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.

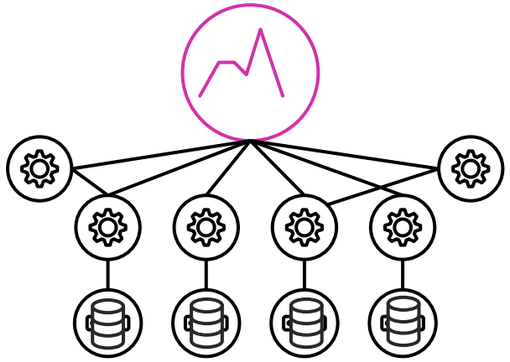
2010

Trends that impact the way you think about data

Explosion of data

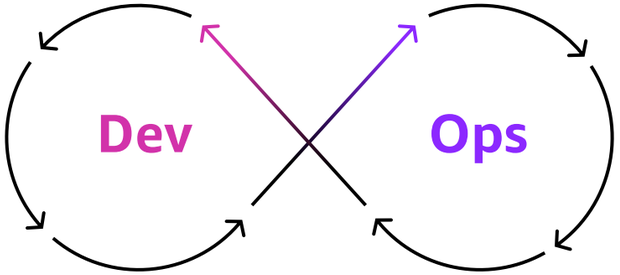
Data grows 10x every 5 years driven by network connected smart devices

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

aws

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

QLDB
Ledger Database

RDS
MySQL, PostgreSQL, MariaDB, Oracle, SQL Server

ElastiCache
Redis, Memcached

Timestream
Time Series

RDS on VMware

Neptune
Graph

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

aws

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
- **QLDB** — Ledger Database
- **RDS** — MySQL, PostgreSQL, MariaDB, Oracle, SQL Server
- **ElastiCache** — Redis, Memcached
- **Timestream** — Time Series
- **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

aws

Three type of projects

Quickly build new apps in the cloud

“Lift and shift” existing apps to the cloud

Gain new insights

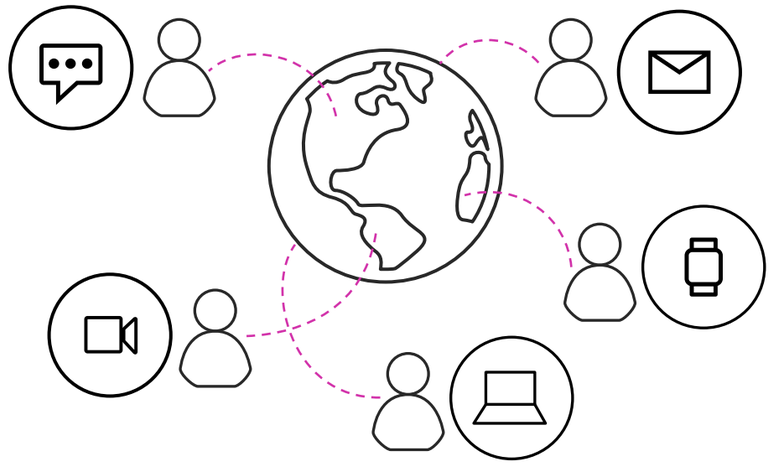
Customers tell us: they have three type of projects

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

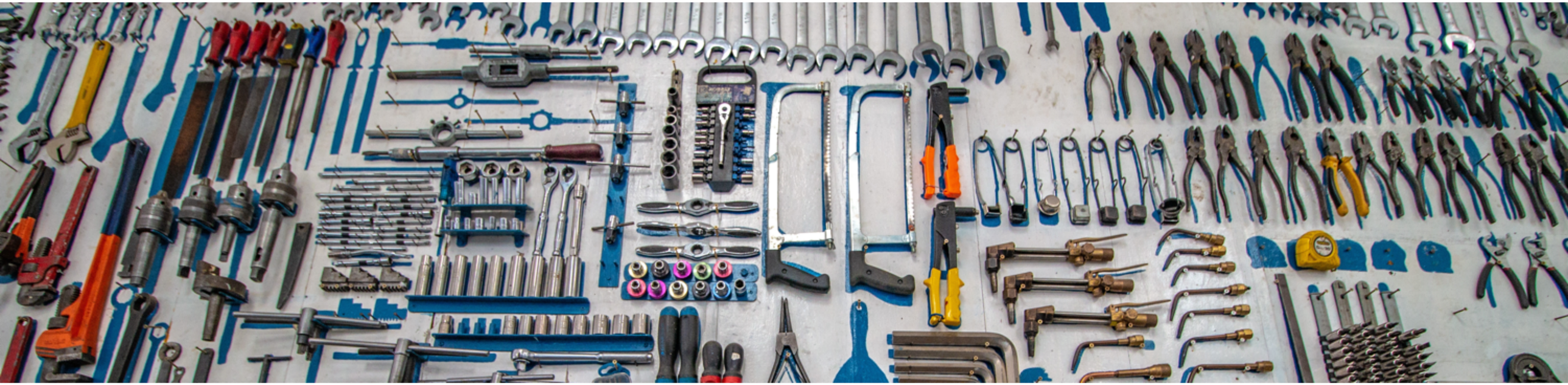
Social media

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

aws

Working backward from the problem you are trying to solve



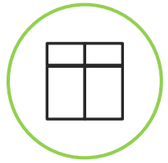
Choosing the right tool for each job

Data categories and common use cases

Relational

Referential integrity, ACID transactions, schema-on-write

Lift and shift, EMR, CRM, finance



Key value

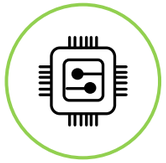
Low-latency, key lookups with high throughput and fast ingestion of data

Real-time bidding, shopping cart, social

Document

Indexing and storing documents with support for query on any attribute

Content management, personalization, mobile



In-memory

Microseconds latency, key-based queries, and specialized data structures

Leaderboards, real-time analytics, caching

Graph

Creating and navigating data relations easily and quickly

Fraud detection, social networking, recommendation engine

Search

Indexing and searching semistructured logs and data

Product catalog, help, and FAQs, full text

Time series

Collect, store, and process data sequenced by time

IoT applications, event tracking

Ledger

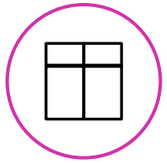
Complete, immutable, and verifiable history of all changes to application data

Systems of record, supply chain, healthcare, registrations, financial

AWS databases services

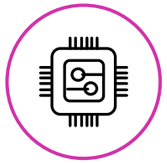
Purpose-built for all your app needs

Relational

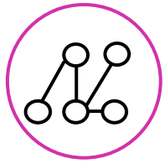


Key-value

Document



In-memory



Graph

Search

Time series

Ledger

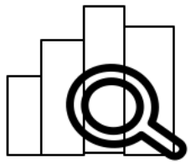
RDS

DynamoDB

Amazon DocumentDB

ElastiCache

Neptune



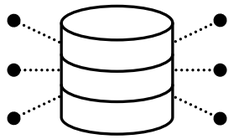
ElasticSearch

Timestream

QLDB

Aurora **Community** **Commercial**

aws



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

Amazon Aurora | MySQL | PostgreSQL | MariaDB | Microsoft SQL Server | ORACLE

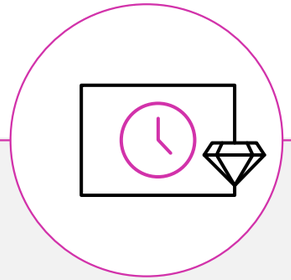
Easy to administer

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

Secure & compliant

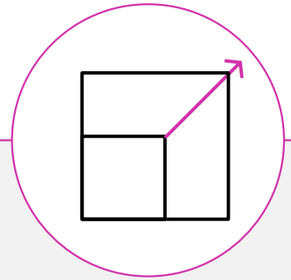
Data encryption at rest and in transit; industry compliance and assurance programs

Available & durable



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

Performant & scalable

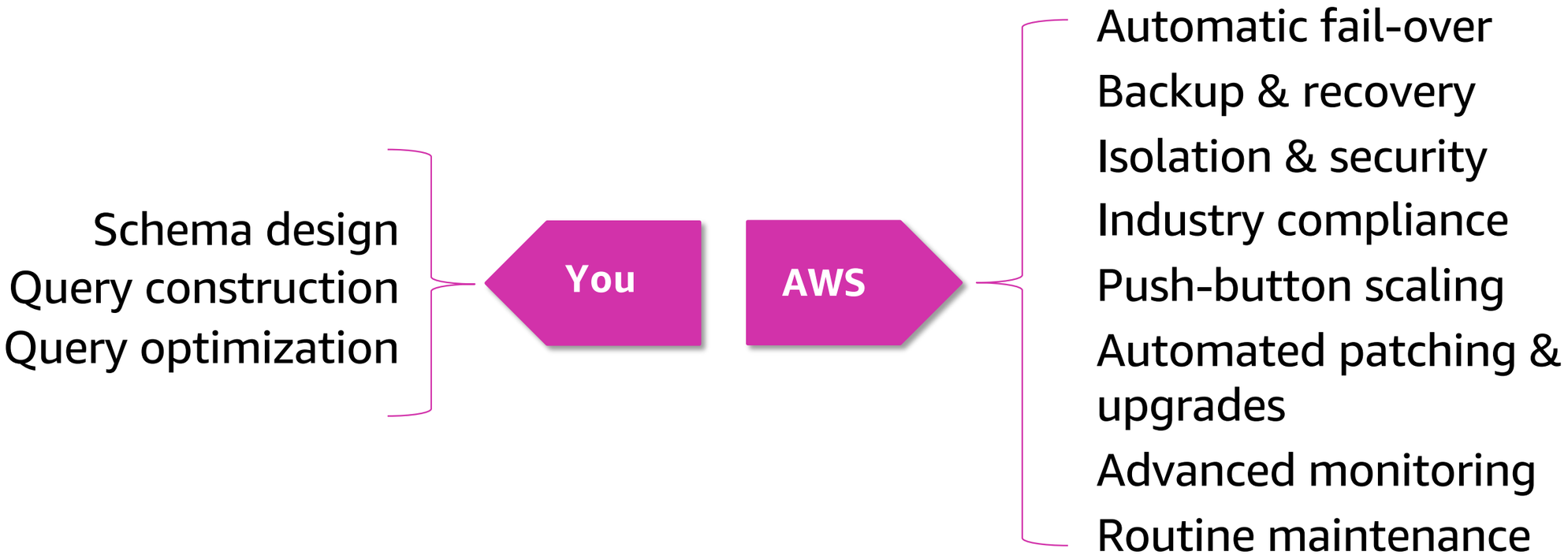


Scale compute and storage with a few clicks; minimal downtime for your application

aws

Amazon RDS - fully managed

Spend time innovating & building new apps, not managing infrastructure



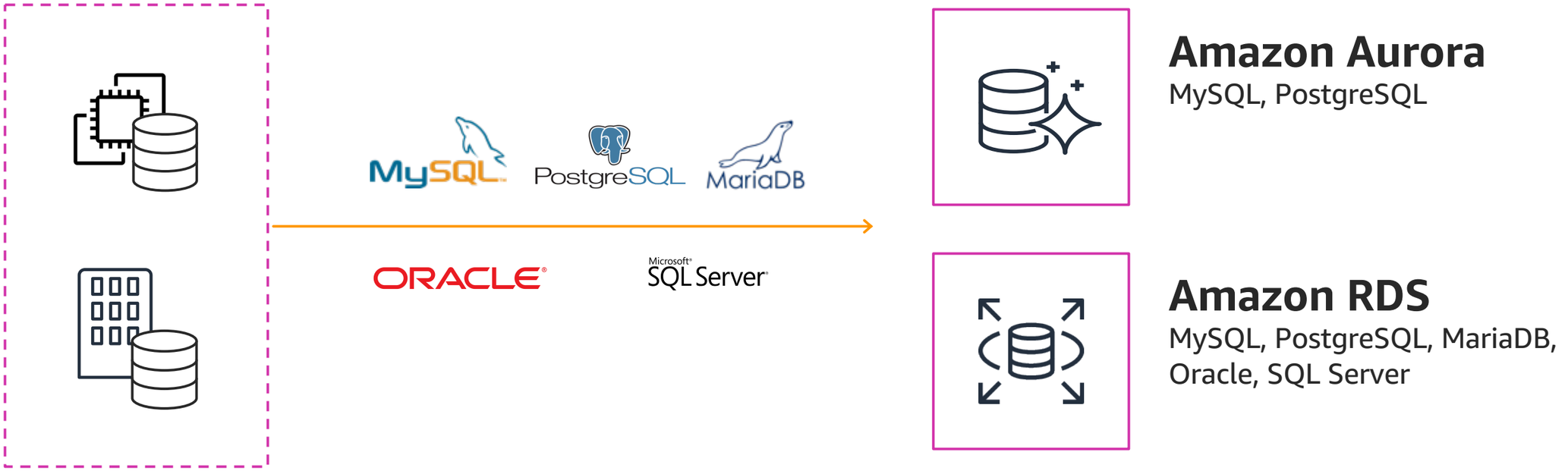
Move to managed relational databases

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

aws

airbnb

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

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

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

"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."

aws

>200,000 databases migrated to AWS

aws

Trimble

Trimble is a global leader in telematics solutions

Challenge:

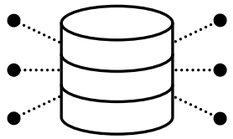
Migrated their on-premise data warehouse to scale, improve availability, and cut cost

Solution:

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

Result:

Trimble's infrastructure costs are projected be to 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

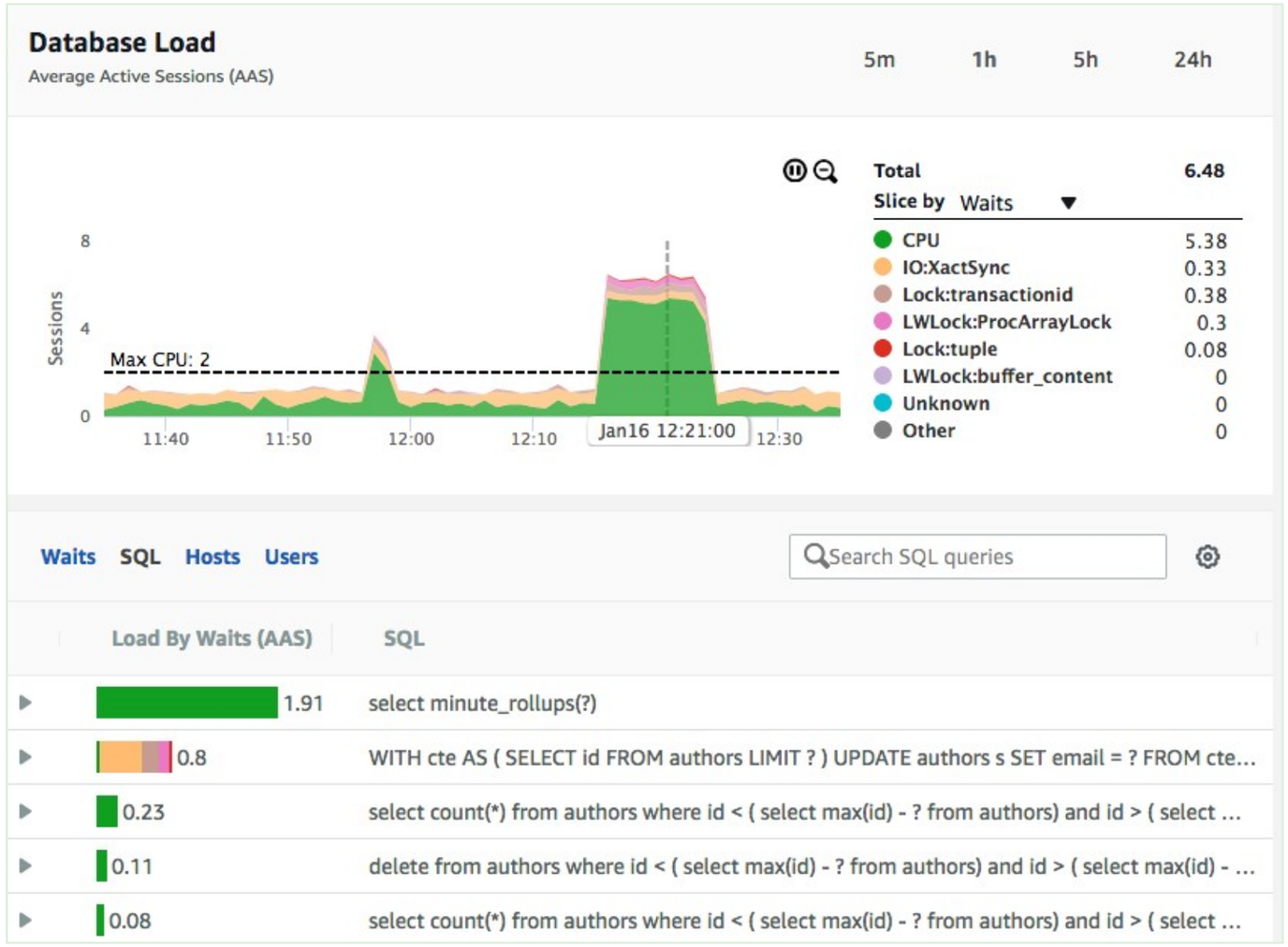
Database Engine

Amazon RDS Performance Insights

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

aws

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

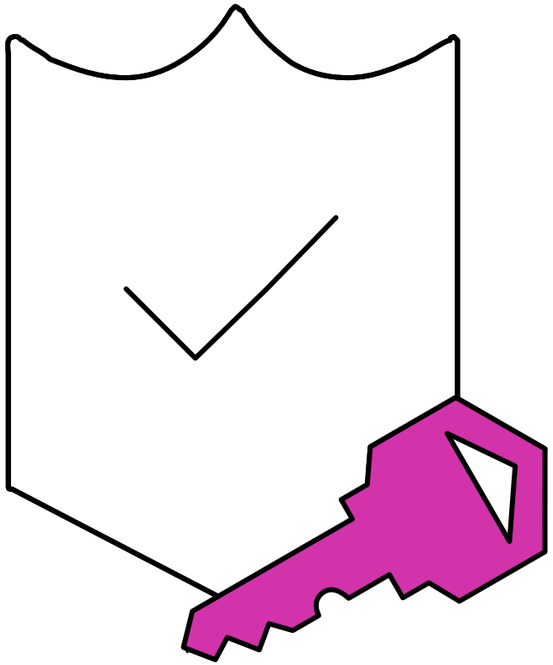
- Hour, day, week, and longer

Available for all Amazon RDS database engines

Secure & compliant

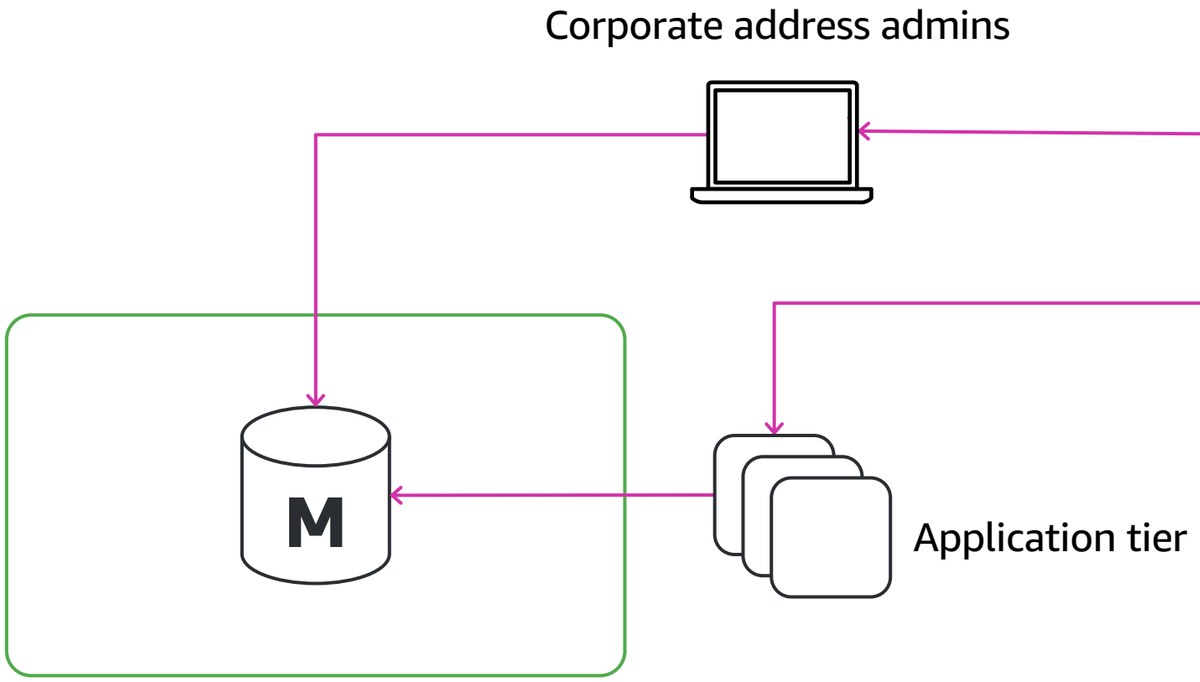
Security and compliance

- 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

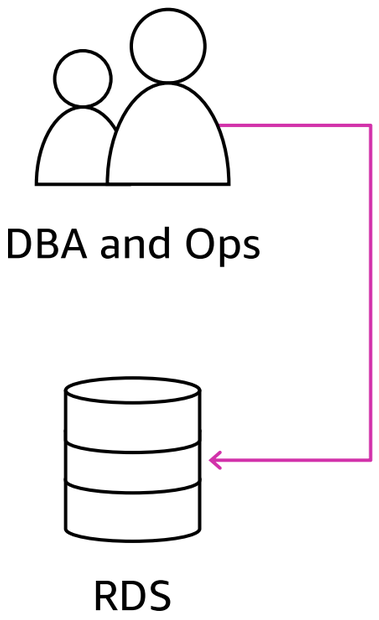


Protocol	Port range	Source
TCP	3306	172.31.0.0/16
TCP	3306	“Application security group”

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

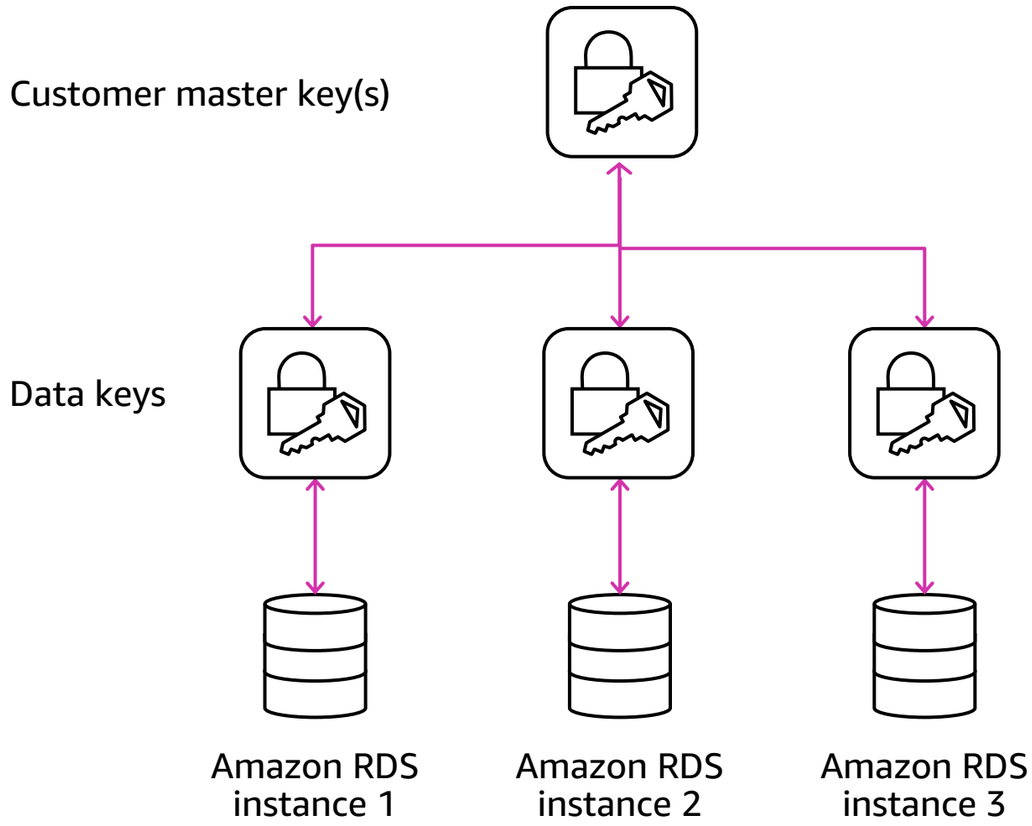
Managed via AWS Key Management Service (KMS)

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

- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- FedRAMP
- HIPAA BAA

MySQL

- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- FedRAMP
- HIPAA BAA
- UK Gov. Programs
- Singapore MTCS

PostgreSQL

- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- FedRAMP
- HIPAA BAA
- UK Gov. Programs
- Singapore MTCS

Oracle

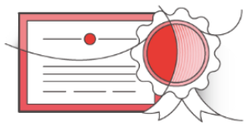
- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- FedRAMP
- HIPAA BAA
- UK Gov. Programs
- Singapore MTCS

MariaDB

- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- HIPAA BAA

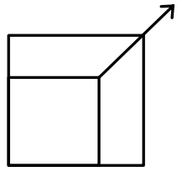
SQL Server

- SOC 1, 2, 3
- ISO 27001/9001
- ISO 27017/27018
- PCI
- HIPAA BAA
- UK Gov. Programs
- Singapore MTCS
- FedRAMP


Singapore MTCS

27001/9001
27017/27018

aws



Highly scalable

Database server instance types

General purpose (T3)

- 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

General Purpose (M5)

- 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)

Memory Optimized (R5)

- 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

Memory Optimized (X1E)

- 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

aws

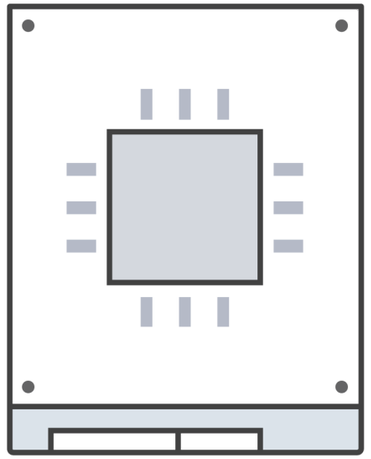
High performance database storage

General purpose (GP2)

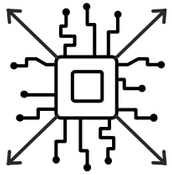
- SSD storage
- Maximum of 64 TiB (16TiB for SQL Server)
- Latency in milliseconds
- IOPS determined by volume size
- Bursts to 3,000 IOPS (applicable below 1.3 TB)
- Affordable performance

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

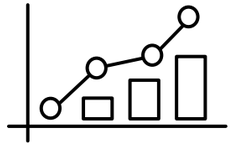


Scale compute and storage with ease



Scale compute to handle increased load

- Up to 96 vCPU and 768 GiB of RAM

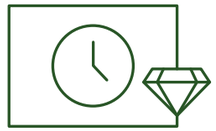


Scale storage for larger data sets

- 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



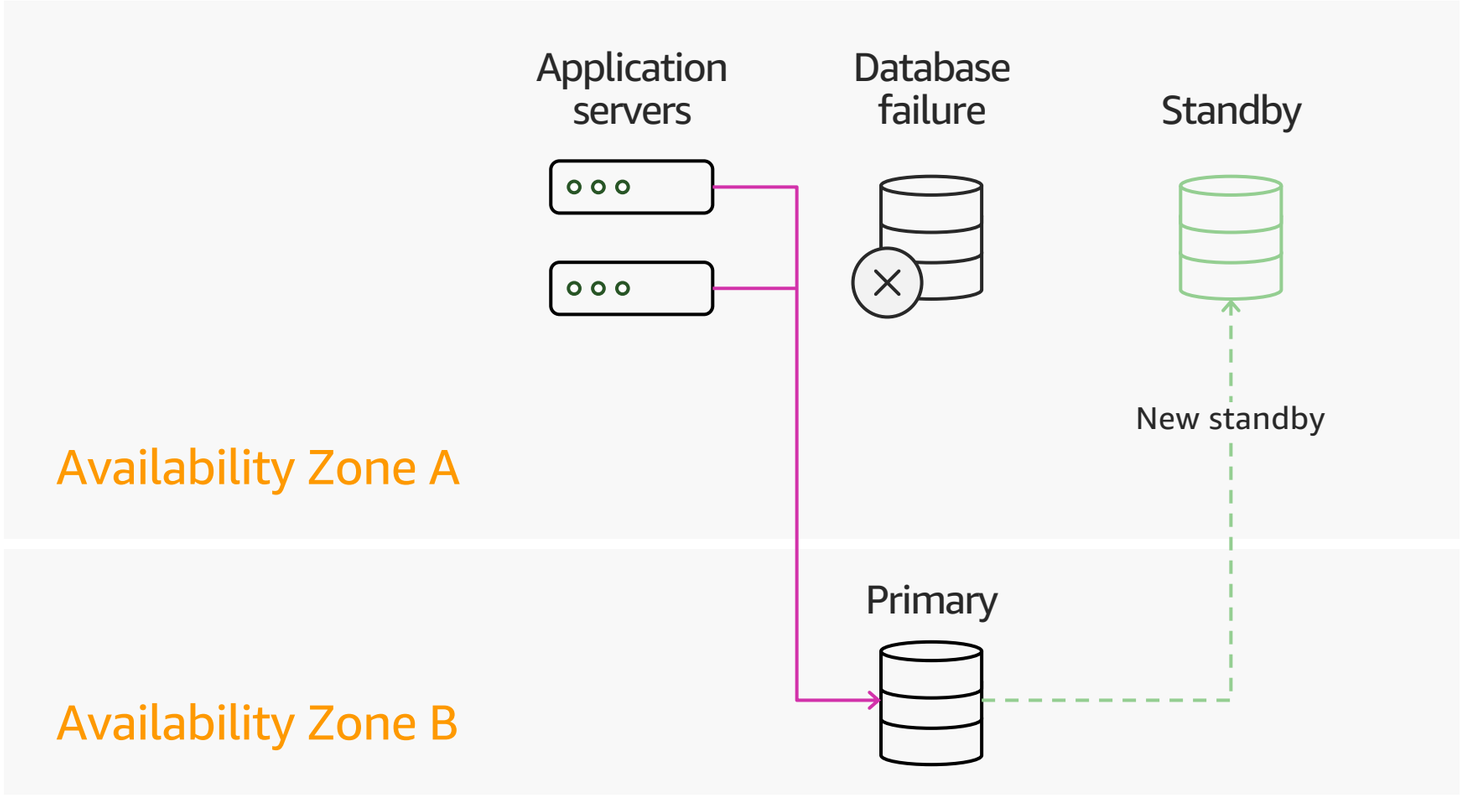
Available & durable

Multi-AZ deployments

Enterprise-grade high availability

Fault tolerance across multiple data centers

- Automatic failover
- Synchronous replication
- Enabled with one click

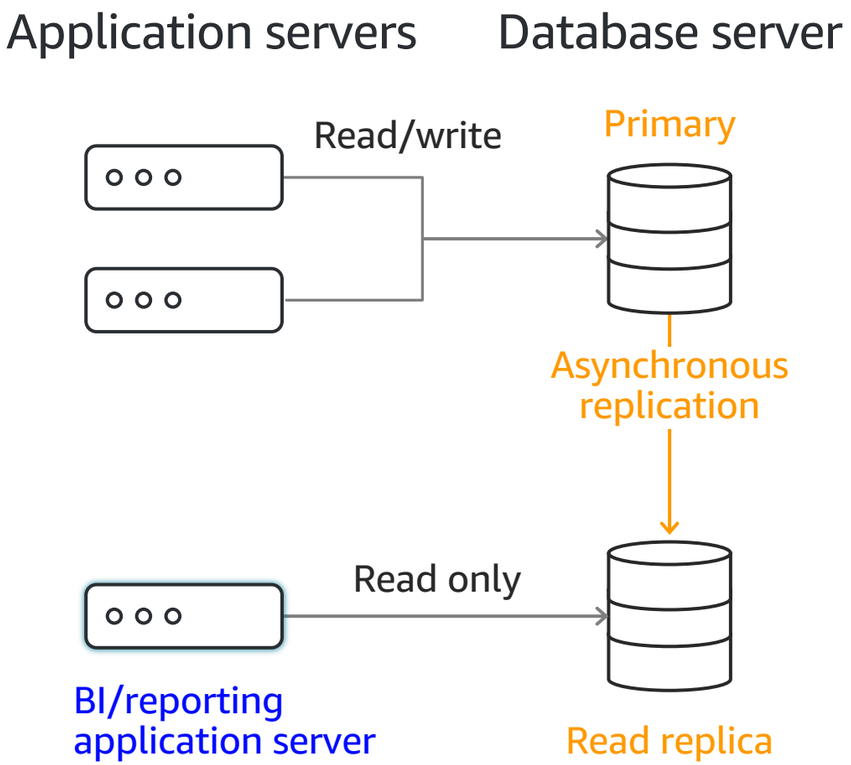


Read Replicas

Read scaling and disaster recovery

RDS for MySQL, PostgreSQL, MariaDB, and Oracle

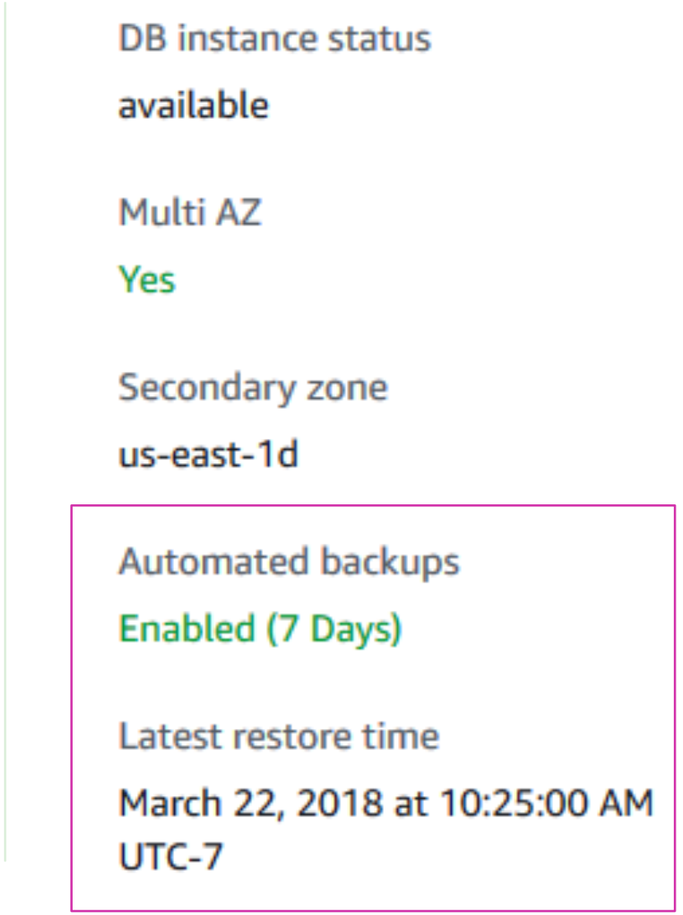
- Relieve pressure on your master node with additional read capacity
- Bring data close to your applications in different regions
- Promote a read replica to a master for faster recovery in the event of disaster



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



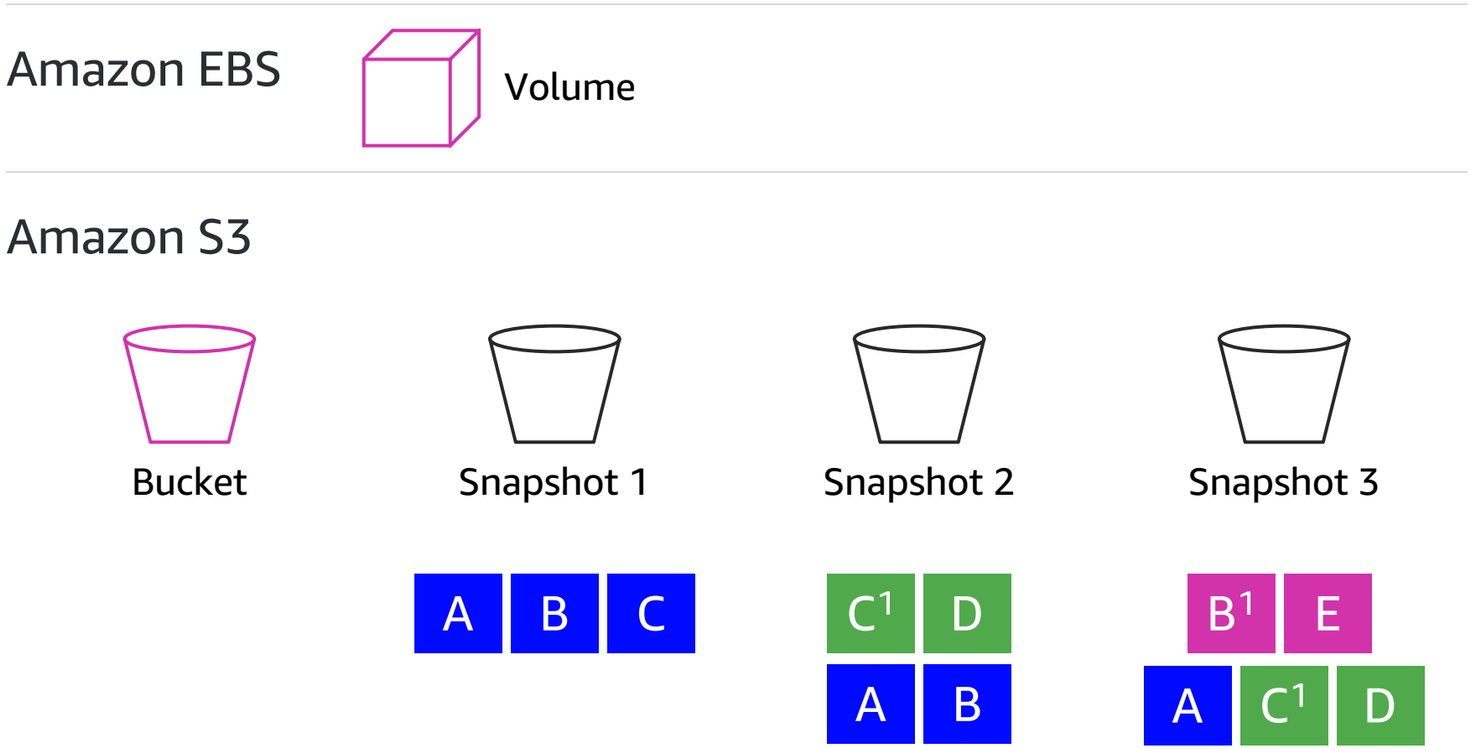
Every day during your backup window, RDS creates a storage volume snapshot of your instance

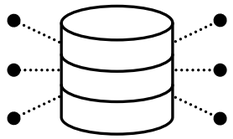
Every five minutes, RDS backs up the transaction logs of your database

Database snapshots

Backups of your entire DB instance in Amazon S3

- Always incremental
- Amazon S3 → 99.999999999% durability
- Supports encryption
- Copy across accounts, across regions





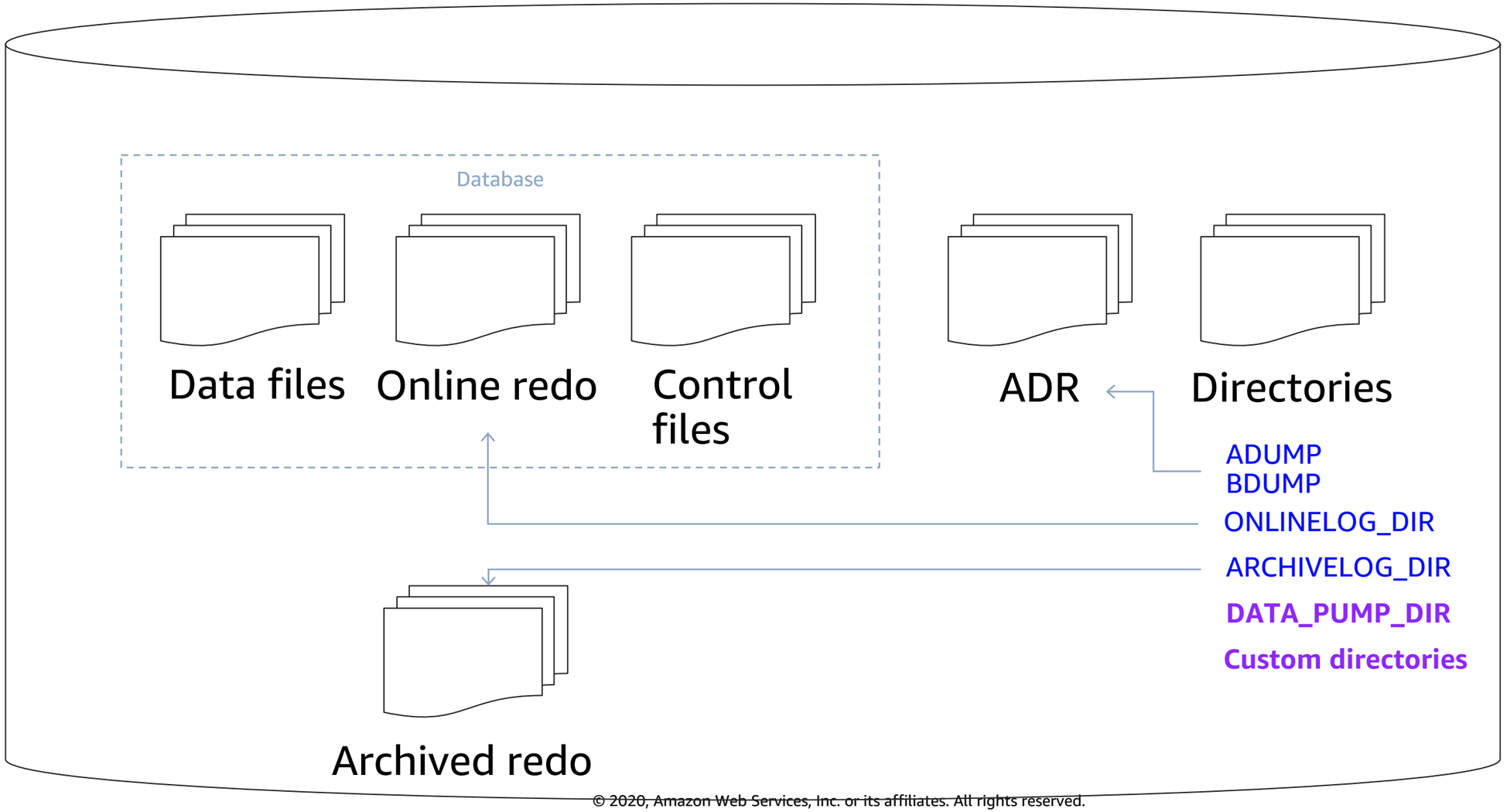
Oracle RDS

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)

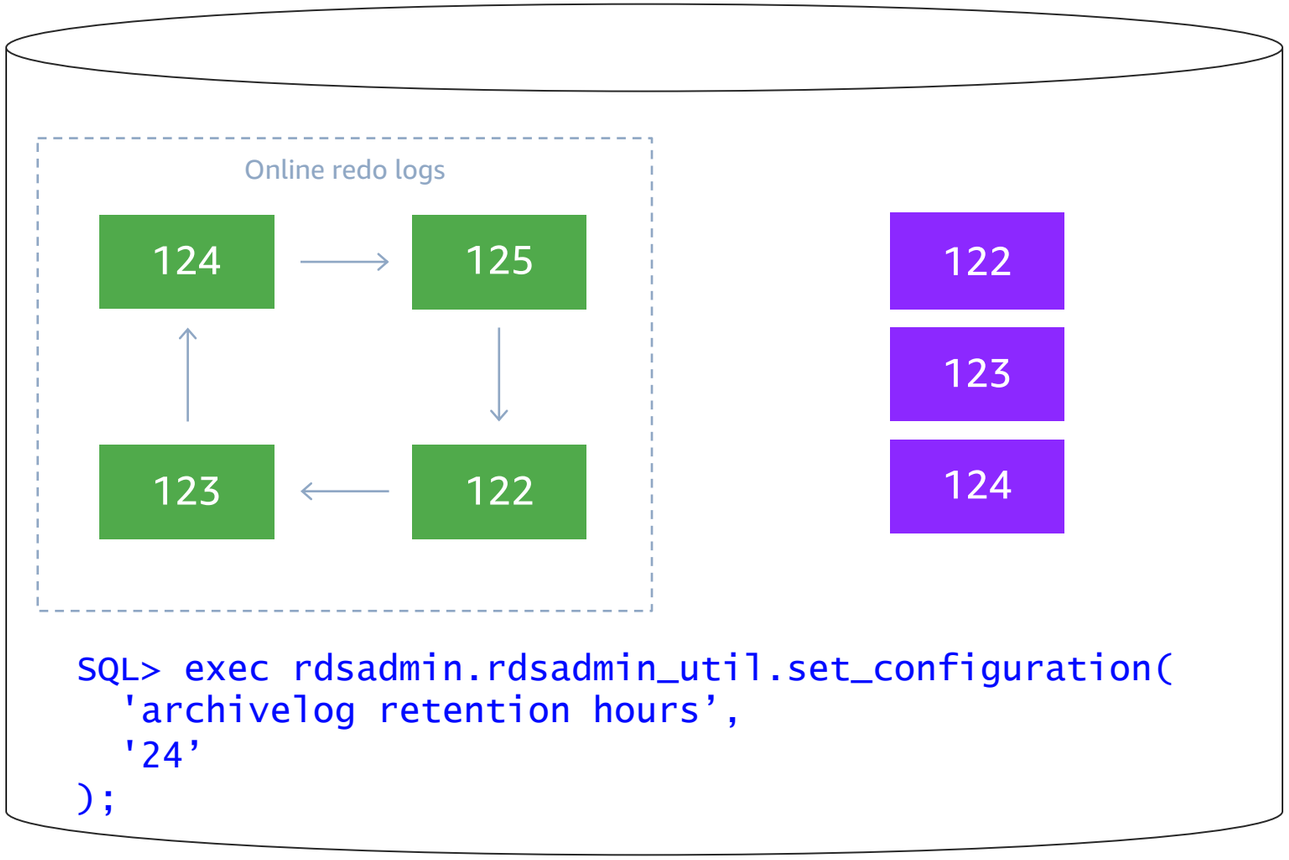
Amazon RDS storage



Multi-AZ Secondary

Amazon EBS Snapshot

Archived redo logs

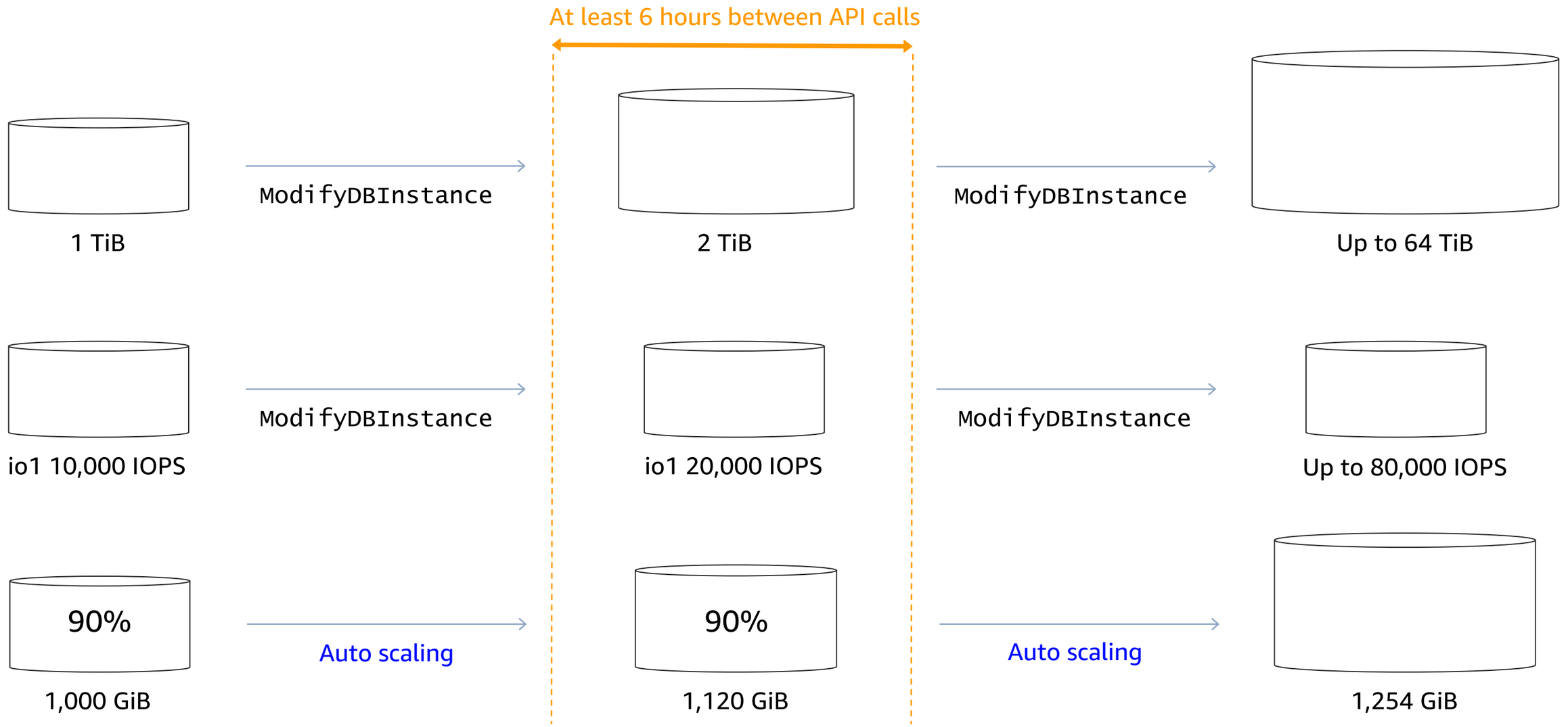



```
SQL> exec rdsadmin.rdsadmin_util.set_configuration(
  'archivelog retention hours',
  '24'
);
```

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

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

Amazon RDS storage scaling

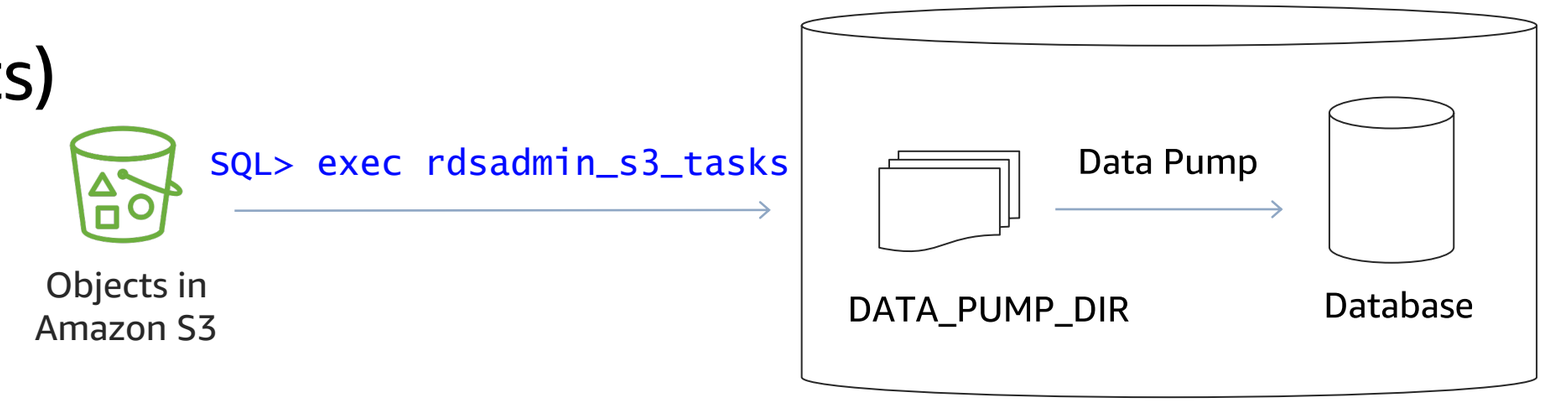


NEW! Amazon RDS storage auto scaling – greater of 5 GiB or 12% when less than 10% free storage for 5 minutes

Loading data

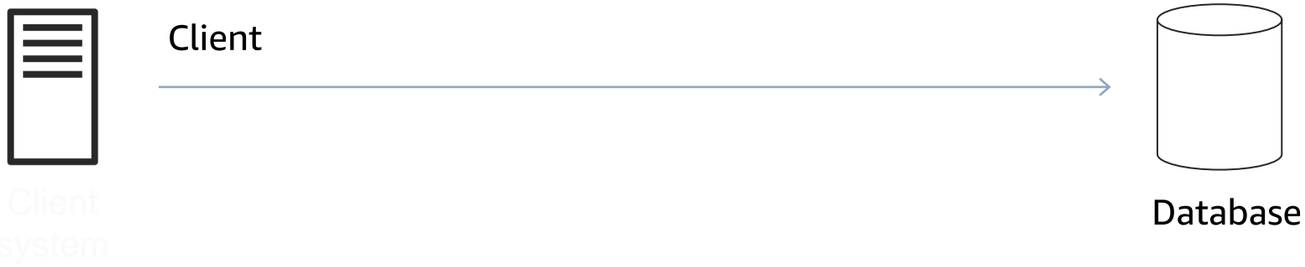
Local data (directory objects)

- Data Pump
- External tables



Remote data (client systems)

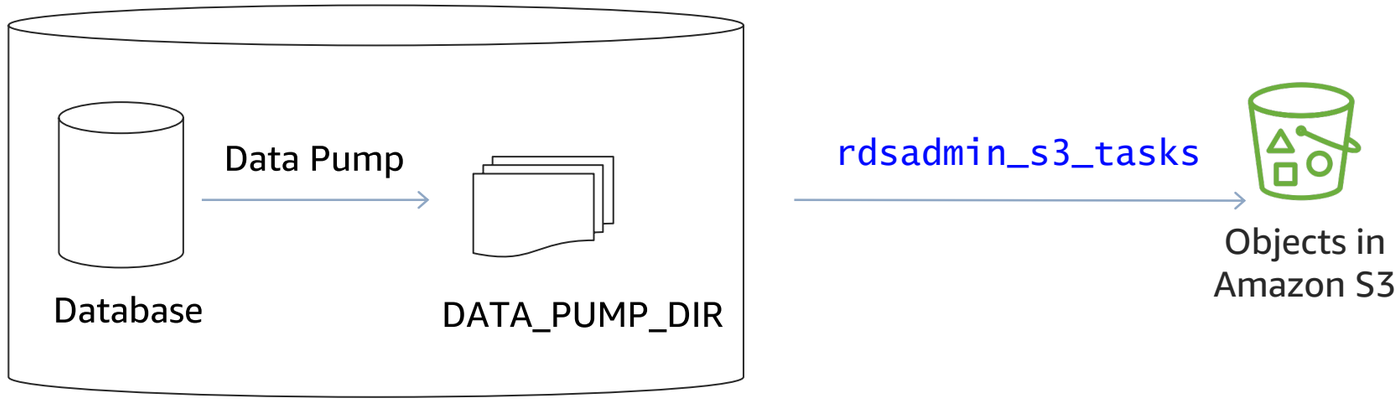
- Legacy imp utility
- Oracle SQL*Loader
- Oracle SQL Developer
- Client applications



Unloading data

Local data (directory objects)

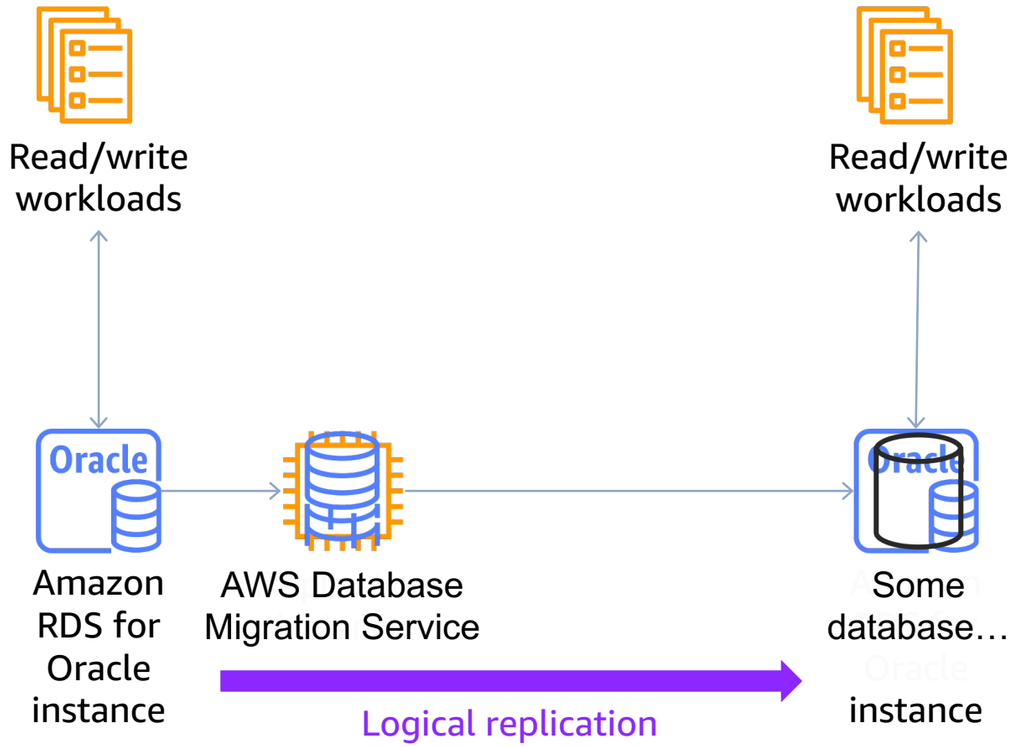
- Data Pump
- External tables
- RMAN



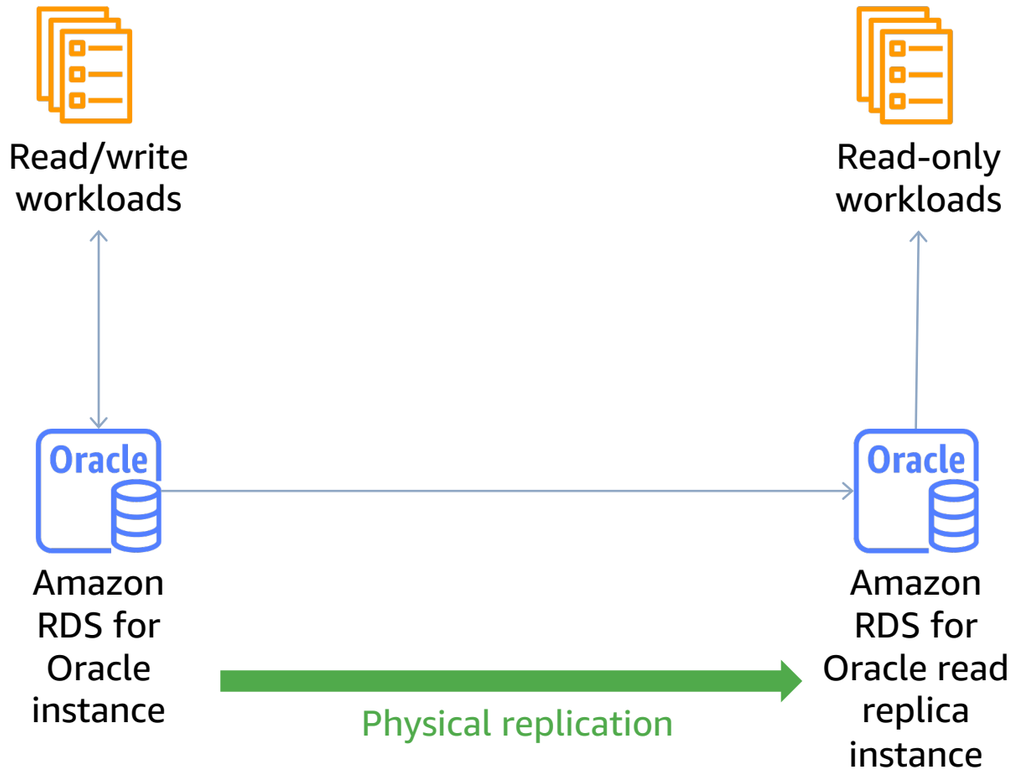
Remote data (client systems)

- Legacy exp utility
- Oracle SQL Developer
- Client applications

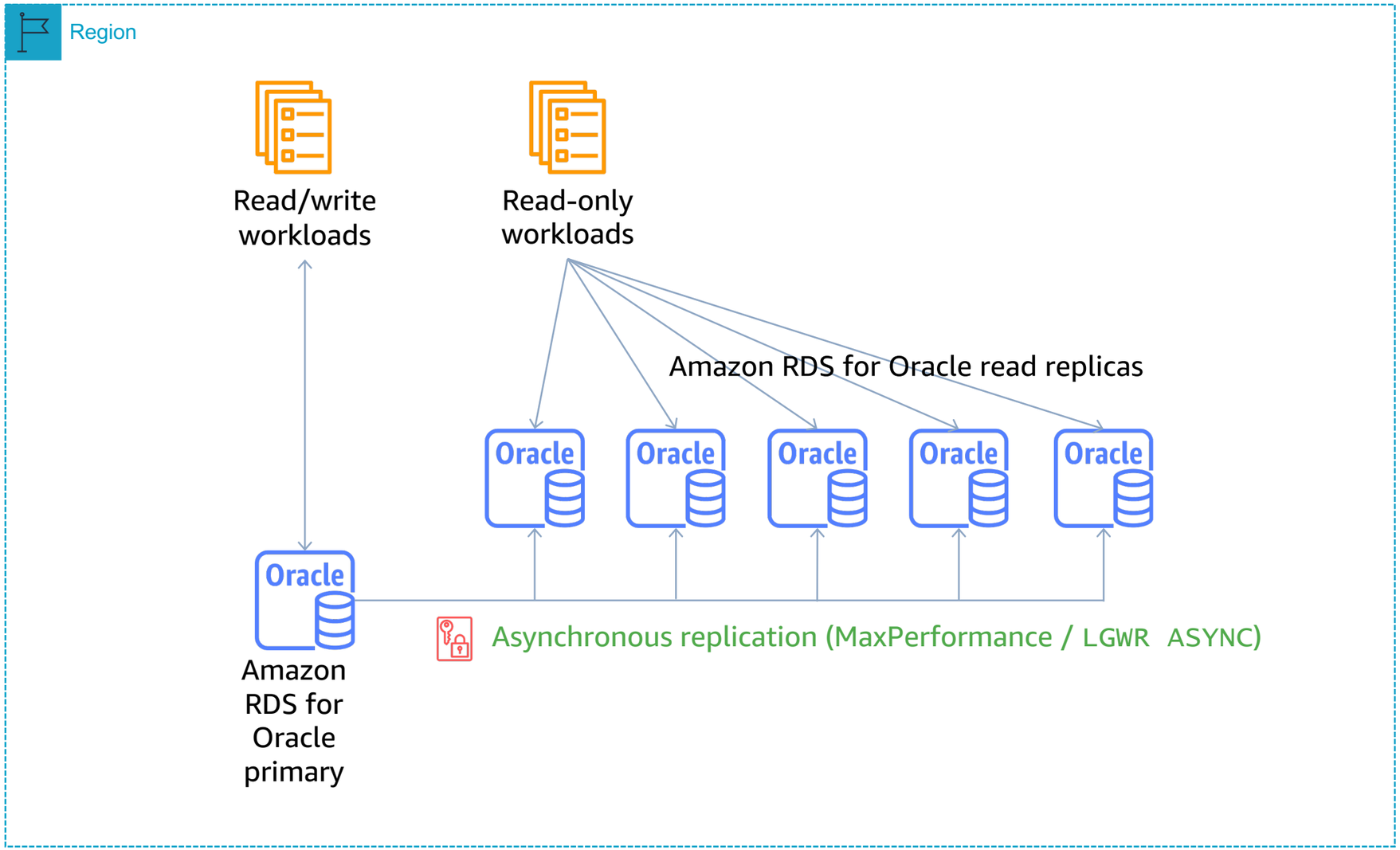
Amazon RDS for Oracle logical replication



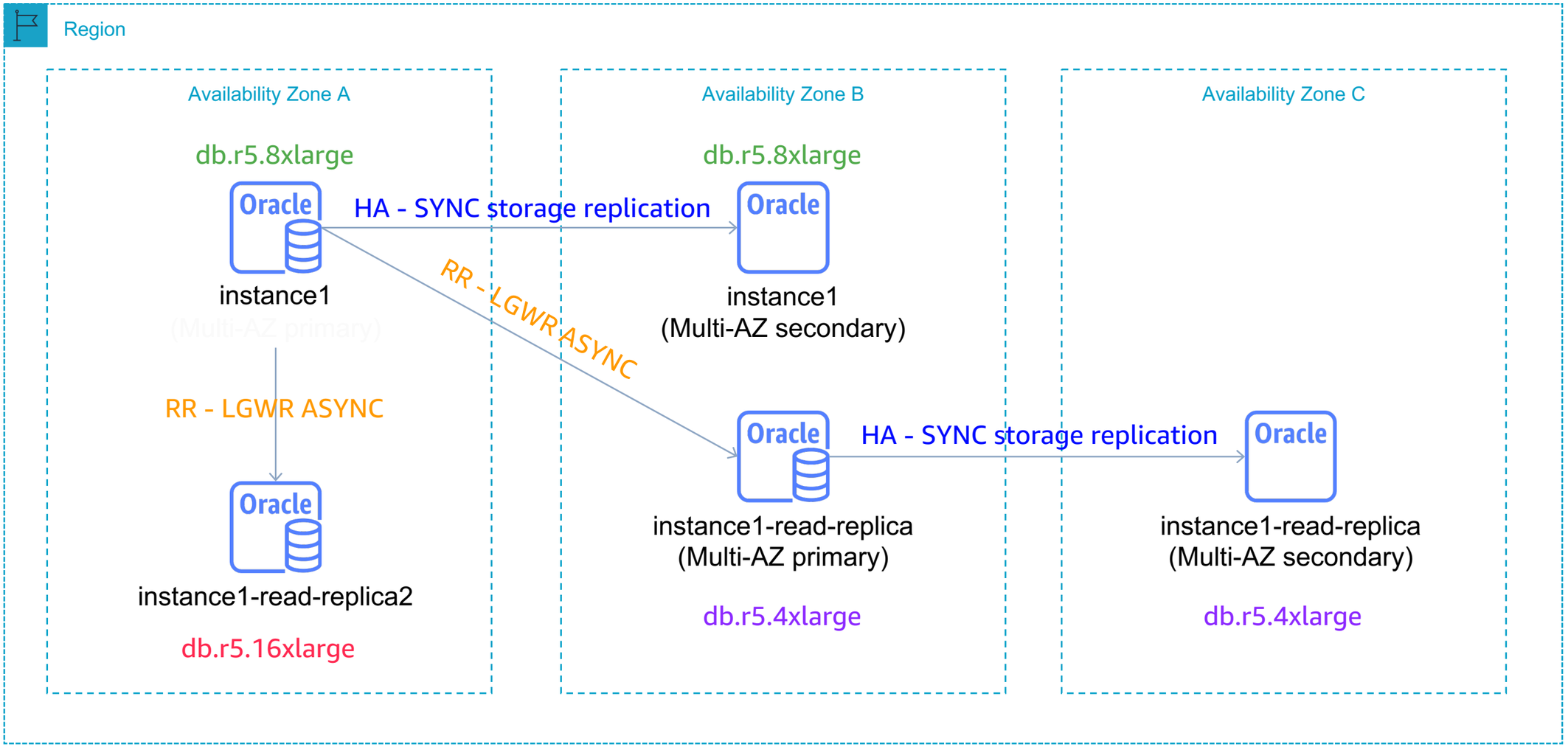
Amazon RDS for Oracle physical replication

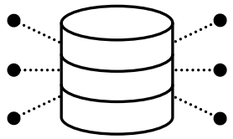


New! Amazon RDS for Oracle read replicas



Amazon RDS for Oracle read replicas





Sql Server RDS

Supported Versions

Versions

- 2012
- 2014
- 2016
- 2017

Editions

- Express
- Web
- Standard
- Enterprise

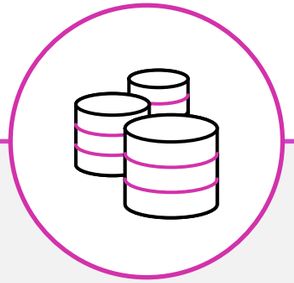
Amazon Aurora

Amazon Aurora

MySQL and PostgreSQL-compatible relational database built for the cloud

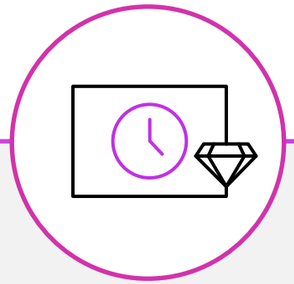
Performance and availability of commercial-grade databases at 1/10th the cost

Performance and scalability



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

Availability and durability

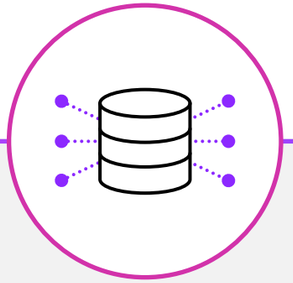


Fault-tolerant, self-healing storage; six copies of data across three Availability Zones; continuous backup to Amazon S3

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

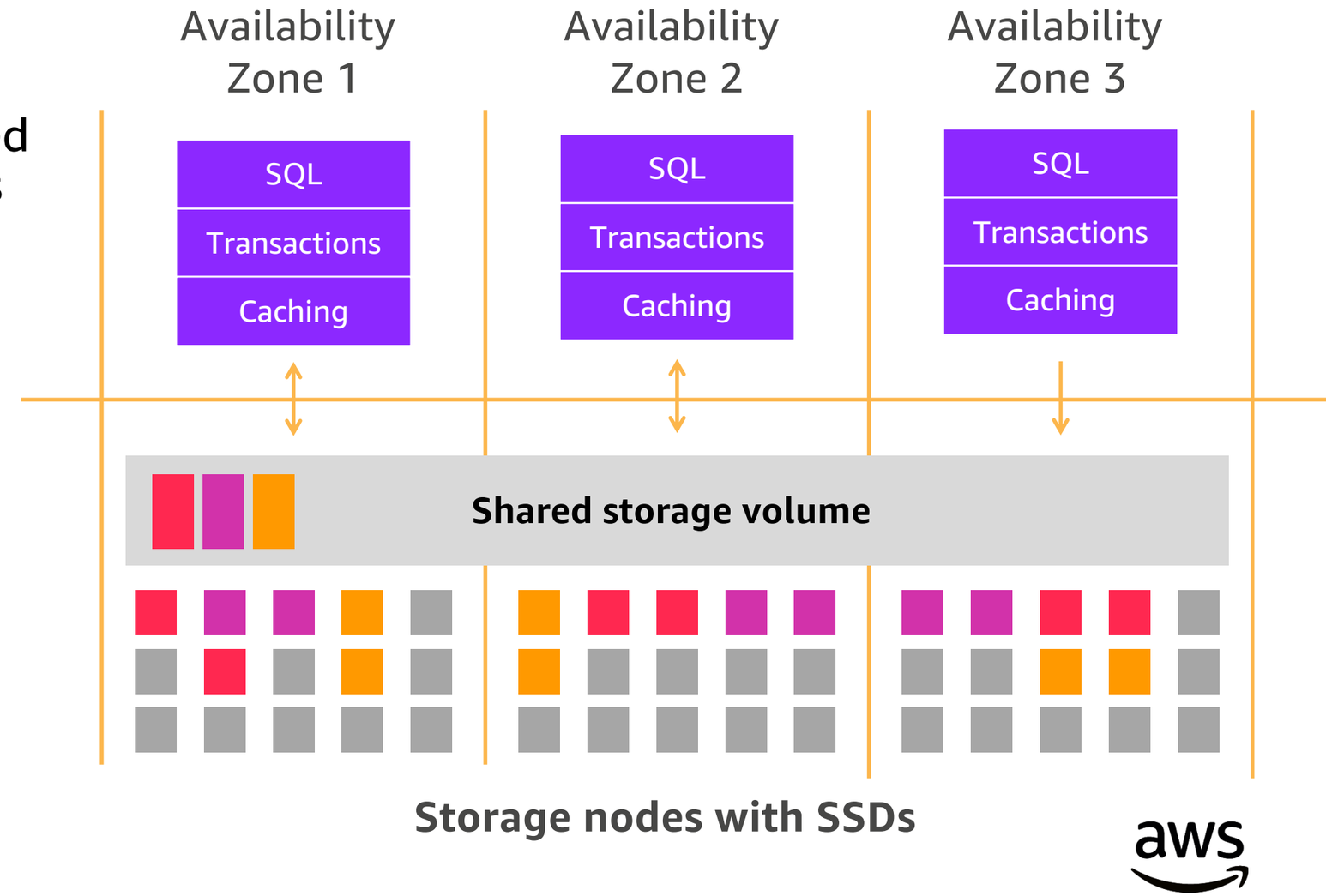
The compute and storage layers are separated in a scale-out, distributed, and multi-tenant architecture

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

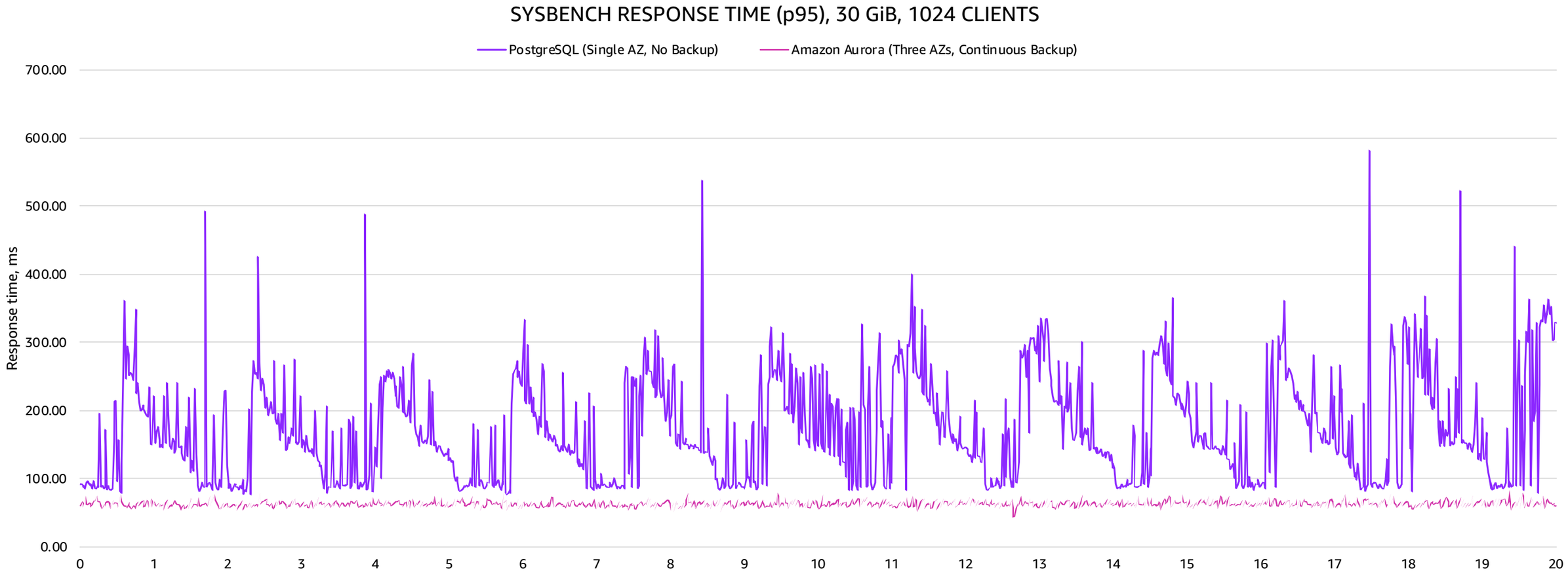
NETWORK-ATTACHED STORAGE IS ALL ABOUT PACKETS/SECOND

HIGH-THROUGHPUT PROCESSING NEEDS CPU AND MEMORY OPTIMIZATIONS

aws

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

aws

verizon

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.

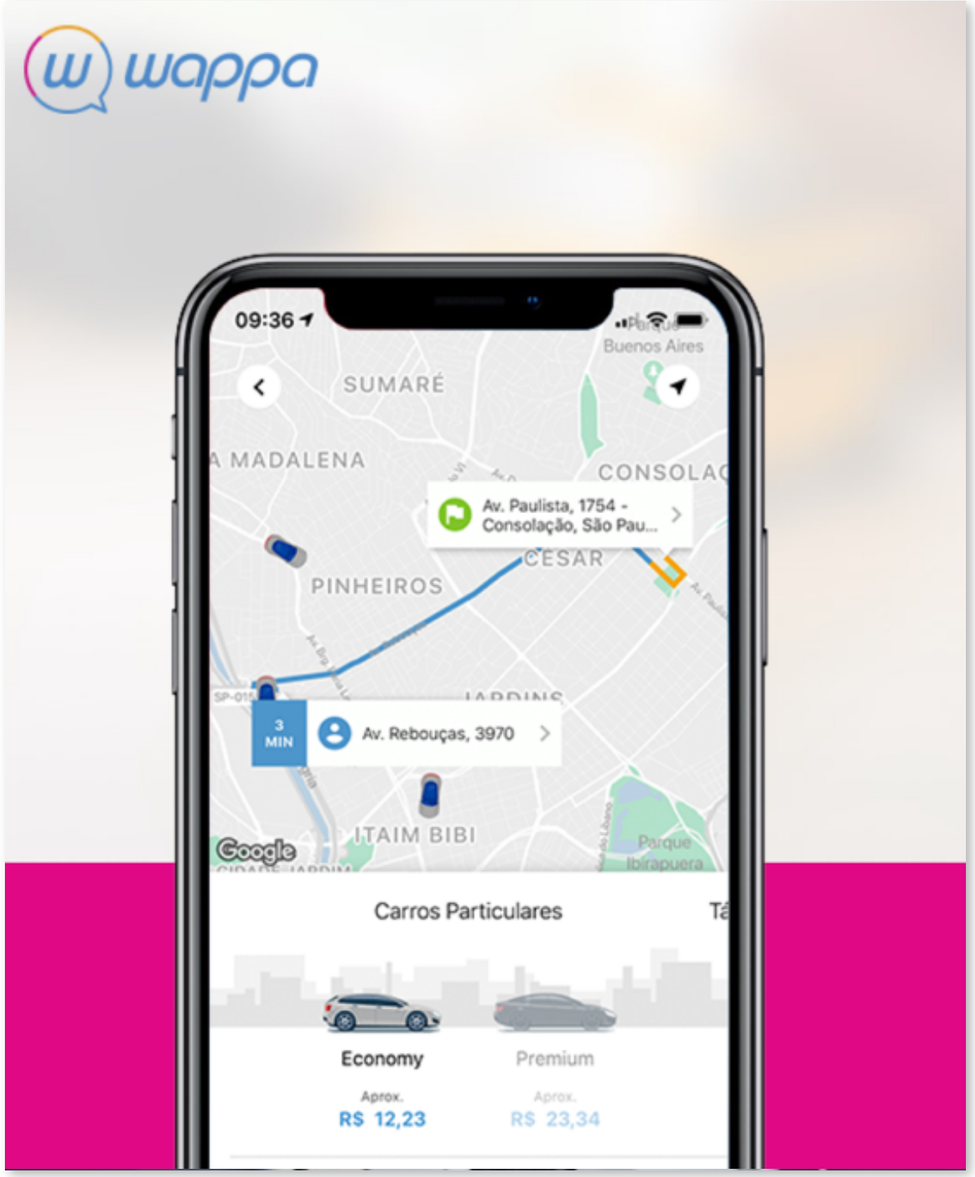
Solution

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

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:

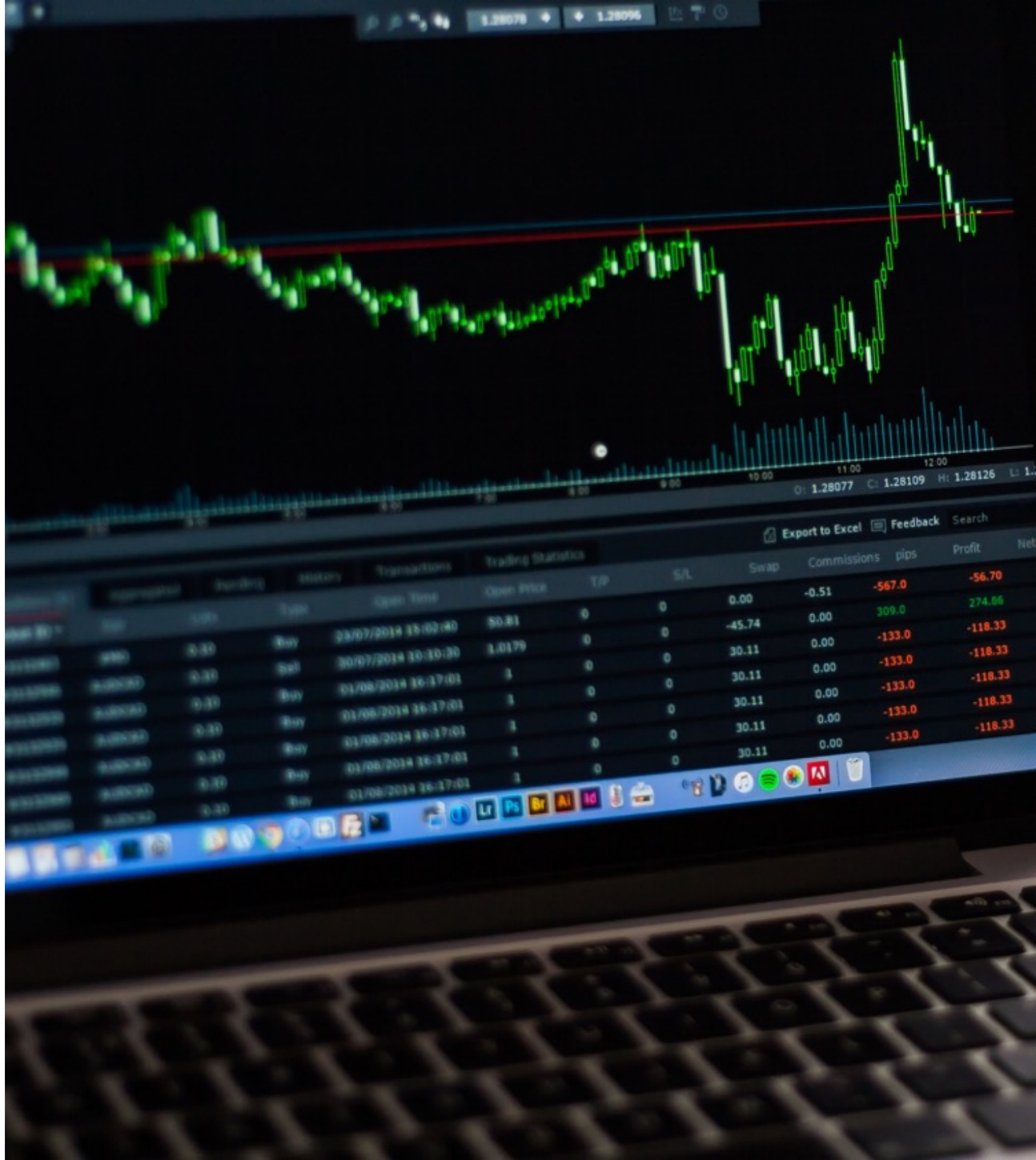
Running on Oracle posed challenges for TRMS, including complicated database administration, latency levels, and lengthy server provisioning work.

Solution:

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

Result:

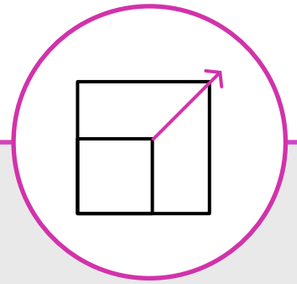
- Efficiency: Reduced administrative overhead by 70%.
- Cost: cut cost by half.
- Performance: scaled up to 900 transactions per second per shard now.



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

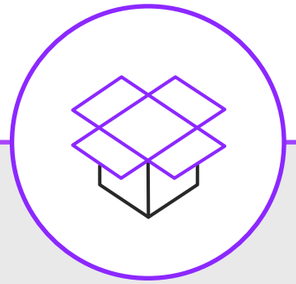
Serverless

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

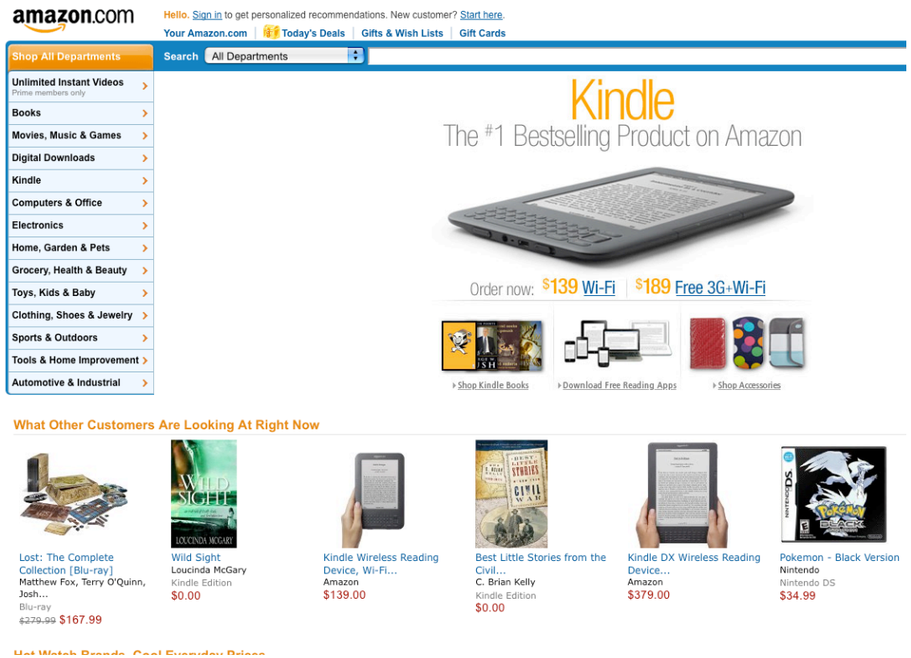
Global database for global users and apps



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

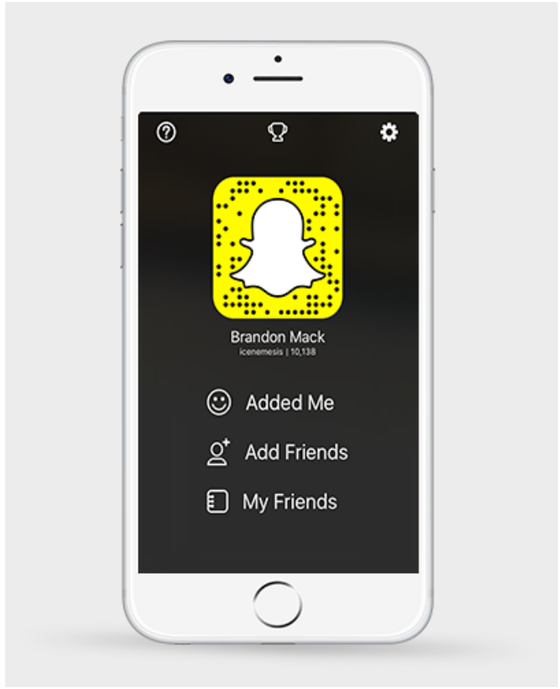
DynamoDB powers the world's largest applications

amazon



>20M requests per second

snapchat



>150M active users

SAMSUNG

Data >1 PB

aws

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

CHALLENGE

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

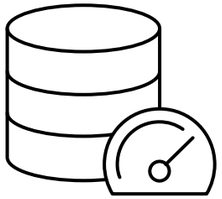
SOLUTION

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.

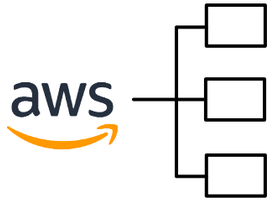
Amazon ElastiCache

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



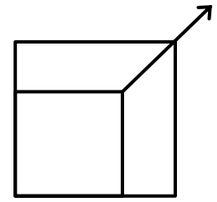
Extreme Performance

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

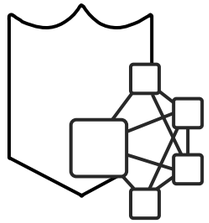
VPC for cluster isolation, encryption at rest/transit, HIPAA compliance

*: https://db-engines.com/en/ranking

ElastiCache Memcached

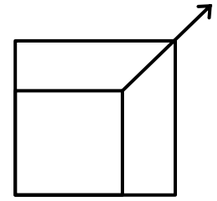
Fully Managed Memcached

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

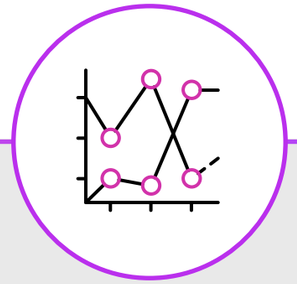
1,000x faster and 1/10th the cost of relational databases

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



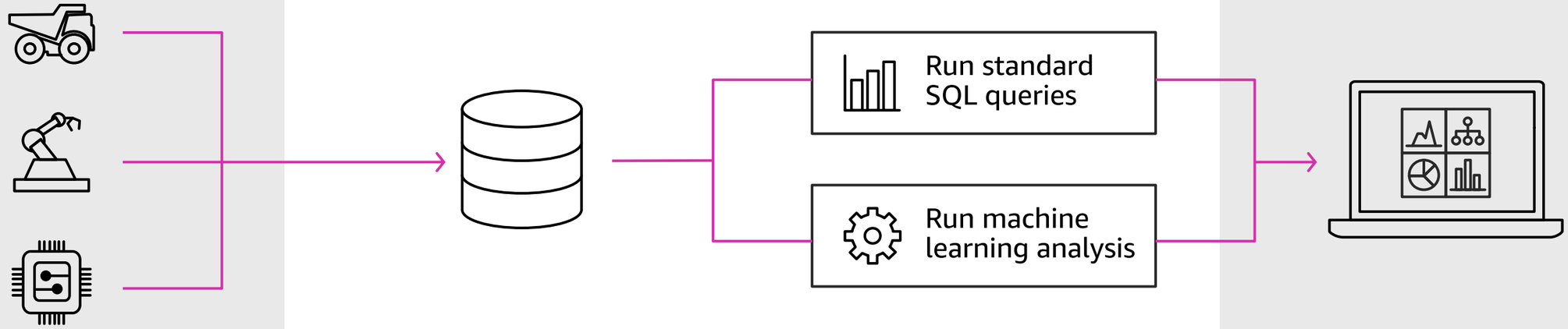
Built-in functions for interpolation, smoothing, and approximation

Serverless

Automated setup, configuration, server provisioning, software patching

Amazon Timestream: How it works

Time-series application



Collect

Collect device data in a variety of formats and frequencies

Store

Store and processes time-series data at scale with Amazon Pyramid

Analyze

Run SQL-like queries with pre-built functions or use machine learning to identify trends, patterns, and anomalies

Build

Build DevOps, IoT, and industrial applications with time-series data management and analytics

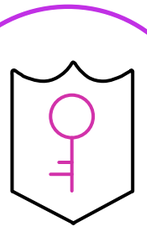
Amazon Quantum Ledger Database

Fully managed ledger database
Track and verify history of all changes made to your application's data

Immutable

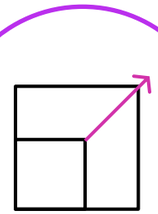
Maintains a sequenced record of all changes to your data, which cannot be deleted or modified; you have the ability to query and analyze the full history

Cryptographically verifiable



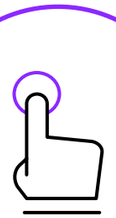
Uses cryptography to generate a secure output file of your data's history

Highly scalable



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

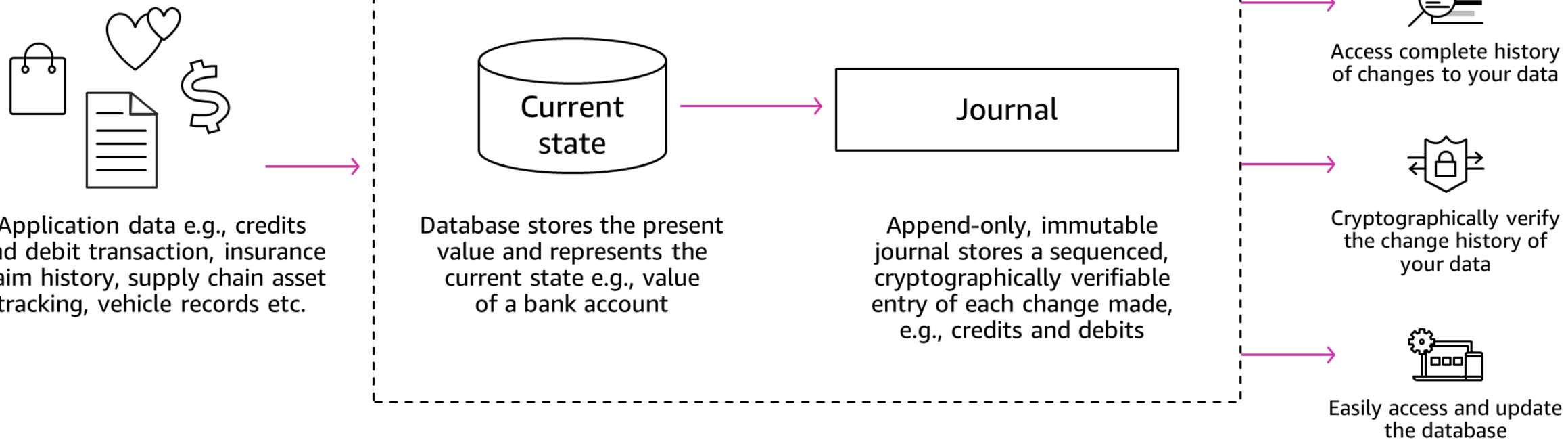
Easy to use



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

Amazon QLDB: How it works

Centralized ledgers



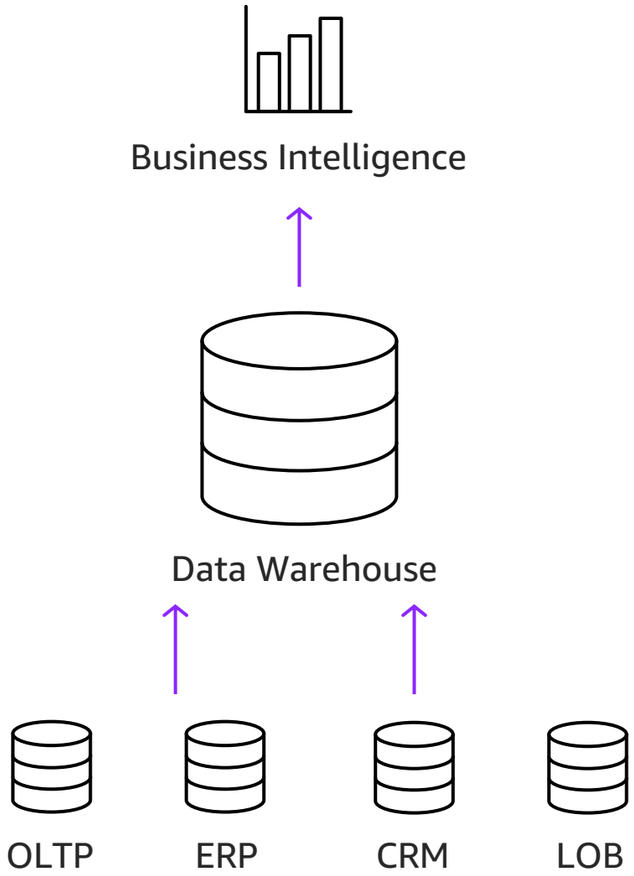
Three type of projects

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

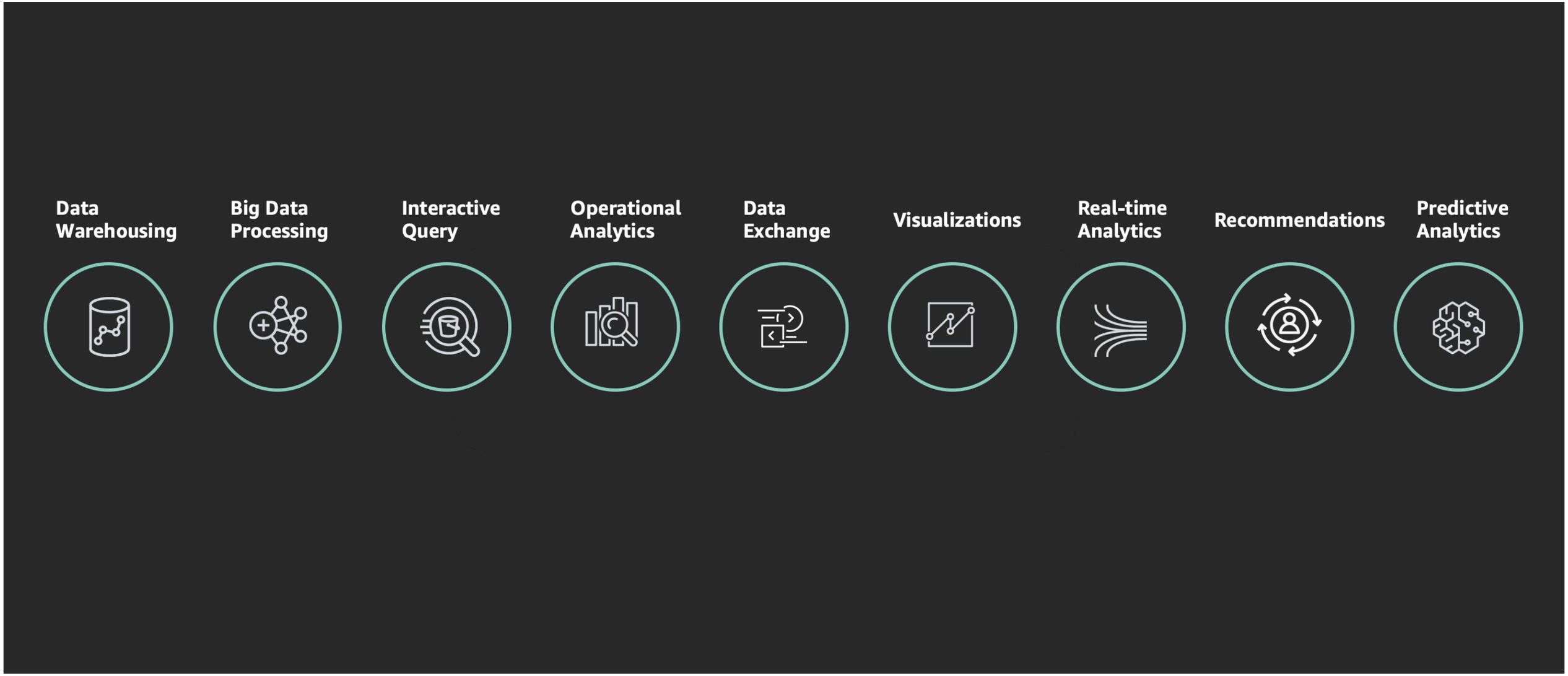
Three key trends are transforming the world of Analytics

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**

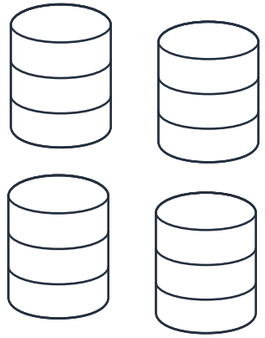
Analytics Use Cases across Enterprises are Growing



Data infrastructures needs new Architecture to meet new demands

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

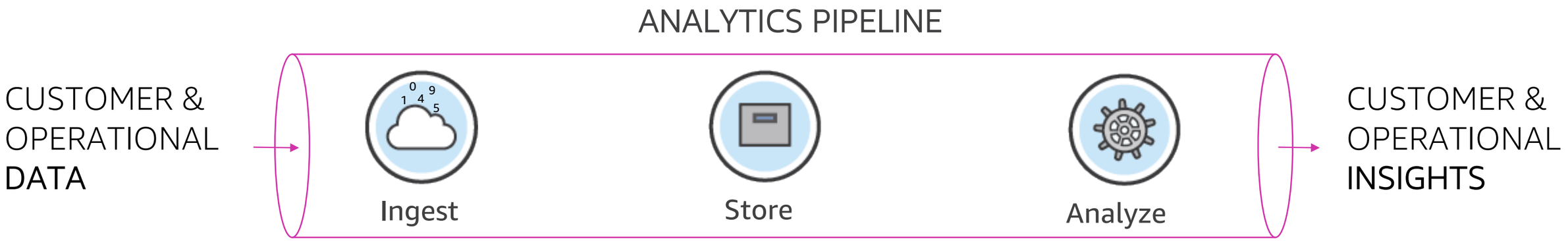
Rigid data schemas prevent access to source data and limit the use of advanced analytics and machine learning



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

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

On-premise Big Data still challenges most enterprises



Custom pipelines

Delays on-boarding of new data

- Proliferation of tools
- Custom-built schema Management is hard

Data doesn't "connect"

Siloed data prevents single source of truth

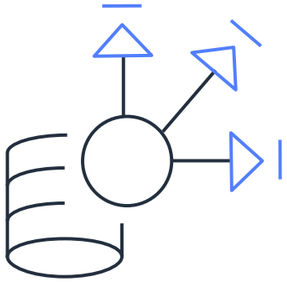
- Data needs to move for cross-domain analysis
- Partial cataloging hampers data availability/transparency

Limited analytics platforms

Inability to generate new types of insights

- Separate analytics environments promotes data movement and delays

Data lakes can accelerate Data Analytics



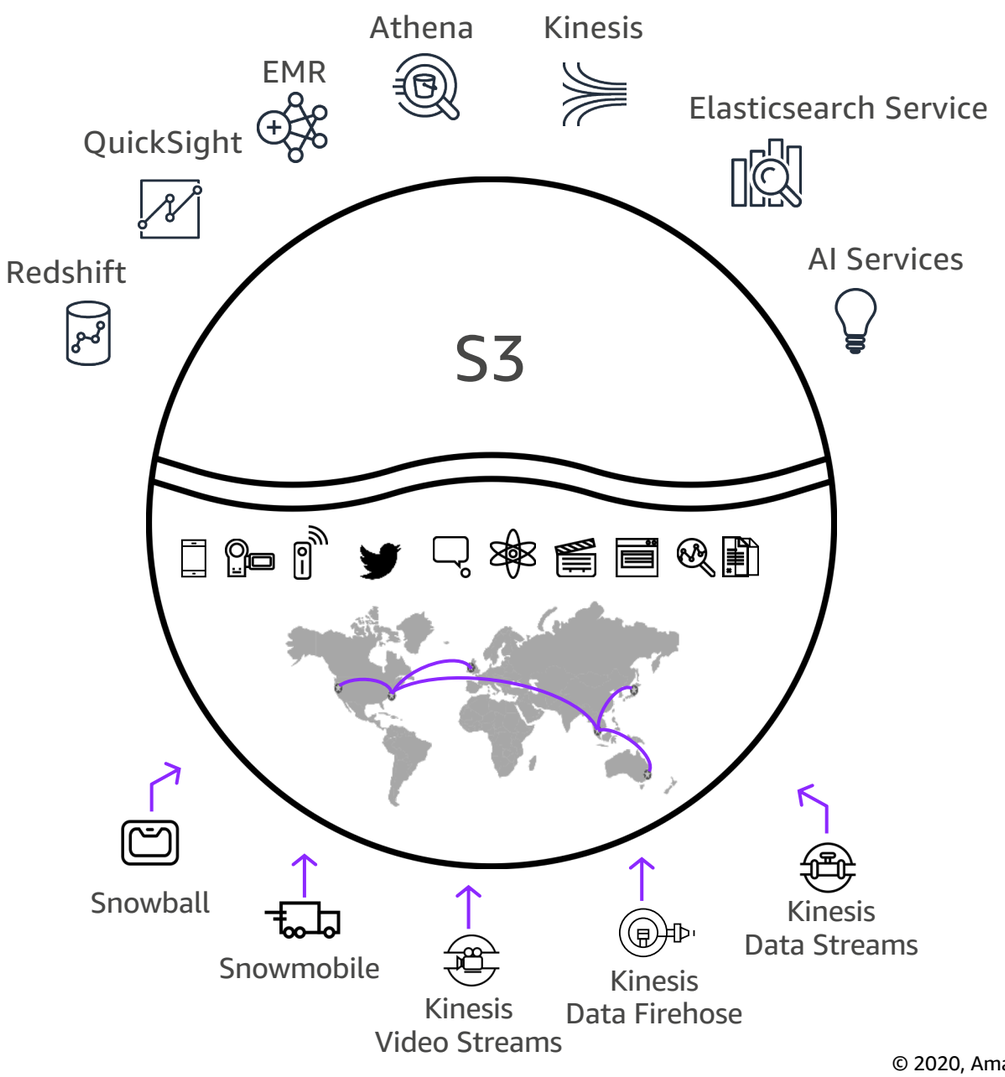
Centralized repository that allows structured and unstructured data to be **stored at any scale**

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

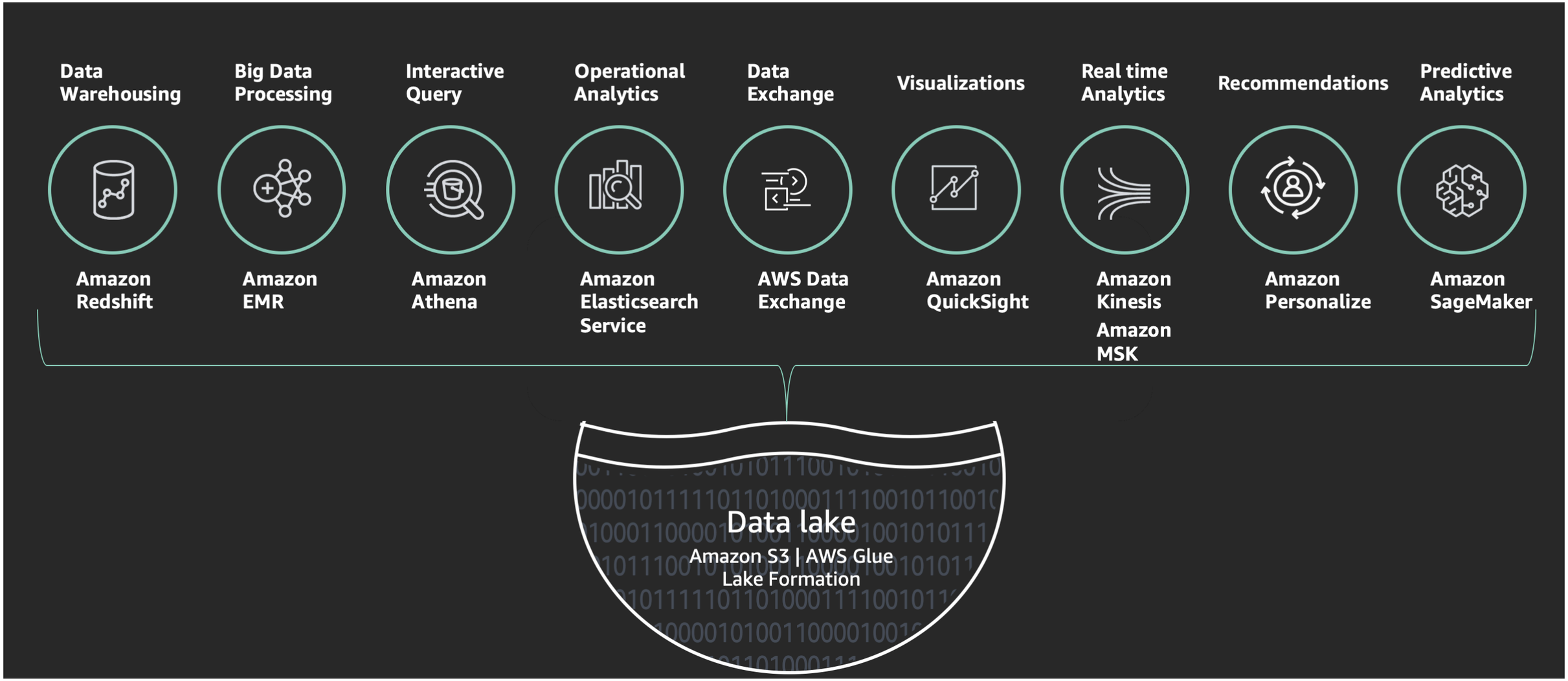
Cost effective

- Store at 2.3 cents per GB-month in Amazon S3
- Query with Amazon Athena at ½ cent per GB scanned
- DW with Amazon Redshift for $1,000/TB/year

Give access to everyone

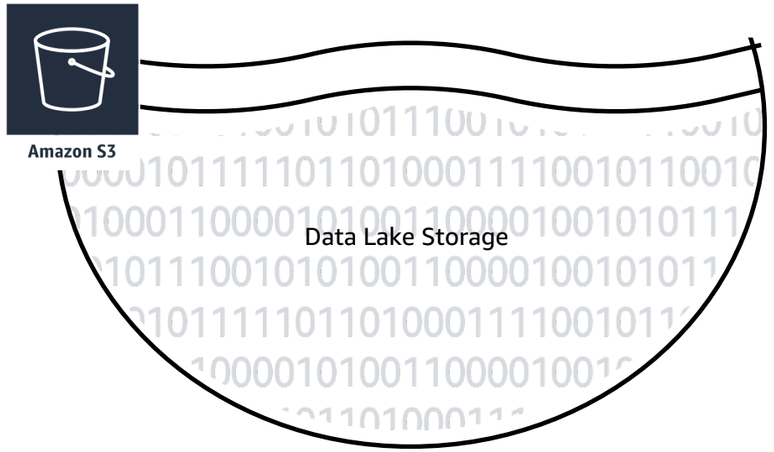
- Amazon QuickSight: $0.30 for 30 minutes of use

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

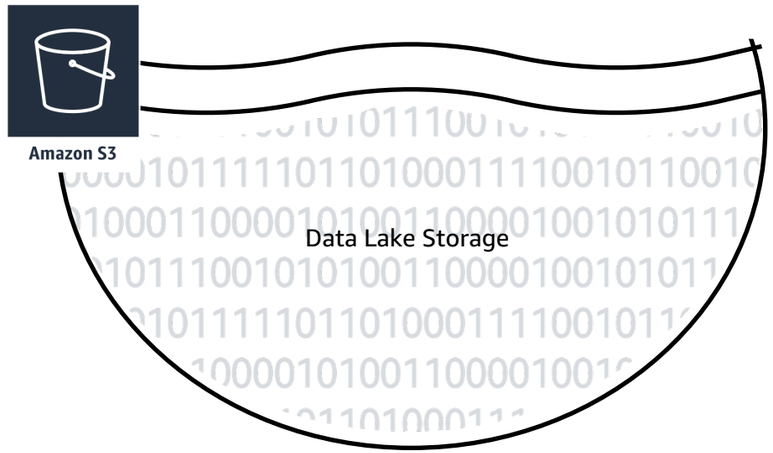
Store data in the format you want:

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 your datacenters

AWS Direct Connect
AWS Snowball
AWS Snowmobile
AWS Database Migration Service
AWS Storage Gateway

Data movement from real-time sources

AWS IoT Core
Amazon Kinesis Data Firehose
Amazon Kinesis Data Streams
Amazon Kinesis Video Streams

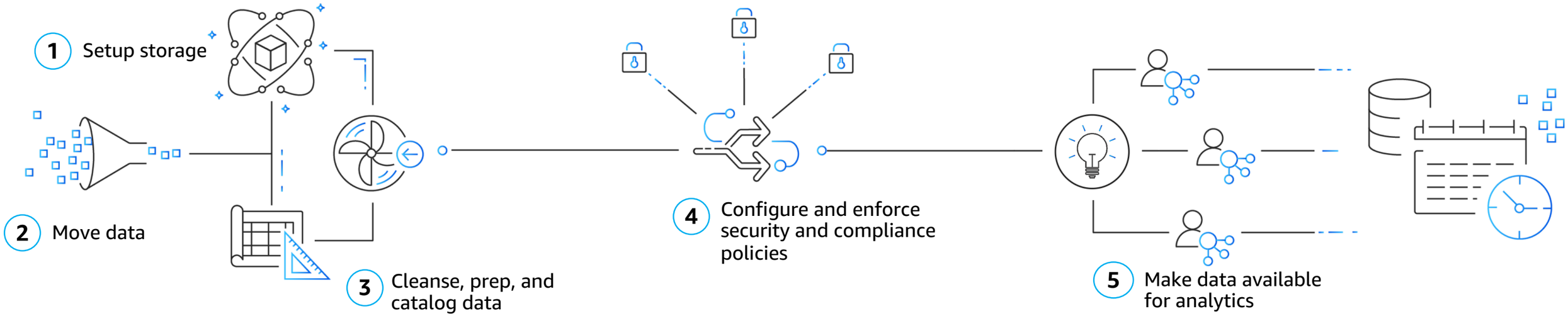
Data movement from on-premises datacenters

- 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

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

Building data lakes can still take months

Build Data Lake in Days using Lake Formation

AWS Lake Formation

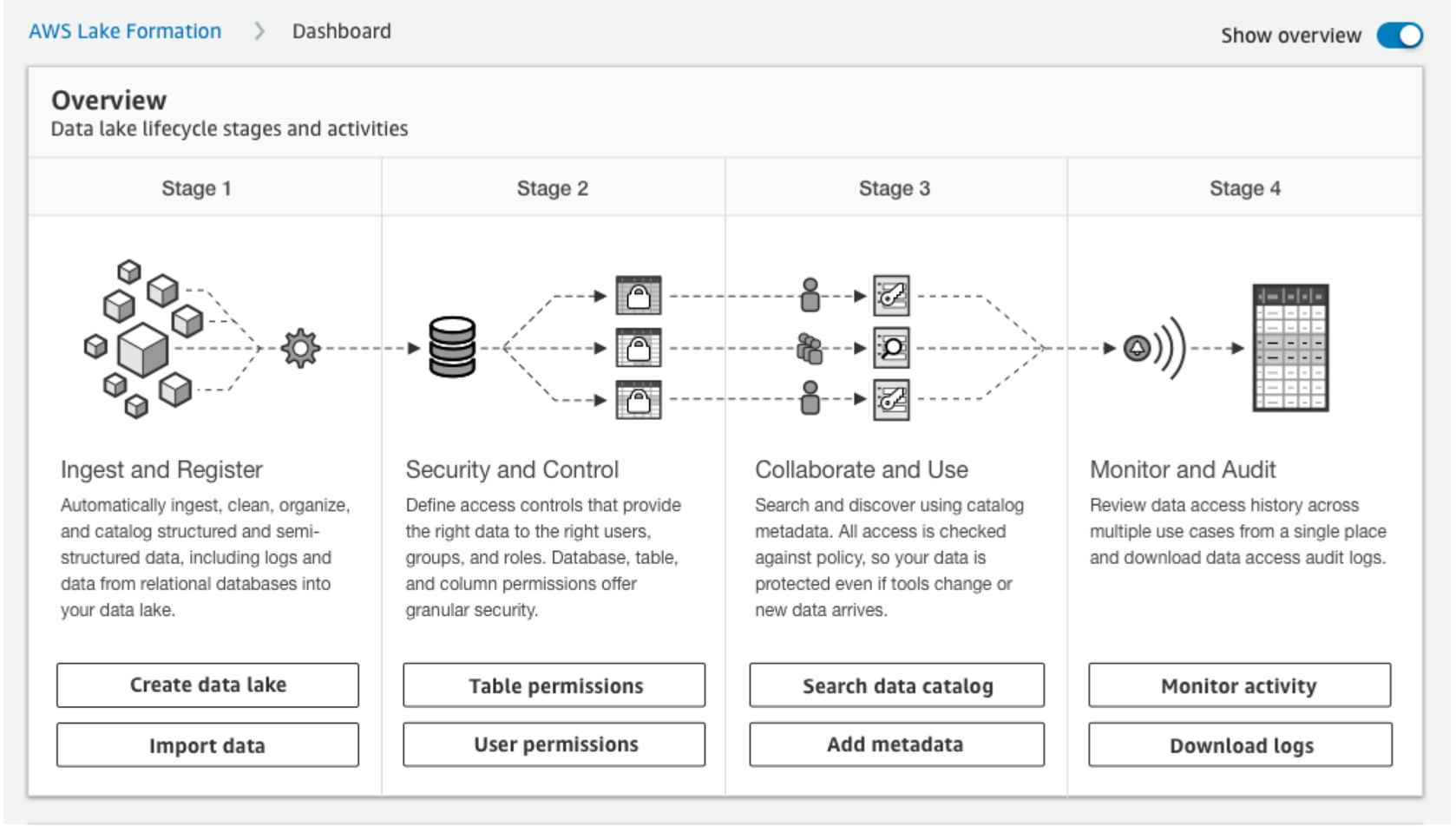
Build a secure data lake in days

Identify, ingest, clean, and transform data

Enforce security policies across multiple services

Gain and manage new insights

All Steps to Build Data Lake streamlined



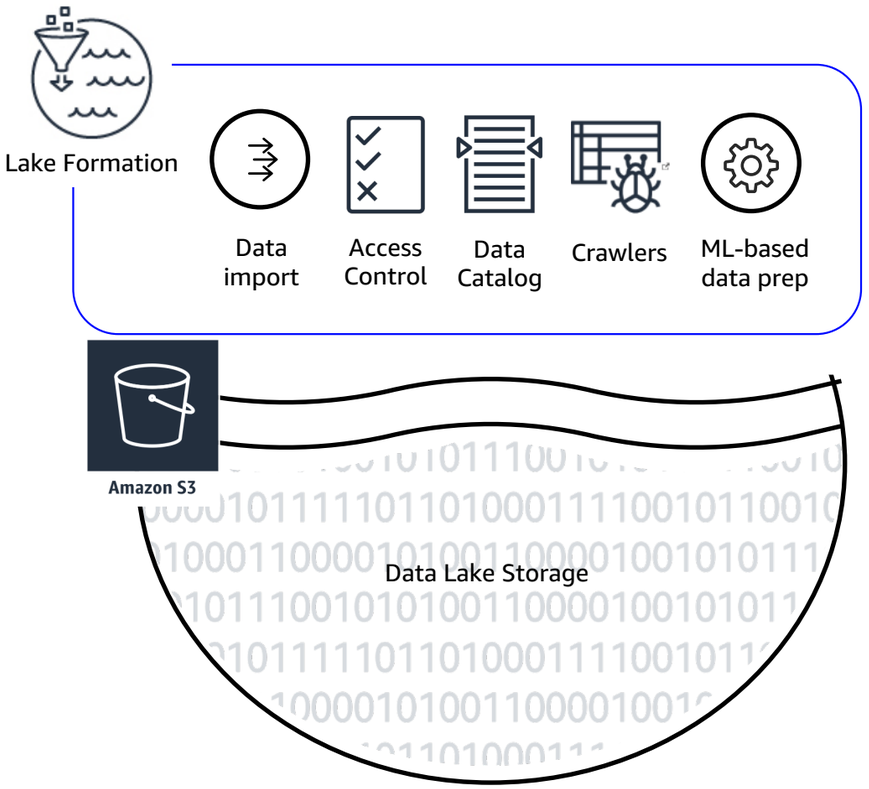
Register existing data or import new

Amazon S3 forms the storage layer for Lake Formation

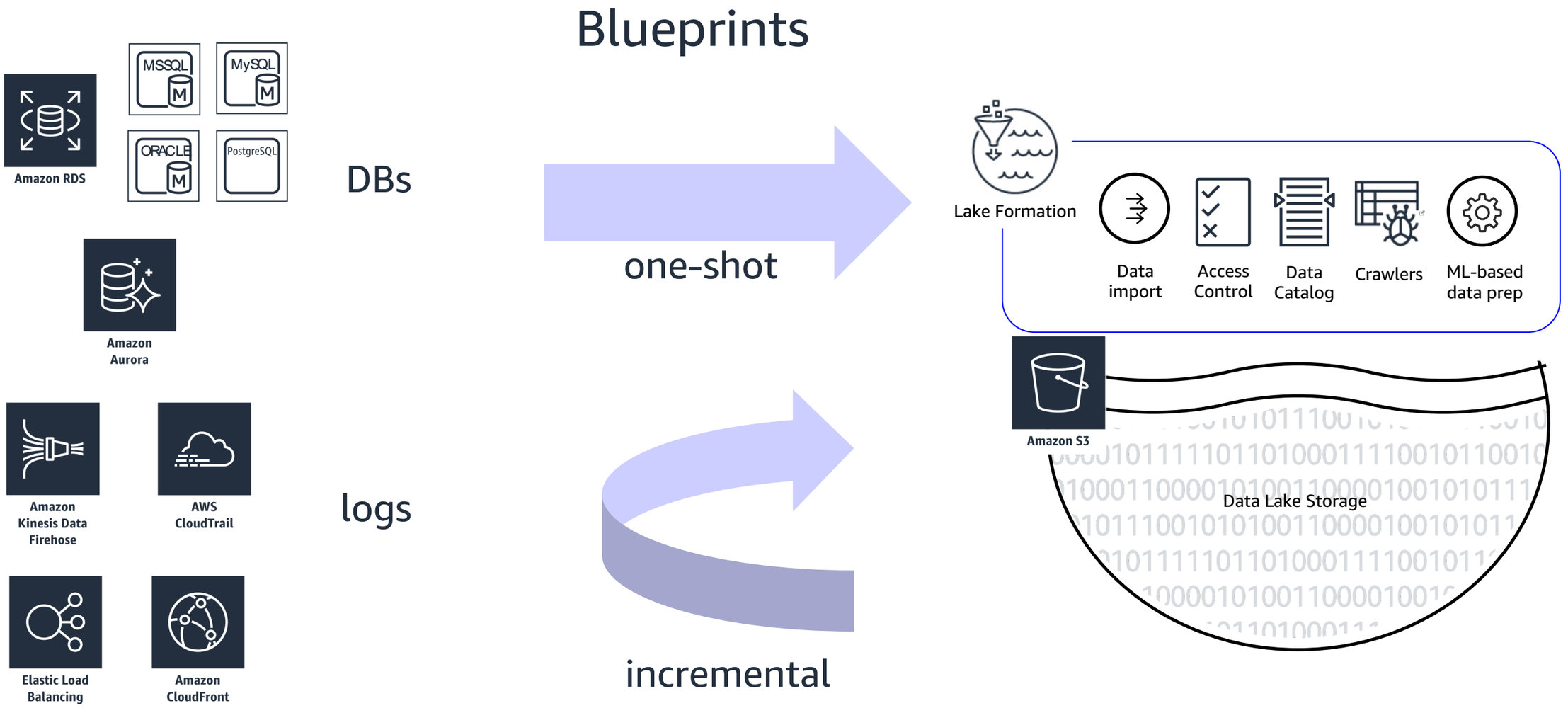
Register existing S3 buckets that contain your data

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



With blueprints

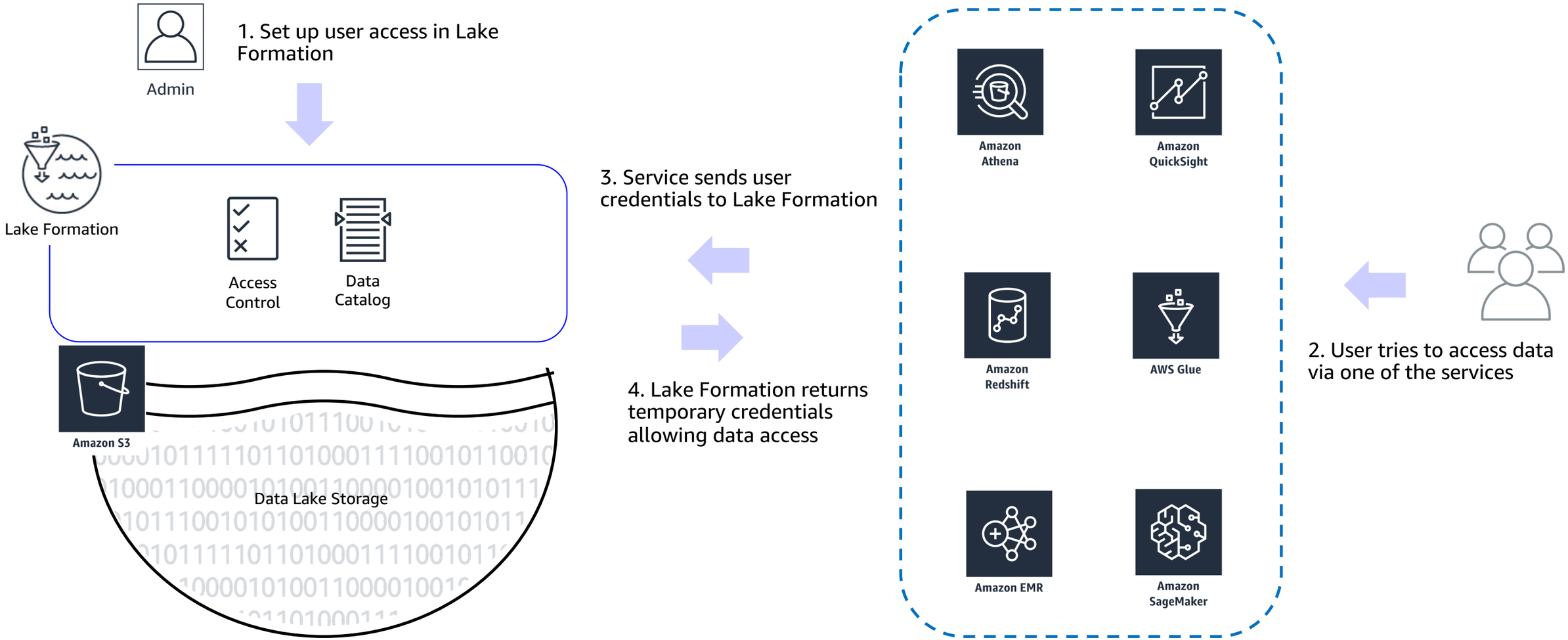
You

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



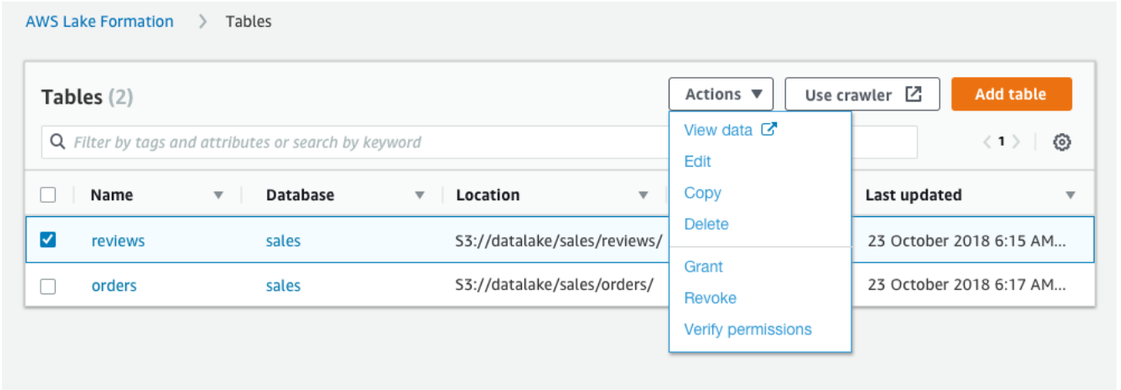
Security permissions in Lake Formation

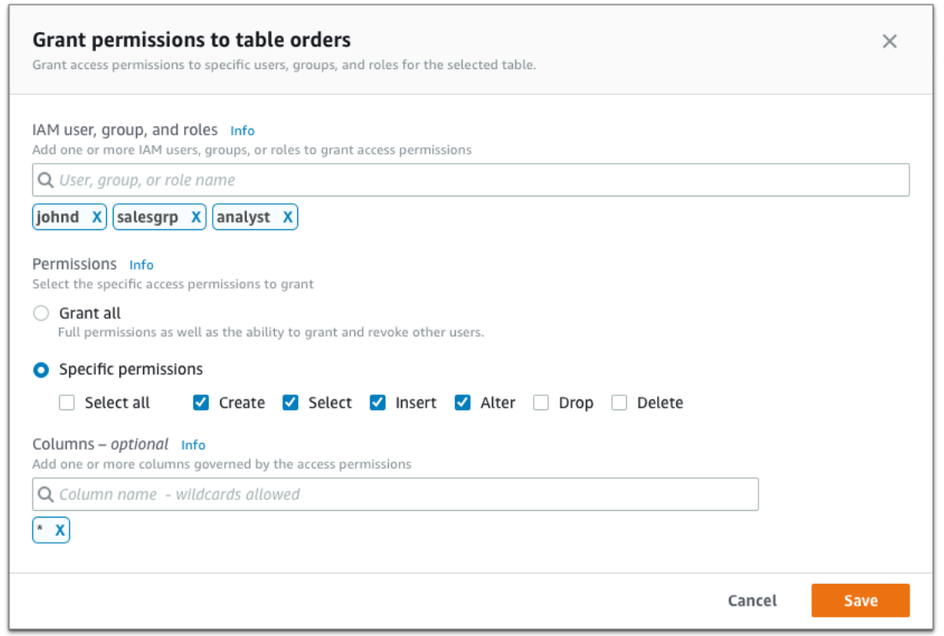
Control data access with simple grant and revoke permissions

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



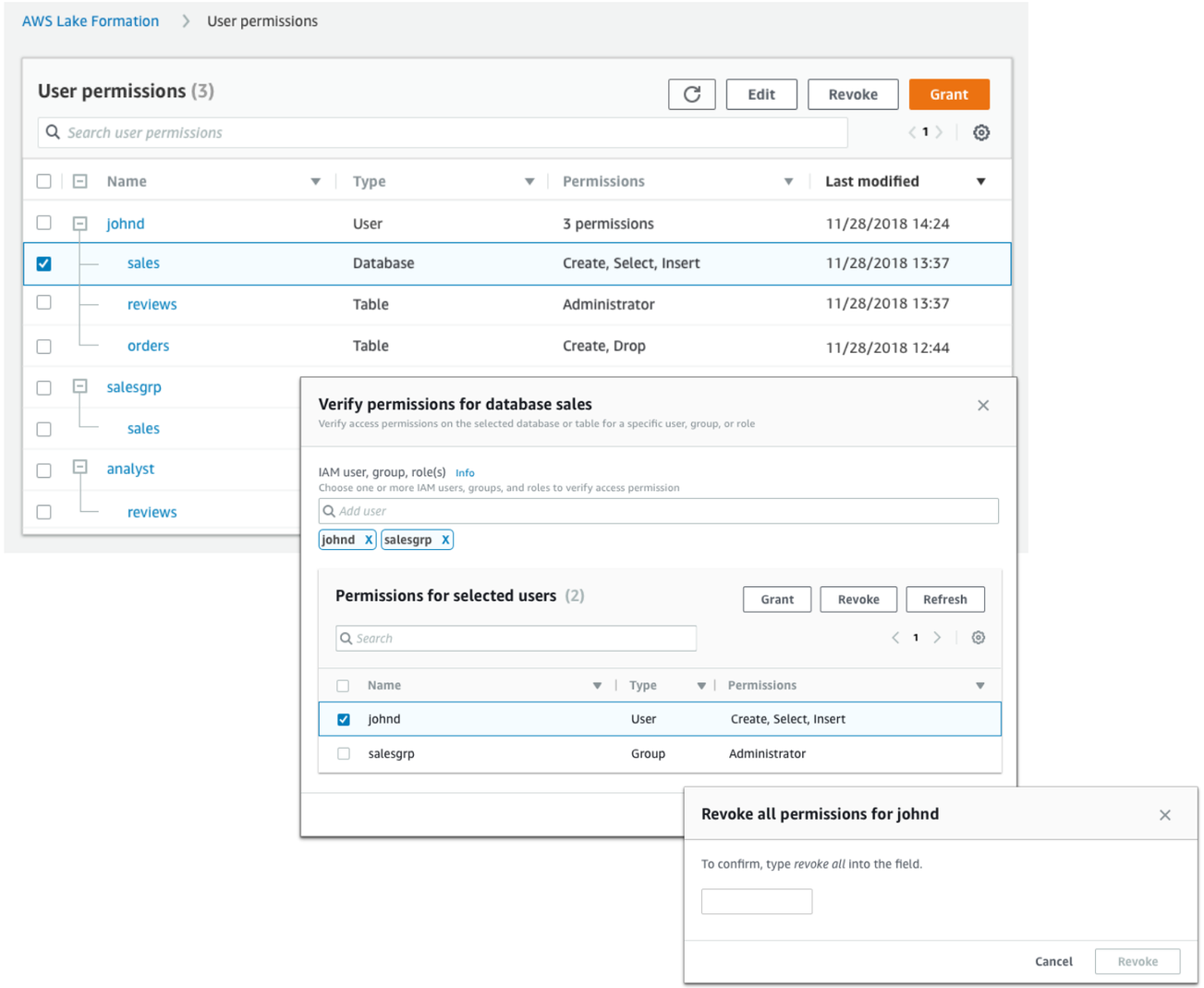


Security permissions in Lake Formation

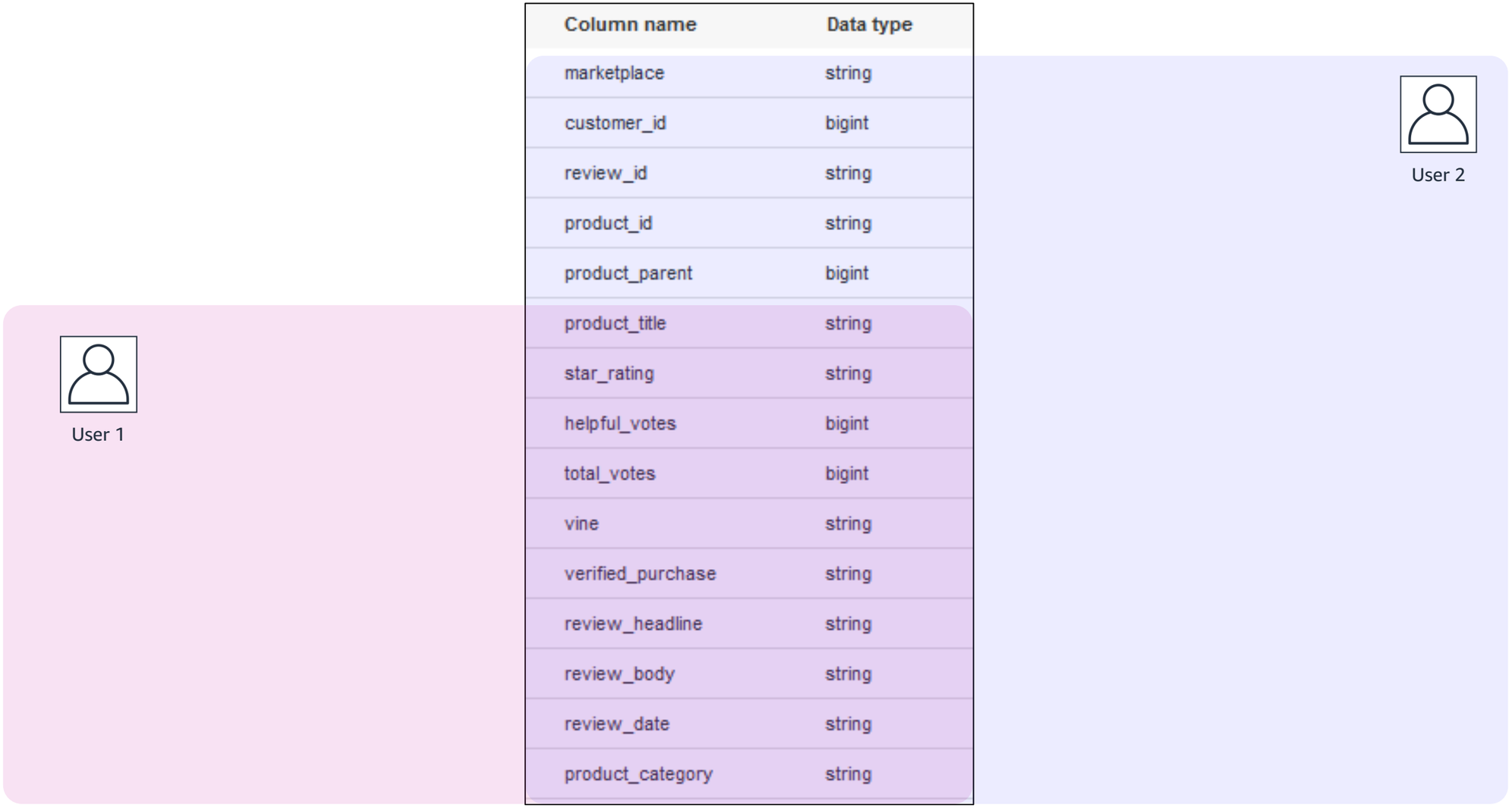
Search and view permissions granted to a user, role, or group in one place

Verify permissions granted to a user

Easily revoke policies for a user



Grant table and column-level permissions

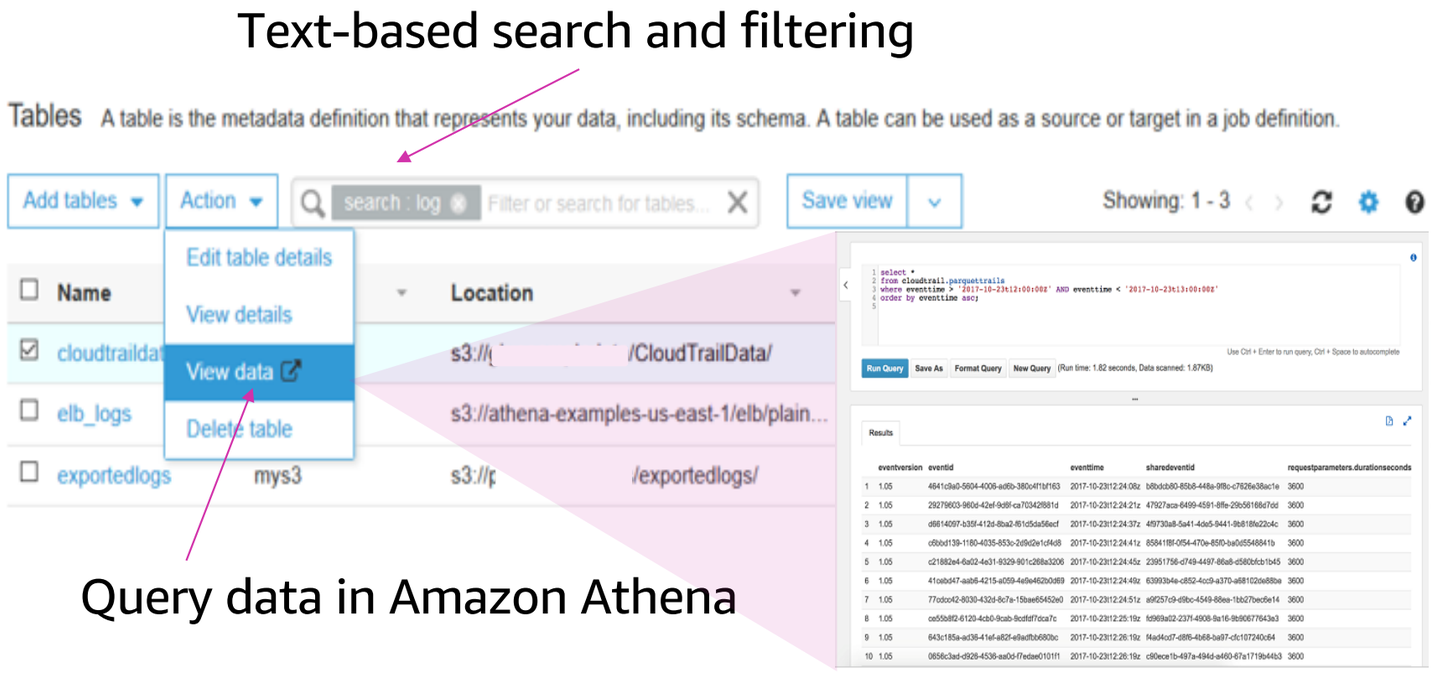


Search and collaborate across multiple users

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

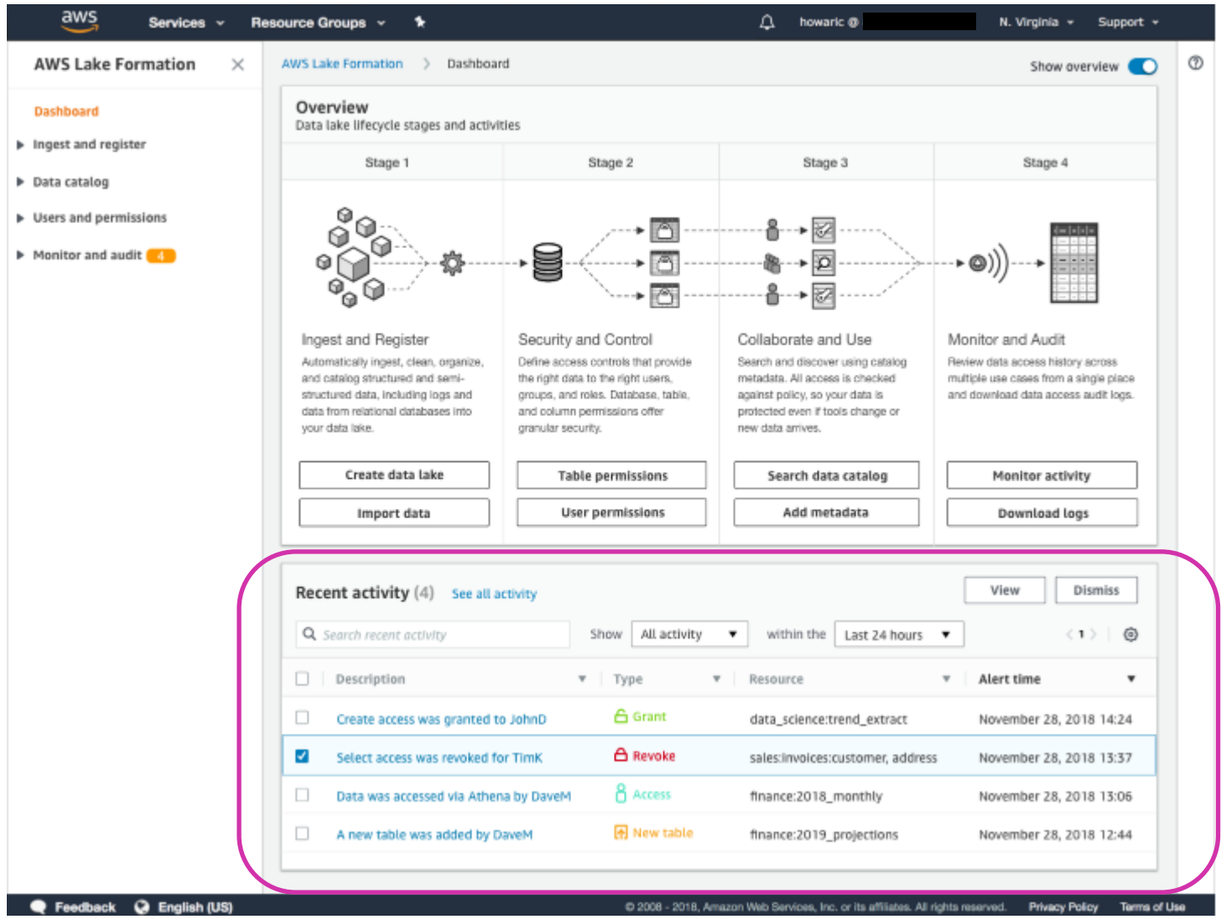


Audit and monitor in real time

See detailed alerts in the console

Download audit logs for further analytics

Data ingest and catalog notifications also published to Amazon CloudWatch events



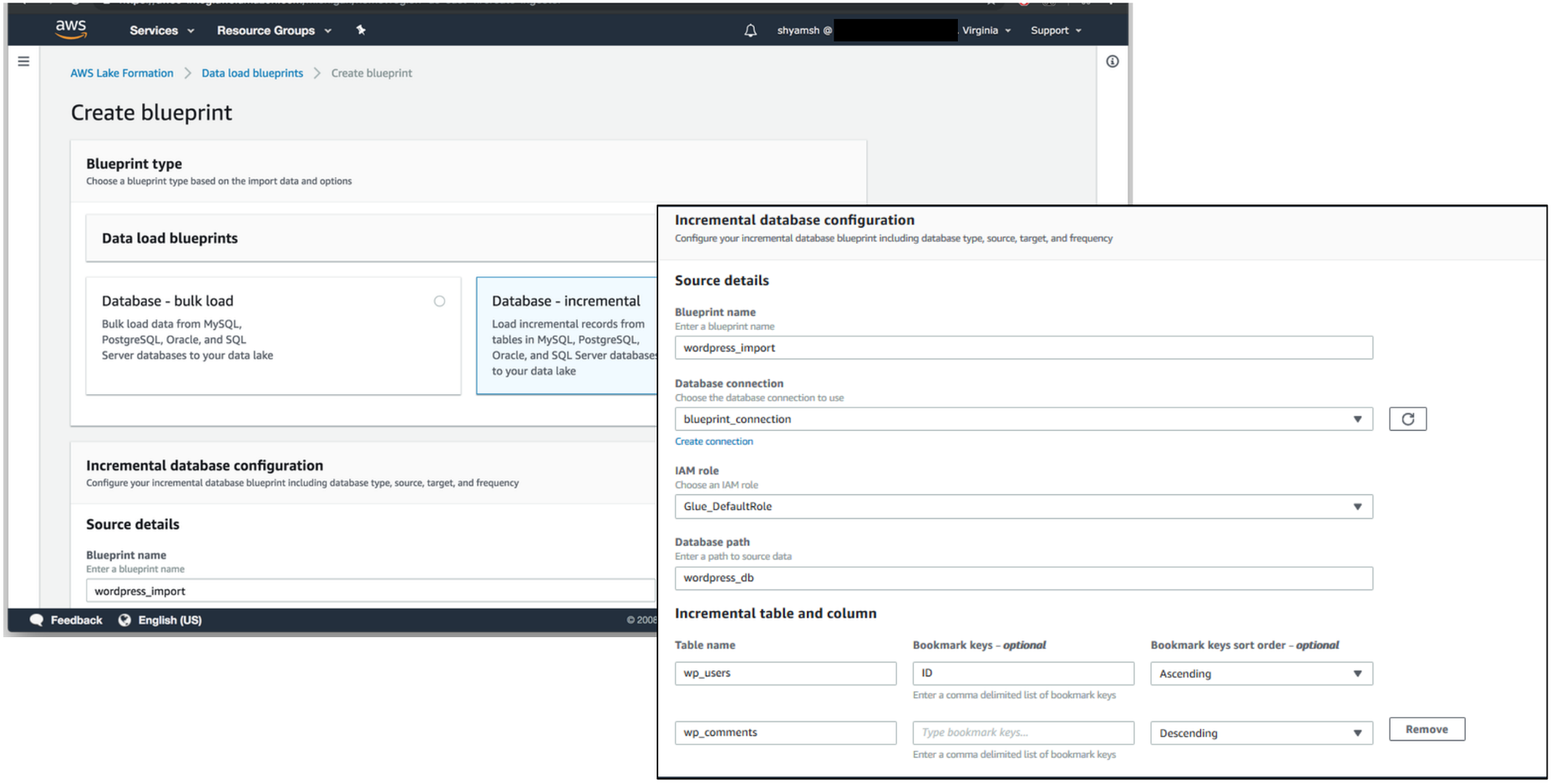
To Summarize : Data Lake in 3 easy steps

1. Use Blueprints to Ingest data

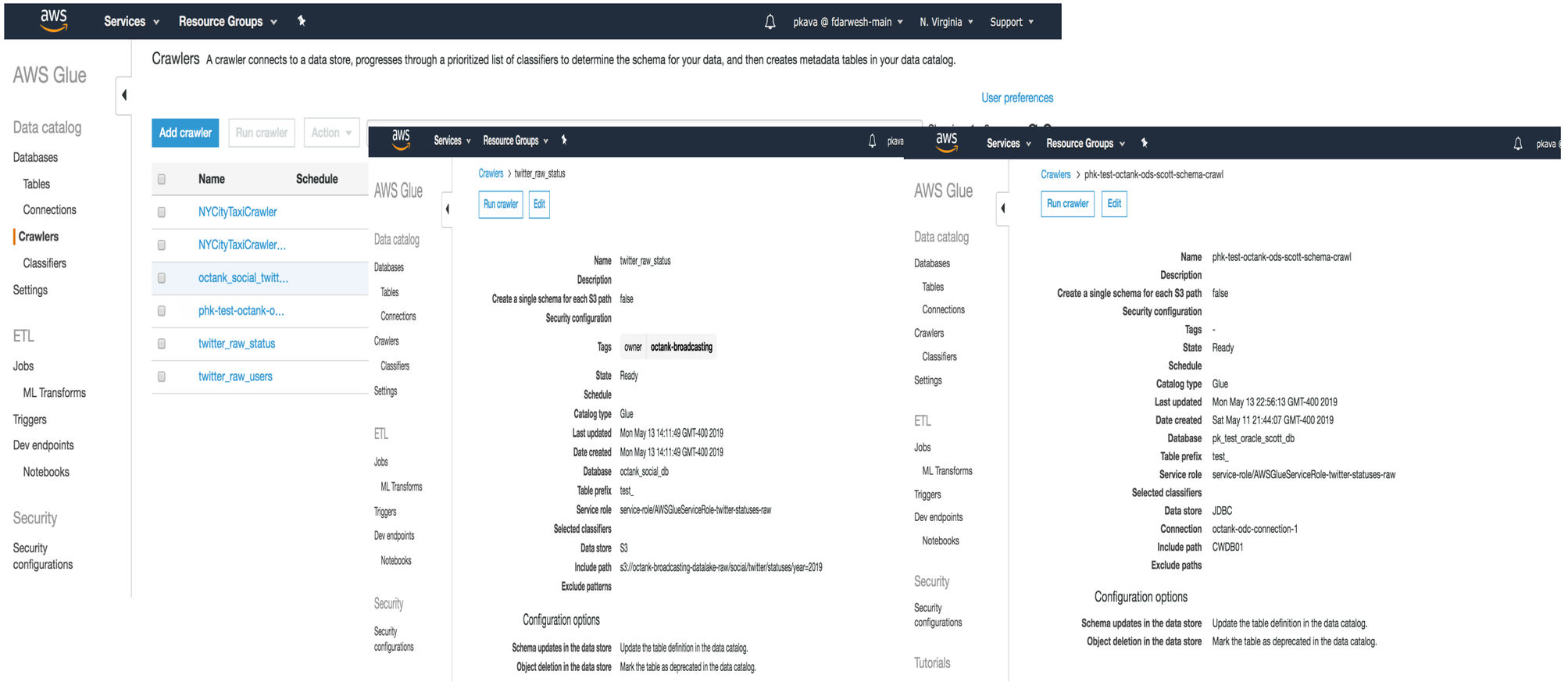
2. Use Glue to Catalog and Transform data

3. Grant Permissions to securely share data

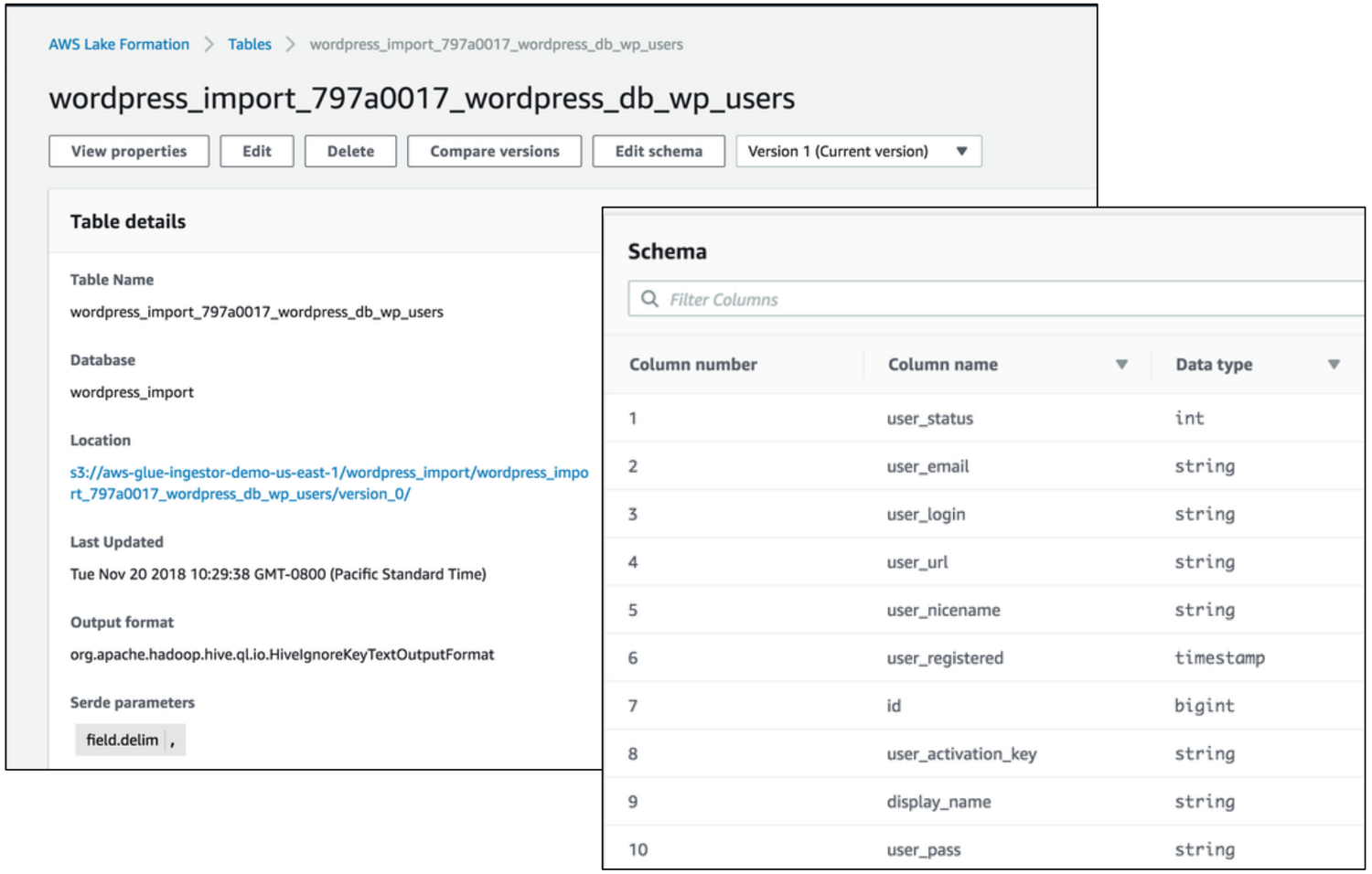
Step 1: Blueprints to Ingest data



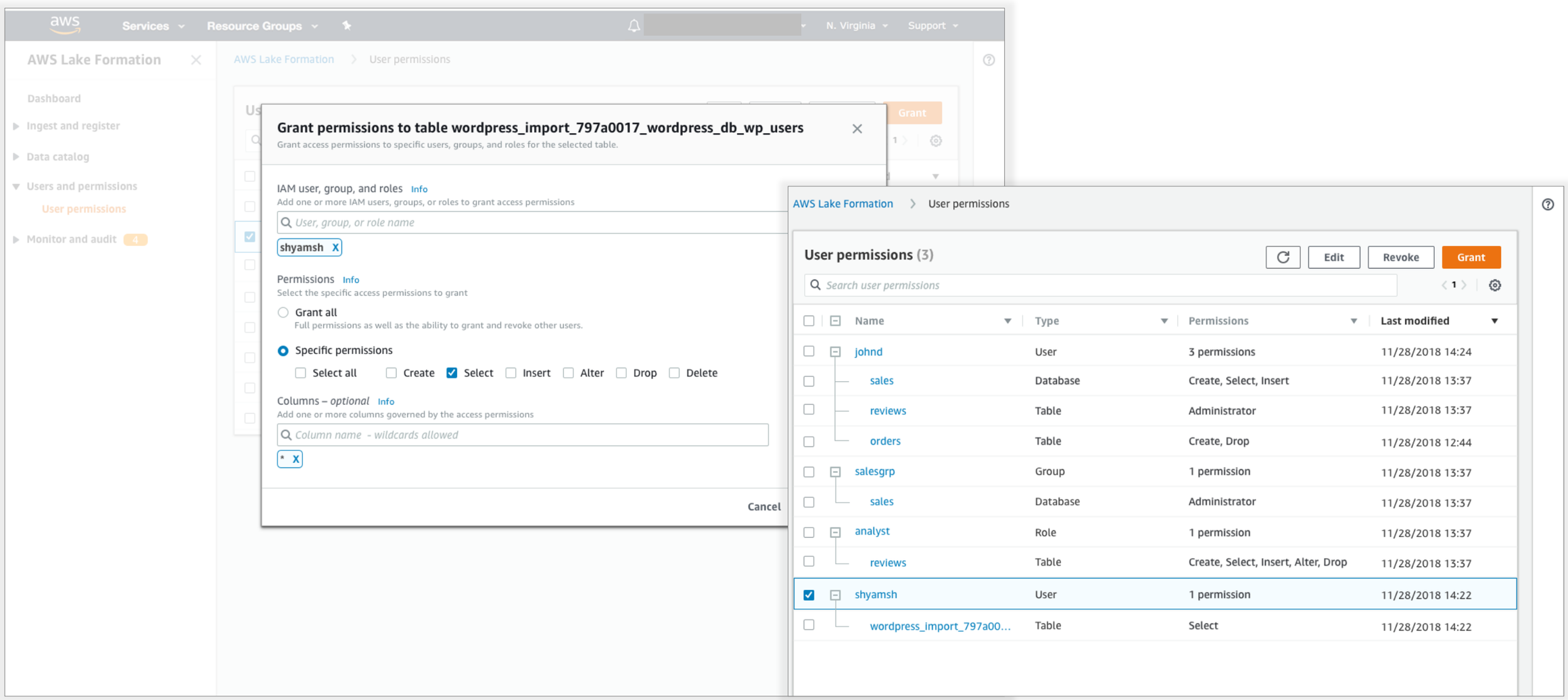
Step 2: Glue to Catalog and Transform data



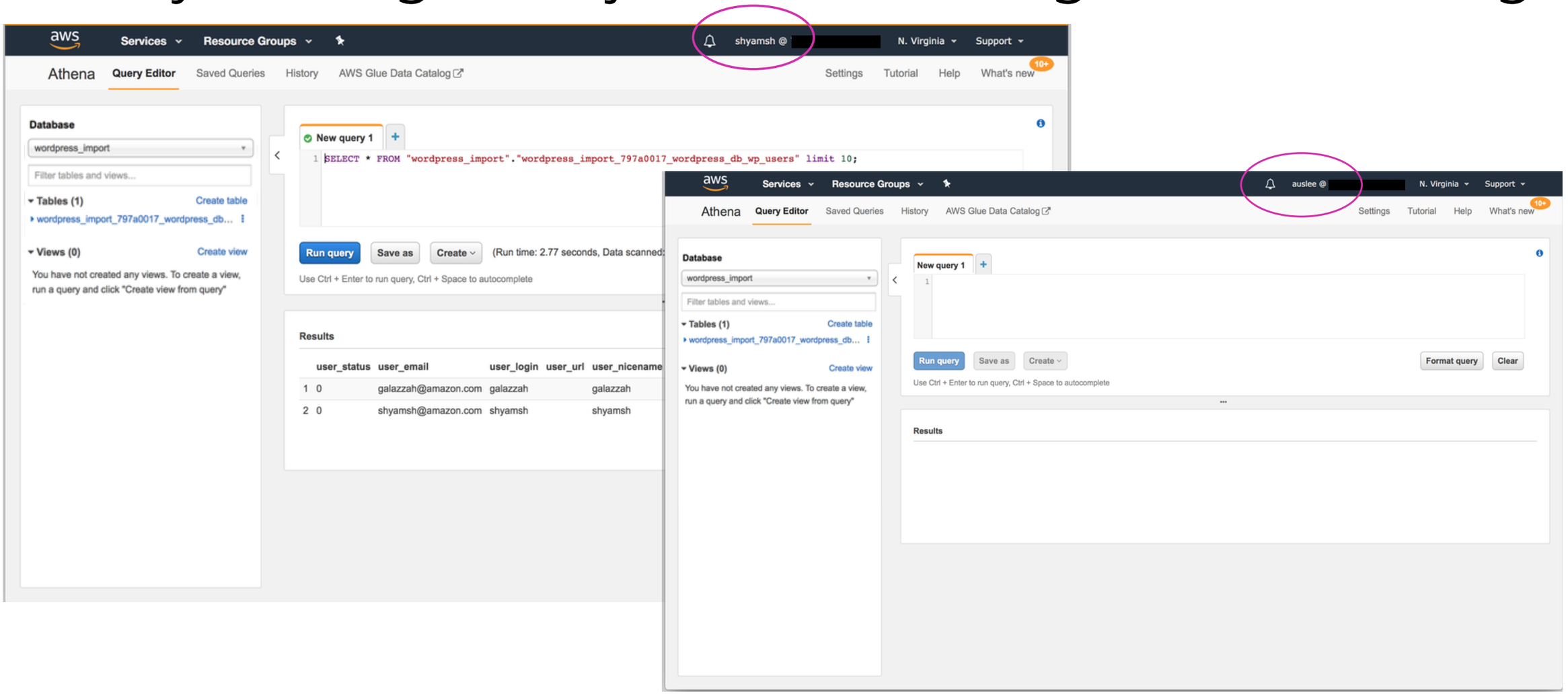
Ingested dataset as table in the data lake



Step 3: Grant Permissions to securely share data



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

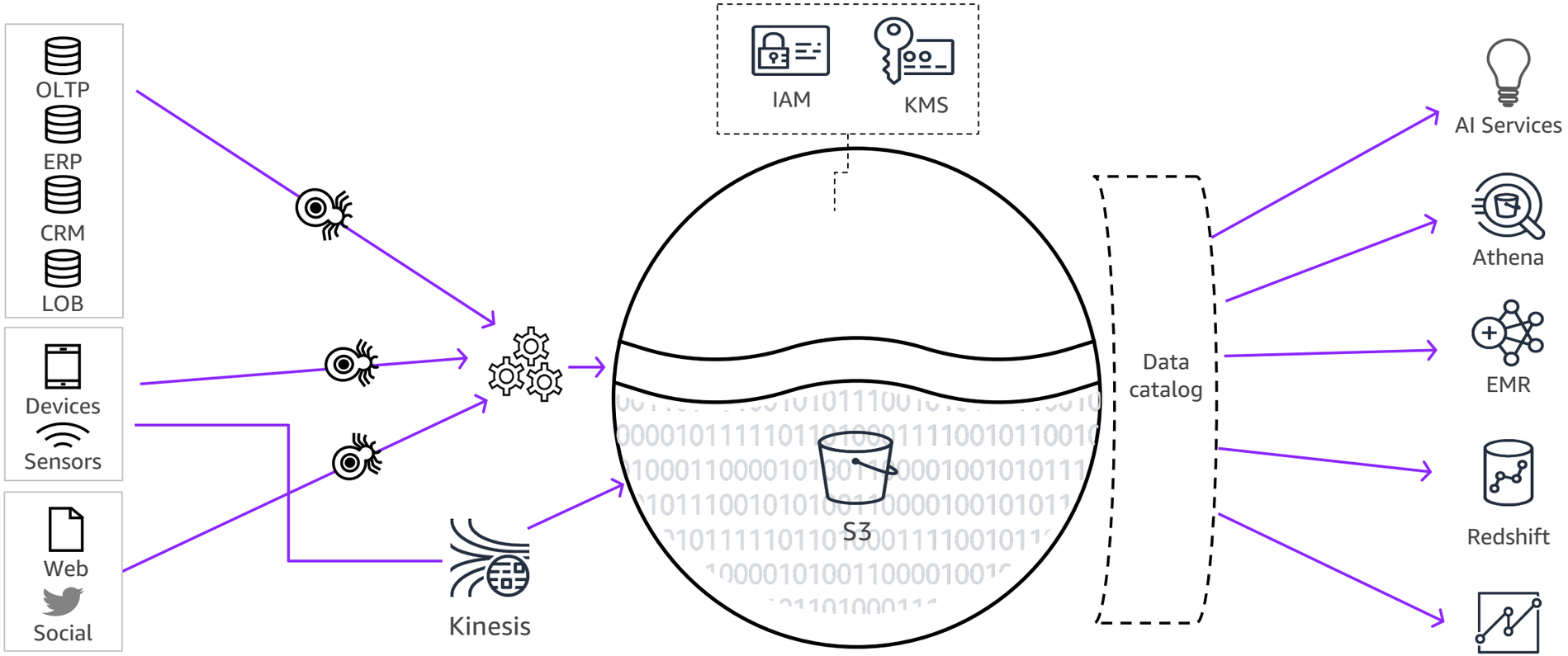


Lake Formation : End to End flow

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

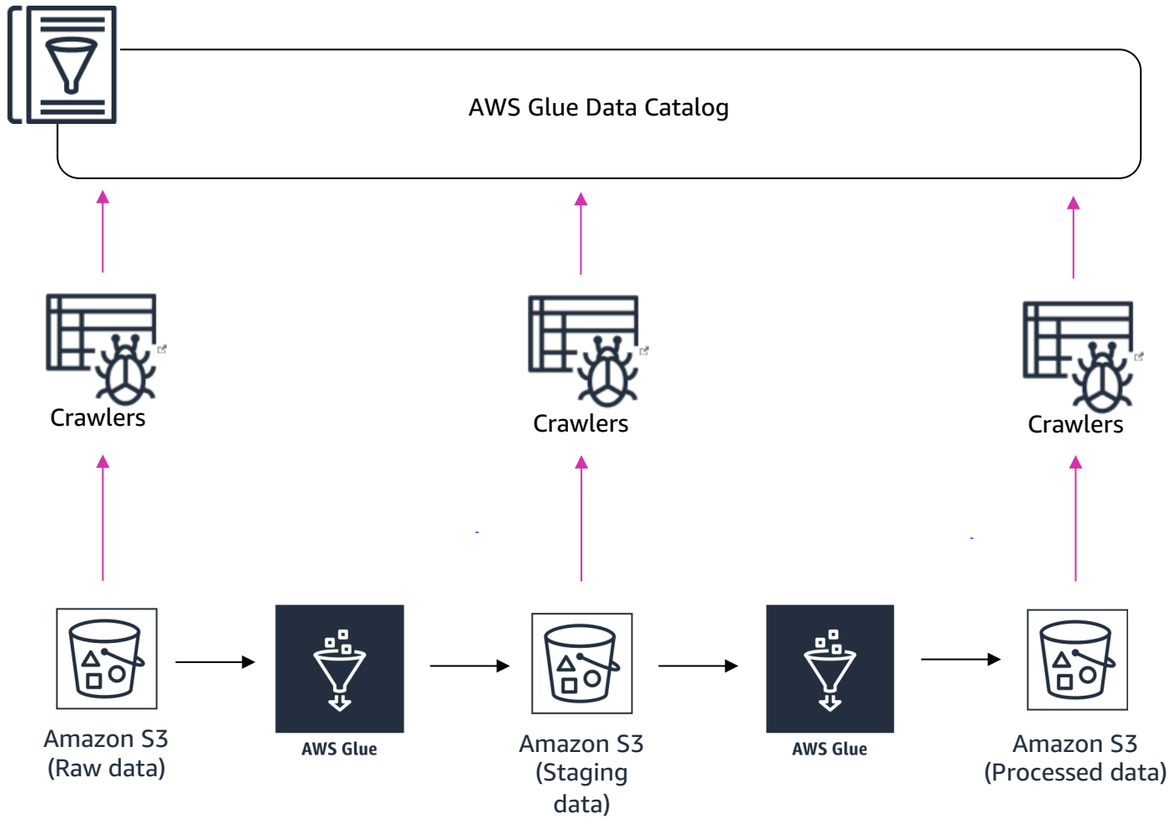
Enable self-service and combined analytics
- Analysts discover all data available for analysis from a single data catalog
- Use multiple analytics tools over the same data



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

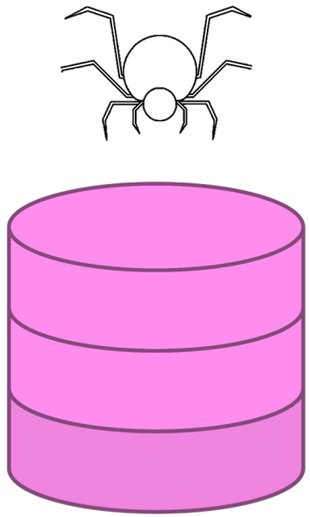
Ability to automatically crawl and **classify your data**

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

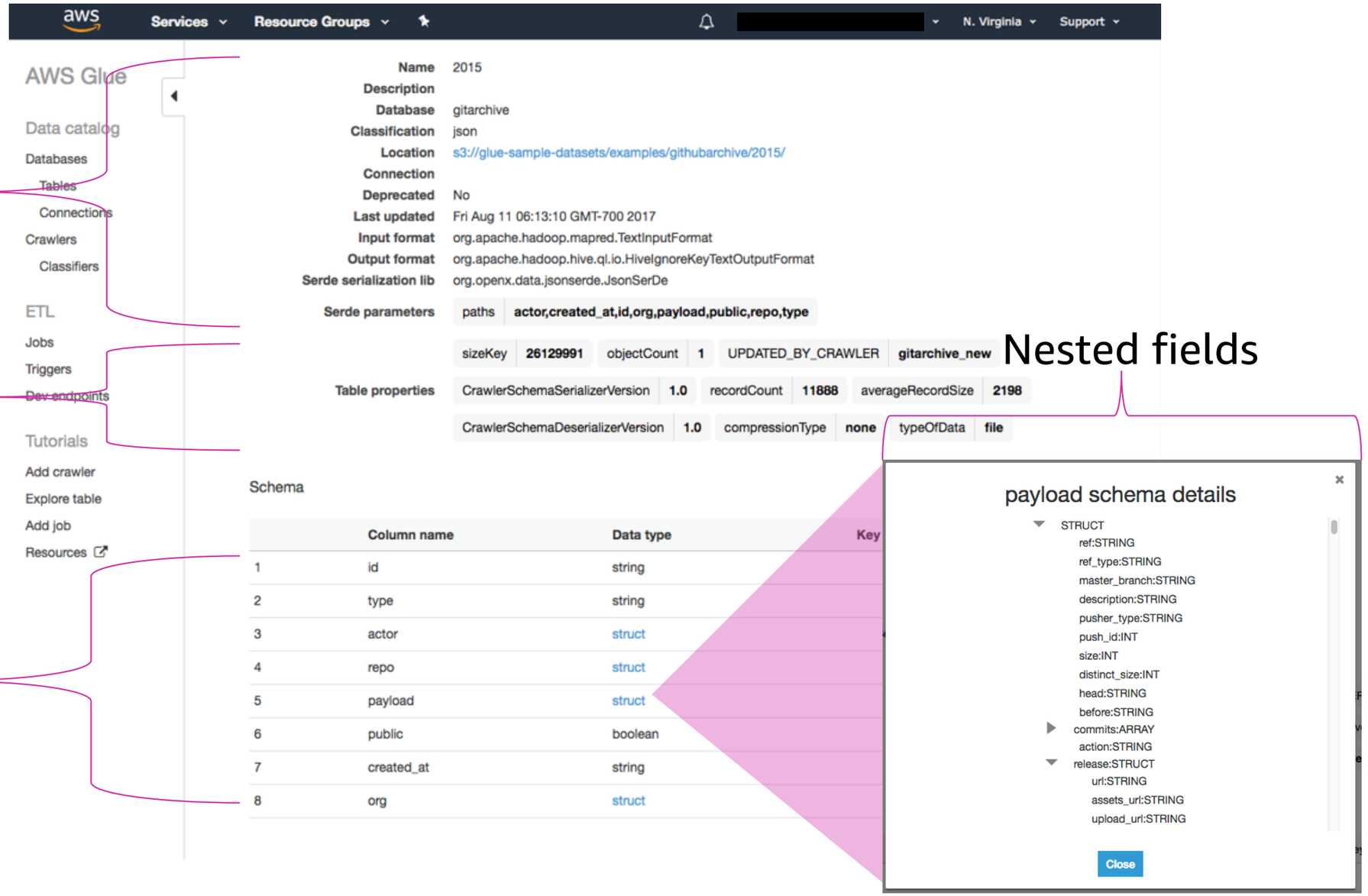
- Crawlers automatically build your Data Catalog and keep it in sync.
- Automatically discover new data, extracts schema definitions
  - Detect schema changes and version tables
  - Detect Hive style partitions on Amazon S3
- Built-in classifiers for popular types; custom classifiers using Grok expression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

A table in the Glue Data Catalog

Table properties

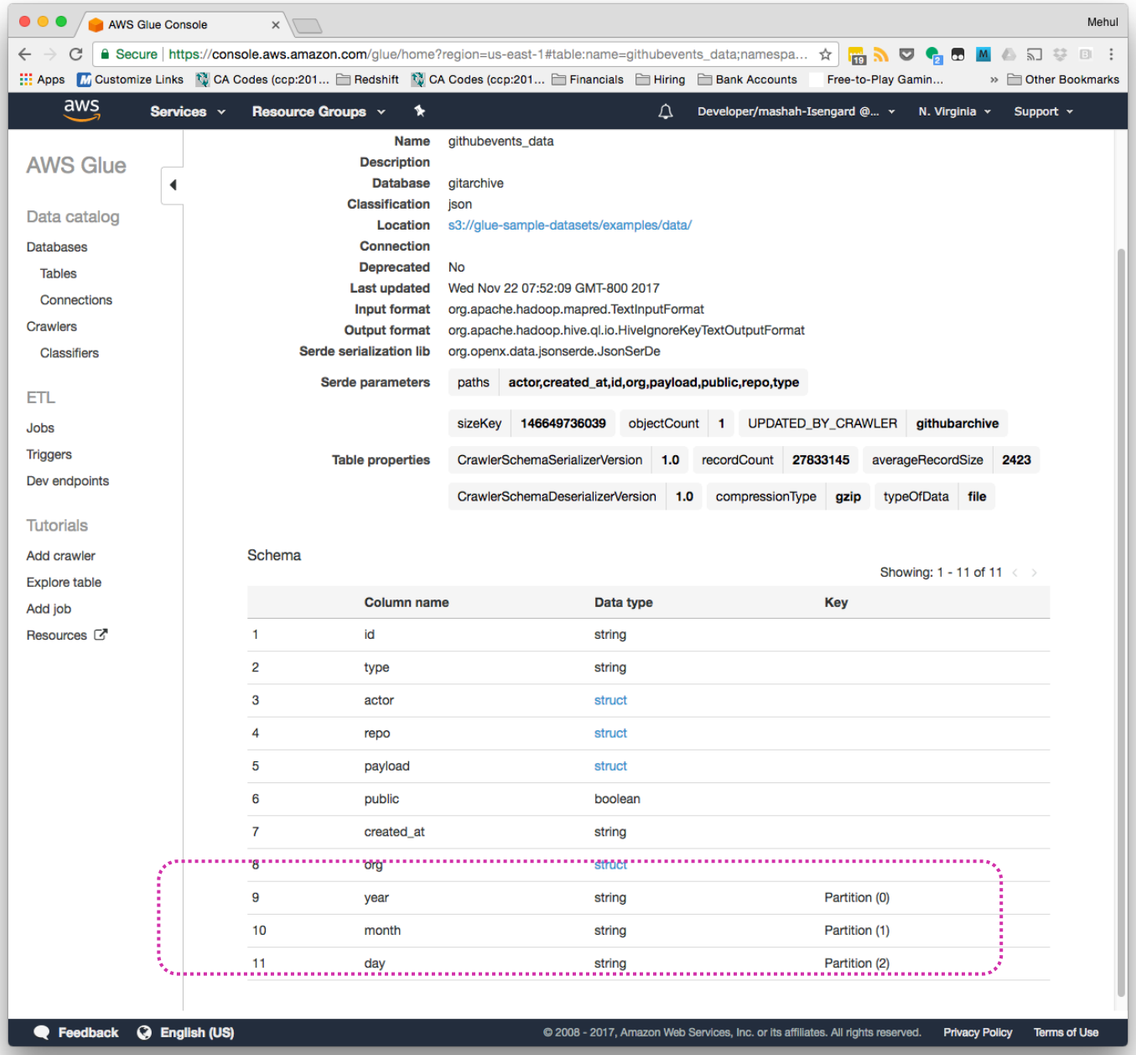
Data statistics

Table schema

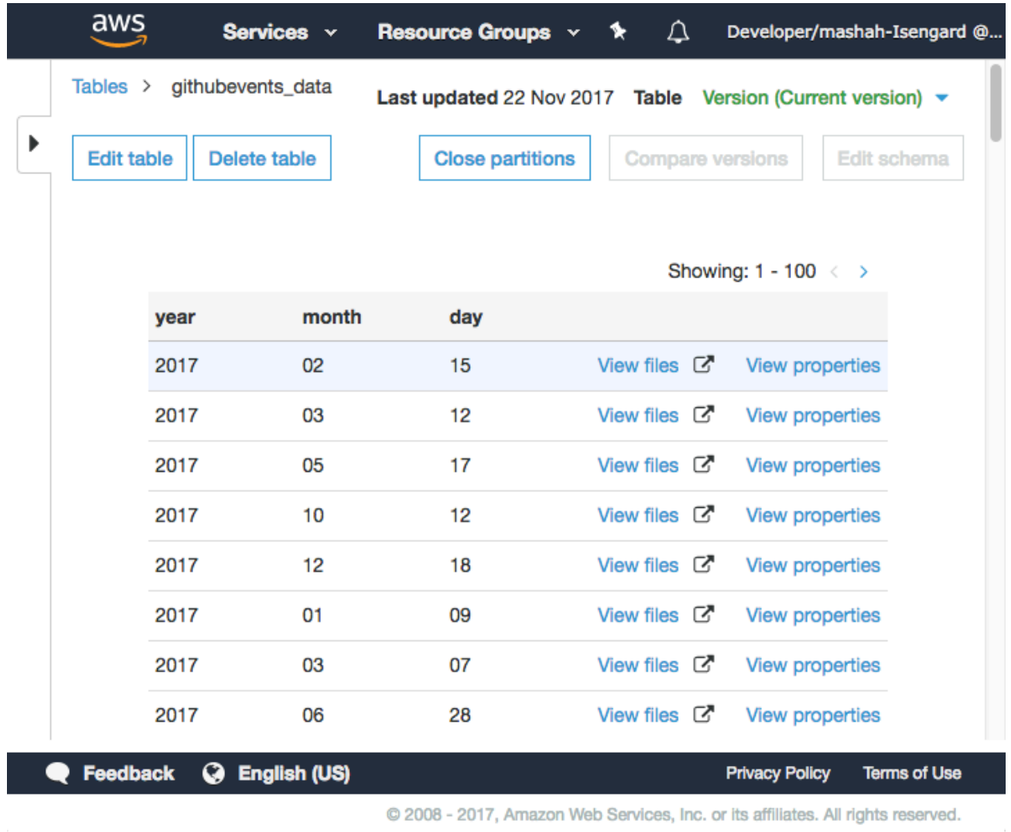


Nested fields

Automatically detected partitions



Available partitions



Automatic schema versioning

Automatically update table version as data evolves

Last updated 21 Aug 2017 **Table** Version 1

Name	simpletweets_json
Description	
Database	simpletweets_json
Classification	json
Location	s3://gluesampledata/simpletweets.json
Connection	
Deprecated	No
Last updated	Mon Aug 21 15:23:42 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 **entities,id,retweeted,text,user**
Table properties	sizeKey **456580** objectCount **1** UPDATED_BY_CRAWLER **TestS3Crawler** mycustom **abc** CrawlerSchemaSerializerVersion **1.0** recordCount **1001** averageRecordSize **456** CrawlerSchemaDeserializerVersion **1.0** compressionType **none** typeOfData **file**

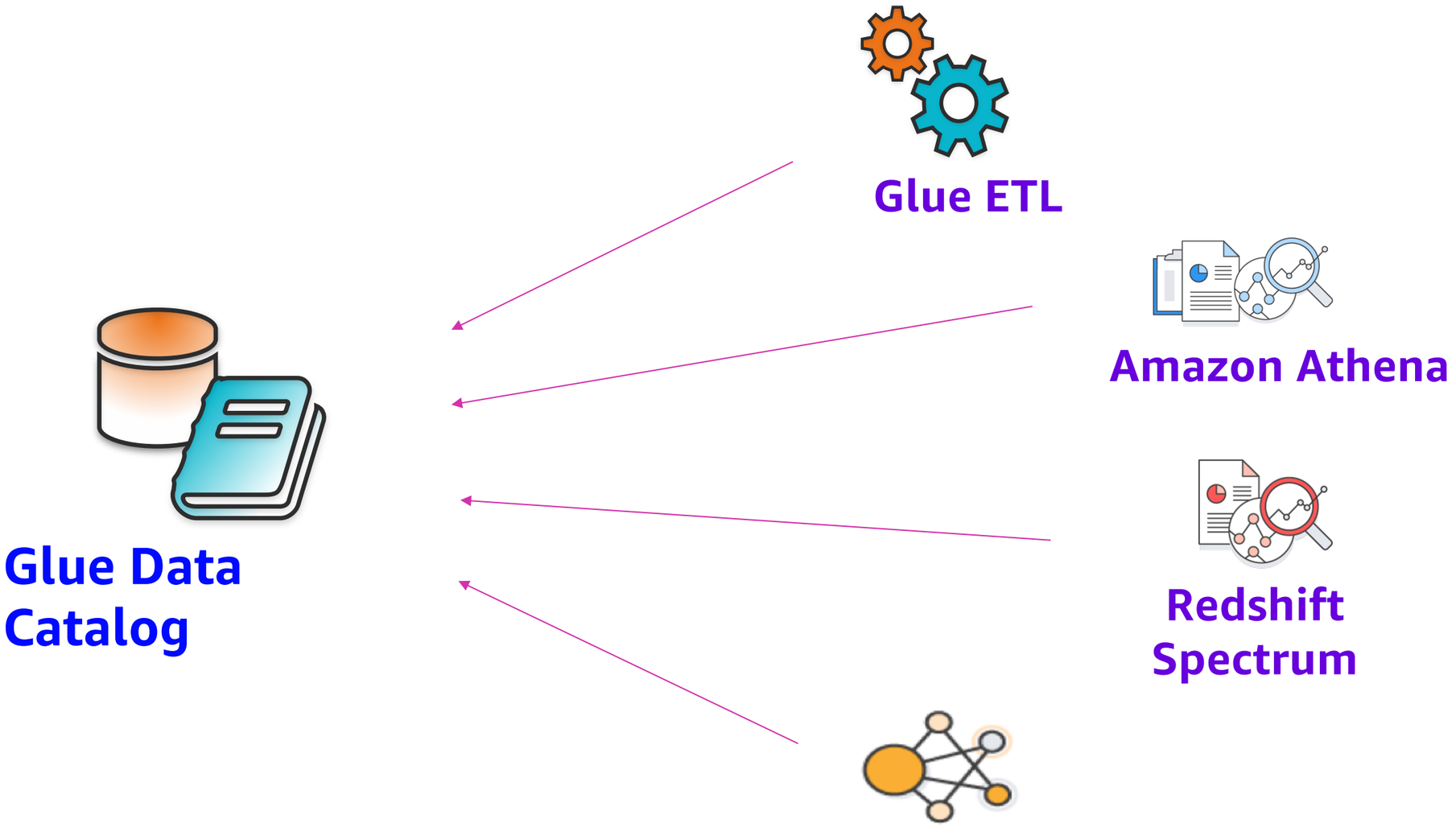
Change	Column name	Data type	Key
	id	bigint	
	retweeted	boolean	
	text	string	
	user	struct	

Last updated 25 Nov 2017 **Table** Version 2

Name	simpletweets_json
Description	
Database	simpletweets_json
Classification	json
Location	s3://gluesampledata/simpletweets.json
Connection	
Deprecated	No
Last updated	Sat Nov 25 12:30:26 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 **entities,id,retweeted,text,user**
Table properties	sizeKey **456580** objectCount **1** UPDATED_BY_CRAWLER **TestS3Crawler** mycustom **abc** CrawlerSchemaSerializerVersion **1.0** recordCount **1001** averageRecordSize **456** CrawlerSchemaDeserializerVersion **1.0** compressionType **none** typeOfData **file**

Change	Column name	Data type	Key
	id	bigint	
	retweeted	boolean	
	text	string	
	user	struct	
Added	url	string	

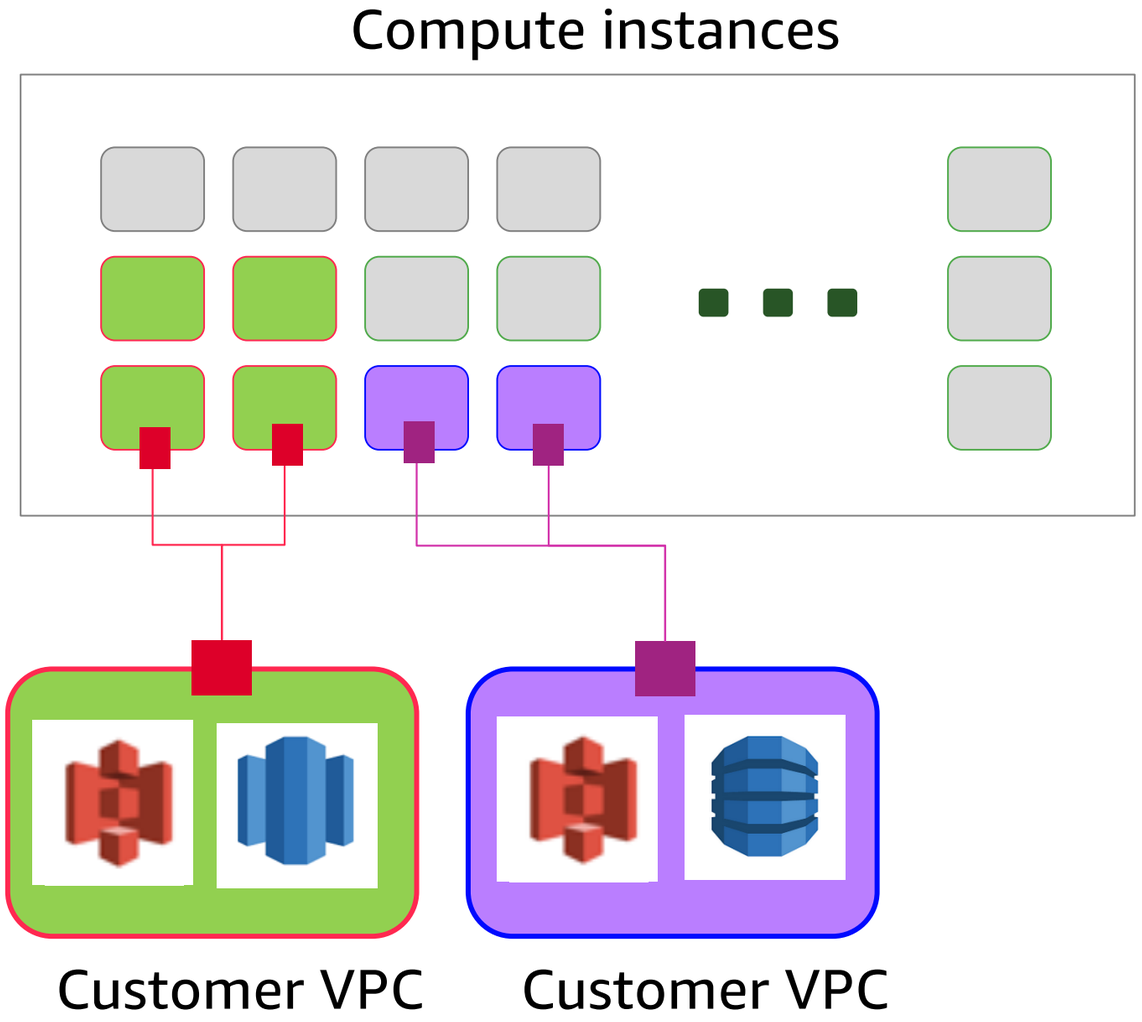
Glue: Data Catalog – Queryable by Many Services



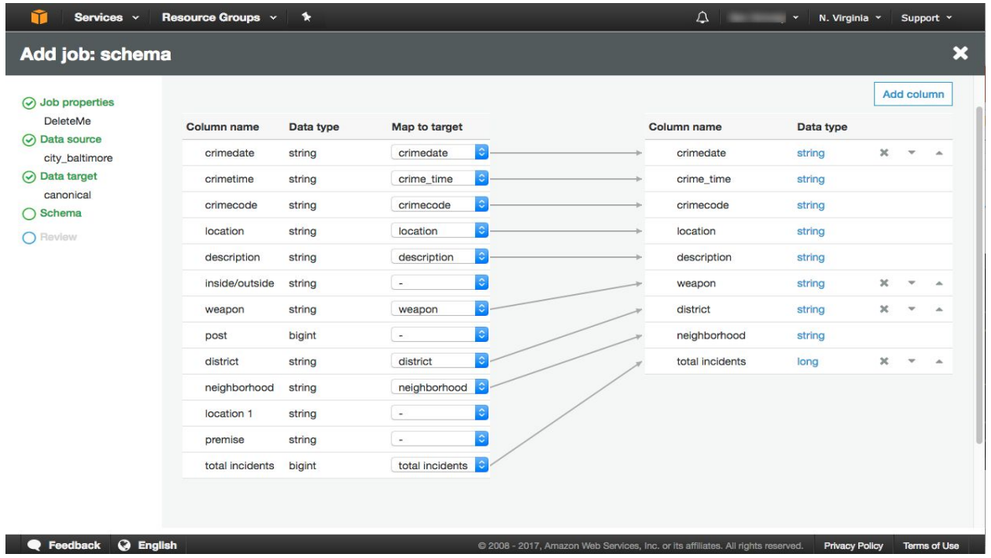
Transform data using Serverless ETL

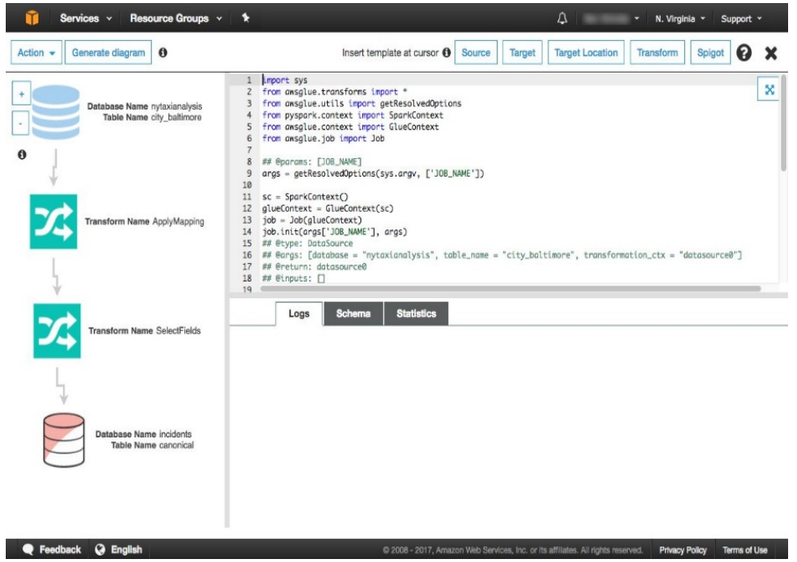
No need to provision, configure, or manage servers

- 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

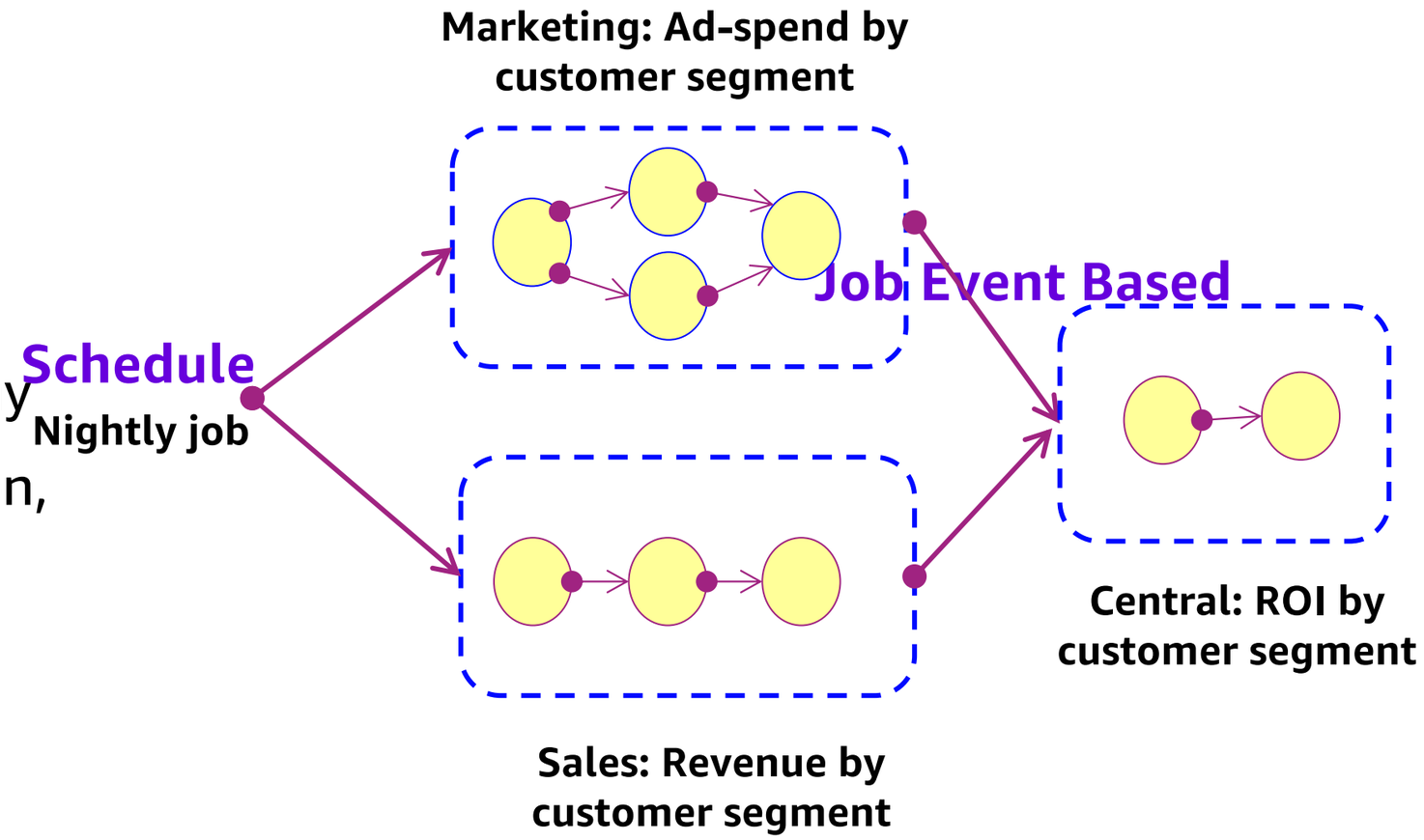
Compose jobs globally with event-based dependencies

- Easy to reuse and leverage work across organization boundaries

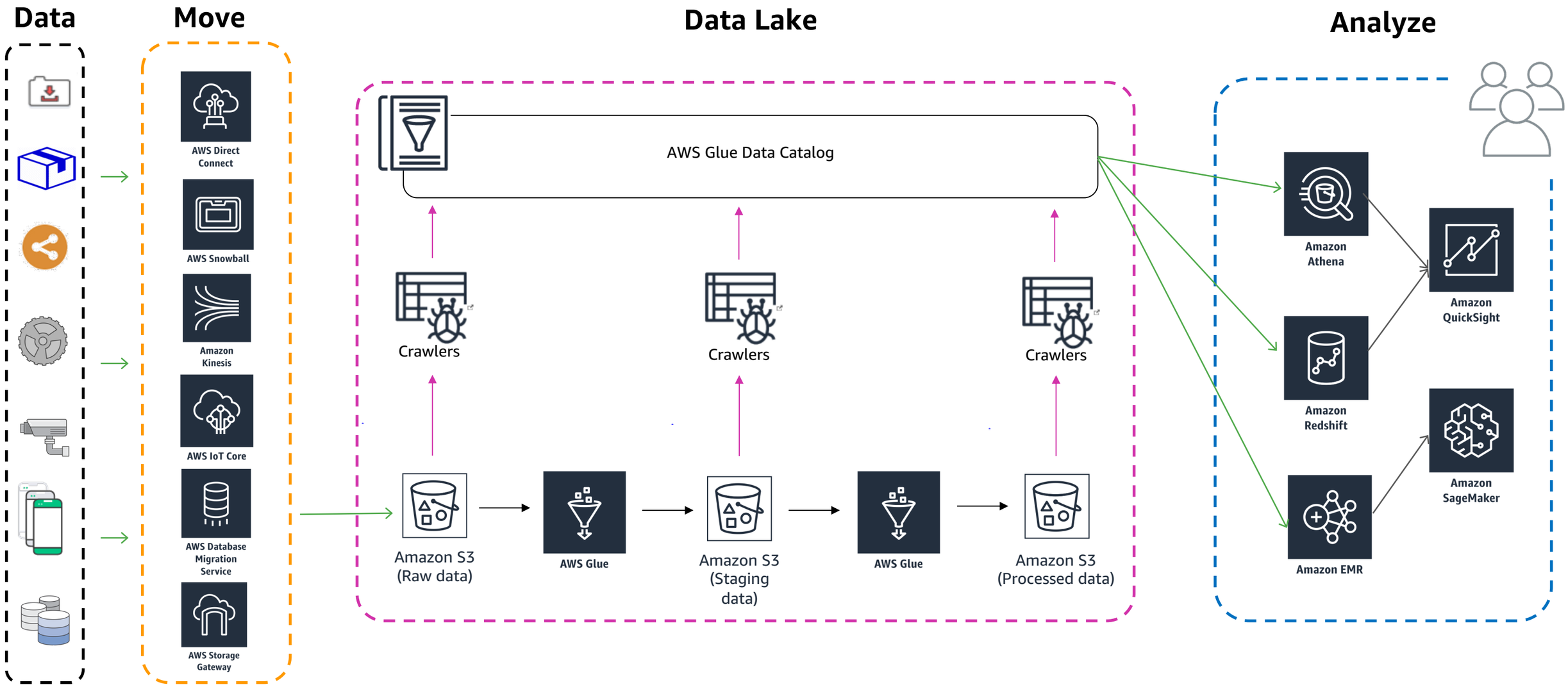
Multiple triggering mechanisms

- **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