



AWS
re:Invent

D A T 3 5 5 - R

How to choose between Amazon Aurora MySQL and PostgreSQL

Vlad Vlasceanu

Principal Database Solutions
Architect

Amazon Web Services

Gowri Balasubramanian

Principal Database Solutions
Architect

Amazon Web Services

Agenda

Introduction to Amazon Aurora

Key decision criteria

Amazon Aurora

MySQL and PostgreSQL compatible relational database built for the cloud

Performance and availability of commercial-grade databases at one-tenth the cost



Performance and scalability

Up to 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 AZs; continuous backup to Amazon S3



Highly secure

Network isolation, encryption at rest and in transit



Fully managed

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

Customer success: Aurora for peak/global demand

Intuit

Commerce platform migration to Aurora MySQL, handling the peak demands during tax season. Global database enables them to achieve low-latency read scaling coast to coast.

Customer success: Aurora for Inventory Database



Challenge:

- Large maintenance burden of an unmanaged Oracle database
- Large team of database engineers dedicated to maintenance activities such as OS patching and upgrades
- Complicated and inefficient hardware provisioning

Solution:

- Aurora PostgreSQL was selected as the database because it could handle complex write transactions at speeds comparable to the Oracle system
- The scaling effort has been reduced by as much as 95 percent.

Use cases for Aurora



Migrations from
commercial
engines



SaaS applications



Geospatial
workloads



Gaming
workloads



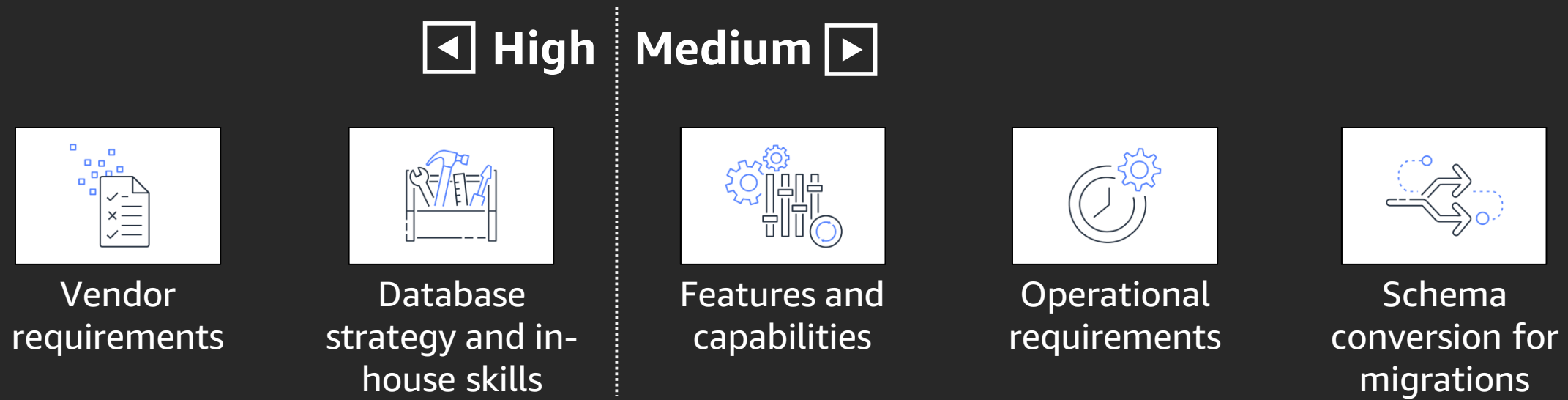
Enterprise
applications

Workload characteristics

- OLTP Workloads + light analytics/reporting (HTAP)
- Reaching throughput or other scaling limits with open-source engines
- High volumes of concurrent connections
- High durability and HA requirements
- Workload consolidation
- On-demand scaling

Key decision criteria

How to choose?



Tiebreaker: Proof of concept

Database strategy and in-house skills

In-house skills of DB technology

- PostgreSQL vs. MySQL/MariaDB
 - Retraining in-house DBAs vs. recruitment considerations
-



Cloud adoption strategy

- Rearchitected vs. lift-and-shift
 - Microservices vs. monolithic applications
-



Database strategy

- Upgrade and patching strategy

Feature parity: Aurora common features

Features	Aurora
Storage	Purpose-built log structured distributed storage Six copies of data across three Availability Zones
Failover	Under 30 seconds Survivable cache
Features	Up to 15 readers Auto scaling with readers Aurora serverless
Performance	Up to 5x throughput of MySQL Up to 3x throughput of PostgreSQL
Manageability	Fast database cloning Continuous backup Amazon RDS performance insights
Security	Encryption at rest, in-transit, AWS Identity and Access Management (IAM) authentication
Integration with other AWS services	AWS DMS, Amazon S3, Amazon CloudWatch Logs

Feature parity: Aurora differences

	Aurora MySQL (5.6, 5.7 compatible)	Aurora PostgreSQL (9.6.X, 10.X, 11.X compatible)
Availability	Aurora multi-master (5.6 only) Zero-downtime patching on writer (ZDP) Zero-downtime restarts due to lag (ZDR)	
Replication	Aurora global database Binary log based	Logical replication (≥ 10.6)
Performance	Asynchronous key prefetch Parallel query (5.6 only) Overhauled query cache Adaptive thread pool Latch-free lock manager	Cluster cache management Query plan management
Data recovery	Database backtrack	
Integration	Amazon S3 import/export Lambda function invocation Amazon Aurora machine learning (ML) integration	Amazon S3 import (≥ 10.7)
Security		Database activity streams (≥ 10.7)

Feature parity: Open-source engines

	Aurora MySQL (5.6, 5.7 compatible)	Aurora PostgreSQL (9.6.X, 10.X, 11.X compatible)
Connection scaling	Thread-based model	Process-based model
Native partitioning	Range, list, hash, key, subpartitions	Range, list, Hash
Analytic features	Minimal	Comprehensive, windowing
Geospatial	Core capabilities (z-ordered curves)	PostGIS
Indexing	Clustered, b-tree, adaptive hash indexing	B-tree, hash, GiST, GIN, BRIN, partial, expressions
Extensions	Minimal	Comprehensive
Stored procedures	Supported	pl/pgpsql, pl/v8, plperl
JSON	Supported in 5.7, indexes via generated columns	Built-in support with indexing capabilities
Others		Materialized views, CTE

Operational requirements

Performance

- High connection counts and variability

Sizing

- Large tables/wide tables

- Single vs. consolidated databases

Database development

- Stored procedure vs. application logic

Security

- DB authentication/access control

Change management

- Frequent DDLs

- Blue/green deployments

Maintenance

- Major versions and minor version upgrades

Data integration

- ETL, data exchange with upstream/downstream systems

Schema conversion for migrations

What were the results of the AWS Schema Conversion Tool (AWS SCT) assessment report?

- Close call vs. wide gap
- Keep in mind: Migration is a one-time event
- AWS SCT conversion is not the end state, must optimize
- There are always repeatable patterns

Proof of concept: Best practices

Define success criteria

- Establish baseline and measure the application performance (reads/sec, writes/sec, latency, throughput)
- Operational efficiency metrics such as failover time, data refresh/load metrics

Identify workload characteristics

- Run with representative production workload
- Right mix of OLTP vs. HTAP
- Query or DML concurrency
- Number of clients or users

Proof of concept: Best practices

Quantify the importance of your success criteria using a scoring model

Illustrative only

Criteria	Criticality (3: High, 2: Medium, 1: Low)	Aurora MySQL score	Aurora PostgreSQL score
Availability	3	2	2
Query performance	3	2	3
Schema changes	2	1	1

Rank degree criteria is met

- 3: Exceeded expectation
- 2: Met expectation out of the box
- 1: Met expectation with extra work
- 0: Fail

Rank importance of the criteria

Related breakouts

DAT328-R – Deep dive on Amazon Aurora with PostgreSQL compatibility

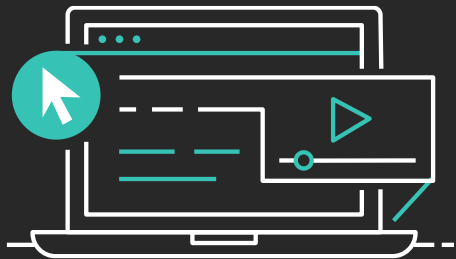
DAT309-R – Amazon Aurora storage demystified: How it all works

DAT404-R – Amazon Aurora Multi-Master: Scaling out database write performance

DAT202-R – What's new in Amazon Aurora

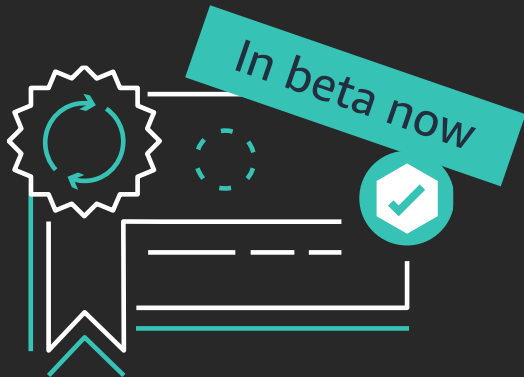
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

Thank you!

Gowri

gowrishb@amazon.com

Vlad

vladv@amazon.com



Please complete the session
survey in the mobile app.