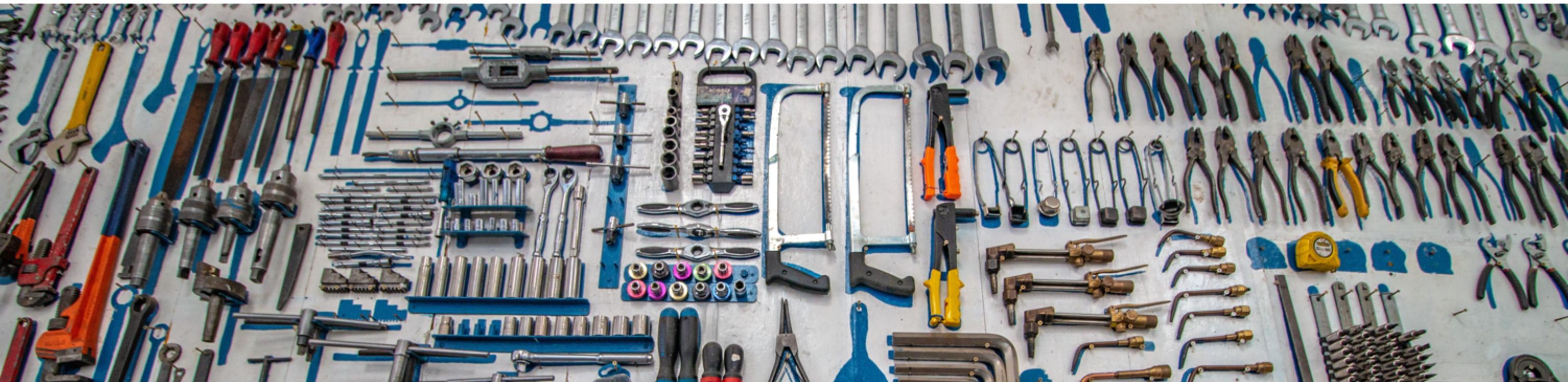
Social media



Dating

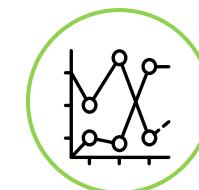
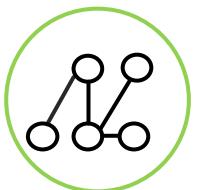
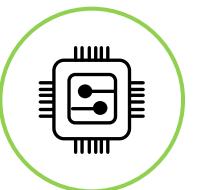
- Users:** 1 million+
- Data volume:** TB–PB–EB
- Locality:** Global
- Performance:** Milliseconds–microseconds
- Request rate:** Millions
- Access:** Web, mobile, IoT, devices
- Scale:** Up-down, Out-in
- Economics:** Pay for what you use
- Developer access:** No assembly required

Working backward from the problem you are trying to solve

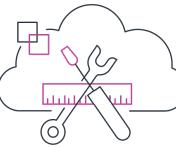


Choosing the right tool for each job

Data categories and common use cases

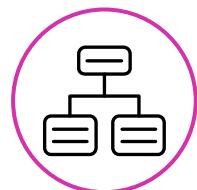


Relational	Key value	Document	In-memory	Graph	Search	Time series	Ledger
Referential integrity, ACID transactions, schema-on-write	Low-latency, key lookups with high throughput and fast ingestion of data	Indexing and storing documents with support for query on any attribute	Microseconds latency, key-based queries, and specialized data structures	Creating and navigating data relations easily and quickly	Indexing and searching semistructured logs and data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data
Lift and shift, EMR, CRM, finance	Real-time bidding, shopping cart, social	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	Product catalog, help, and FAQs, full text	IoT applications, event tracking	Systems of record, supply chain, healthcare, registrations, financial

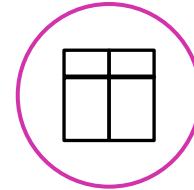


AWS databases services

Purpose-built for all your app needs



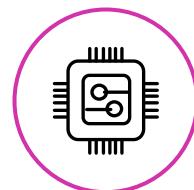
Relational



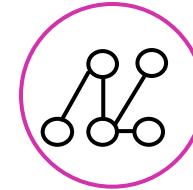
Key-value



Document



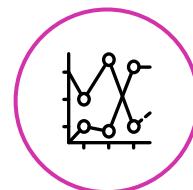
In-memory



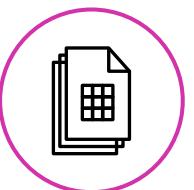
Graph



Search



Time series



Ledger



RDS



DynamoDB



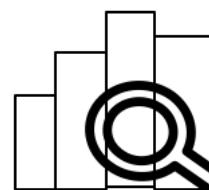
Amazon DocumentDB



ElastiCache



Neptune



ElasticSearch



Timestream



QLDB

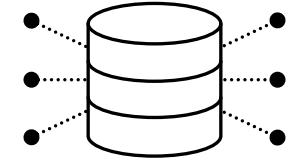
Aurora Community Commercial



ORACLE®

PostgreSQL PostgreSQL Microsoft® SQL Server®





What is Amazon Relational Database Service (RDS)?

Relational databases are complex



Our experience running Amazon.com taught us that relational databases are challenging to manage and operate with high availability

It's expensive and complex to manage administrative functions including **regular patching cycles, performance optimization, and backup and disaster recovery—all for constantly changing applications**

Self managing relational databases is time consuming, complex, and expensive

- Hardware & software installation, configuration, patching, and backups
- Performance and high availability issues
- Capacity planning, and scaling clusters for compute and storage

Security and compliance

Amazon RDS

Managed relational database service with a choice of six popular database engines



Easy to administer



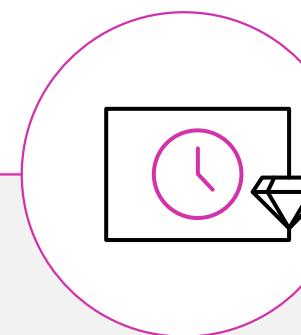
Easily deploy and maintain hardware, OS and DB software; built-in monitoring

Secure & compliant



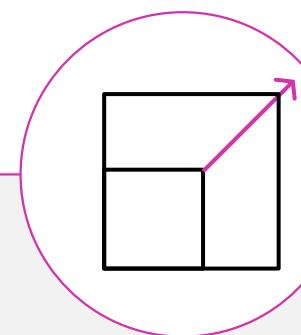
Data encryption at rest and in transit; industry compliance and assurance programs

Available & durable



Automatic Multi-AZ data replication; automated backup, snapshots, failover

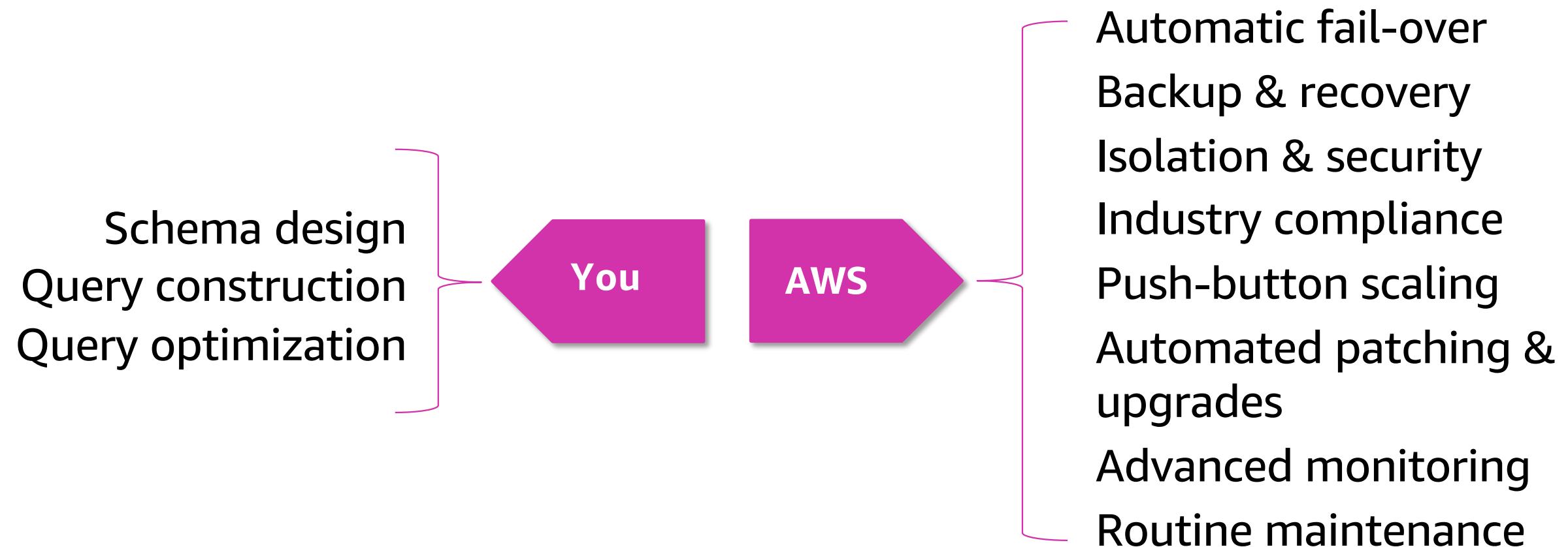
Performant & scalable



Scale compute and storage with a few clicks; minimal downtime for your application

Amazon RDS - fully managed

Spend time innovating & building new apps, not managing infrastructure



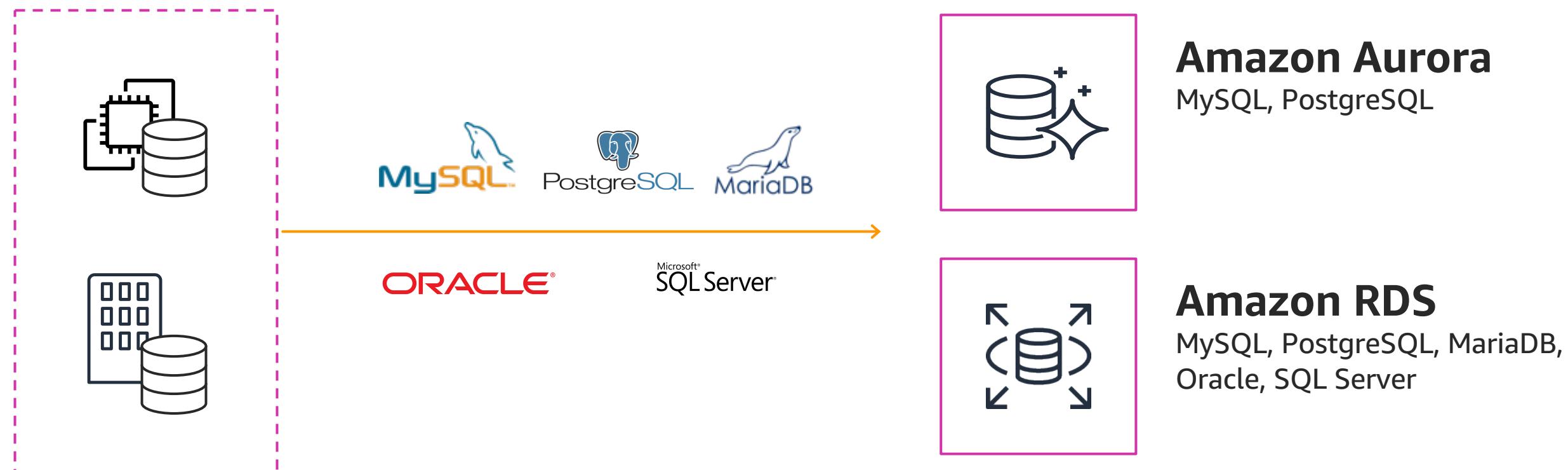
Move to managed relational databases

Migrate on-premises or cloud-hosted relational databases to managed services

Reduce DB administrative burden

No need to rearchitect existing applications

Get better performance, availability, scalability, and security



Hundreds of thousands of customers use Amazon RDS





A community marketplace that allows property owners and travelers to connect with each other for the purpose of renting unique vacation spaces around the world.

Challenge:

Airbnb experienced service challenges with its original provider and wanted to scale their internet business to the next level.

Solution:

Airbnb moved its main MySQL database to Amazon RDS for MySQL with only 15 minutes of downtime.

Result

- RDS simplifies time-consuming database administrative tasks so engineers can spend more time on features.
- Airbnb uses asynchronous replication to improve website performance launched via the RDS console or an API call.
- Multi-Availability Zone (Multi-AZ) provides Airbnb with high availability.



Extended Stay America moves to RDS for SQL Server



Company: Extended Stay America

Industry: Hospitality

Website: <https://www.extendedstayamerica.com/>

Extended Stay America, Inc. is the operator of an economy, extended-stay hotel chain consisting of 627 properties in the United States

Challenge

The largest Extended Stay hotel chain was looking to increase reliability and scale their e-commerce and guest Wi-Fi application for their 627 hotels

Approach

Leveraging the fully managed RDS for SQL Server and Aurora helped Extended Stay increase scalability of their e-commerce and guest Wi-Fi applications, which allowed them to distribute their applications across more availability zones with higher availability

Results

Improved availability and added redundancy to e-commerce & guest Wi-Fi applications

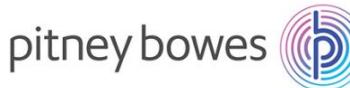
Reduced operational costs by 25% allowing them to reinvest and deliver more services to their core business faster

Scaling our e-commerce applications with RDS SQL Server, we saw higher availability, scalability and were able to reduce operational cost by 25%, allowing us to reinvest savings in our core business.

>200,000 databases migrated to AWS



U.S. Department
of Veterans Affairs



THOMSON REUTERS



Dokter Anda, Kapan Saja.



Sotheby's





Trimble is a global leader in telematics solutions

Challenge:

Migrated their on-premise data warehouse to scale, improve availability, and cut cost

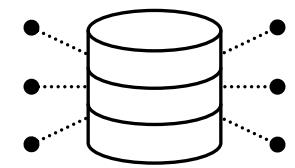
Solution:

Trimble migrated from Oracle to Amazon RDS for PostgreSQL using AWS Database Migration Service (DMS)

Result:

Trimble's infrastructure costs are projected to be less than one-quarter of their privately hosted infrastructure. Trimble expects these to reduce operational overhead





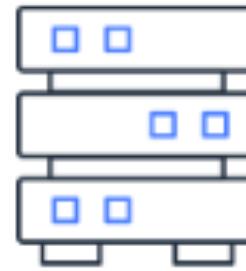
Easy to
administer

Ease of administration



- Single console and API for managing all your relational databases
- Hardware provisioning, patching, backup/restore, scaling, and high availability with a few clicks
- Security and monitoring is built in

Monitoring RDS/Aurora databases



Instance

Amazon CloudWatch

- CPU / Memory / IOPS / Network
- Per minute metric storage in Amazon CloudWatch



Operating System

Amazon RDS Enhanced Monitoring

- Process / Thread list
- Per second metric storage in Amazon CloudWatch Logs

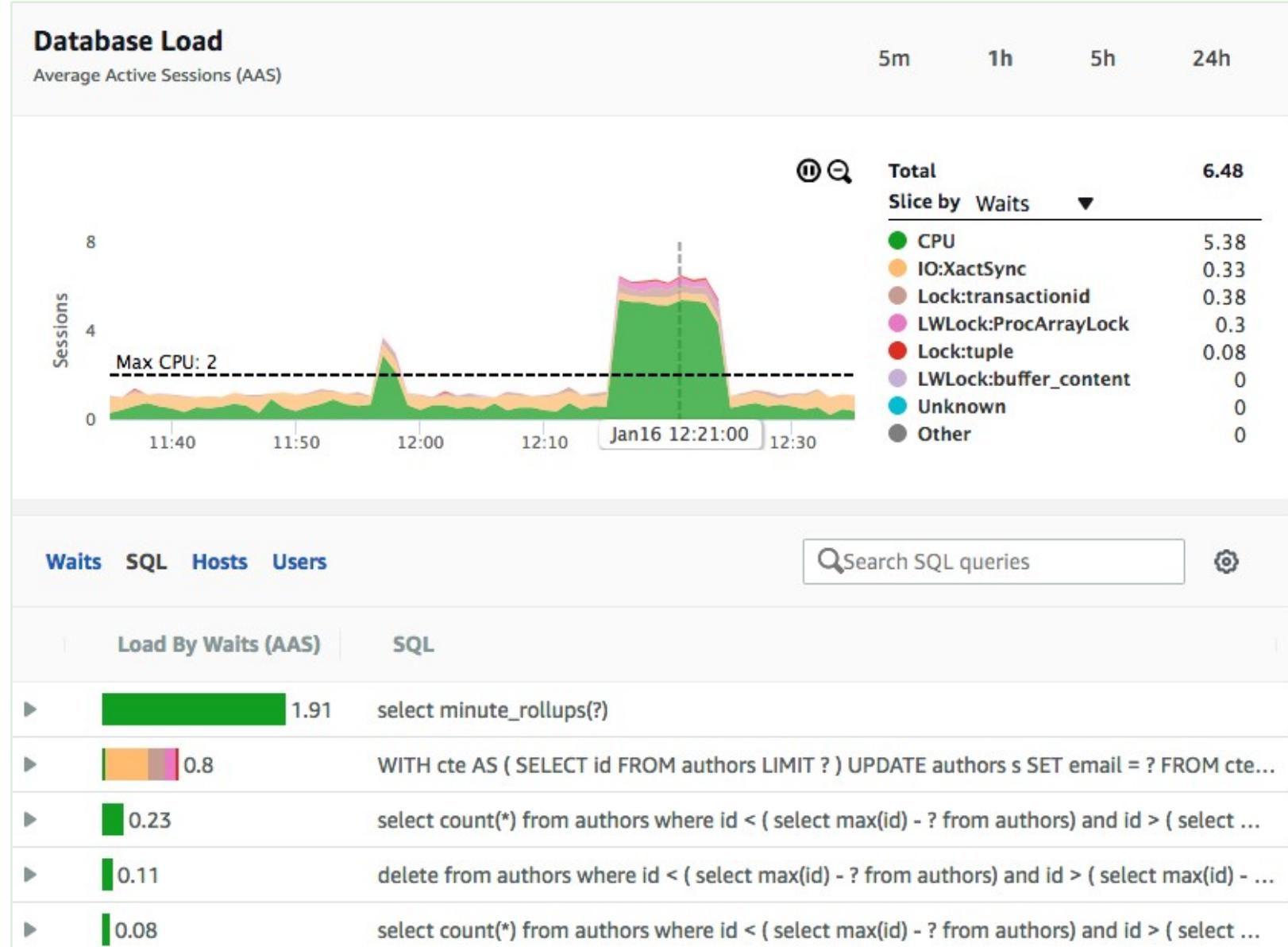


Database Engine

Amazon RDS Performance Insights

- SQL / State / User / Host (“Database Load”)
- Per second metric storage in Amazon RDS

Performance Insights increases productivity



Amazon RDS Performance Insights measures database load over time

Easy to identify database bottlenecks

- Top SQL/most intensive queries

Enables problem discovery

Adjustable timeframe

- Hour, day, week, and longer

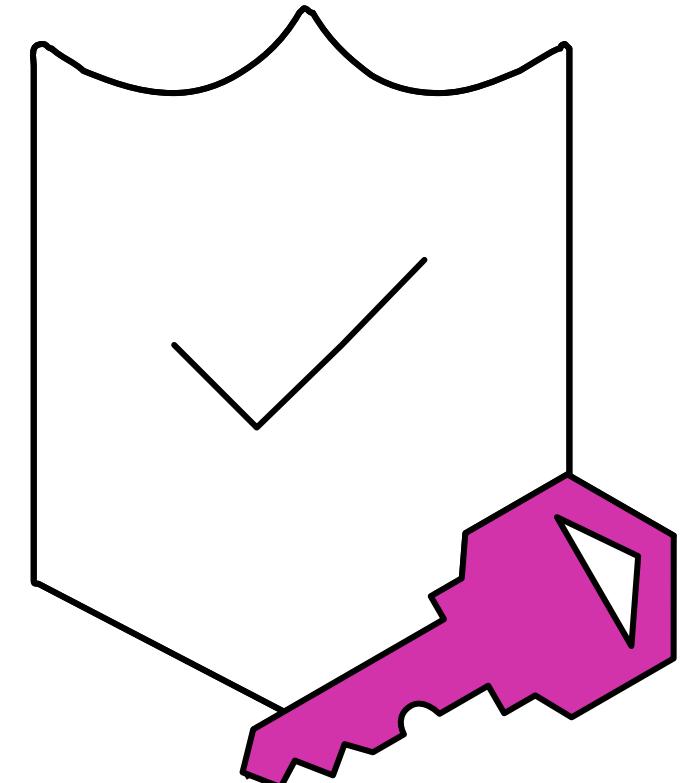
Available for all Amazon RDS database engines



Secure &
compliant

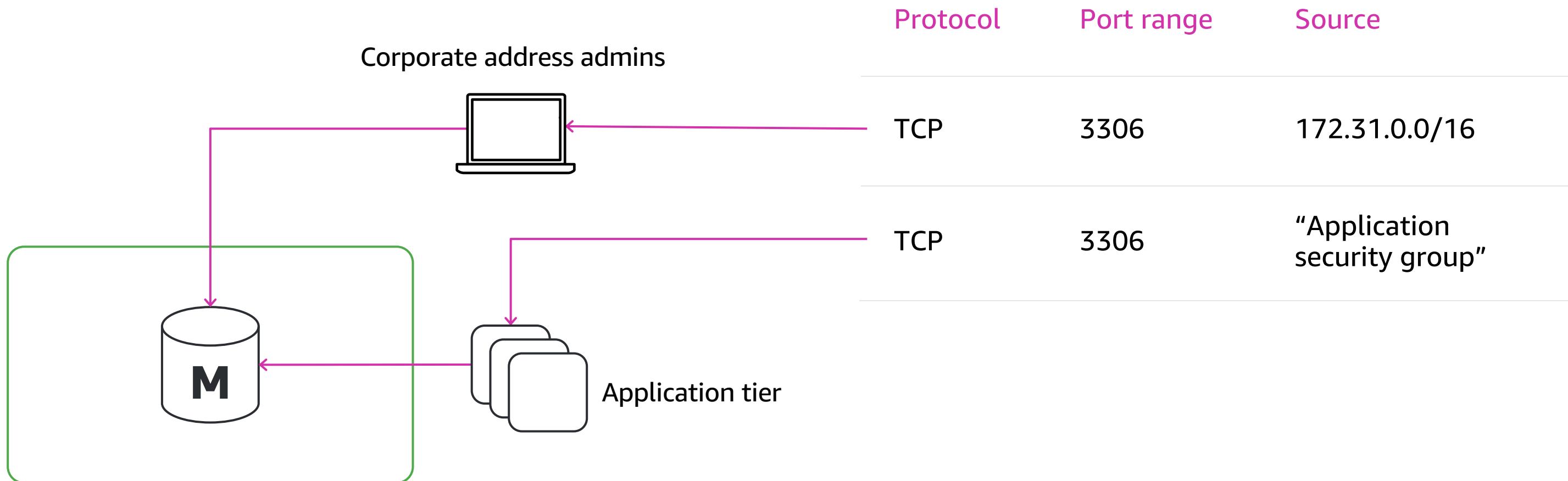
Security and compliance

- Network security
 - Amazon Virtual Private Cloud (VPC) security groups act as a virtual firewall to control inbound and outbound traffic
- Resource access permissions
 - AWS Identity and Access Management (IAM) provides resource-level role permission controls
- Data encryption
 - Encryption at rest using AWS KMS or Oracle/Microsoft TDE
 - SSL protection for data in transit
- Compliance and assurance programs for finance, healthcare, government, and more
 - HIPAA eligibility under a Business Associate Agreement (BAA) with AWS
- Active Directory / Kerberos integration
 - RDS for Oracle, SQL Server, PostgreSQL



Secure network access

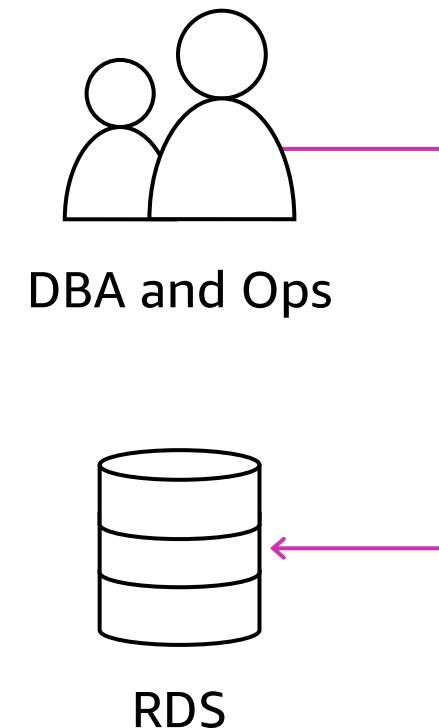
Controlled through Amazon Virtual Private Cloud (VPC) security groups



Resource-level role permissions

Enabled through AWS Identity and Access Management (IAM)

Governed access:
use IAM to control who
can perform actions
with Amazon RDS



```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowCreateDBInstanceOnly",  
            "Effect": "Allow",  
            "Action": [  
                "rds>CreateDBInstance"  
            ],  
            "Resource": [  
                "arn:aws:rds:*:123456789012:db:test*",  
                "arn:aws:rds:*:123456789012:og:default*",  
                "arn:aws:rds:*:123456789012:pg:default*",  
                "arn:aws:rds:*:123456789012:subgrp:default"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "rds:DatabaseEngine": "mysql",  
                    "rds:DatabaseClass": "db.t2.micro"  
                }  
            }  
        }  
    ]  
}
```

Encryption of data at rest

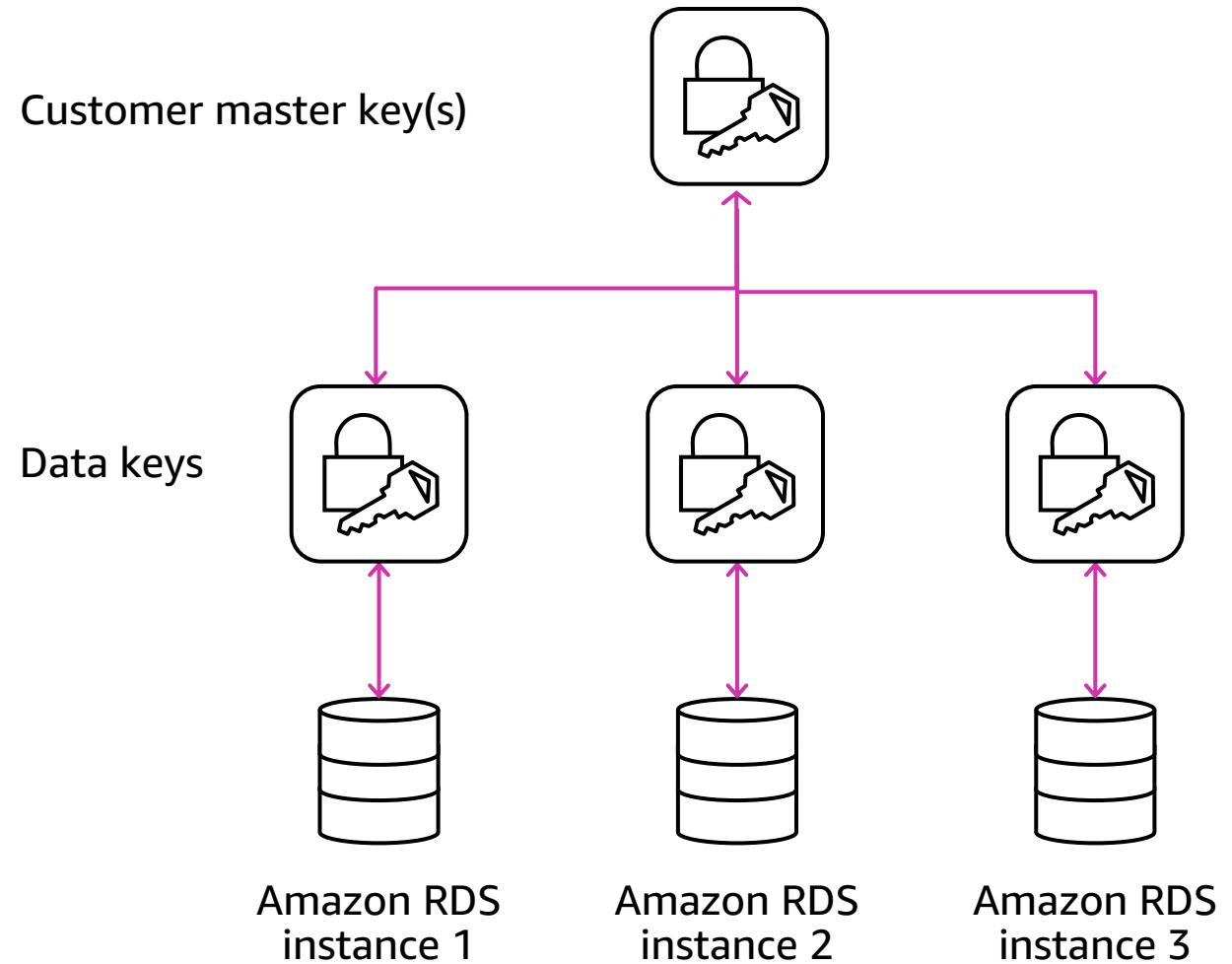
Managed via AWS Key Management Service (KMS)

Two-tiered key hierarchy using envelope encryption

- Unique data key encrypts customer data
- AWS KMS master keys encrypt data keys
- Available for all RDS engines

Benefits

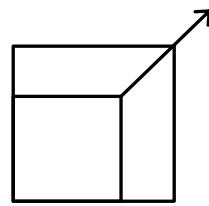
- Limits risk of compromised data key
- Better performance for encrypting large data
- Easier to manage small number of master keys than millions of data keys
- Centralized access and audit of key activity



Compliance

Aurora	MySQL	PostgreSQL	Oracle	MariaDB	SQL Server
• SOC 1, 2, 3	• SOC 1, 2, 3	• SOC 1, 2, 3	• SOC 1, 2, 3	• SOC 1, 2, 3	• SOC 1, 2, 3
• ISO 27001/9001	• ISO 27001/9001	• ISO 27001/9001	• ISO 27001/9001	• ISO 27001/9001	• ISO 27001/9001
• ISO 27017/27018	• ISO 27017/27018	• ISO 27017/27018	• ISO 27017/27018	• ISO 27017/27018	• ISO 27017/27018
• PCI	• PCI	• PCI	• PCI	• PCI	• PCI
• FedRAMP	• FedRAMP	• FedRAMP	• FedRAMP	• HIPAA BAA	• HIPAA BAA
• HIPAA BAA	• HIPAA BAA	• HIPAA BAA	• HIPAA BAA		• UK Gov. Programs
	• UK Gov. Programs	• UK Gov. Programs	• UK Gov. Programs		• Singapore MTCS
	• Singapore MTCS	• Singapore MTCS	• Singapore MTCS		• FedRAMP





Highly scalable

Database server instance types

General purpose (T3)	General Purpose (M5)	Memory Optimized (R5)	Memory Optimized (X1E)
<ul style="list-style-type: none">• 1 vCPU / 1 GB RAM > 8 vCPU 32 GB RAM• Moderate networking performance• Built on the AWS Nitro System• Unlimited and Standard mode• Good for smaller or variable workloads	<ul style="list-style-type: none">• 2 vCPU / 8 GiB RAM > 96 vCPU 384 GiB RAM• High performance networking• Built on the AWS Nitro System• Good for running CPU intensive workloads (e.g. WordPress)	<ul style="list-style-type: none">• 2 vCPU / 16 GiB RAM > 96 vCPU 768 GiB RAM• High performance networking• Built on the AWS Nitro System• Good for query intensive workloads or high connection counts	<ul style="list-style-type: none">• RDS Oracle and SQL Server only• 4 vCPU / 122 GiB RAM > 128 vCPU 3904 GiB RAM• High performance networking• X1e instances offer the highest memory per vCPU among instance types and one of the lowest price per GiB of memory• Good for query intensive workloads or high connection counts

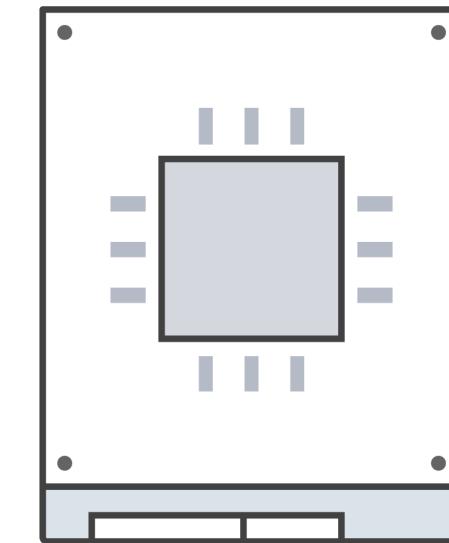
High performance database storage

General purpose (GP2)

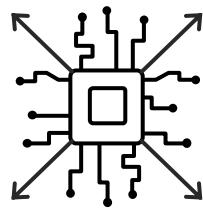
- SSD storage
- Maximum of 64 TiB (16TiB for SQL Server)
- Latency in milliseconds
- IOPS determined by volume size
- Bursts to 3,000 IOPS (applicable below 1.3 TB)
- Affordable performance

Provisioned IOPS (IO1)

- SSD storage
- Maximum of 64 TiB (16TiB for SQL Server)
- Single digit millisecond latencies
- Maximum of 80K IOPS (40K for SQL Server)
- Delivers within 10% of the IOPS performance 99.9% of the time
- High performance and consistency

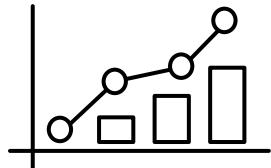


Scale compute and storage with ease



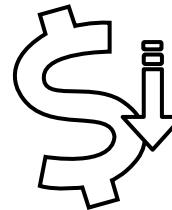
Scale compute to handle increased load

- Up to 96 vCPU and 768 GiB of RAM



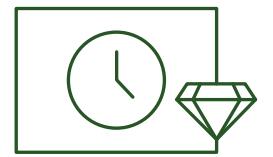
Scale storage for larger data sets

- Quickly scale EBS storage up to 64TiB (16TiB for SQL Server)
- No downtime for storage scaling



Scale down to control costs

- As little as 1vCPU / 1 GiB of RAM



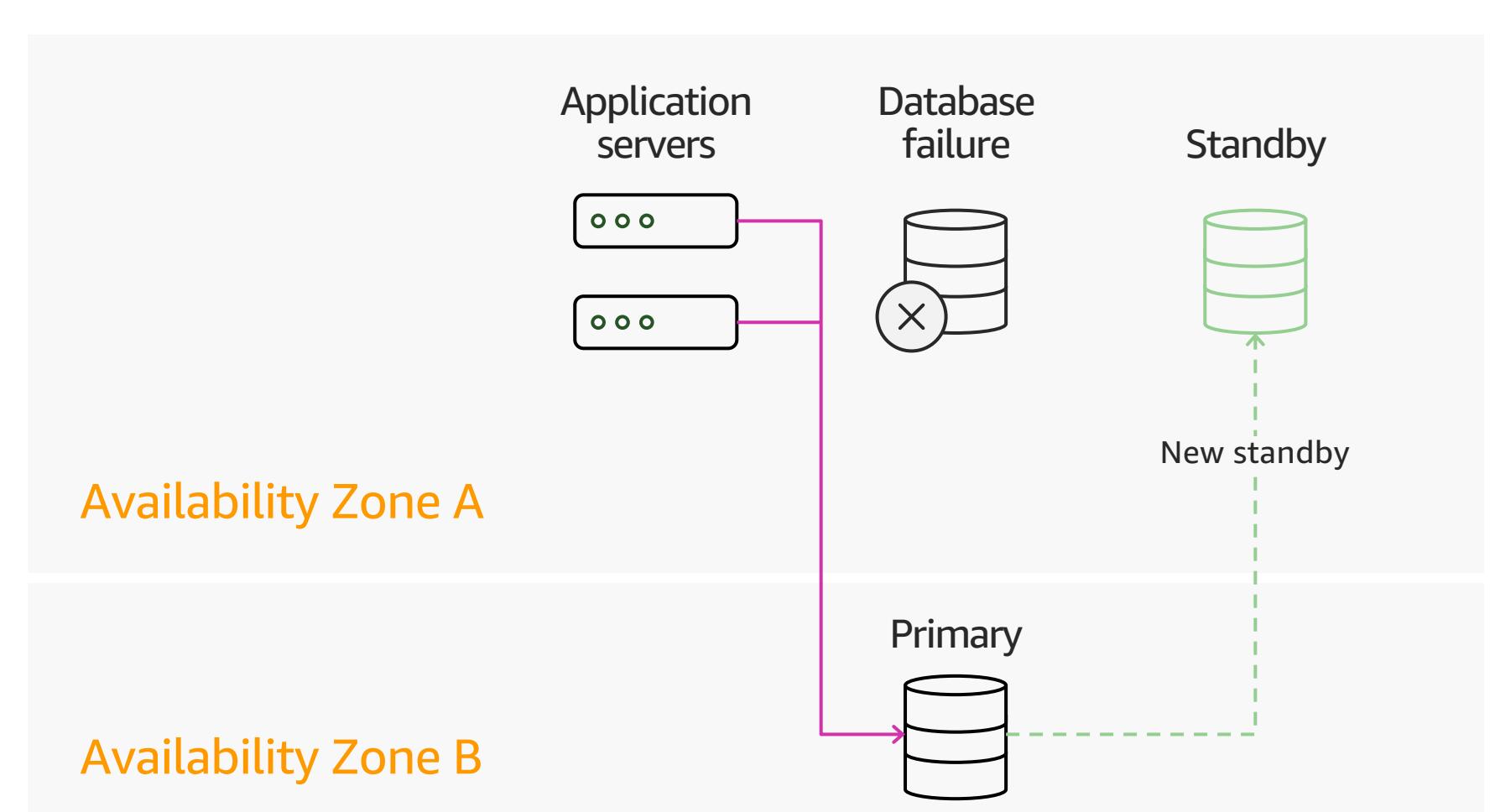
Available &
durable

Multi-AZ deployments

Enterprise-grade high availability

Fault tolerance across multiple data centers

- Automatic failover
- Synchronous replication
- Enabled with one click

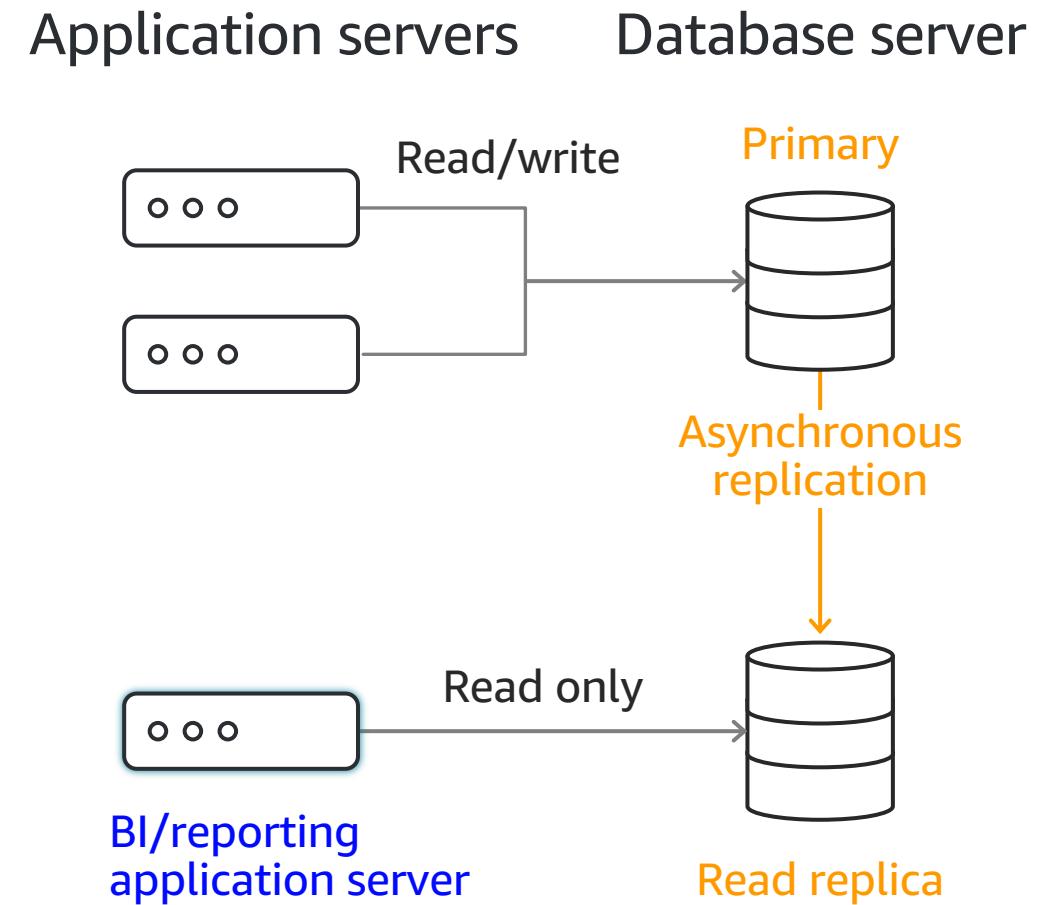


Read Replicas

Read scaling and disaster recovery

RDS for MySQL, PostgreSQL, MariaDB, and Oracle

- Relieve pressure on your master node with additional read capacity
- Bring data close to your applications in different regions
- Promote a read replica to a master for faster recovery in the event of disaster



Automated backups

Point-in-time recovery for your DB instance

- Scheduled daily volume backup of entire instance
- Archive database change logs
- 35-day maximum retention
- Minimal impact on database performance
- Taken from standby when running Multi-AZ

DB instance status

available

Multi AZ

Yes

Secondary zone

us-east-1d

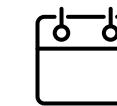
Automated backups

Enabled (7 Days)

Latest restore time

March 22, 2018 at 10:25:00 AM

UTC-7



Every day during your backup window, RDS creates a storage volume snapshot of your instance



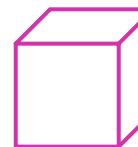
Every five minutes, RDS backs up the transaction logs of your database

Database snapshots

Backups of your entire DB instance in Amazon S3

- Always incremental
- Amazon S3 → 99.99999999% durability
- Supports encryption
- Copy across accounts, across regions

Amazon EBS



Volume

Amazon S3



Bucket



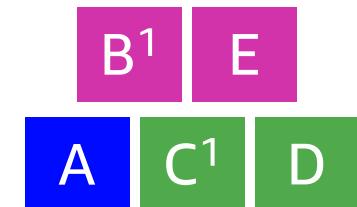
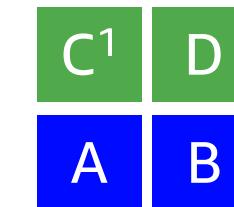
Snapshot 1

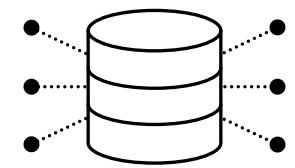


Snapshot 2



Snapshot 3





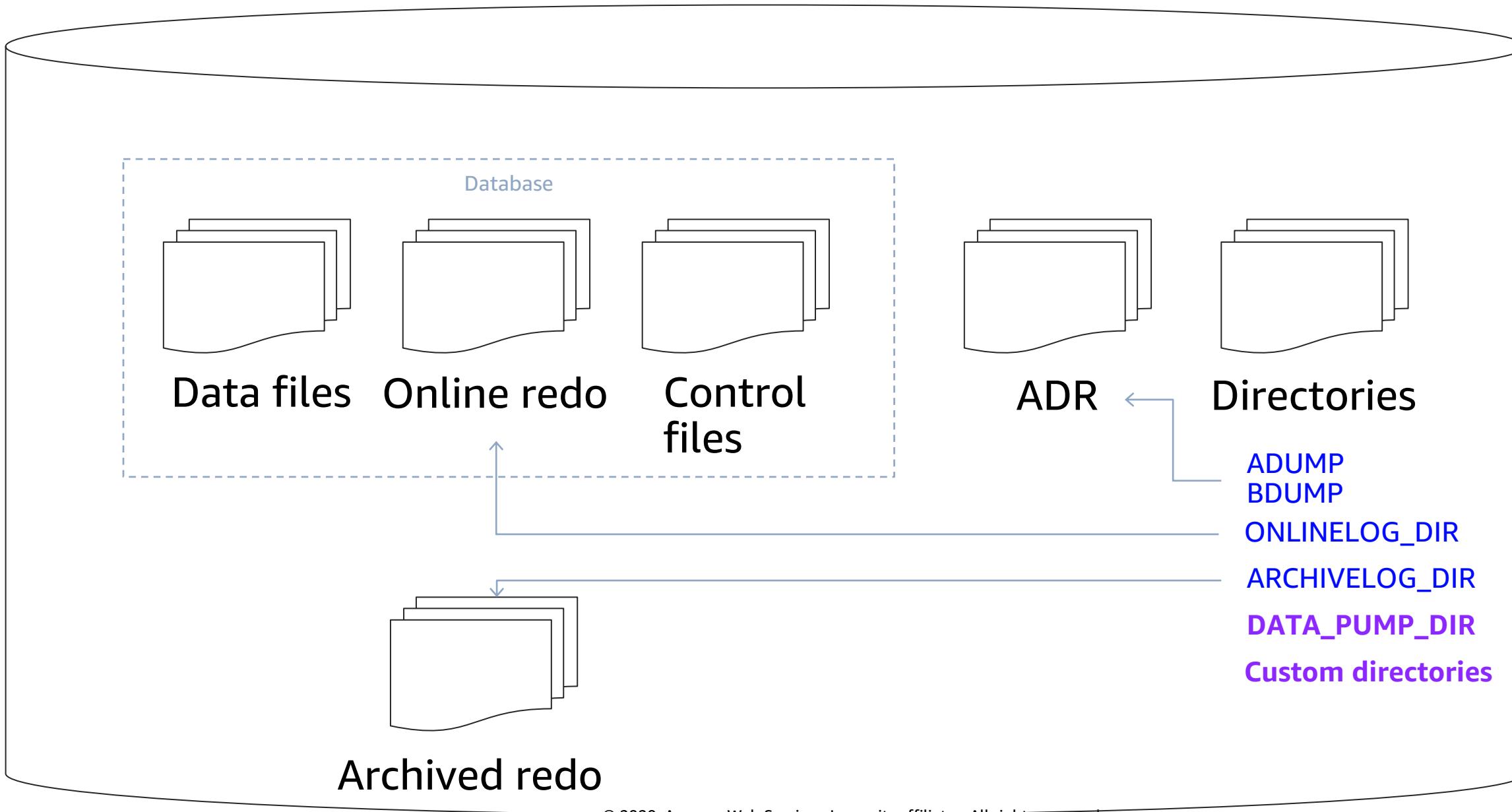
Oracle RDS

Supported Versions

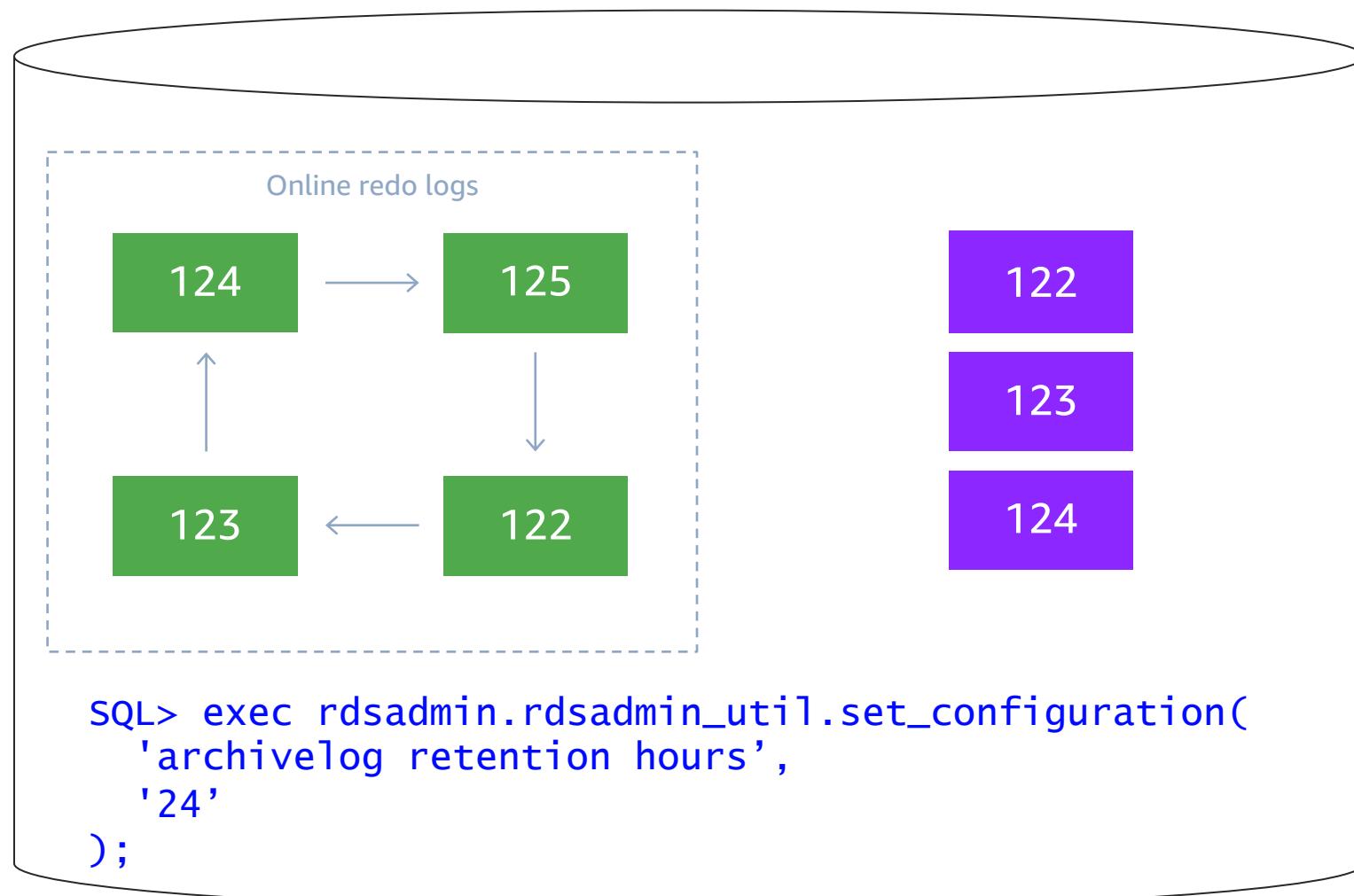
Versions

- 11.2.0.4
- 12.1.0.2
- 12.2.0.1
- 18c (12.2.0.2)
- 19c (12.2.0.3)

Amazon RDS storage

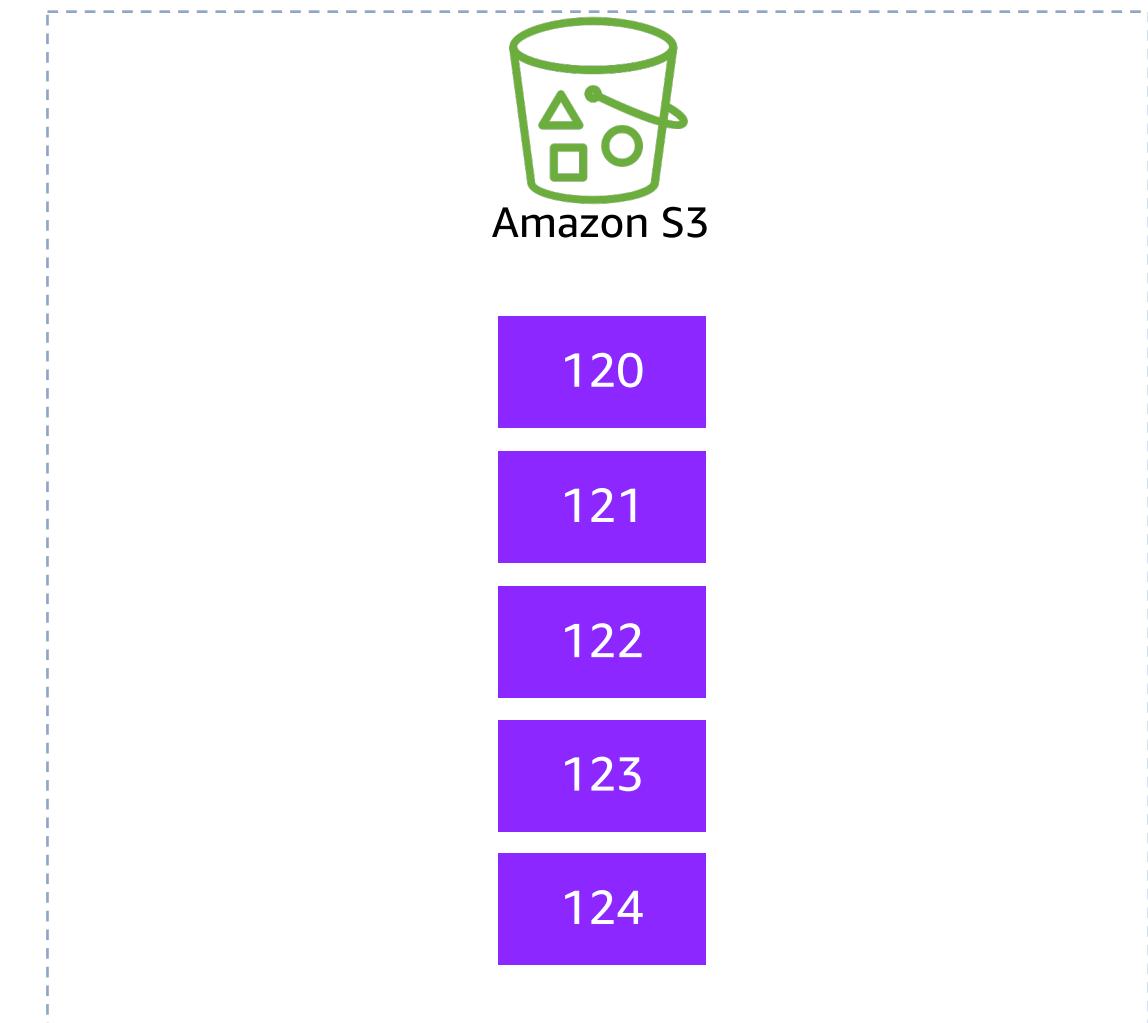


Archived redo logs



On-host retention: set with **rdsadmin.rdsadmin_util** package
(space dependent; default 0 hours)

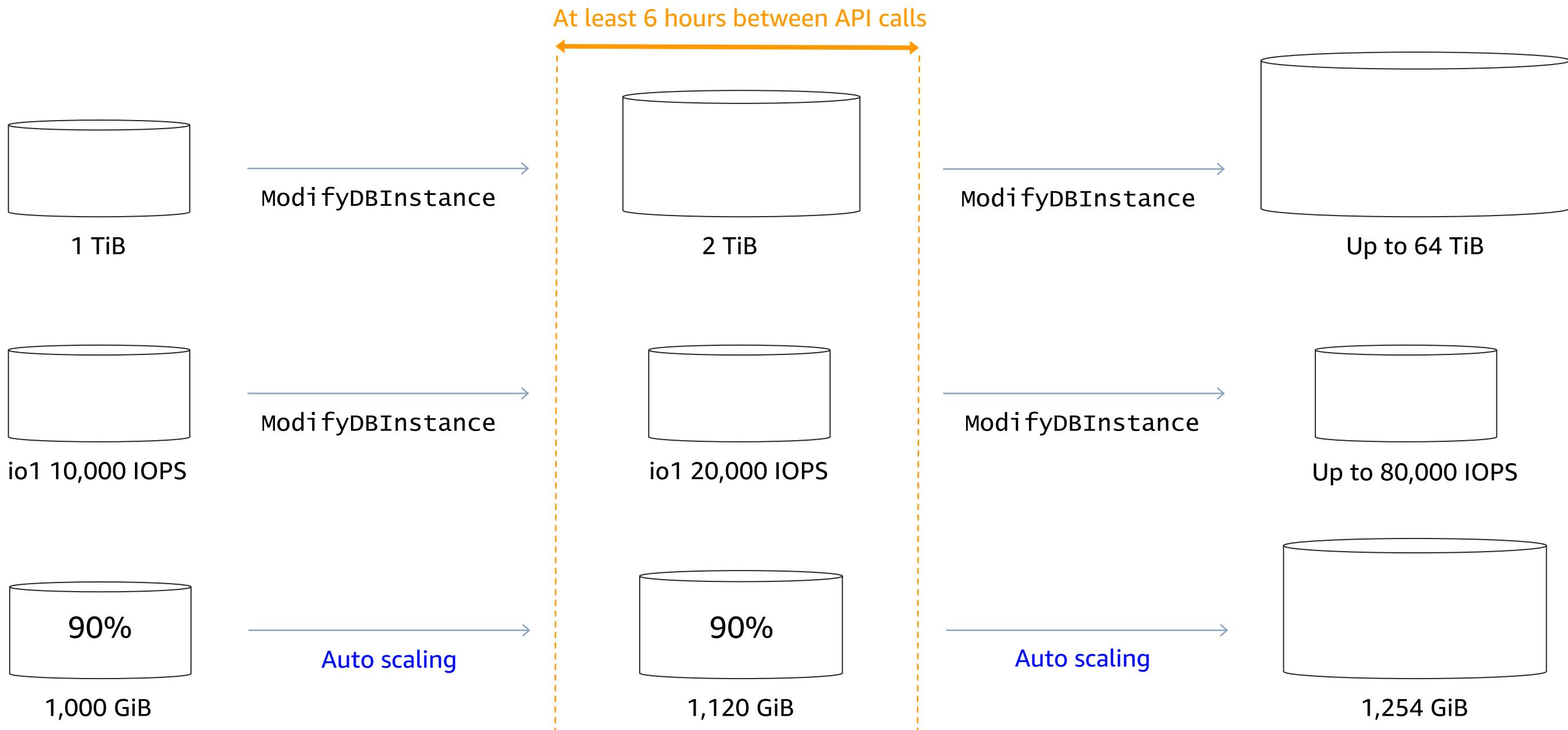
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Amazon S3 retention: Amazon RDS instance
BackupRetentionPeriod
(1–35 days, default 1 day)



Amazon RDS storage scaling



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

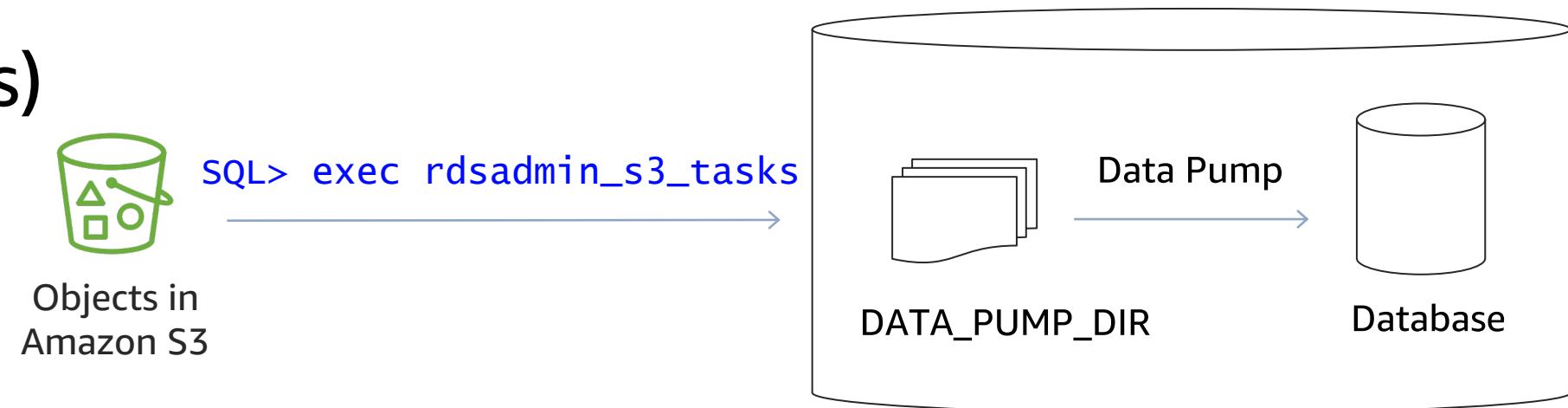
NEW! Amazon RDS storage auto scaling – greater of 5 GiB or 12% when less than 10% free storage for 5 minutes



Loading data

Local data (directory objects)

- Data Pump
- External tables



Remote data (client systems)

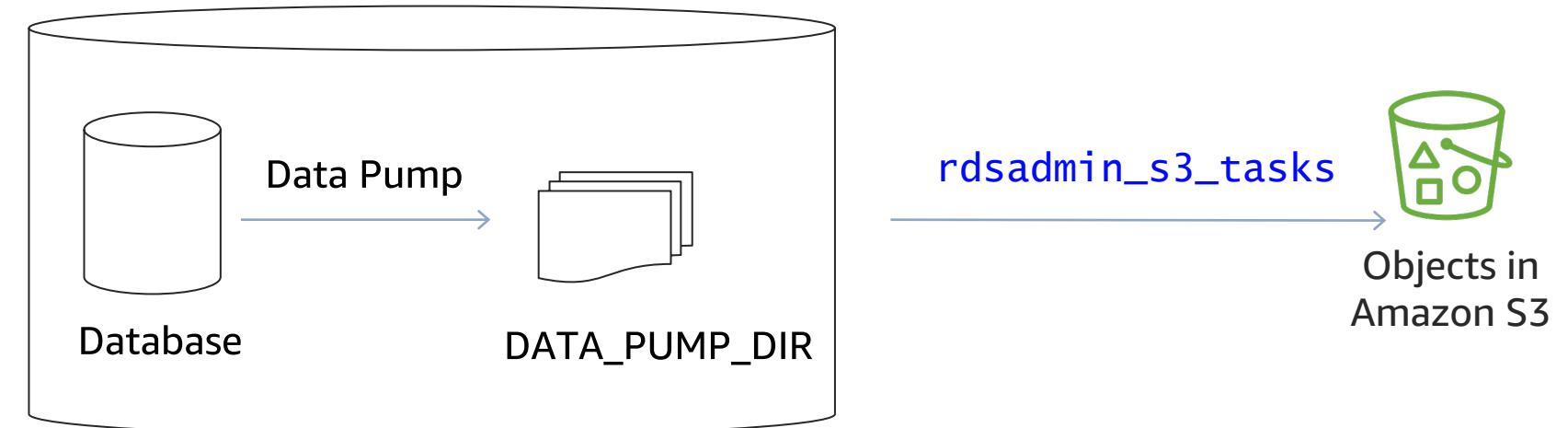
- Legacy imp utility
- Oracle SQL*Loader
- Oracle SQL Developer
- Client applications



Unloading data

Local data (directory objects)

- Data Pump
- External tables
- RMAN

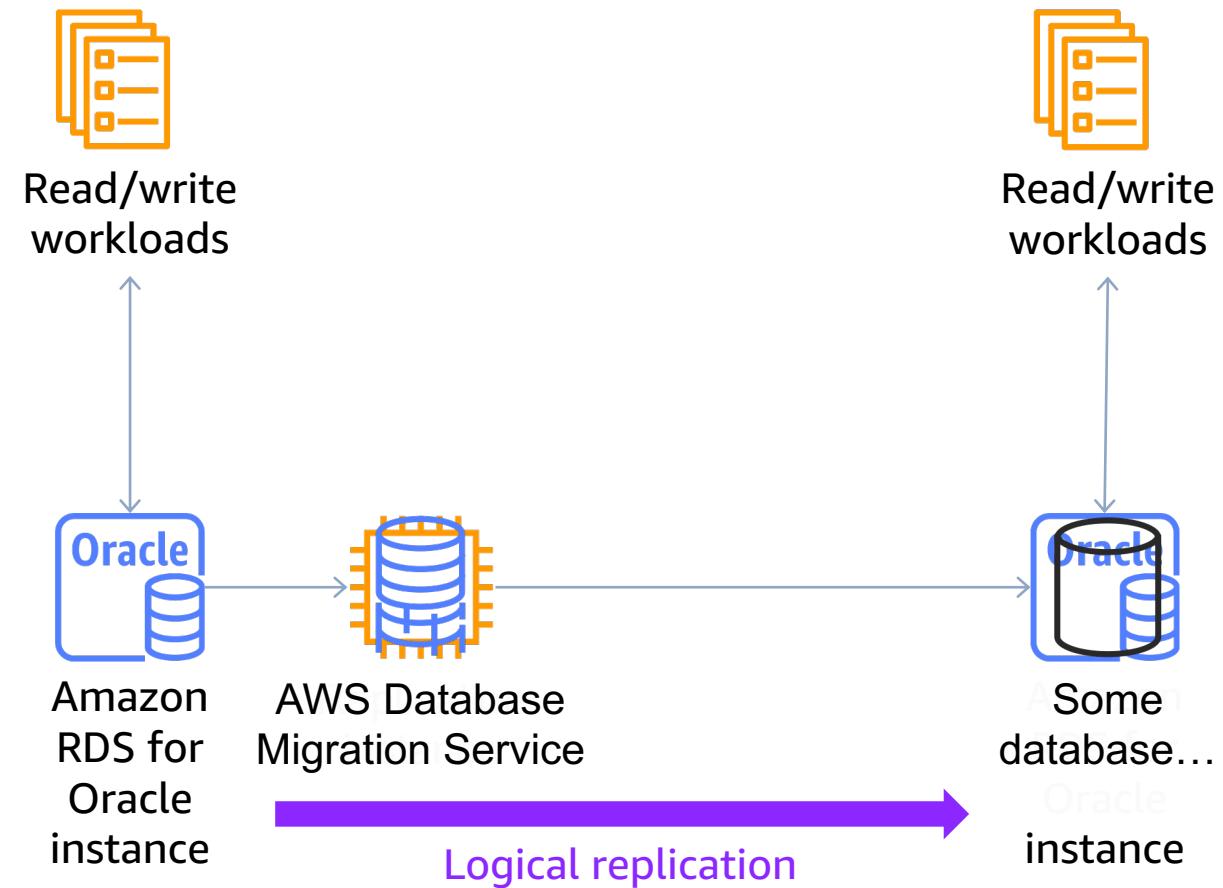


Remote data (client systems)

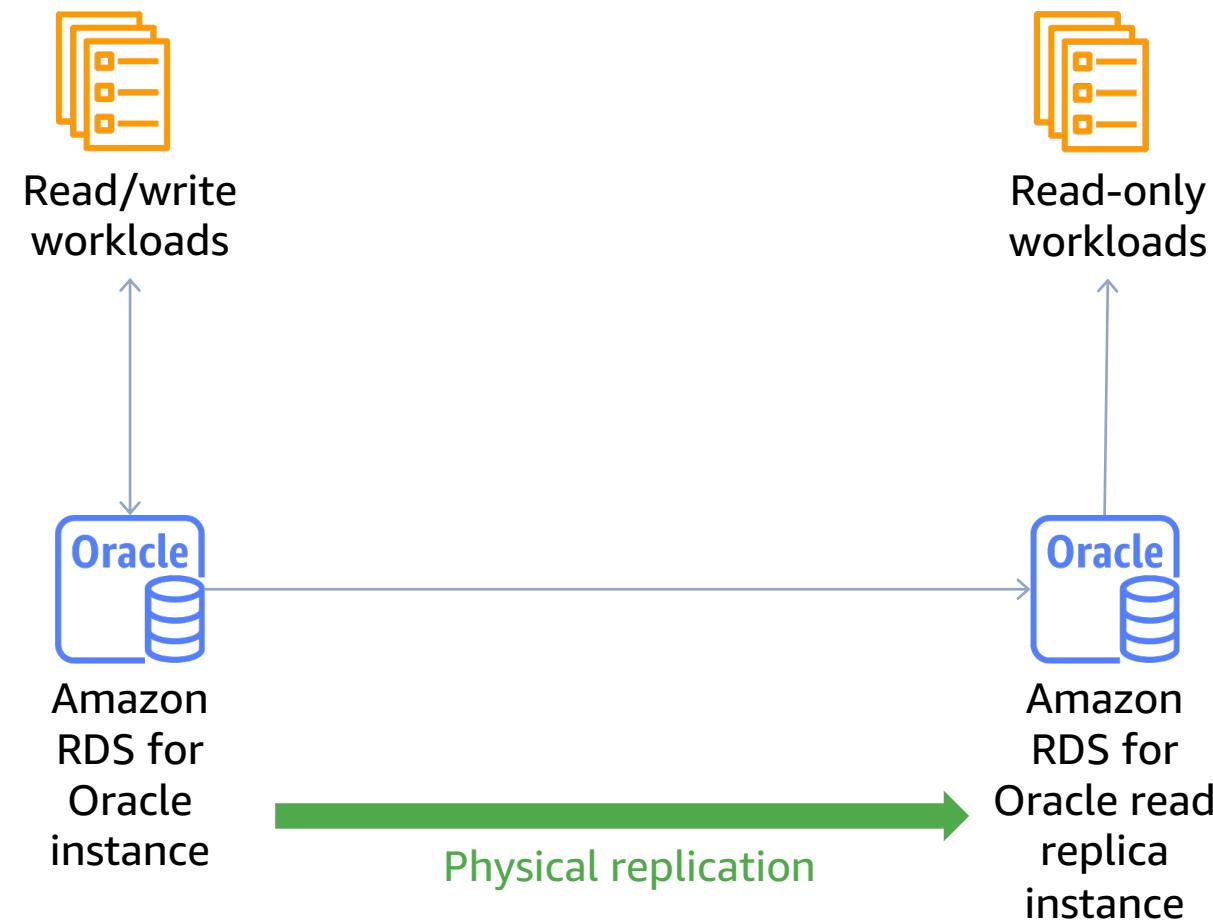
- Legacy exp utility
- Oracle SQL Developer
- Client applications



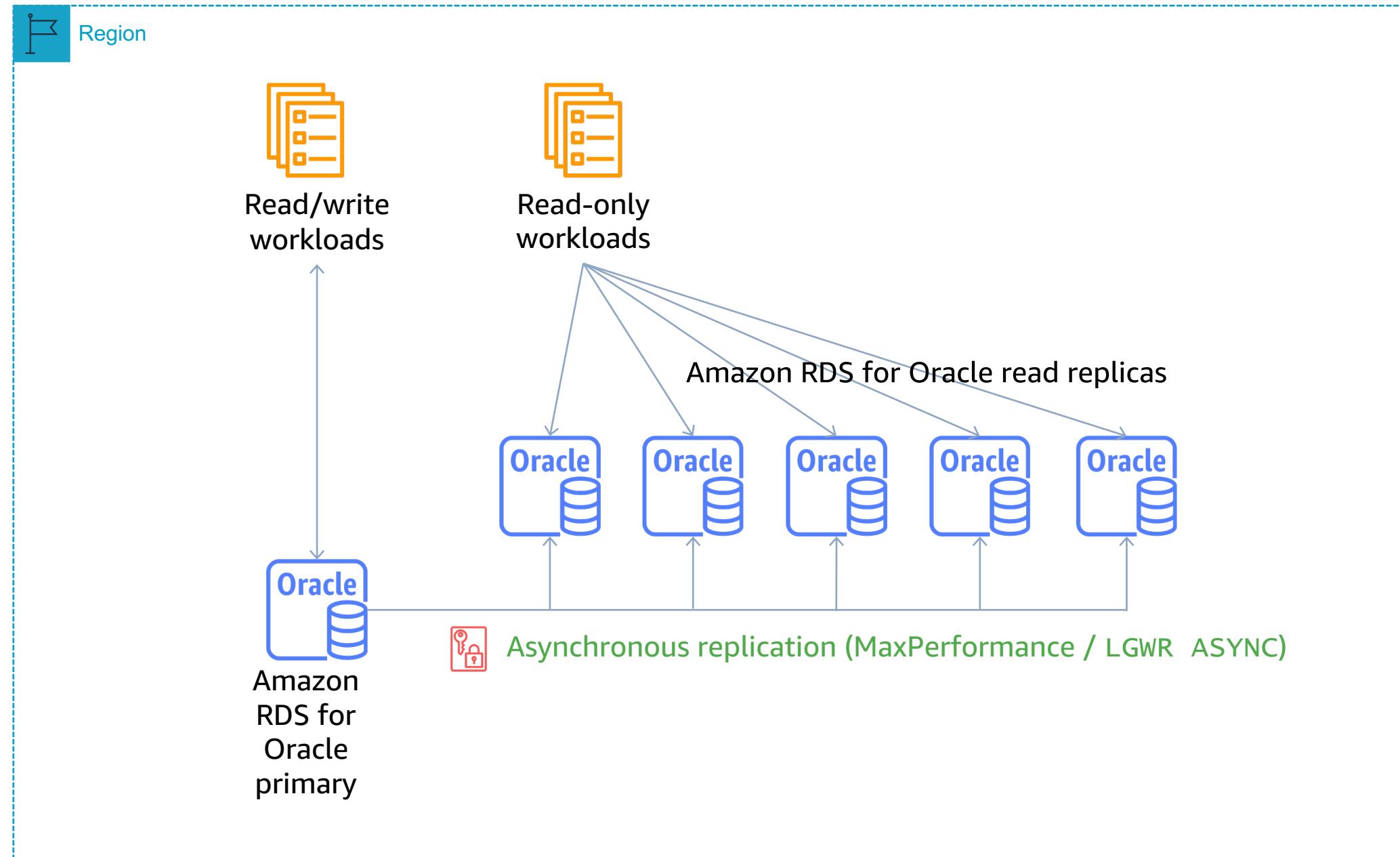
Amazon RDS for Oracle logical replication



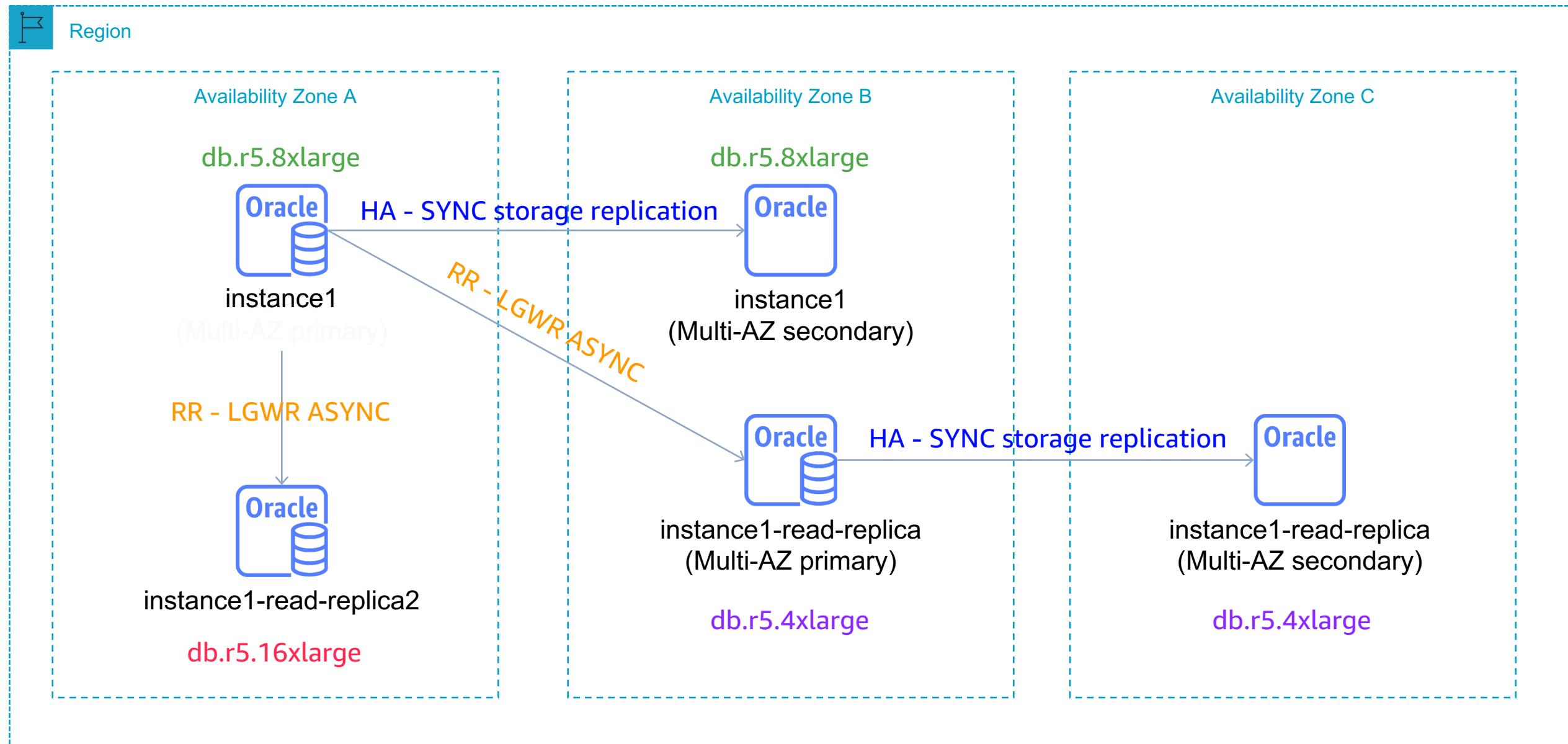
Amazon RDS for Oracle physical replication

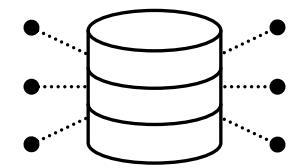


New! Amazon RDS for Oracle read replicas



Amazon RDS for Oracle read replicas





Sql Server RDS

Supported Versions

Versions

- 2012
- 2014
- 2016
- 2017

Editions

- Express
- Web
- Standard
- Enterprise



Amazon Aurora

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

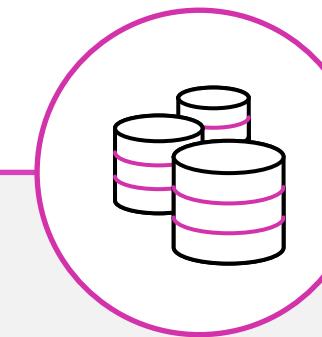


Amazon Aurora

MySQL and PostgreSQL-compatible relational database built for the cloud

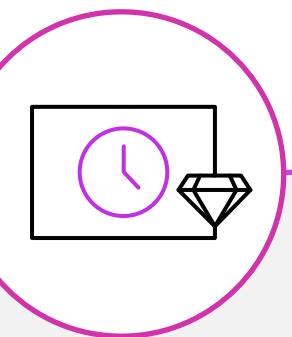
Performance and availability of commercial-grade databases at 1/10th the cost

Performance and scalability



5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas

Availability and durability



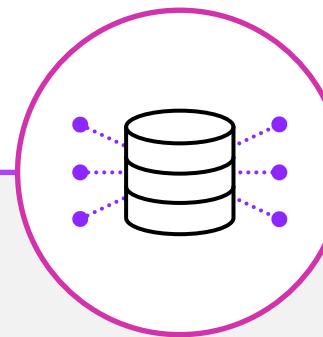
Fault-tolerant, self-healing storage; six copies of data across three Availability Zones; continuous backup to Amazon S3

Highly secure



Network isolation, encryption at rest/transit, compliance and assurance programs

Fully managed



Managed by RDS: No server provisioning, software patching, setup, configuration, or backups

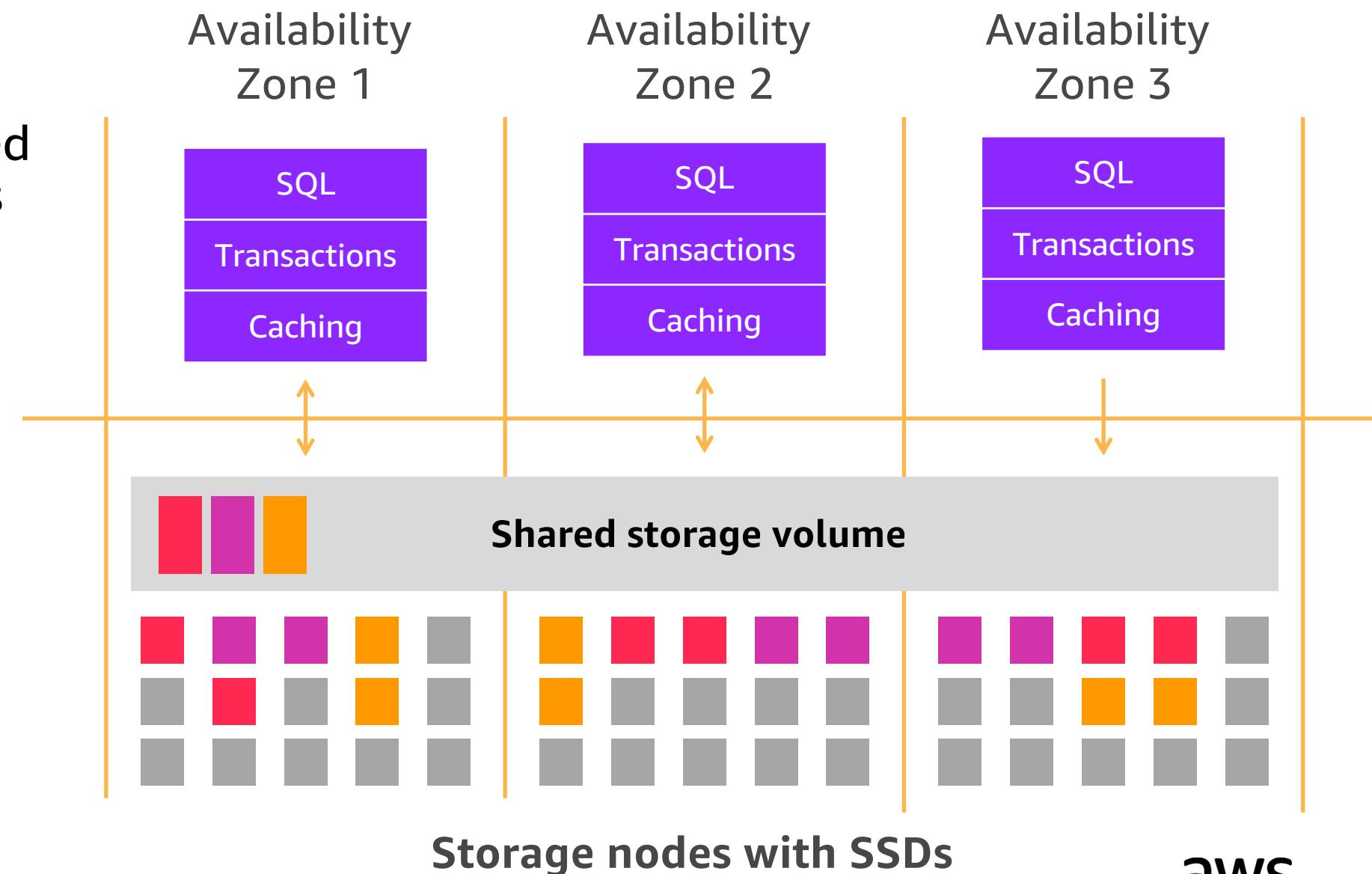
The compute and storage layers are separated in a scale-out, distributed, and multi-tenant architecture

Purpose-built log-structured distributed storage system designed for databases

Storage volume is striped across hundreds of storage nodes distributed over 3 different availability zones

Six copies of data, two copies in each availability zone to protect against AZ+1 failures

Plan to apply same principles to other layers of the stack



How Does Amazon Aurora Achieve High Performance?

DO LESS WORK

Do fewer IOs

Minimize network packets

Offload the database engine

BE MORE EFFICIENT

Process asynchronously

Reduce latency path

Use lock-free data structures

Batch operations together

DATABASES ARE ALL ABOUT IO

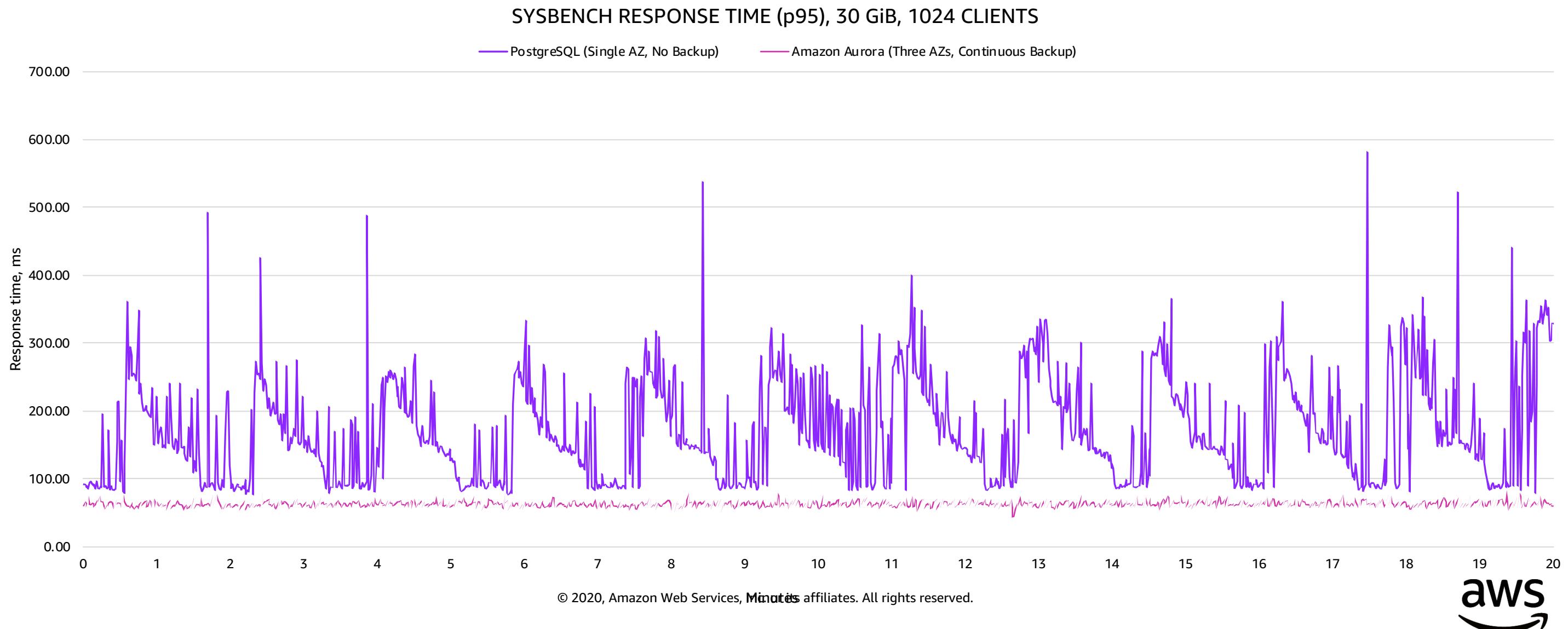
NETWORK-ATTACHED STORAGE IS ALL ABOUT PACKETS/SECOND

HIGH-THROUGHPUT PROCESSING NEEDS CPU AND MEMORY OPTIMIZATIONS

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



Amazon Aurora provides >2x Faster Response Times
Response time under heavy write load >2x faster than PostgreSQL
and variance reduced by 99%



Aurora: fastest growing service in AWS history

NASDAQ



DOW JONES



zendesk

AUTODESK



pumpkin



Blackboard



DOW JONES



FICO



PERSONAL CAPITAL

AstraZeneca



PAGELY



PeopleAdmin



NEW INNOVATIONS

SysAid



FIRST FUEL



CAL POLY





Challenge:

Wanted to move away from expensive, legacy self-managed database solutions to more efficient and cost-effective managed options

Solution:

Moved on-premises databases to Aurora PostgreSQL

Result:

Aurora PostgreSQL runs 40% faster compared to databases





Wappa is the pioneer and market leader in taxi expense management, headquartered in Brazil.

Challenge

The platform needs to find rides quickly and accelerate the budgeting, payment, and reporting processes to help customers reduce corporate travel expenses.

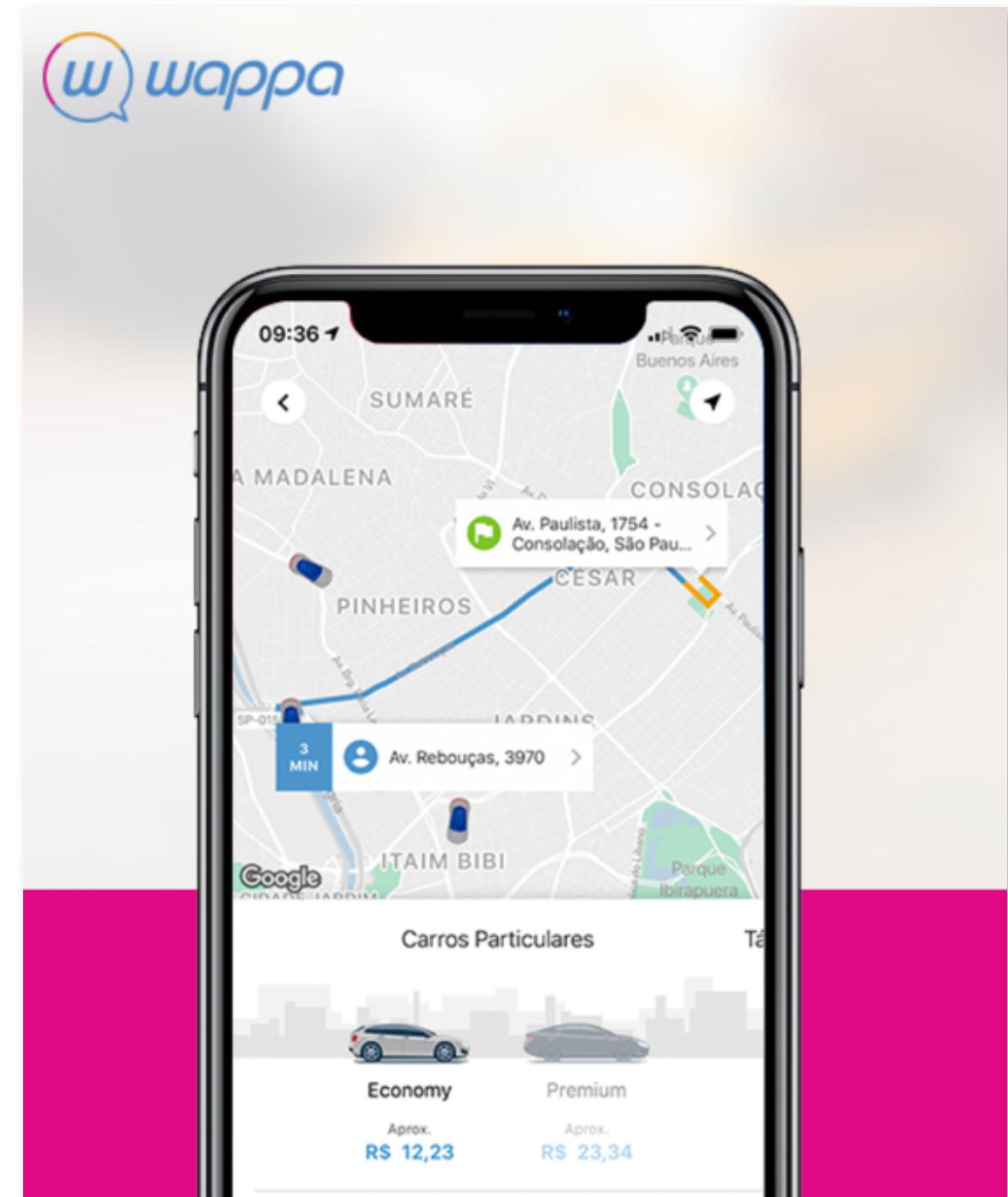
Solution

Wappa migrated their Oracle database to Amazon Aurora.

Result

“Our user validation process has become 60 percent faster, reporting time per user has dropped 75 percent, and the payment process is 70 percent faster. We’re clearly seeing the results in our user growth numbers and user satisfaction ratings of our application.”

—Cesar Matias, Chief Technology Officer, Wappa





Amazon Transaction Risk Management (TRMS) collects real-time and historical data points to detect and prevent millions of dollars in fraudulent transaction each year.

Challenge:

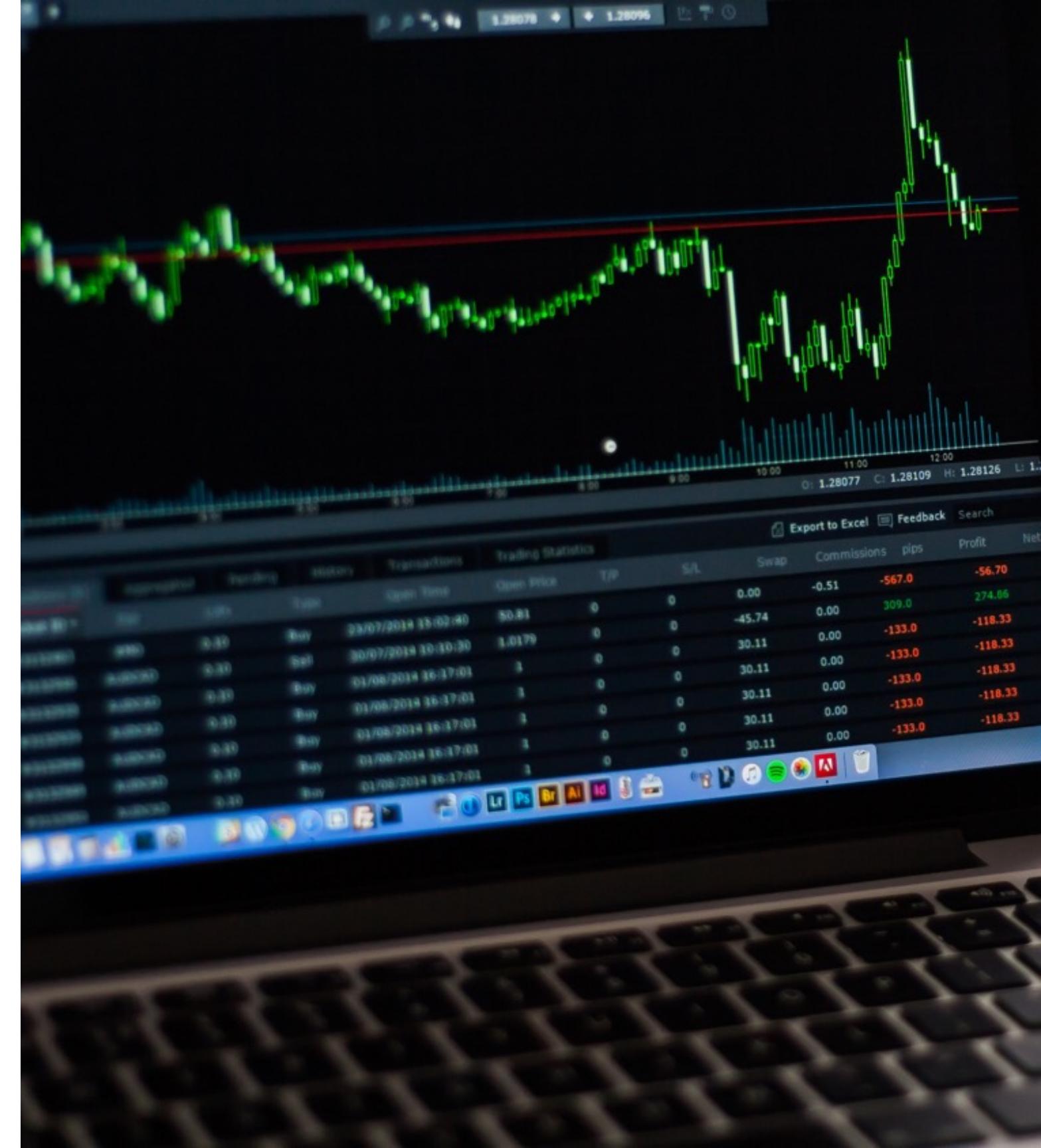
Running on Oracle posed challenges for TRMS, including complicated database administration, latency levels, and lengthy server provisioning work.

Solution:

Migrated 100+ on-premises Oracle databases to **Amazon Aurora** with **AWS Database Migration Service (DMS)**.

Result:

- Efficiency: Reduced administrative overhead by 70%.
- Cost: cut cost by half.
- Performance: scaled up to 900 transactions per second per shard now.

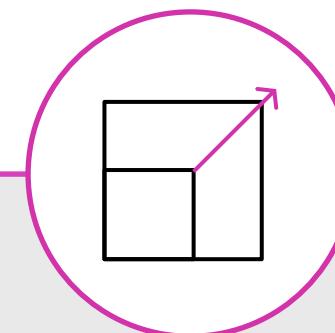




Amazon DynamoDB

Fast and flexible key value database service for any scale

Performance at scale



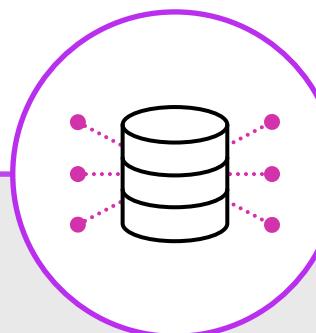
Consistent, single-digit millisecond response times at any scale; build applications with virtually unlimited throughput

Serverless



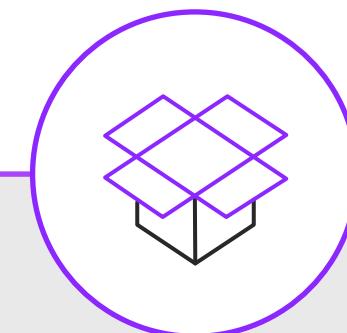
No hardware provisioning, software patching, or upgrades; scales up or down automatically; continuously backs up your data

Comprehensive security



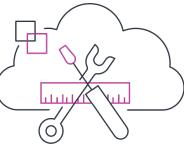
Encrypts all data by default and fully integrates with AWS Identity and Access Management for robust security

Global database for global users and apps



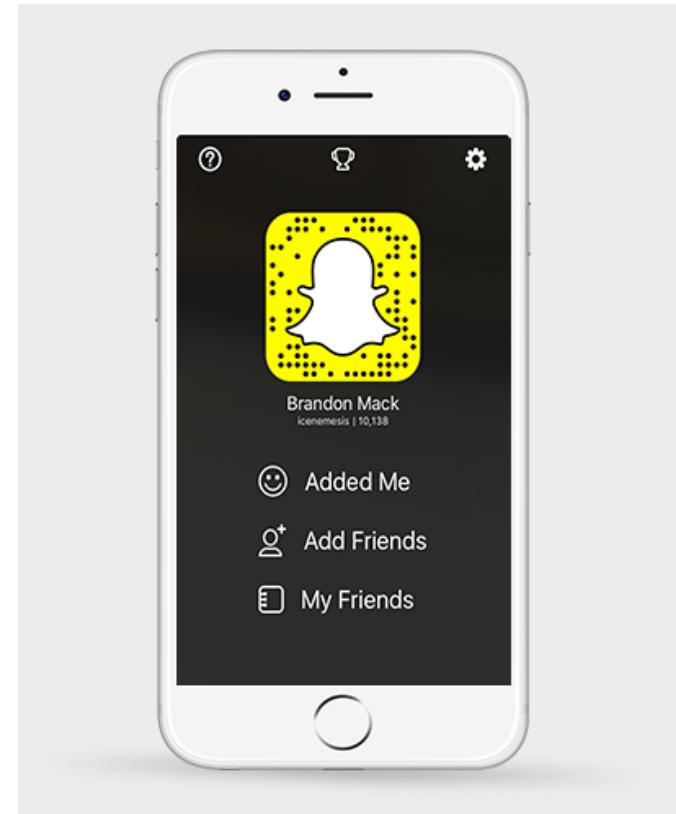
Build global applications with fast access to local data by easily replicating tables across multiple AWS Regions

DynamoDB powers the world's largest applications



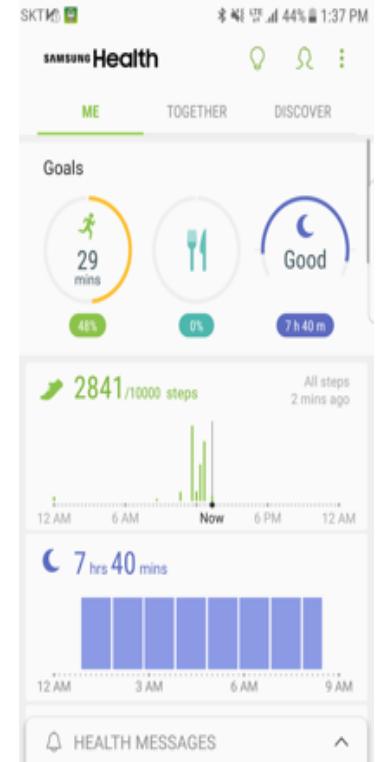
The screenshot shows the Amazon.com homepage. At the top, there is a search bar and a navigation menu with categories like "Shop All Departments", "Unlimited Instant Videos", "Books", "Movies, Music & Games", "Digital Downloads", "Kindle", "Computers & Office", "Electronics", "Home, Garden & Pets", "Grocery, Health & Beauty", "Toys, Kids & Baby", "Clothing, Shoes & Jewelry", "Sports & Outdoors", "Tools & Home Improvement", and "Automotive & Industrial". Below the menu, a large banner for the Kindle device is displayed, stating "Kindle The #1 Bestselling Product on Amazon". It includes a price of \$139 Wi-Fi and \$189 Free 3G+Wi-Fi, along with links to "Shop Kindle Books", "Download Free Reading Apps", and "Shop Accessories". A section titled "What Other Customers Are Looking At Right Now" shows products like "Lost: The Complete Collection [Blu-ray]", "Wild Sight" by Loucinda McGary, and various Kindle devices. The footer contains a "Use Watch Device" link.

>20M requests per second



>150M active users

SAMSUNG



Data >1 PB



Lyft

>1M rides/day,
8x traffic in peak hours

CHALLENGE

Needed a solution that scales and manages up to 8x more riders during peak times.

SOLUTION

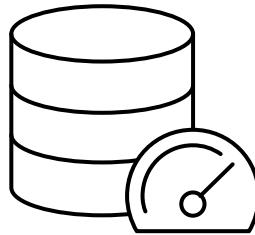
DynamoDB stores GPS coordinates of all rides.

With AWS, Lyft saves on infrastructure costs and enables massive growth of ridesharing platform. There are now 23M people who use Lyft worldwide.



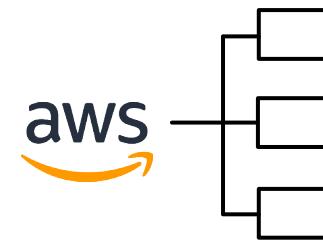
Amazon ElastiCache

Fully-managed, Redis or Memcached compatible, low-latency, in-memory data store



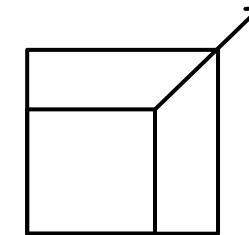
Extreme Performance

In-memory data store and cache for sub-millisecond response times



Fully Managed

AWS manages all hardware and software setup, configuration, monitoring



Easily Scalable

Read scaling with replicas. Write and memory scaling with sharding. Non disruptive scaling

ElastiCache Redis

#1 Key-Value Store*

Fast in-memory data store in the cloud. Use as a database, cache, message broker, queue

Highly Available & Reliable

Read replicas, multiple primaries, multi-AZ with automatic failover

Fully Managed & Hardened

AWS manages hardware, software, setup, configuration, monitoring, failure recovery, and backups

Easily Scalable

Cluster with up to 6.1 TiB of in-memory data

Read scaling with replicas

Write and memory scaling with sharding

Scale out or in

Secure & Compliant

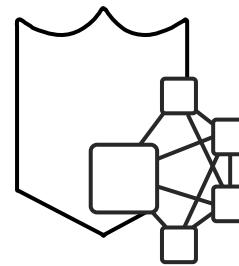
VPC for cluster isolation, encryption at rest/transit, HIPAA compliance

ElastiCache Memcached



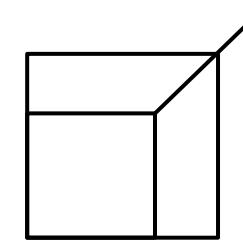
Fully Managed Memcached

Fast in-memory data store in the cloud. Use as a cache to reduce latency and improve throughput



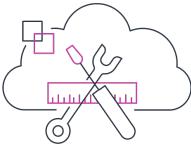
Secure & Hardened

VPC for cluster isolation



Easily Scalable

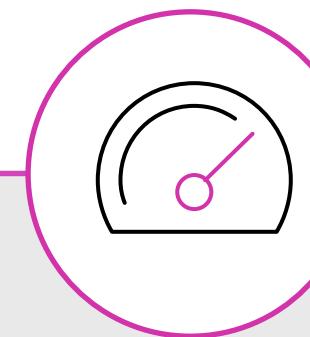
Sharding to scale in-memory cache with up to 20 nodes and 8.14 TiB per cluster



Amazon Timestream

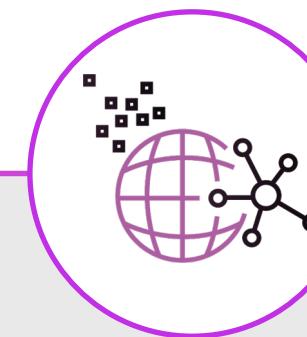
Fast, scalable, fully managed time series database

1,000x faster and 1/10th the cost of relational databases



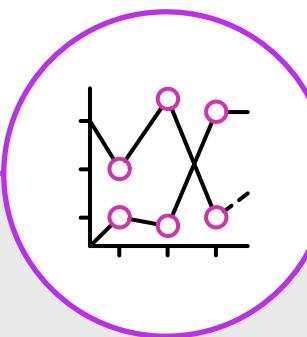
Collect data at the rate of millions of inserts per second (10M/second)

Trillions of daily events



Adaptive query processing engine maintains steady, predictable performance

Analytics optimized for time series data



Built-in functions for interpolation, smoothing, and approximation

Serverless

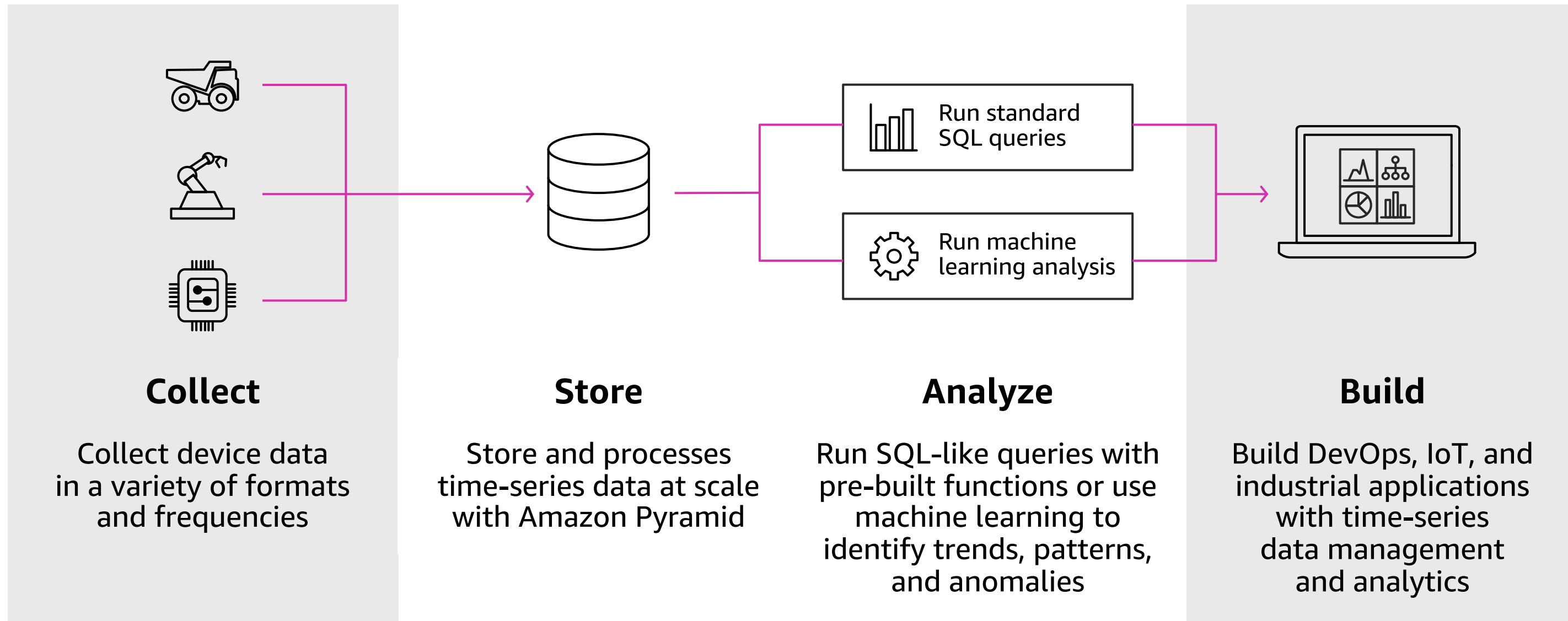


Automated setup, configuration, server provisioning, software patching



Amazon Timestream: How it works

Time-series application



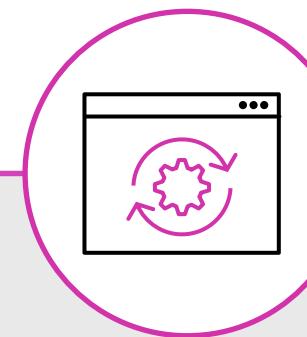


Amazon Quantum Ledger Database

Fully managed ledger database

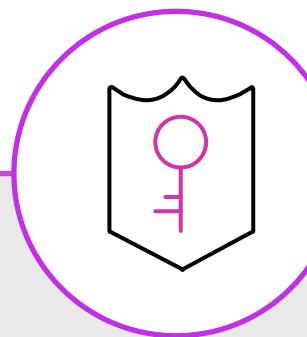
Track and verify history of all changes made to your application's data

Immutable



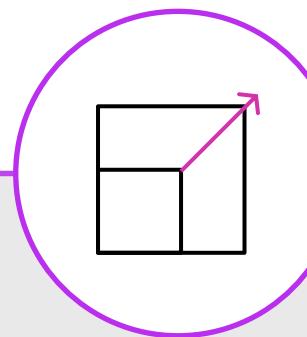
Maintains a sequenced record of all changes to your data, which cannot be deleted or modified; you have the ability to query and analyze the full history

Cryptographically verifiable



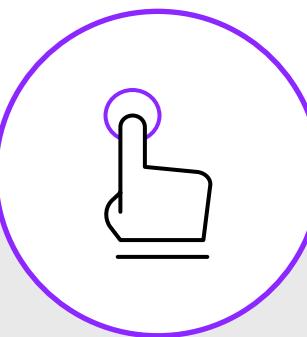
Uses cryptography to generate a secure output file of your data's history

Highly scalable



Executes 2–3x as many transactions than ledgers in common blockchain frameworks

Easy to use

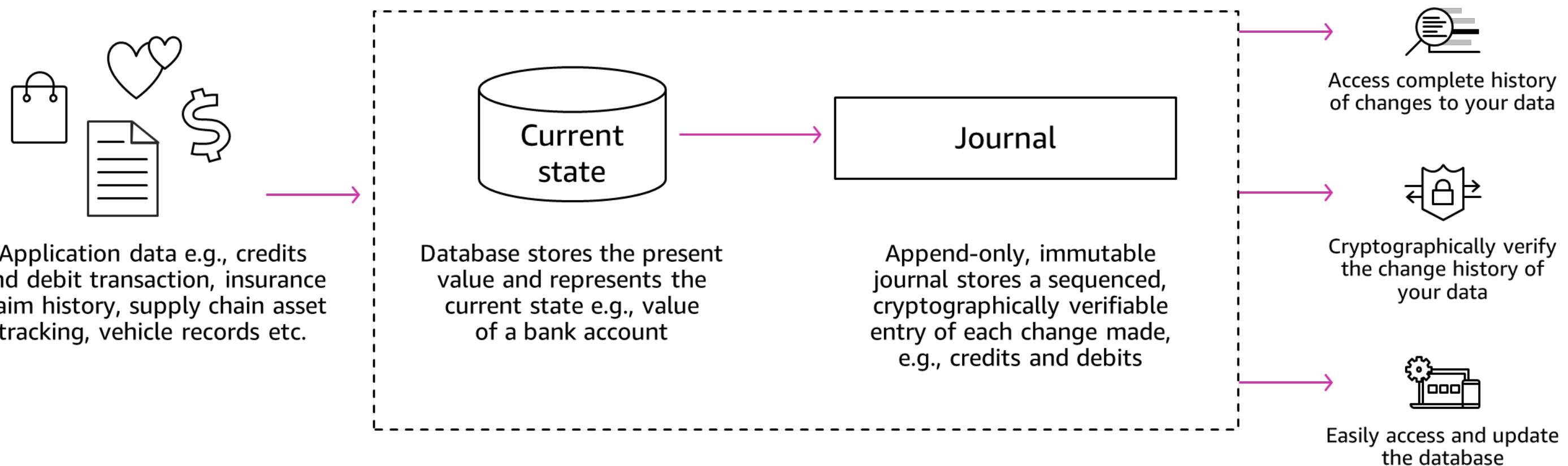


Easy to use, letting you use familiar database capabilities like SQL APIs for querying the data



Amazon QLDB: How it works

Centralized ledgers



Three type of projects



Quickly build new
apps in the cloud

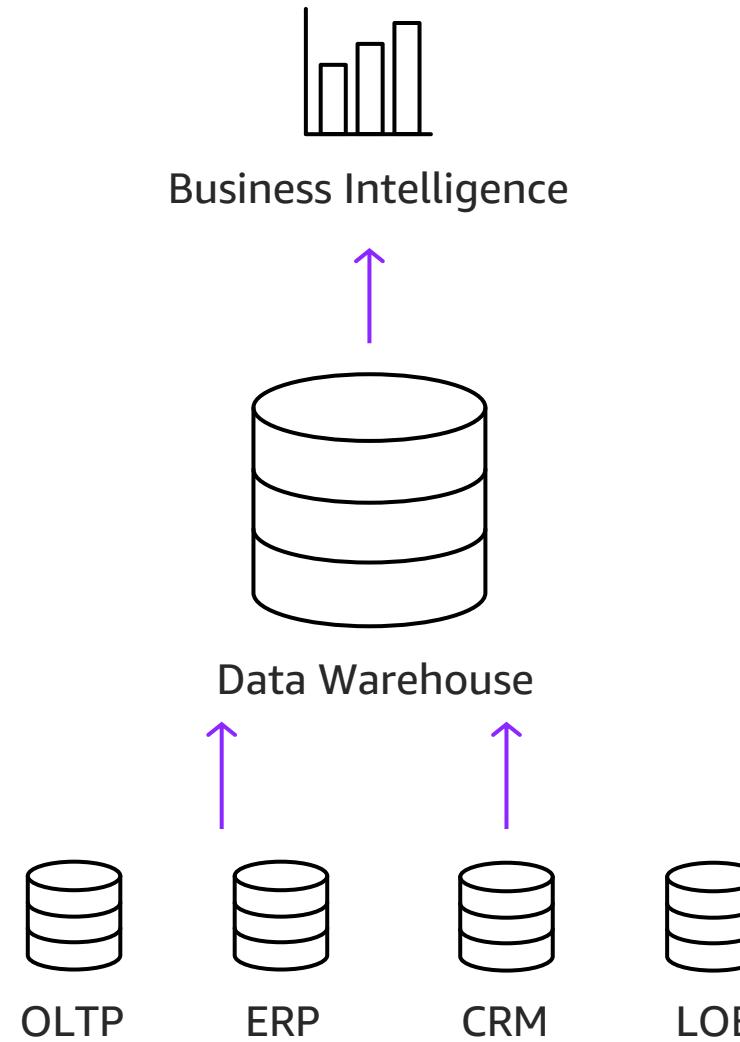


“Lift and shift” existing
apps to the cloud



Gain new
insights

Traditionally, analytics looked like this



Relational data

GBs-TBs scale [not designed for PB/EBs]

Expensive: Large initial capex + \$10K-\$50K/TB/year

90% of data was thrown away because of cost

Three key trends are transforming the world of Analytics

1

There is more data than traditional data platforms can hold

2

There are more ways to analyze data than ever before

3

Democratization is much more common: there are more people working with data than ever before

Analytics Use Cases across Enterprises are Growing

Data Warehousing



Big Data Processing



Interactive Query



Operational Analytics



Data Exchange



Visualizations



Real-time Analytics



Recommendations



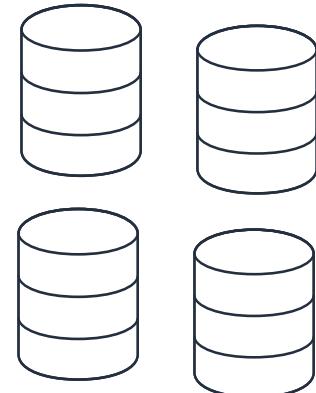
Predictive Analytics



Data infrastructures needs new Architecture to meet new demands



On-premises data infrastructures **do not scale** to meet variable and increasing volumes of data



Multiple disconnected data silos with inconsistent formats obscure data lineage and prevent a consolidated view of activity

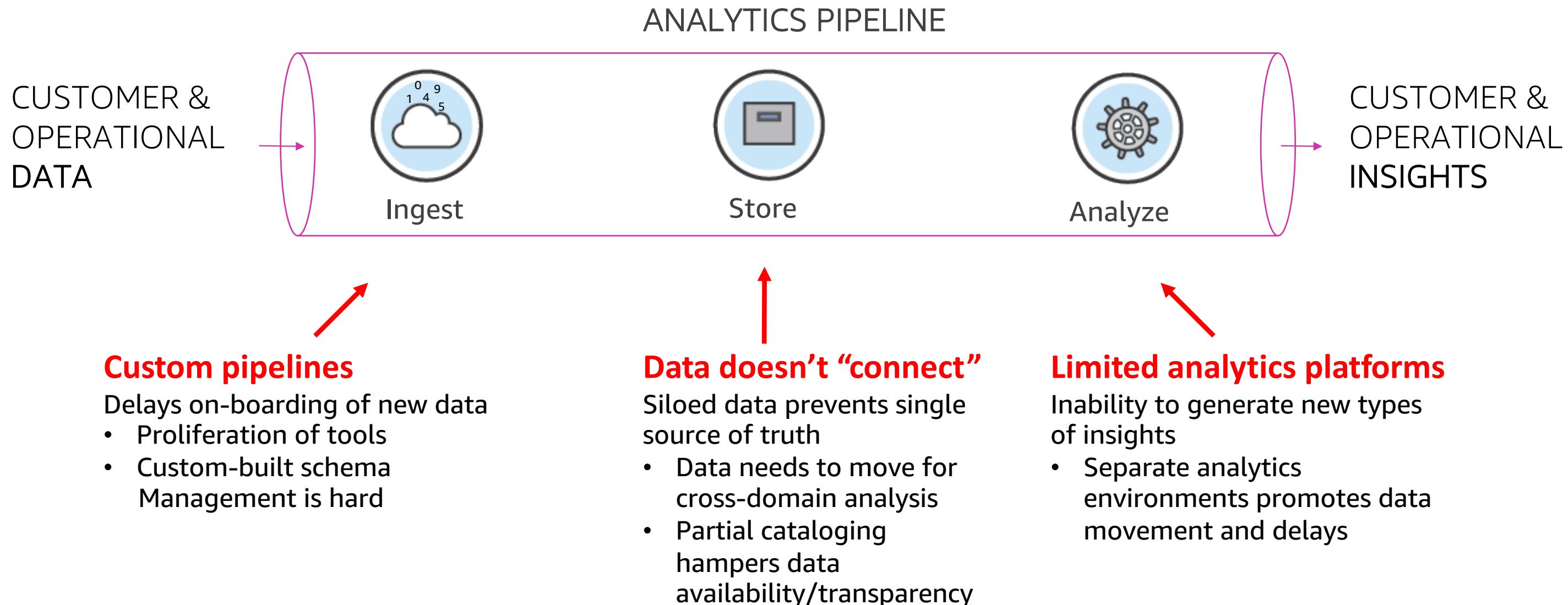


Rigid data schemas prevent access to source data and limit the use of advanced analytics and machine learning

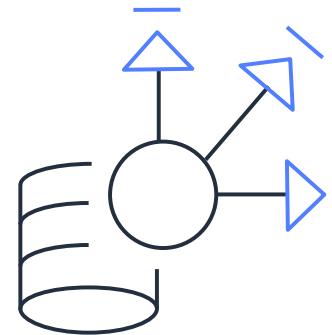


The **high costs of legacy data warehouses** limit access to historical data

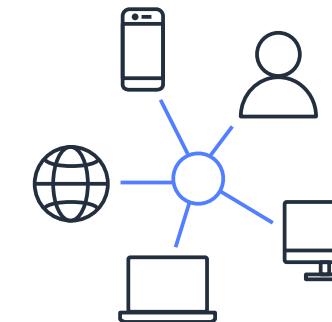
On-premise Big Data still challenges most enterprises



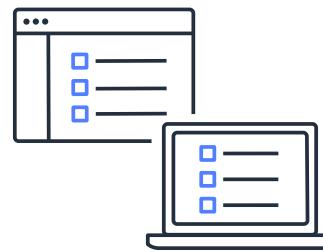
Data lakes can accelerate Data Analytics



Centralized repository
that allows structured and
unstructured data to be
stored at any scale



Used for all use cases including
machine learning, **real-time streaming analytics**, data discovery,
and **business intelligence**



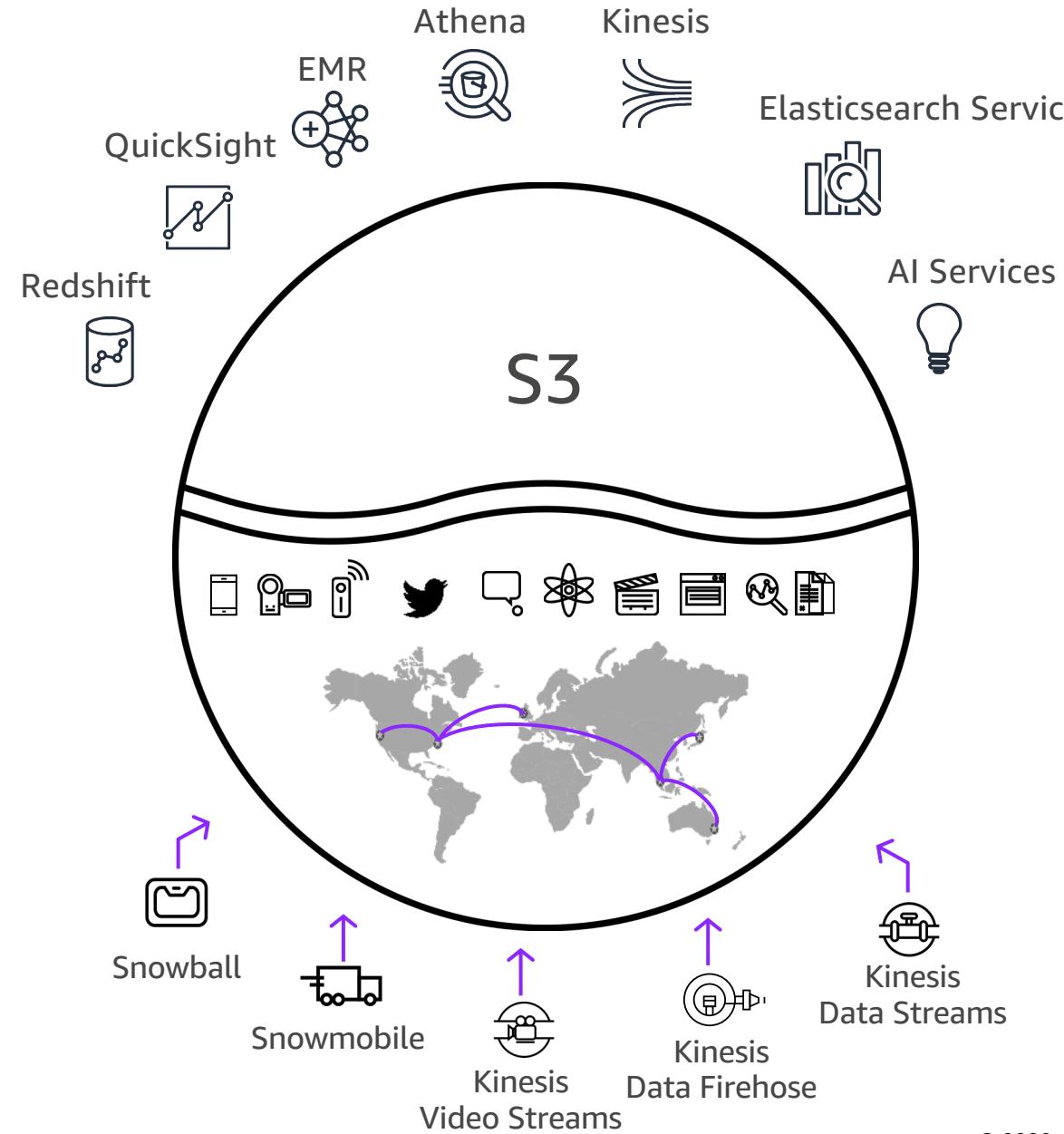
Data is stored as-is,
without having to first
structure the data



Access to historical data **within seconds** without the cost of
managing infrastructure



Data lakes on AWS



Exabyte scale

Store and analyze relational and non-relational data

Purpose-built analytics tools

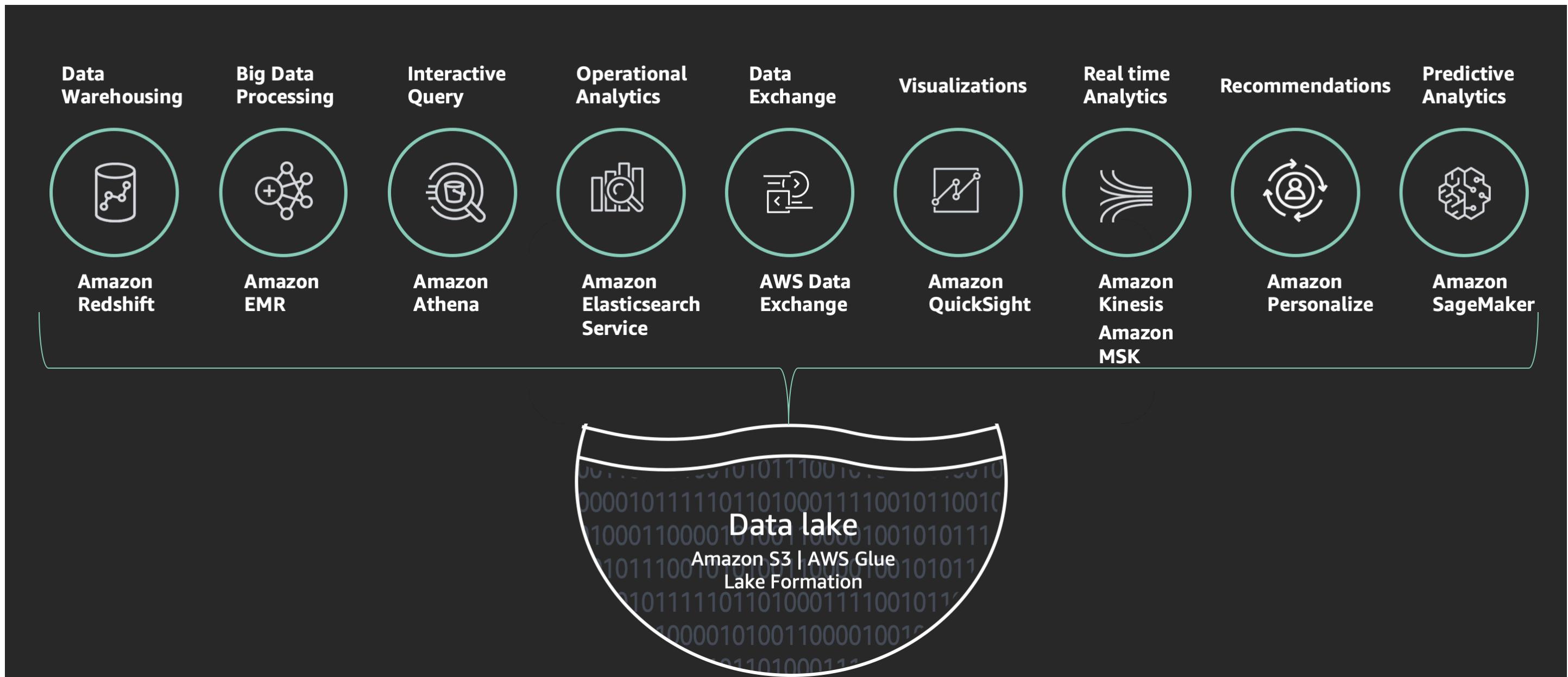
Cost effective

- Store at 2.3 cents per GB-month in Amazon S3
- Query with Amazon Athena at $\frac{1}{2}$ cent per GB scanned
- DW with Amazon Redshift for \$1,000/TB/year

Give access to everyone

- Amazon QuickSight: \$0.30 for 30 minutes of use

Purpose-built Analytics Services on Data in the Lake



Building Serverless Data Lake on AWS

Data Lake storage : S3



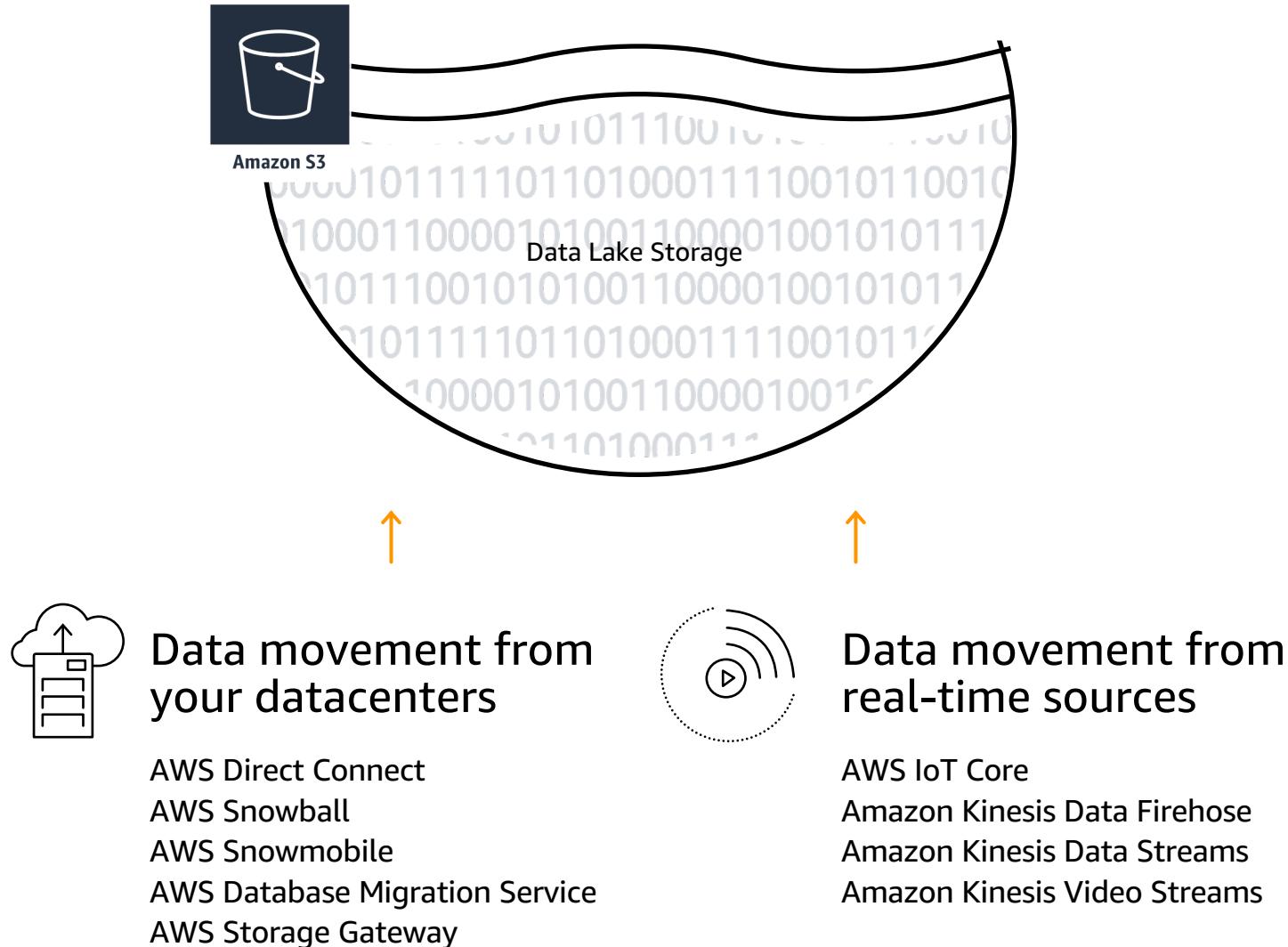
Secure, highly scalable, durable object storage with millisecond latency for data access

Store any type of data—web sites, mobile apps, corporate applications, and IoT sensors, at any scale

Store data in the format you want:
Unstructured (logs, dump files) | semi-structured (JSON, XML) |
structured (CSV, Parquet)

Storage lifecycle integration
Amazon S3-Standard | Amazon S3-Infrequent Access |
Amazon Glacier

Most ways to move data into the Data Lake



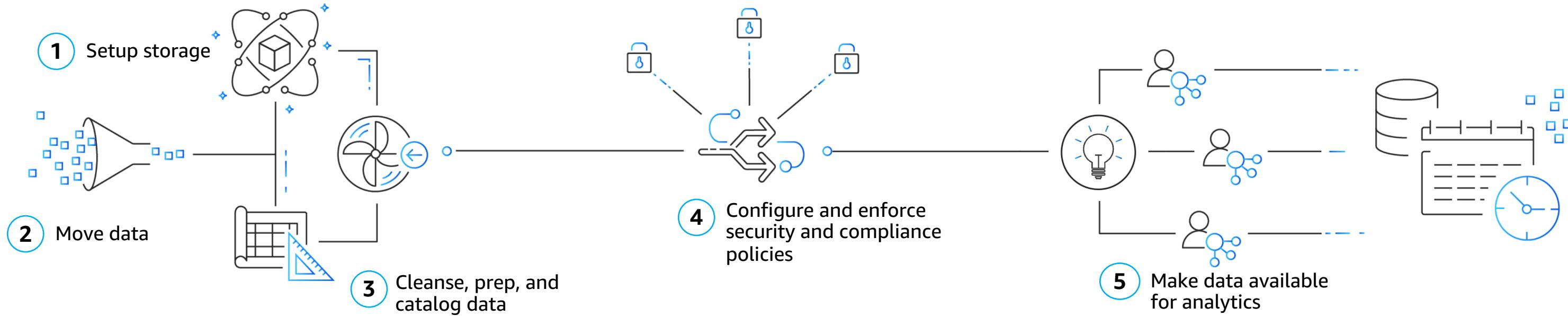
Data movement from on-premises datacenters

- Dedicated Network connection
- Secure appliances
- Ruggedized Shipping Container
- Database migration
- Gateway that lets applications write to the cloud

Data movement from real-time sources

- Connect devices to AWS
- Real-time data streams
- Real-time video streams

Typical steps of building a data lake



Sample of steps required

Configure access from analytics services



Ama

Dash

Insta

Cluste

Query

Perfor

Snaps

Auton

Reserv

Subne

Param

Optio

Event

Event

Recon

<https://con>

Rinse and repeat for other:
data sets, users, and end-services

And more:

- manage and monitor ETL jobs
- update metadata catalog as data changes
- update policies across services as users and permissions change
- manually maintain cleansing scripts
- create audit processes for compliance

...

Manual | Error-prone | Time consuming

[Feedback](#) [English \(US\)](#)

[Documentation](#) [Policy generator](#)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

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



Building data lakes can still take **months**

Build Data Lake in Days using Lake Formation

AWS Lake Formation

Build a secure data lake in days



Identify, ingest, clean, and transform data

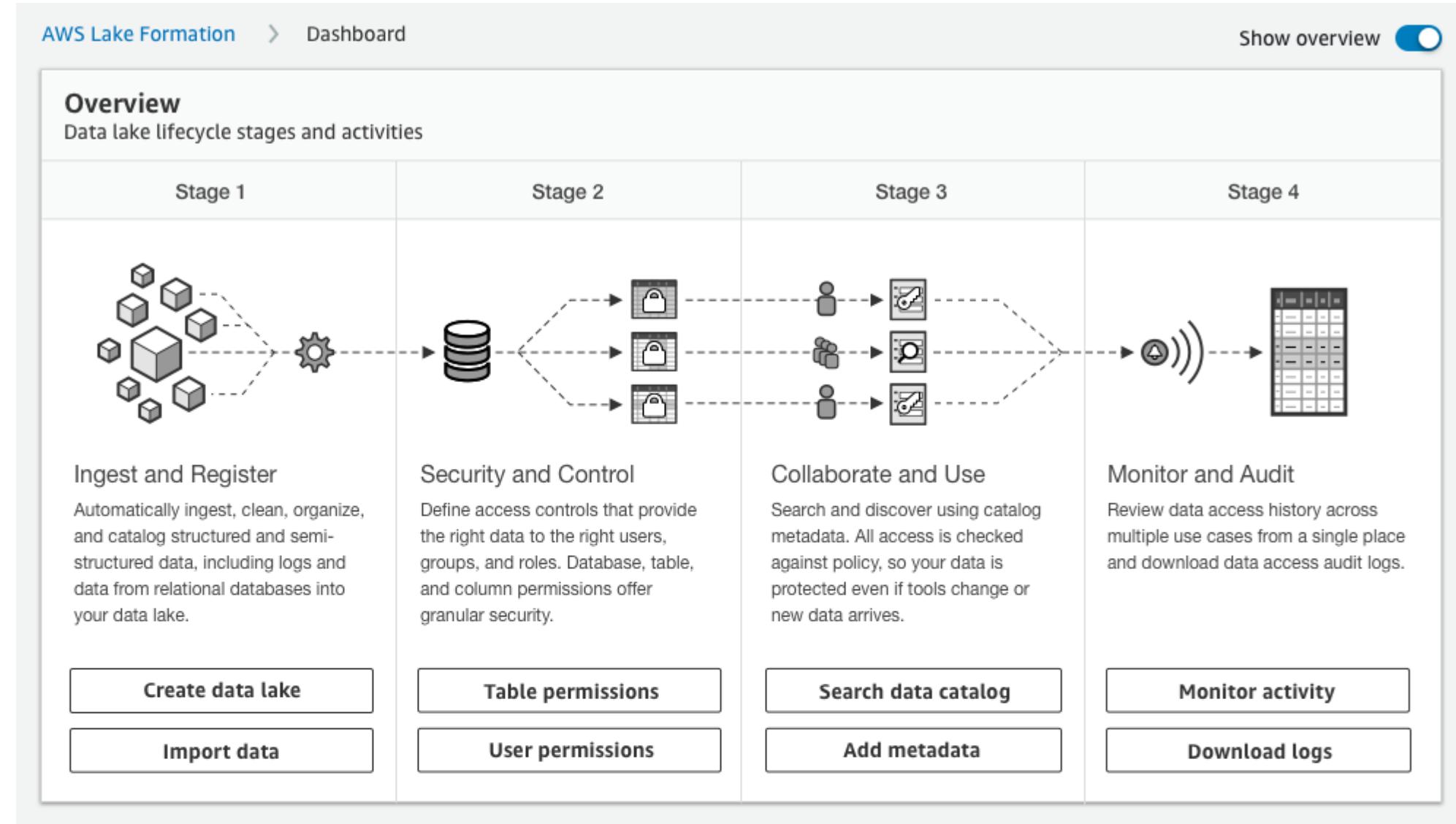


Enforce security policies across multiple services



Gain and manage new insights

All Steps to Build Data Lake streamlined



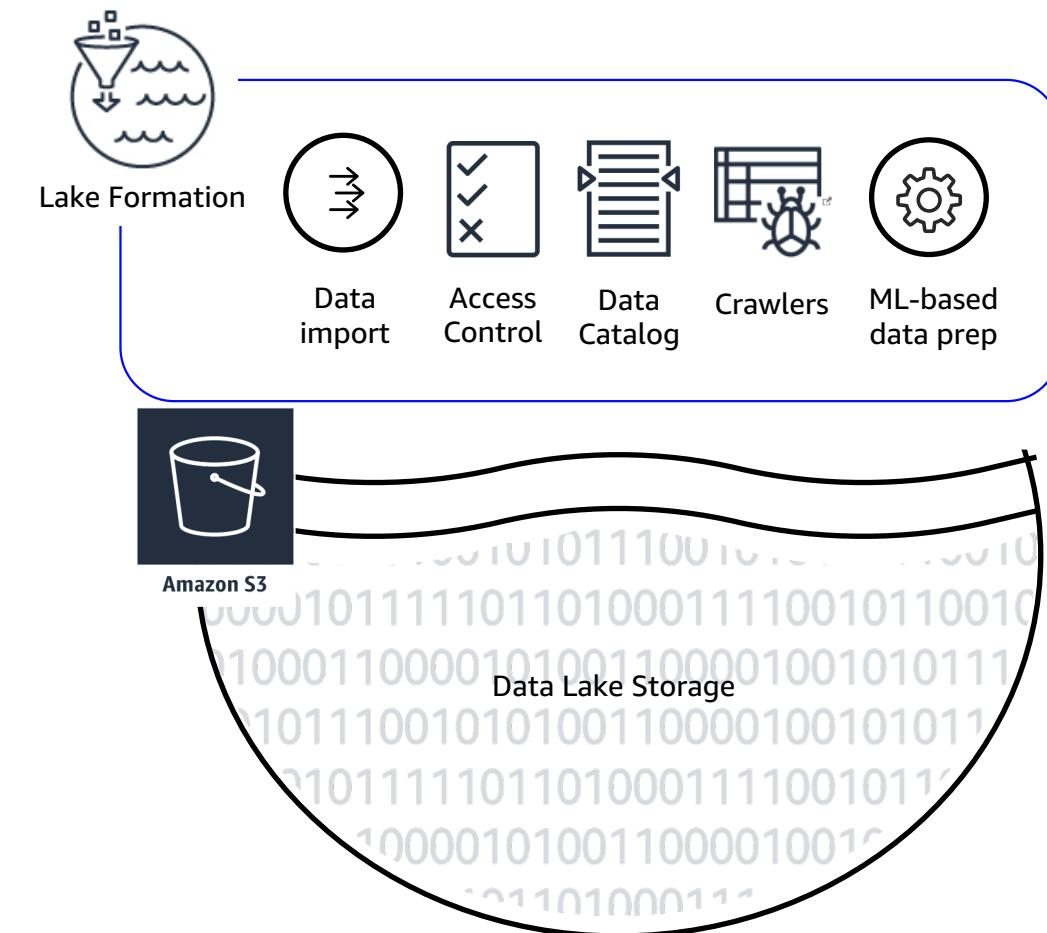
Register existing data or import new

Amazon S3 forms the storage layer for Lake Formation

Register existing S3 buckets that contain your data

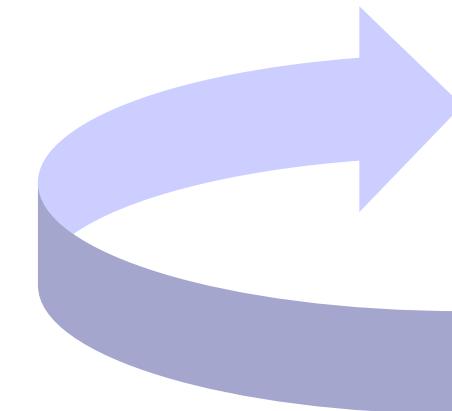
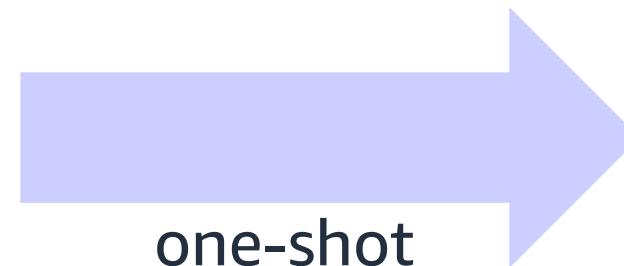
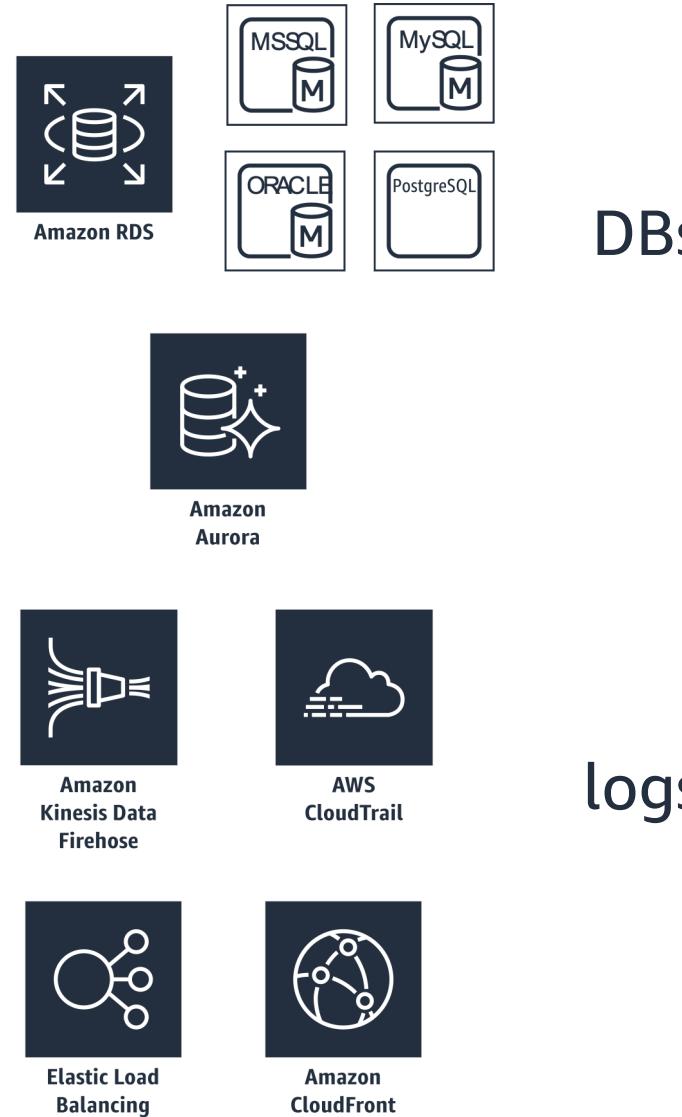
Ask Lake Formation to create required S3 buckets and import data into them

Data is stored in your account. You have direct access to it. No lock-in.

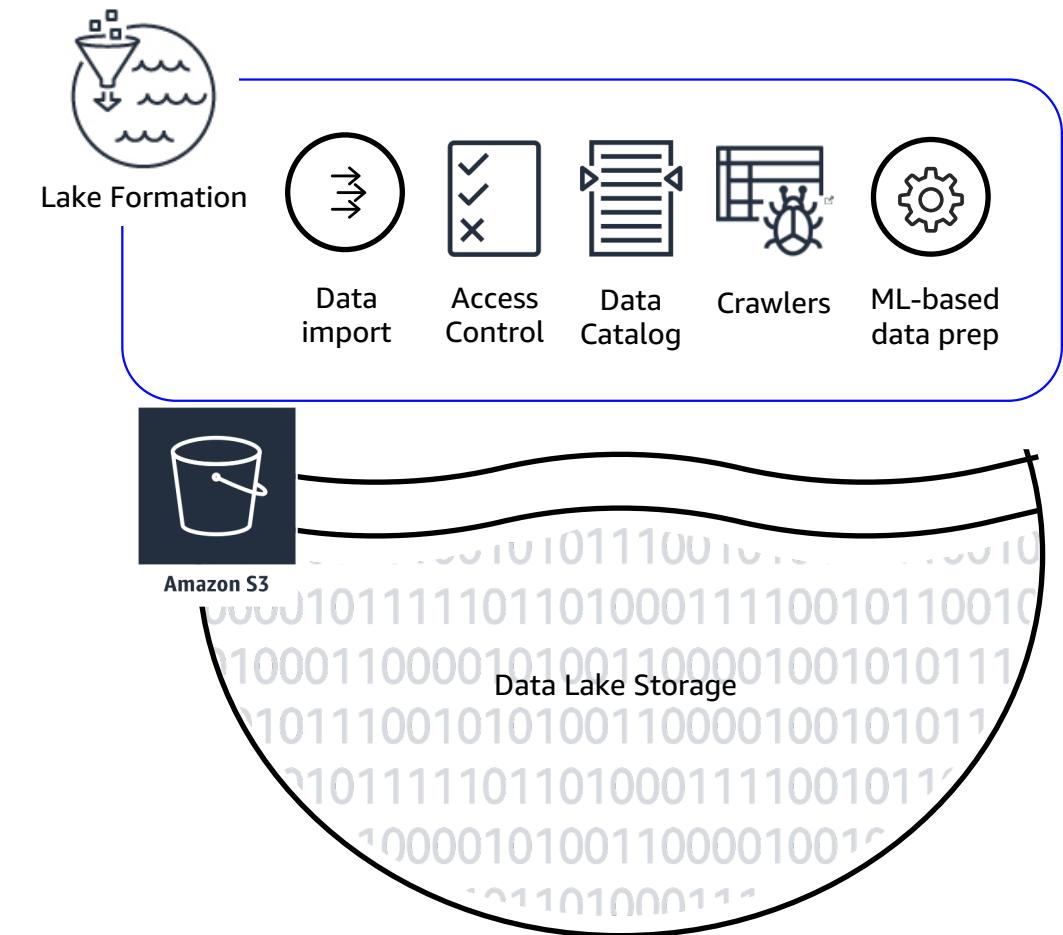


Easily load data to your data lake

Blueprints



incremental



With blueprints

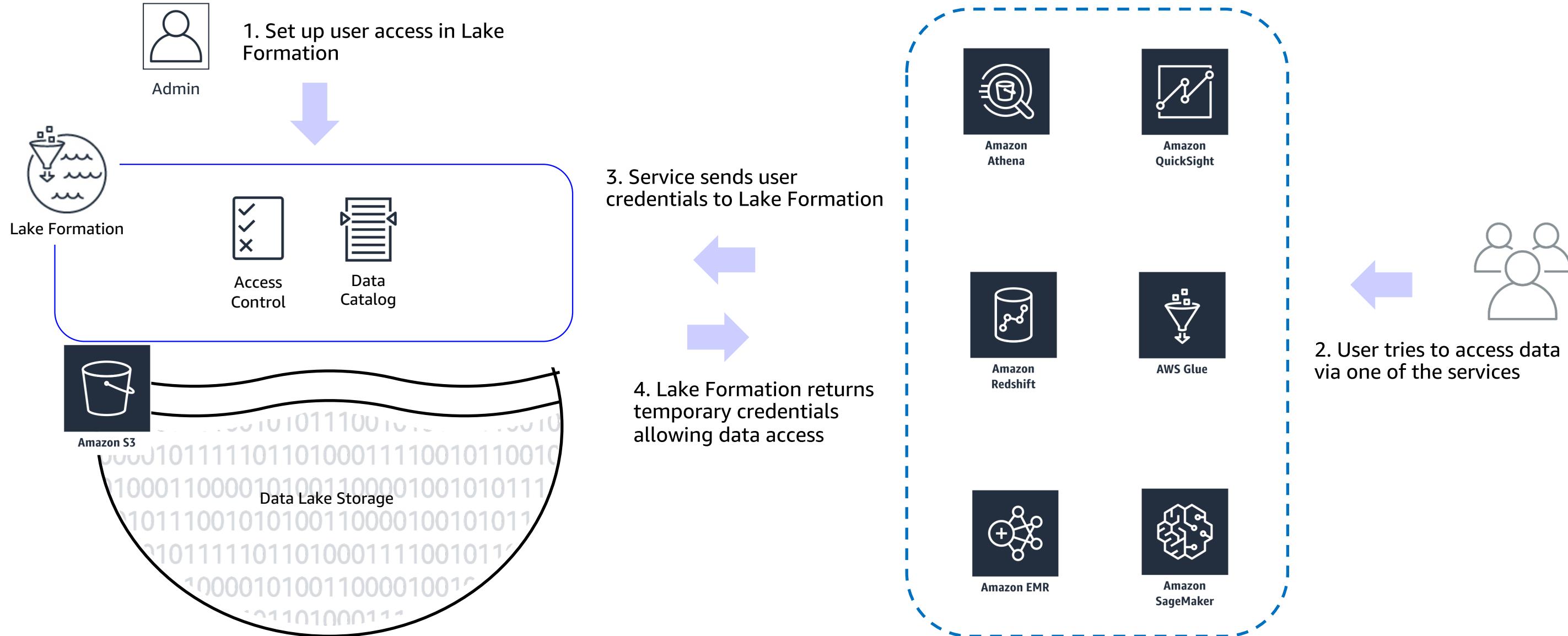
You

1. Point us to the source
2. Tell us the location to load to in your data lake
3. Specify how often you want to load the data

Blueprints

1. Discover the source table(s) schema
2. Automatically convert to the target data format
3. Automatically partition the data based on the partitioning schema
4. Keep track of data that was already processed
5. You can customize any of the above

Secure once, access in multiple ways



Security permissions in Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on tables and columns rather than on buckets and objects

Easily view policies granted to a particular user

Audit all data access at one place

The image shows two screenshots of the AWS Lake Formation interface. The top screenshot is titled 'AWS Lake Formation > Tables' and displays a list of two tables: 'reviews' and 'orders'. The 'reviews' table is selected, showing its database ('sales') and location ('S3://datalake/sales/reviews/'). The bottom screenshot is a modal window titled 'Grant permissions to table orders' with the sub-instruction 'Grant access permissions to specific users, groups, and roles for the selected table.' It contains sections for 'IAM user, group, and roles' (with 'johnnd', 'salesgrp', and 'analyst' listed), 'Permissions' (with 'Grant all' and 'Specific permissions' options selected, and various permission checkboxes like 'Select', 'Create', etc.), and 'Columns – optional' (with a search bar for column names). A 'Save' button is at the bottom right.

Security permissions in Lake Formation

Search and view permissions granted to a user, role, or group in one place

Verify permissions granted to a user

Easily revoke policies for a user

The screenshot illustrates the AWS Lake Formation User permissions interface, demonstrating its features for managing security permissions.

User permissions (3)

Name	Type	Permissions	Last modified
johnd	User	3 permissions	11/28/2018 14:24
<input checked="" type="checkbox"/> sales	Database	Create, Select, Insert	11/28/2018 13:37
<input type="checkbox"/> reviews	Table	Administrator	11/28/2018 13:37
<input type="checkbox"/> orders	Table	Create, Drop	11/28/2018 12:44
<input type="checkbox"/> salesgrp			
<input type="checkbox"/> sales			
<input type="checkbox"/> analyst			
<input type="checkbox"/> reviews			

Verify permissions for database sales

IAM user, group, role(s) [Info](#)
Choose one or more IAM users, groups, and roles to verify access permission
 johnd salesgrp

Permissions for selected users (2)

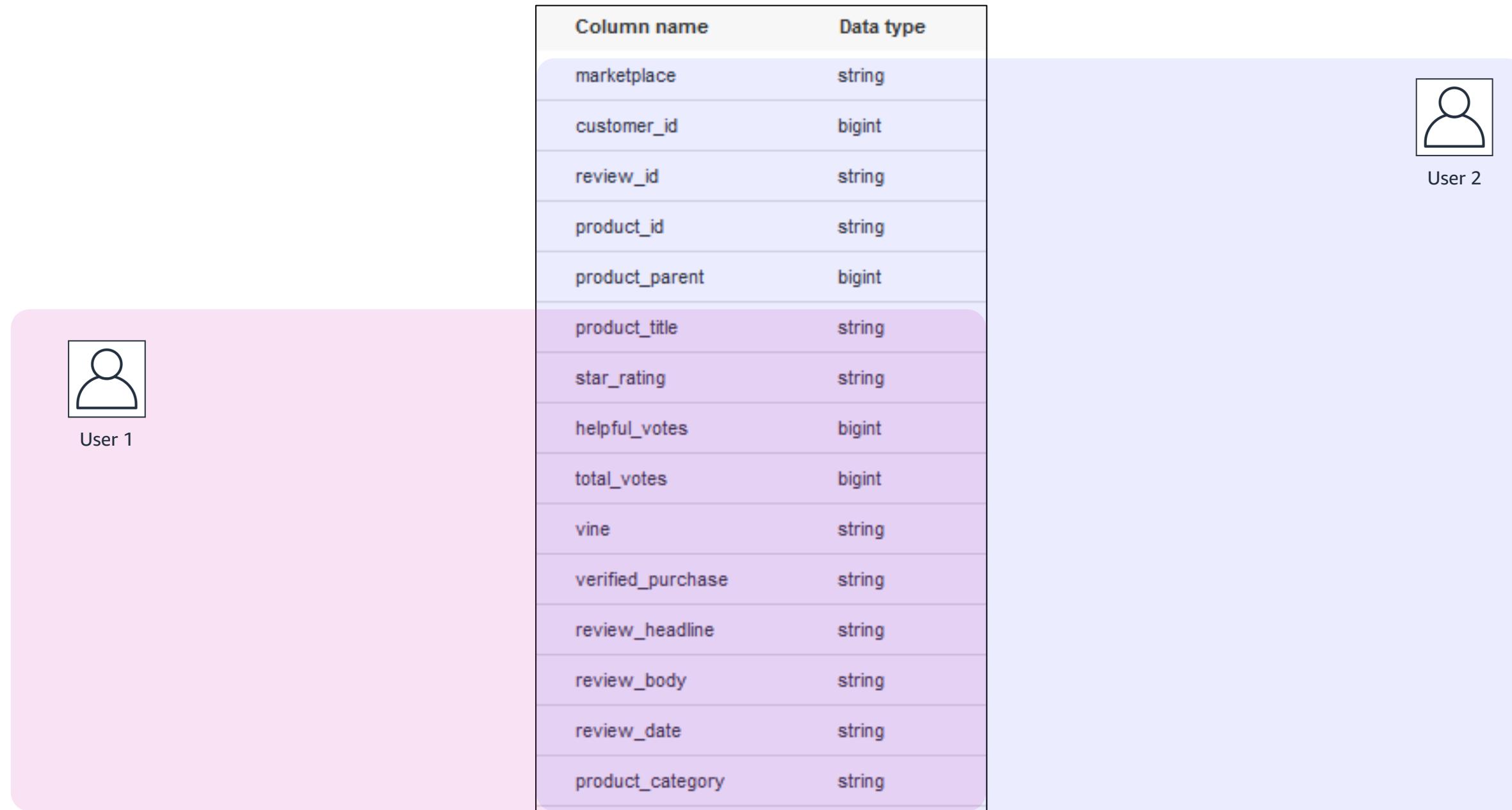
Name	Type	Permissions
<input checked="" type="checkbox"/> johnd	User	Create, Select, Insert
<input type="checkbox"/> salesgrp	Group	Administrator

Revoke all permissions for johnd

To confirm, type `revoke all` into the field.

Buttons: Grant, Revoke, Refresh, Cancel, Revoke

Grant table and column-level permissions



Search and collaborate across multiple users

Text-based, faceted search
across all metadata

Add attributes like Data
owners, stewards, and other as
table properties

Add data sensitivity level,
column definitions, and others
as column properties

Text-based search and filtering

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

The screenshot shows the AWS Glue Data Catalog interface. At the top, there's a navigation bar with 'Add tables', 'Action', a search bar ('search : log'), 'Filter or search for tables...', 'Save view', and some status indicators ('Showing: 1 - 3'). Below the navigation is a table list. One row is selected, showing 'clouptraildata' as the name, with a checked checkbox. To the right of the table name are columns for 'Location' (s3://.../CloudTrailData/) and 'mys3'. A context menu is open over the 'clouptraildata' row, with 'Edit table details', 'View details', 'View data' (which is highlighted in blue), and 'Delete table' as options. In the background, there's a preview pane showing an SQL query and its results. The results table has columns: eventversion, eventid, eventtime, sharedevident, and requestparameters.durationseconds. It lists 10 rows of data, each with a timestamp between 2017-10-23T12:24:08Z and 2017-10-23T12:26:19Z, and a duration of 3600 seconds.

eventversion	eventid	eventtime	sharedevident	requestparameters.durationseconds
1	1.05	4641c1e0-5604-4006-ad6b-380c4ff1bf163	b6bd0e80-85f8-448a-9fb0-c7826e38ac1e	3600
2	1.05	29279e03-9e0b-42ef-9d8f-ca70342881fd	47927aca-6499-4591-8fe-29b56168d7dd	3600
3	1.05	d6614097-b35f-412d-8ba2-f81d5da56ed	4f9730a8-5a41-4de5-9441-8b81fe22a4c	3600
4	1.05	c6bb139-1180-4035-853c-20f92e1cf468	65841ff1054-470e-850-a0d554841b	3600
5	1.05	c21882e4-6a02-4e31-9329-901c288a3206	23951756-d749-4497-86a8-d580fcbb1b45	3600
6	1.05	410ab47-ab5b-4215-a059-4e9e462009fb	6399304e-c852-4c59-a370-68f1202e889e	3600
7	1.05	77odcof2-8030-432d-8c7a-15aae65452e0	a9257c9-d9bc-4549-88ea-1bb27bece614	3600
8	1.05	ca55bbff2-6120-4cb0-9cab-9cdff7fca7c	f0999a02-2371-4900-9a16-9b90677643a3	3600
9	1.05	643c185a-ad38-41ef-a82f-e9adfb680bc	f4ed4d07-d8f6-4b68-be97-cfc107240e64	3600
10	1.05	0656c3ad-d928-4536-aa0d-f7edae0101f1	c90ec01b-497a-494d-a460-67a1719b44b3	3600

Audit and monitor in real time

See detailed alerts in the console

Download audit logs for further analytics

Data ingest and catalog notifications also published to Amazon CloudWatch events

The screenshot shows the AWS Lake Formation Dashboard. On the left, there's a sidebar with 'Monitor and audit' selected, showing 4 notifications. The main area has four stages: Stage 1 (Ingest and Register), Stage 2 (Security and Control), Stage 3 (Collaborate and Use), and Stage 4 (Monitor and Audit). Each stage has associated buttons like 'Create data lake', 'Table permissions', etc. A pink box highlights the 'Recent activity' section at the bottom right, which lists four audit events:

Description	Type	Resource	Alert time
Create access was granted to JohnD	Grant	data_science:trend_extract	November 28, 2018 14:24
Select access was revoked for TimK	Revoke	sales:invoices:customer, address	November 28, 2018 13:37
Data was accessed via Athena by DaveM	Access	finance:2018_monthly	November 28, 2018 13:06
A new table was added by DaveM	New table	finance:2019_projections	November 28, 2018 12:44

To Summarize : Data Lake in 3 easy steps

1. Use Blueprints to Ingest data
2. Use Glue to Catalog and Transform data
3. Grant Permissions to securely share data

Step 1: Blueprints to Ingest data

The screenshot shows the 'Create blueprint' page in the AWS Lake Formation console. The user is configuring an 'Incremental database configuration' for a 'Database - incremental' blueprint.

Blueprint type: Database - incremental

Source details:

- Blueprint name:** wordpress_import
- Database connection:** blueprint_connection
- IAM role:** Glue_DefaultRole
- Database path:** wordpress_db

Incremental table and column:

Table name	Bookmark keys - optional	Bookmark keys sort order - optional
wp_users	ID	Ascending
wp_comments	Type bookmark keys...	Descending

Feedback: English (US)

Step 2: Glue to Catalog and Transform data

The screenshot shows the AWS Glue Data Catalog interface. On the left, a sidebar navigation includes: AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers (selected), Classifiers, Settings, ETL, Jobs, ML Transforms, Triggers, Dev endpoints, Notebooks, Security, and Security configurations.

The main area displays two crawler configurations:

- Crawler 1 (Left):** Name: twitter_raw_status, Schedule: Daily. Configuration details include: Description, Create a single schema for each S3 path: false, Security configuration, Tags: owner octank-broadcasting, State: Ready, Schedule: Daily, Catalog type: Glue, Last updated: Mon May 13 14:11:49 GMT-400 2019, Date created: Mon May 13 14:11:49 GMT-400 2019, Database: octank_social_db, Table prefix: test_, Service role: service-role/AWSGlueServiceRole-twitter-statuses-raw, Selected classifiers: Data store: S3, Include path: s3://octank-broadcasting-datalake-raw/social/twitter/statuses/year=2019, Exclude patterns, Configuration options, Schema updates in the data store: Update the table definition in the data catalog, Object deletion in the data store: Mark the table as deprecated in the data catalog.
- Crawler 2 (Right):** Name: phk-test-octank-ods-scott-schema-crawl, Schedule: Daily. Configuration details include: Description, Create a single schema for each S3 path: false, Security configuration, Tags: - (empty), State: Ready, Schedule: Daily, Catalog type: Glue, Last updated: Mon May 13 22:56:13 GMT-400 2019, Date created: Sat May 11 21:44:07 GMT-400 2019, Database: pk_test_oracle_scott_db, Table prefix: test_, Service role: service-role/AWSGlueServiceRole-twitter-statuses-raw, Selected classifiers: Data store: JDBC, Connection: octank-odc-connection-1, Include path: CWDB01, Exclude paths, Configuration options, Schema updates in the data store: Update the table definition in the data catalog, Object deletion in the data store: Mark the table as deprecated in the data catalog.

Ingested dataset as table in the data lake

AWS Lake Formation > Tables > wordpress_import_797a0017_wordpress_db_wp_users

wordpress_import_797a0017_wordpress_db_wp_users

[View properties](#) [Edit](#) [Delete](#) [Compare versions](#) [Edit schema](#) Version 1 (Current version) ▾

Table details		
Table Name	wordpress_import_797a0017_wordpress_db_wp_users	
Database	wordpress_import	
Location	s3://aws-glue-ingestor-demo-us-east-1/wordpress_import/wordpress_import_797a0017_wordpress_db_wp_users/version_0/	
Last Updated	Tue Nov 20 2018 10:29:38 GMT-0800 (Pacific Standard Time)	
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	
Serde parameters	field.delim ,	

Column number	Column name	Data type
1	user_status	int
2	user_email	string
3	user_login	string
4	user_url	string
5	user_nicename	string
6	user_registered	timestamp
7	id	bigint
8	user_activation_key	string
9	display_name	string
10	user_pass	string

Step 3: Grant Permissions to securely share data

The screenshot shows the AWS Lake Formation User permissions interface. A modal window titled "Grant permissions to table wordpress_import_797a0017_wordpress_db_wp_users" is open, prompting the user to grant access to specific users, groups, or roles. The user "shyamsh" has been selected. The "Specific permissions" option is chosen, with "Select" checked. The main pane displays a list of existing user permissions, showing entries for "johnd", "sales", "reviews", "orders", "salesgrp", "sales", "analyst", "reviews", "shyamsh", and "wordpress_import_797a0017_wordpress_db_wp_users". The "shyamsh" entry is highlighted with a blue border, indicating it is selected for granting permissions.

Name	Type	Permissions	Last modified
johnd	User	3 permissions	11/28/2018 14:24
sales	Database	Create, Select, Insert	11/28/2018 13:37
reviews	Table	Administrator	11/28/2018 13:37
orders	Table	Create, Drop	11/28/2018 12:44
salesgrp	Group	1 permission	11/28/2018 13:37
sales	Database	Administrator	11/28/2018 13:37
analyst	Role	1 permission	11/28/2018 13:37
reviews	Table	Create, Select, Insert, Alter, Drop	11/28/2018 13:37
shyamsh	User	1 permission	11/28/2018 14:22
wordpress_import_797a0017_wordpress_db_wp_users	Table	Select	11/28/2018 14:22

Run query in Amazon Athena, Spectrum or use EMR to securely run Big Data jobs (More integrations coming..)

The image displays two side-by-side screenshots of the AWS Athena Query Editor interface. Both screenshots show a user running a query against a database named 'wordpress_import'.

Left Screenshot (User: shyamsh):

- Database:** wordpress_import
- Query:** `1 SELECT * FROM "wordpress_import"."wordpress_import_797a0017_wordpress_db_wp_users" limit 10;`
- Results:** A table showing user data:

	user_status	user_email	user_login	user_url	user_nicename
1	0	galazzah@amazon.com	galazzah		galazzah
2	0	shyamsh@amazon.com	shyamsh		shyamsh

Right Screenshot (User: auslee):

- Database:** wordpress_import
- Query:** `1`
- Results:** An empty table.



Lake Formation : End to End flow

Build data lakes quickly

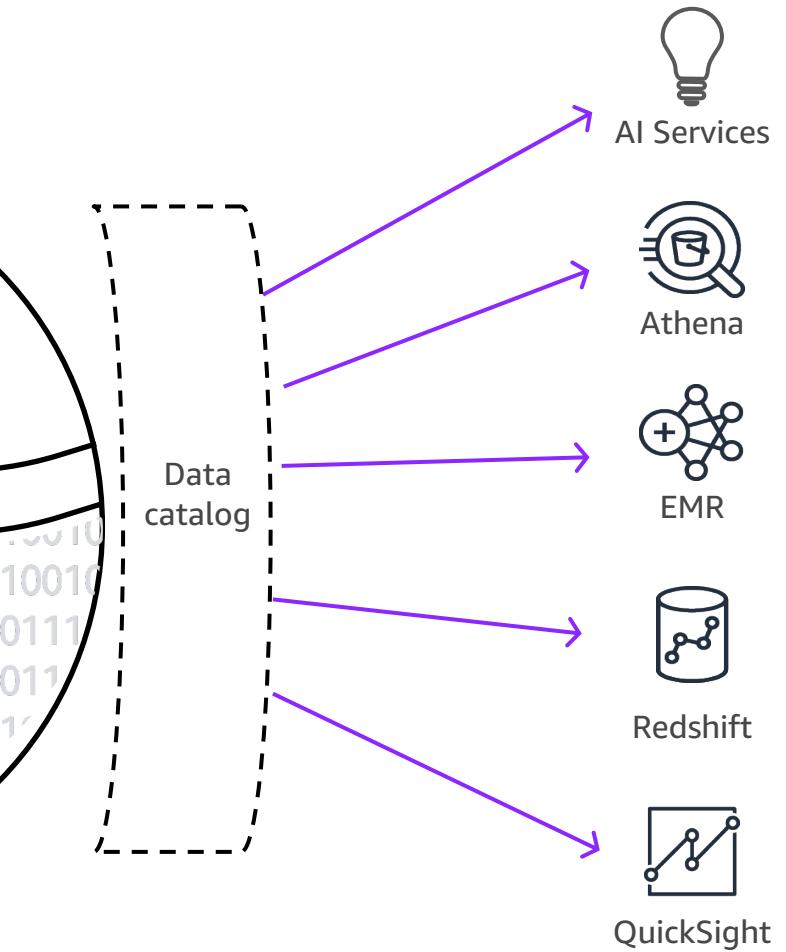
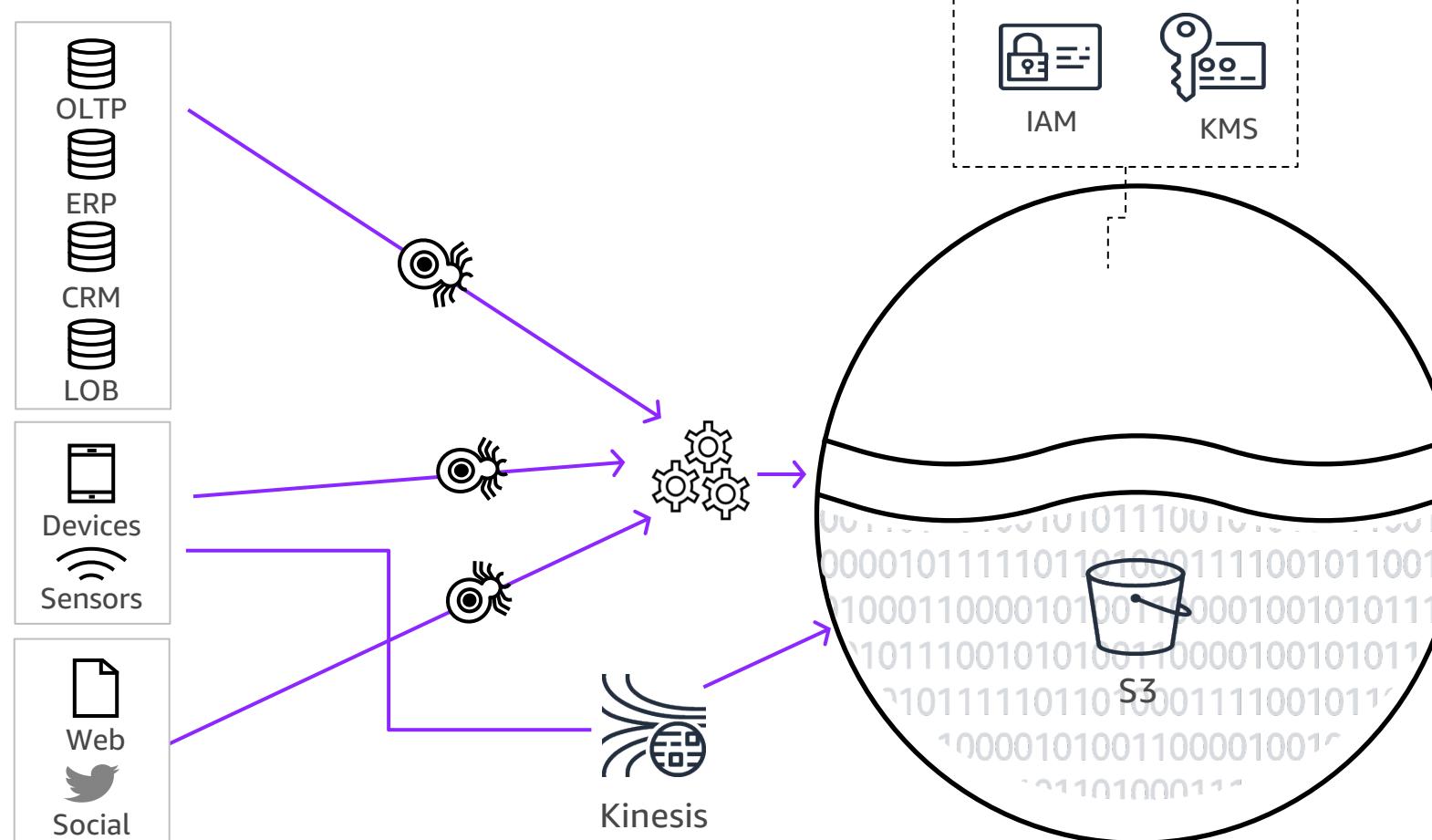
- Identify, crawl, and catalog sources
- Ingest and clean data
- Transform into optimal formats

Simplify security management

- Enforce encryption
- Define access policies
- Implement audit login

Enable self-service and combined analytics

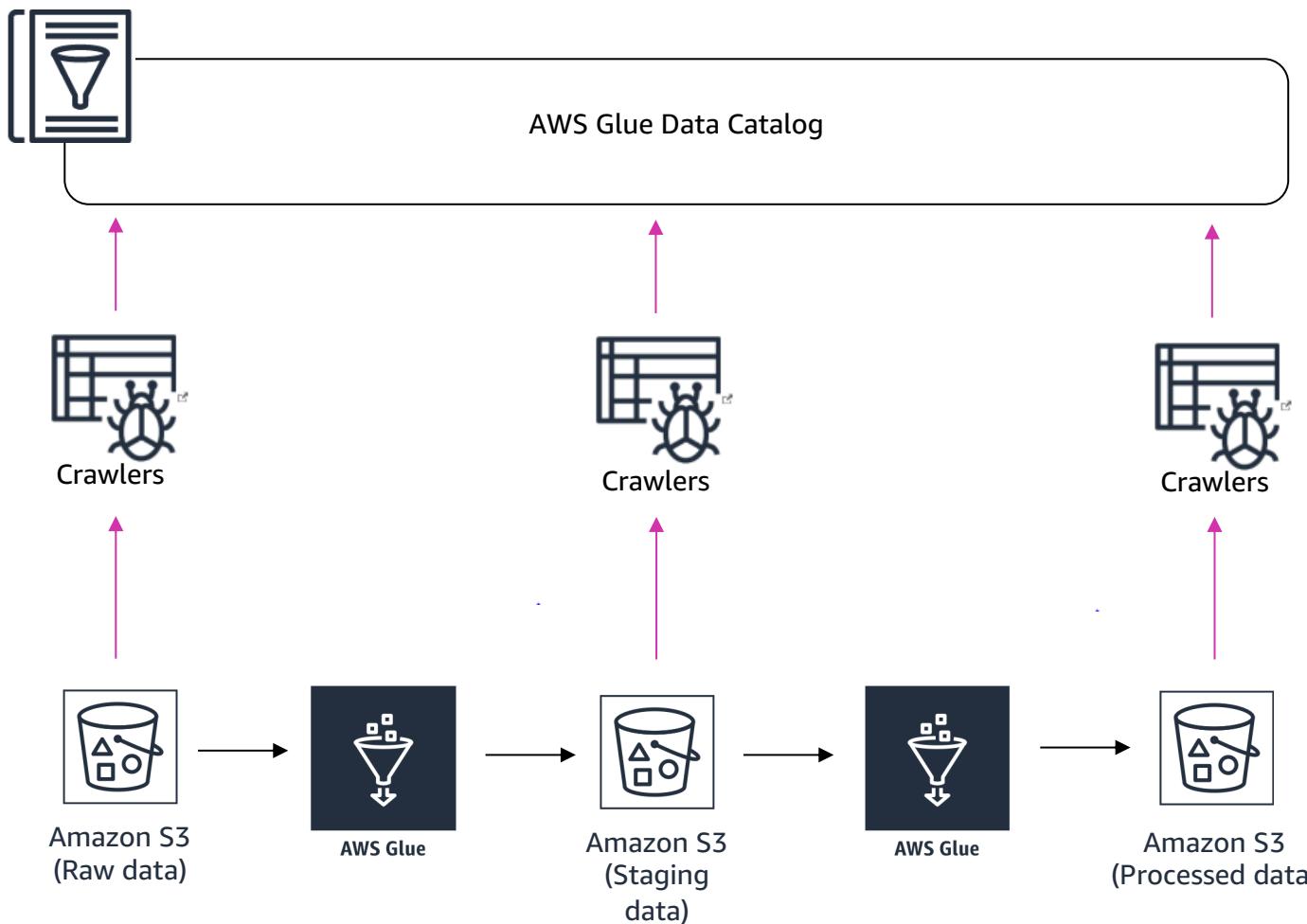
- Analysts discover all data available for analysis from a single data catalog
- Use multiple analytics tools over the same data



Glue

Serverless ETL Engine that powers Lake Formation

Use AWS Glue to cleanse, prep, and catalog data in Lake



AWS Glue Data Catalog - a single view across your data lake

- Automatically **discovers** data and stores schema
- Makes data **searchable**, and available for ETL
- Contains table definitions and custom metadata

Use AWS Glue ETL jobs to cleanse, transform, and store processed data

- **Serverless Apache Spark** environment
- Use Glue ETL libraries or bring your own code
- Write code in **Python or Scala**
- **Call any AWS API** using the AWS boto3 SDK



AWS Glue Data Catalog

Discover and organize your data

What is the AWS Glue Data Catalog?

Unified metadata repository across relational databases either on AWS or on-premises, Amazon RDS, Amazon Redshift, and Amazon S3!

Single searchable view into your data, no matter where it is stored

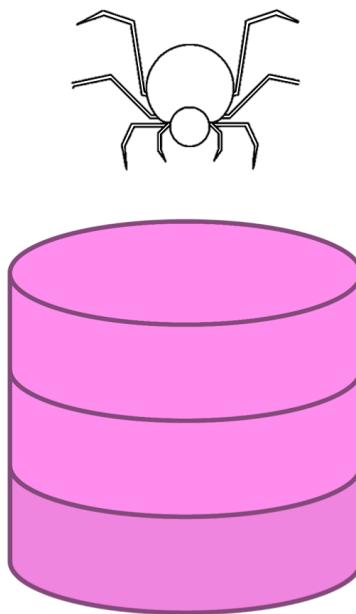
Ability to automatically crawl and **classify your data**

Augment technical metadata with **business metadata** for tables

Track data evolution using **schema versioning**

Apache Hive metastore compatible and integrated with AWS Analytics services

Build Data Lake Catalog using Glue Crawlers



Crawlers

Automatically catalog your data

- Crawlers **automatically build** your Data Catalog and keep it in sync.
- Automatically **discover** new data, extracts schema definitions
 - Detect schema changes and version tables
 - Detect Hive style partitions on Amazon S3
- Built-in classifiers for popular types; custom classifiers using **Grok expression**
- Run ad hoc or on a schedule; **serverless** – only pay when crawler runs

A table in the Glue Data Catalog

Table properties

Data statistics

Table schema

The screenshot shows the AWS Glue Data Catalog interface. On the left, a sidebar lists various options: AWS Glue, Data catalog, Databases, Tables (highlighted with a pink box), Connections, Crawlers, Classifiers, ETL, Jobs, Triggers, Dev endpoints (highlighted with a pink box), Tutorials, Add crawler, Explore table, Add job, and Resources (highlighted with a pink box). The main area displays the properties of a table named '2015' in the 'gitarchive' database. The table's location is 's3://glue-sample-datasets/examples/githubarchive/2015/'. The 'Tables' option in the sidebar is also highlighted with a pink box. The 'Nested fields' section shows a detailed schema for the 'payload' field, which is defined as a STRUCT type containing various sub-fields like 'ref', 'ref_type', 'master_branch', etc. A pink arrow points from the 'Tables' sidebar entry to the 'Tables' section in the main content area.

Name: 2015
Description:
Database: gitarchive
Classification: json
Location: s3://glue-sample-datasets/examples/githubarchive/2015/
Connection:
Deprecated: No
Last updated: Fri Aug 11 06:13:10 GMT-700 2017
Input format: org.apache.hadoop.mapred.TextInputFormat
Output format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib: org.openx.data.jsonserde.JsonSerDe
Serde parameters: paths actor,created_at,id,org,payload,public,repo,type
sizeKey: 26129991 objectCount: 1 UPDATED_BY_CRAWLER gitarchive_new
Table properties: CrawlerSchemaSerializerVersion: 1.0 recordCount: 11888 averageRecordSize: 2198
CrawlerSchemaDeserializerVersion: 1.0 compressionType: none typeOfData: file

payload schema details

Column name	Data type	Key
1 id	string	
2 type	string	
3 actor	struct	
4 repo	struct	
5 payload	struct	
6 public	boolean	
7 created_at	string	
8 org	struct	

aws

Automatically detected partitions

AWS Glue Console - Secure | https://console.aws.amazon.com/glue/home?region=us-east-1#table:name=githubevents_data;namespa...
AWS Services Resource Groups Developer/mashah-Isengard @... N. Virginia Support
Data catalog Databases Tables Connections Crawlers Classifiers ETL Jobs Triggers Dev endpoints Tutorials Add crawler Explore table Add job Resources

Name: githubevents_data
Description: gitarchive
Database: gitarchive
Classification: json
Location: s3://glue-sample-datasets/examples/data/
Connection: CCP:201... Redshift CA Codes (CCP:201... Financials Hiring Bank Accounts Free-to-Play Game... Other Bookmarks
Deprecated: No
Last updated: Wed Nov 22 07:52:09 GMT-800 2017
Input format: org.apache.hadoop.mapred.TextInputFormat
Output format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib: org.openx.data.jsonserde.JsonSerDe
Serde parameters: paths actor,created_at,id,org,payload,public,repo,type
Table properties: CrawlerSchemaSerializerVersion 1.0 recordCount 27833145 averageRecordSize 2423
CrawlerSchemaDeserializerVersion 1.0 compressionType gzip typeOfData file
Schema (Showing: 1 - 11 of 11) Column name Data type Key
1 id string
2 type string
3 actor struct
4 repo struct
5 payload struct
6 public boolean
7 created_at string
8 org struct Partition (0)
9 year string Partition (1)
10 month string Partition (2)
11 day string
Feedback English (US) Privacy Policy Terms of Use © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Available partitions

aws Services Resource Groups Developer/mashah-Isengard @...
Tables > githubevents_data Last updated 22 Nov 2017 Table Version (Current version)
Edit table Delete table Close partitions Compare versions Edit schema
Showing: 1 - 100 < >
year month day
2017 02 15 View files View properties
2017 03 12 View files View properties
2017 05 17 View files View properties
2017 10 12 View files View properties
2017 12 18 View files View properties
2017 01 09 View files View properties
2017 03 07 View files View properties
2017 06 28 View files View properties
Feedback English (US) Privacy Policy Terms of Use © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

year	month	day		
2017	02	15	View files	View properties
2017	03	12	View files	View properties
2017	05	17	View files	View properties
2017	10	12	View files	View properties
2017	12	18	View files	View properties
2017	01	09	View files	View properties
2017	03	07	View files	View properties
2017	06	28	View files	View properties

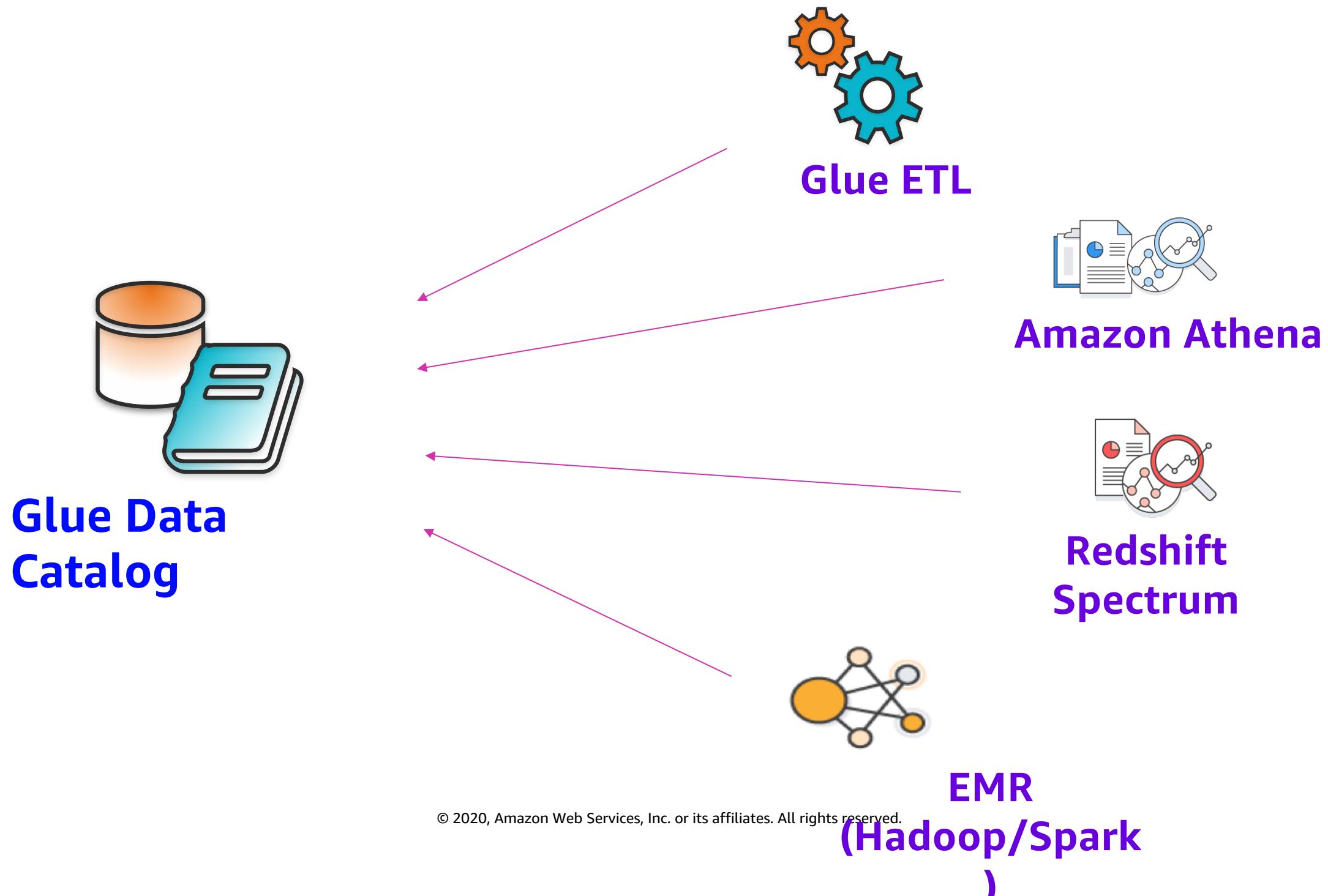
Automatic schema versioning

Automatically update table version as data evolves



Last updated 21 Aug 2017 Table Version 1		Last updated 25 Nov 2017 Table Version 2	
Name	simpletweets_json	Name	simpletweets_json
Description		Description	
Database	simpletweets_json	Database	simpletweets_json
Classification	json	Classification	json
Location	s3://gluesampleddata/simpletweets.json	Location	s3://gluesampleddata/simpletweets.json
Connection		Connection	
Deprecated	No	Deprecated	No
Last updated	Mon Aug 21 15:23:42 GMT-700 2017	Last updated	Sat Nov 25 12:30:28 GMT-800 2017
Input format	org.apache.hadoop.mapred.TextInputFormat	Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.openx.data.jsonserde.JsonSerDe	Serde serialization lib	org.openx.data.jsonserde.JsonSerDe
Serde parameters	paths entities,id,retweeted,text,user	Serde parameters	paths entities,id,retweeted,text,user
	sizeKey 456580 objectCount 1 UPDATED_BY_CRAWLER TestS3Crawler		sizeKey 456580 objectCount 1 UPDATED_BY_CRAWLER TestS3Crawler
Table properties	mycustom abc CrawlerSchemaSerializerVersion 1.0 recordCount 1001	Table properties	mycustom abc CrawlerSchemaSerializerVersion 1.0 recordCount 1001
	averageRecordSize 456 CrawlerSchemaDeserializerVersion 1.0		averageRecordSize 456 CrawlerSchemaDeserializerVersion 1.0
	compressionType none typeOfData file		compressionType none typeOfData file
Change	Column name	Data type	Key
	id	bigint	
	retweeted	boolean	
	text	string	
	user	struct	
Change	Column name	Data type	Key
	id	bigint	
	retweeted	boolean	
	text	string	
	user	struct	
Added	url	string	

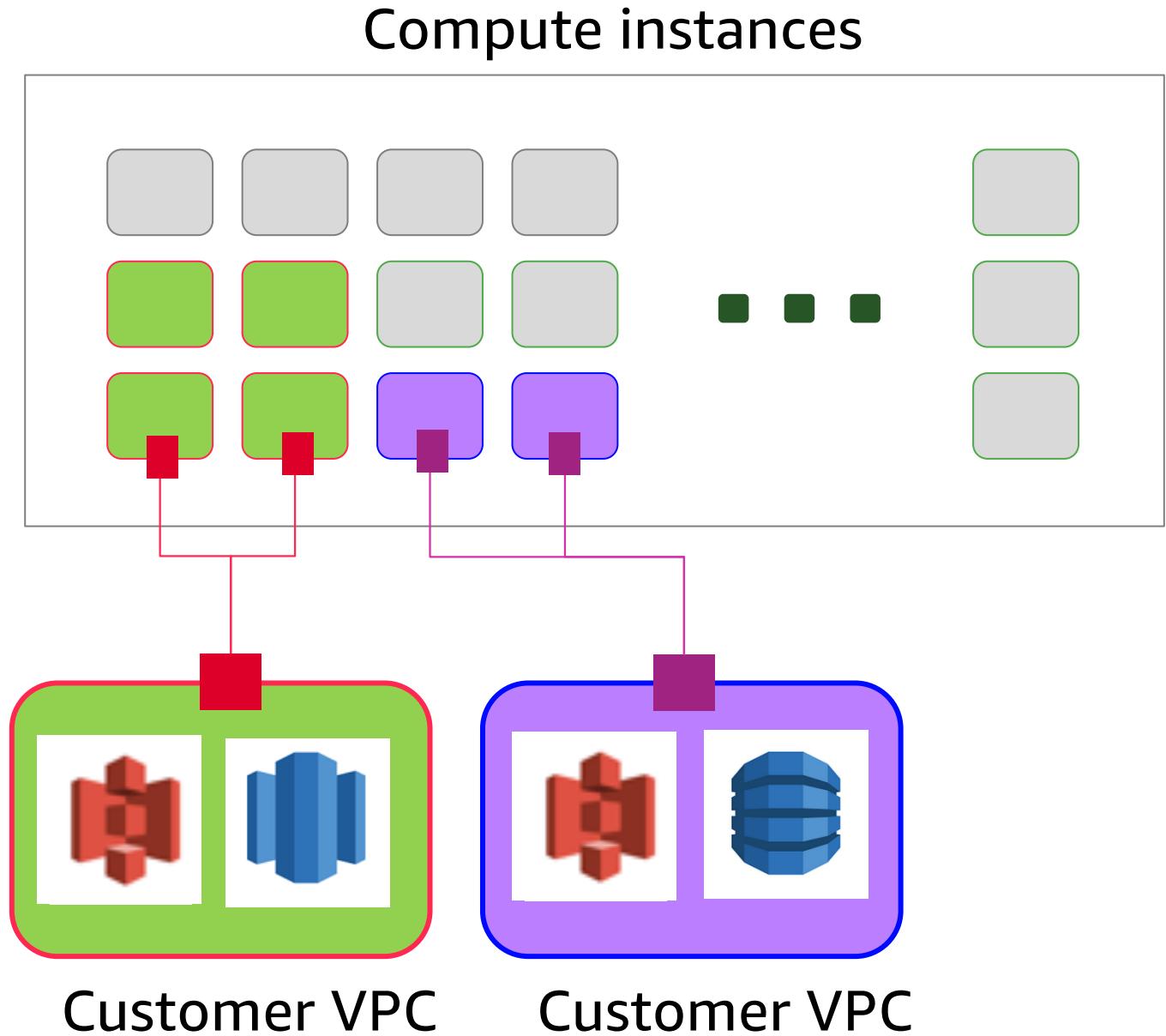
Glue: Data Catalog – Queryable by Many Services



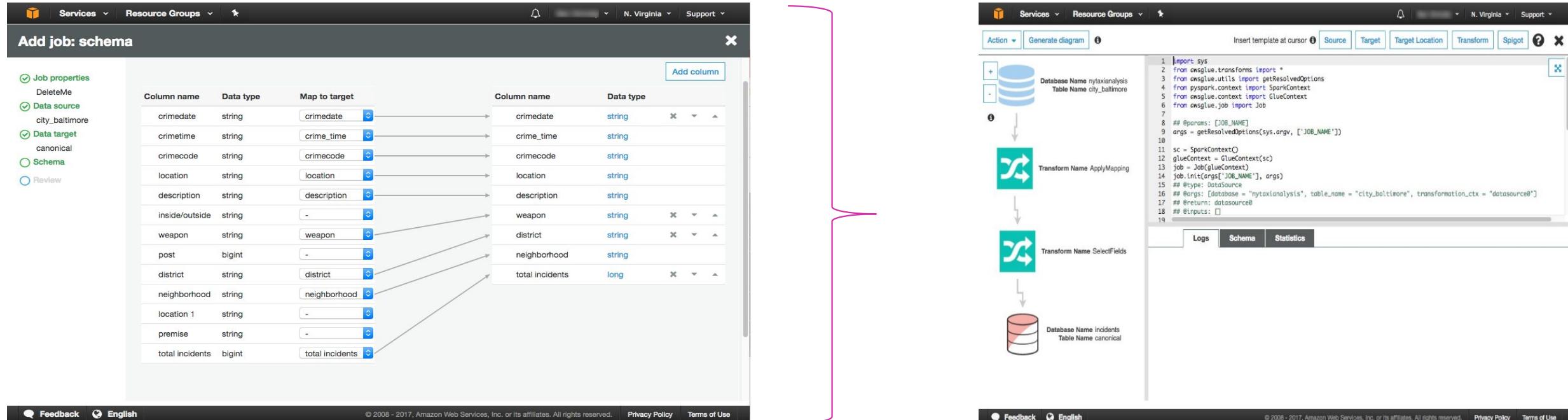
Transform data using Serverless ETL

No need to provision, configure, or manage servers

- Build Python or Spark based ETL code
- Run it in Serverless environment
- Specify the capacity that gets allocated to each ETL job
- Pay only for the resources you consume while consuming them. Stop paying once job is complete
- Auto-configure VPC and role-based access



Job authoring: Automatic code generation



1. Customize the mappings
2. Glue generates transformation graph and **Python or Scala** code
3. Connect your **notebook** to development endpoints to customize your code



Orchestration and resource management

Fully managed, serverless job execution

Job execution: Scheduling and monitoring

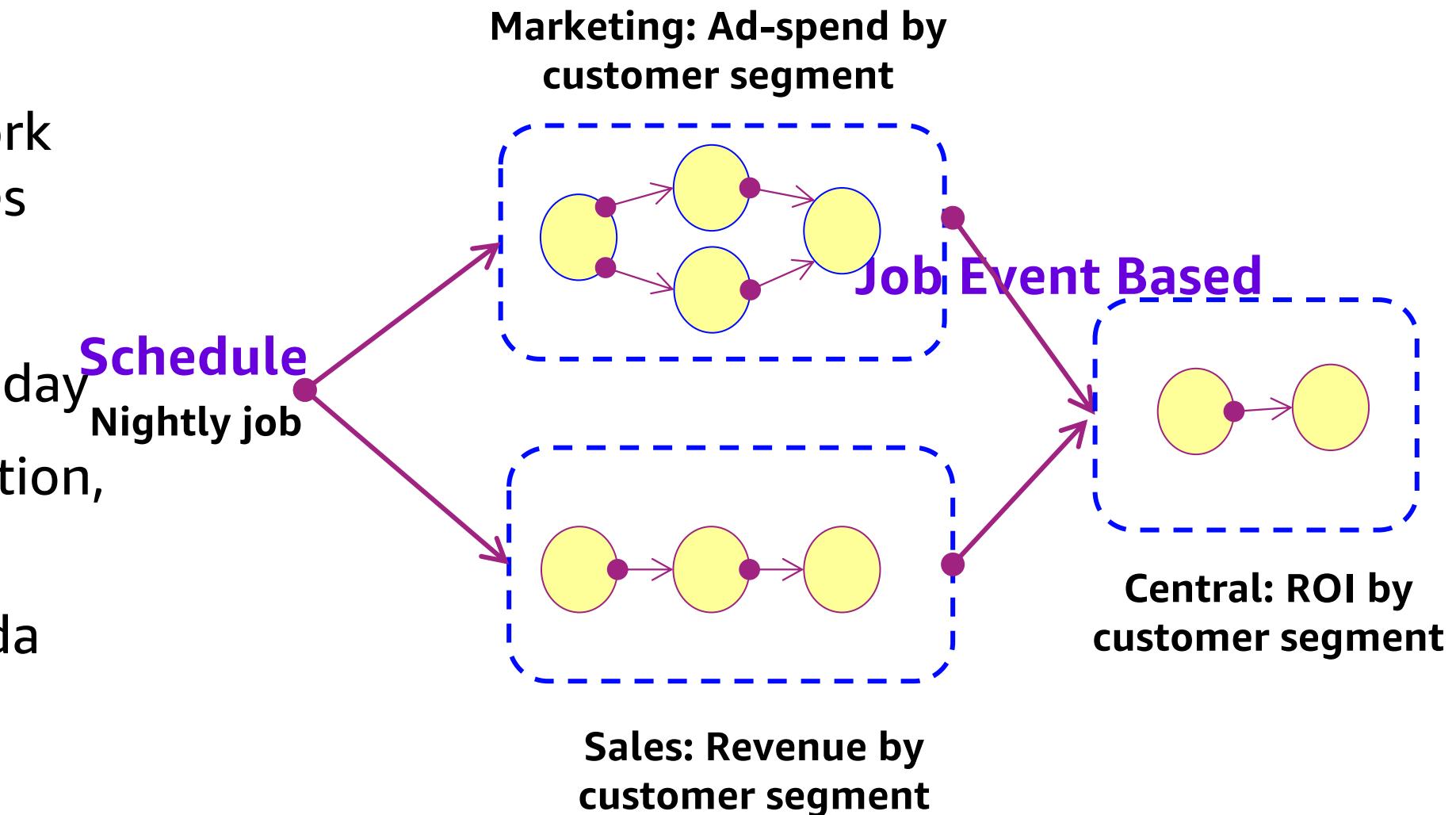
Compose jobs globally with event-based dependencies

- Easy to reuse and leverage work across organization boundaries

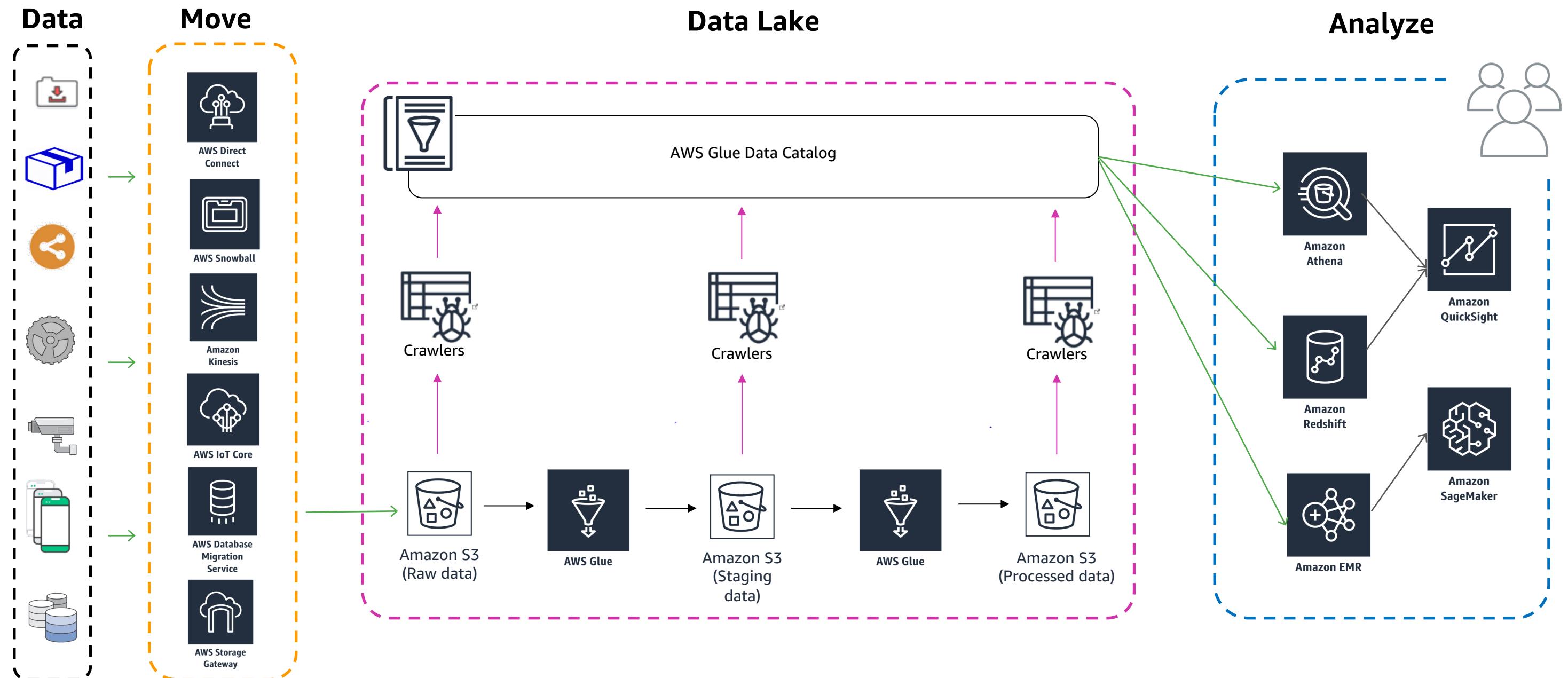
Multiple triggering mechanisms

- **Schedule-based:** e.g., time of day
- **Event-based:** e.g., job completion, job failure, job stopped, etc.
- **On-demand:** e.g., AWS Lambda

Logs and alerts are available in Amazon CloudWatch



Putting it together - Data Lake with AWS Glue



Thank you !

Praful Kava

Solution Architect, Database and Analytics, AWS

pkava@amazon.com

Rajeev Thottathil

Solution Architect, Database and Analytics, AWS

thottr@amazon.com