Aurora PostgreSQL for SQL Server DBAs

Wanda He
Prin. Database Specialist Solutions Architect, AWS

Thuymy Tran
Database Specialist Solutions Architect, AWS

Agenda

- Amazon Aurora Overview
- High Availability and Disaster Recovery
- Backup and Restore
- Scaling and Optimization
- Security
- Performance Monitoring
- Migration Options

Amazon Aurora Overview

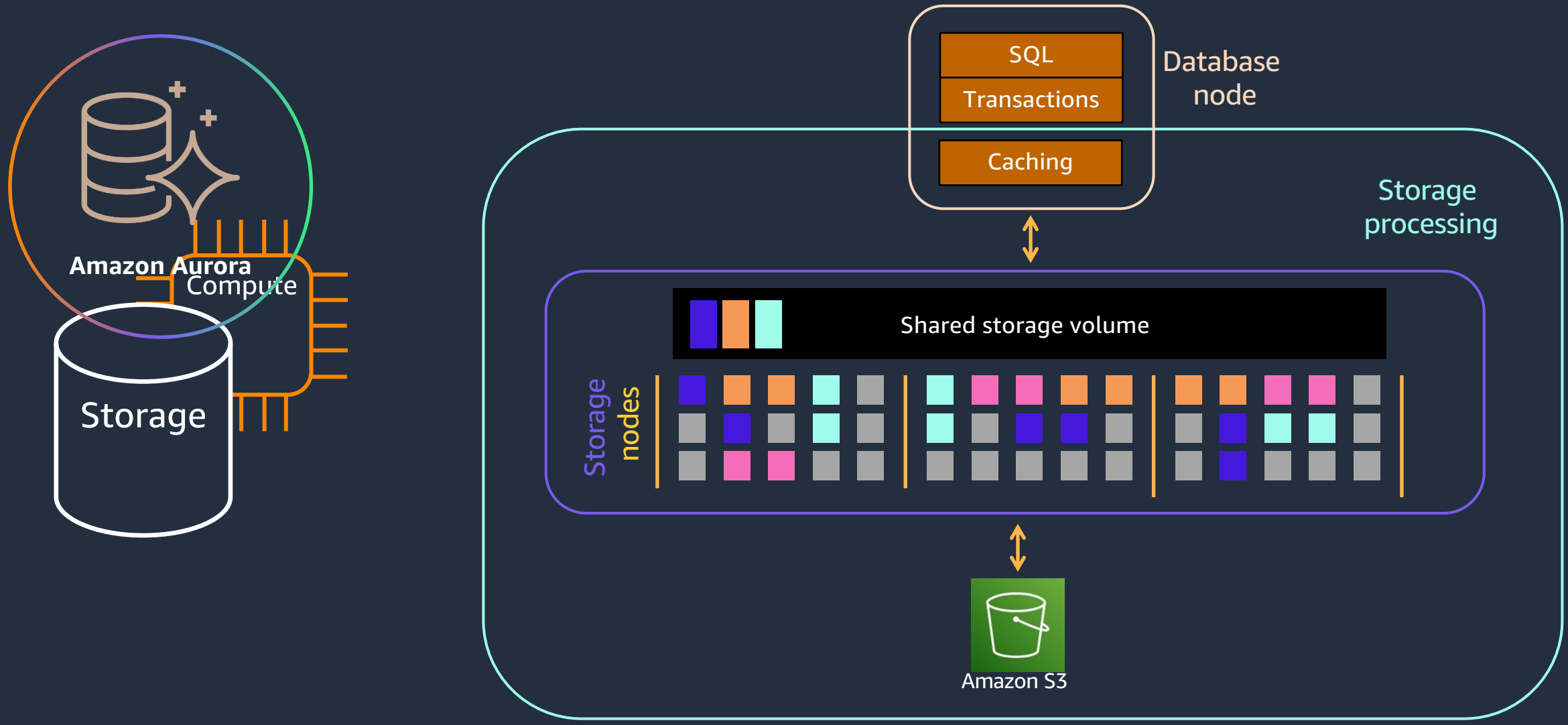
Amazon Aurora

COMMERCIAL-GRADE CLOUD NATIVE DATABASE

Delivered as a **managed** service

- Drop-in compatibility with MySQL and PostgreSQL
- Simplicity and cost-effectiveness of open-source databases
- Throughput and availability of commercial databases
- Simple pay-as-you-go pricing

Aurora decouples storage and query processing



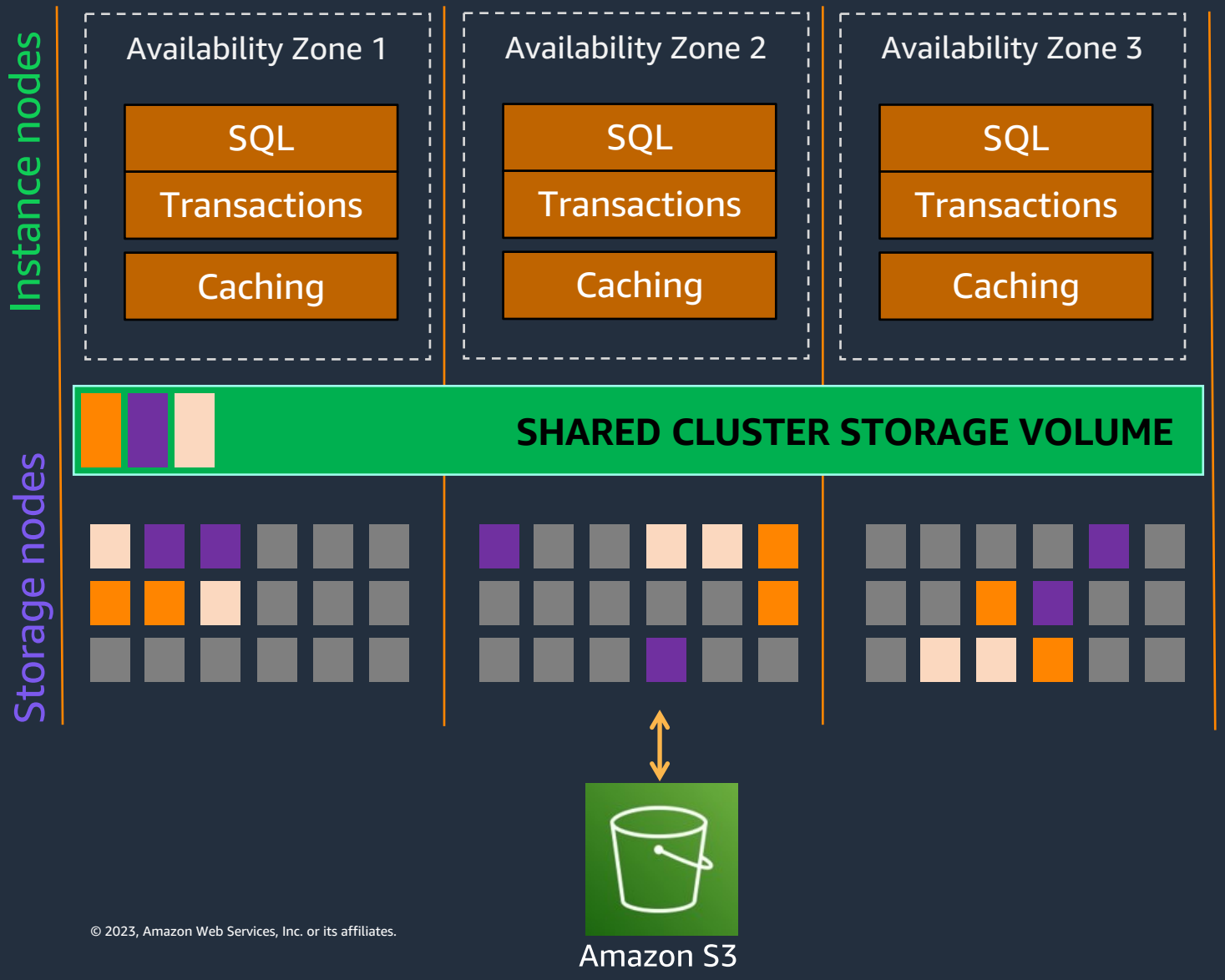
Scale-out, distributed storage processing architecture

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

6 copies of data, 2 copies in each Availability Zone to protect against AZ+1 failures

Data is written in 10 GB “protection groups,” growing automatically when needed



Aurora distributed storage provides:

So the database can process your transactions and SQL queries without impact from storage processing activities

- Redo log processing
- Instant crash recovery
- Fault-tolerant and self-healing storage
- Fast database cloning
- Database snapshots
- Continuous backups and point-in-time restore
- Storage automatic scaling independent of compute
- Read and write scalability
- Warm cache on database restart
- Low-latency replication

High Availability and Disaster Recovery

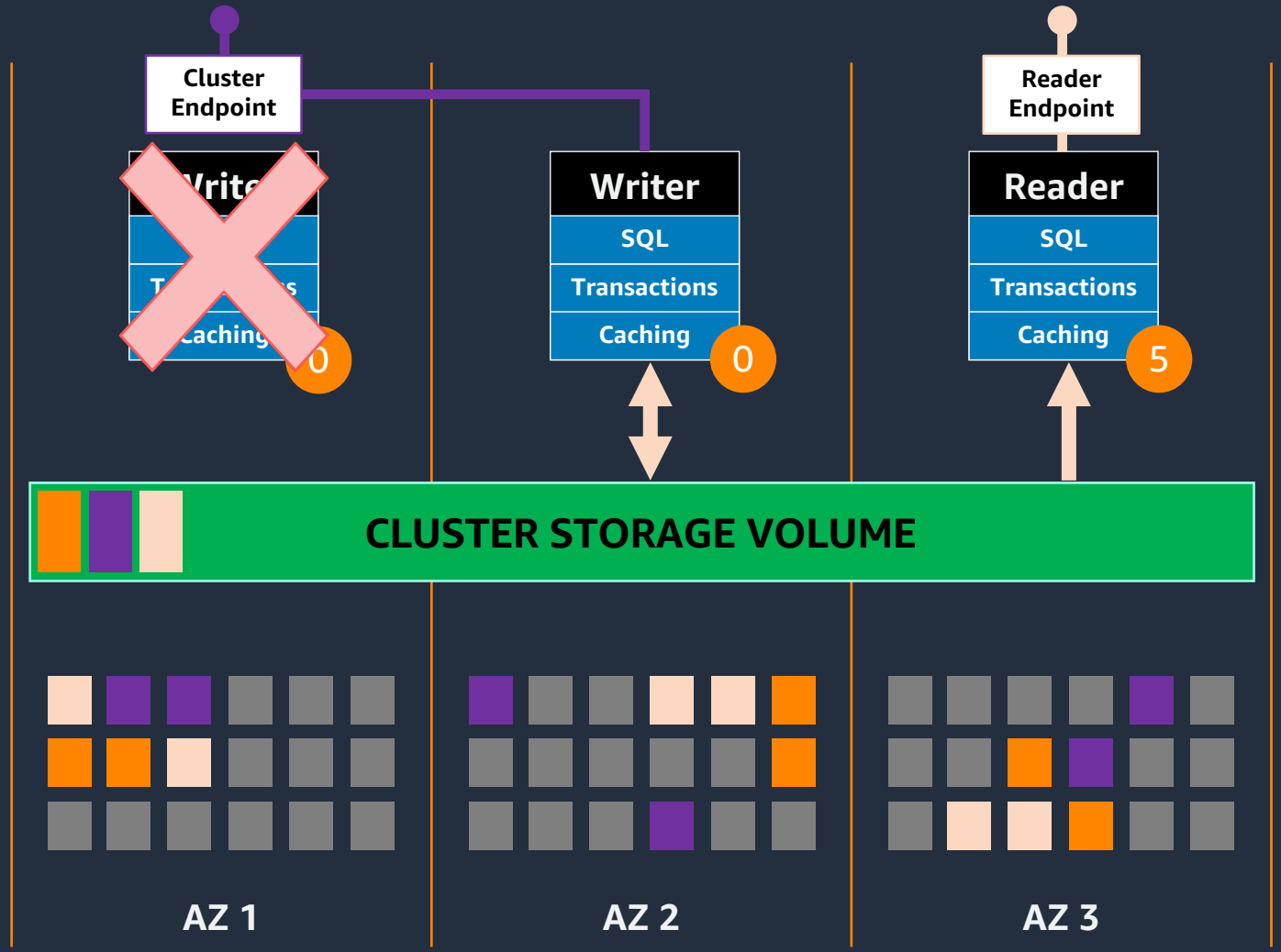
High availability - Automatic failover, zero data loss

SQL Server

- AlwaysOn Availability Groups
- Failover Cluster
- Mirroring

Aurora PostgreSQL

- Multi-AZ
- Six copies of data, two copies in each Availability Zone to protect against AZ+1 failures
- 30s failover time



Disaster Recovery

SQL Server

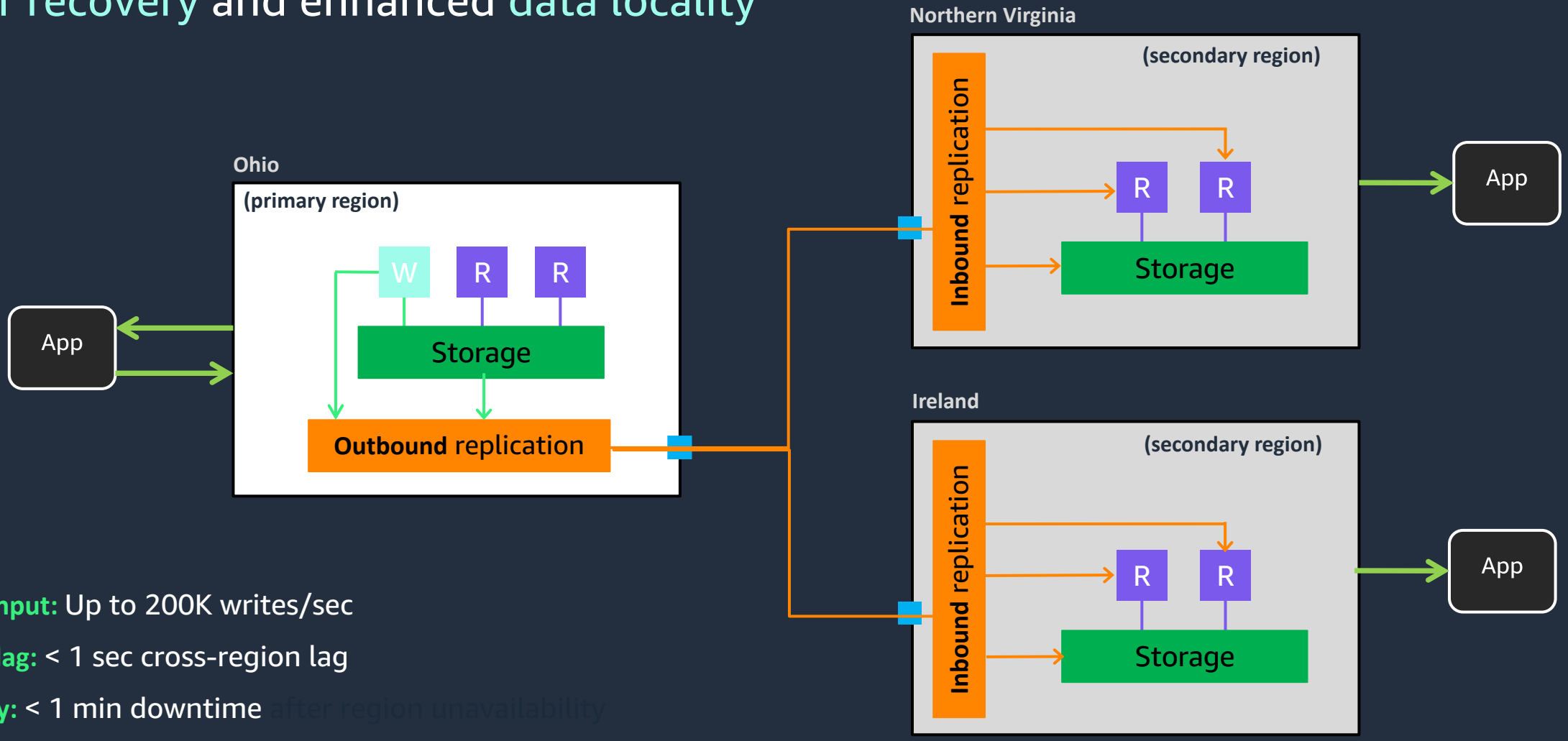
- AlwaysOn Availability Groups (async)
- Failover clustering (geo-clustering)
- Mirroring
- Log shipping
- Replication
- Backup and restore
- 3rd party storage replication (e.g. EMC SRDF)

Aurora PostgreSQL

- Aurora Global Database
- Cross-region snapshot copy & restore
- AWS Data Migration Service (DMS)
- Self-managed logical replication

Aurora Global Database

Disaster recovery and enhanced data locality



High throughput: Up to 200K writes/sec

Low replica lag: < 1 sec cross-region lag

Fast recovery: < 1 min downtime after region unavailability

Backup and Restore

Backup and Restore

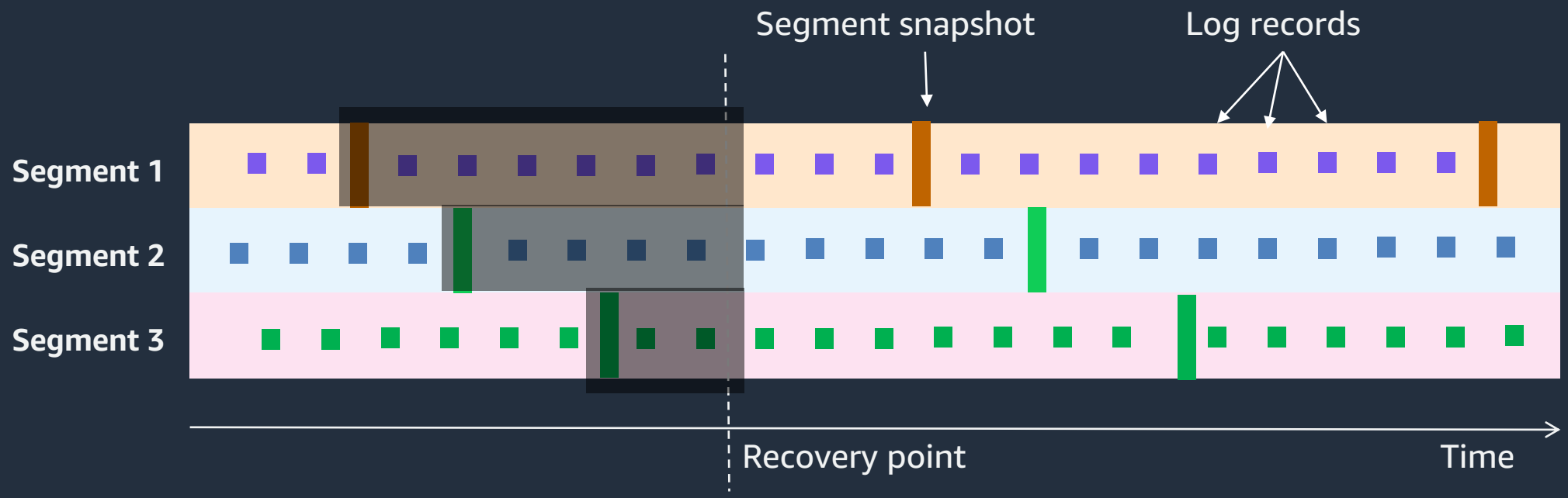
SQL Server

- Native database backup
- T-Log backup (DB level)
- 3rd party storage volume snapshot backup

Aurora PostgreSQL

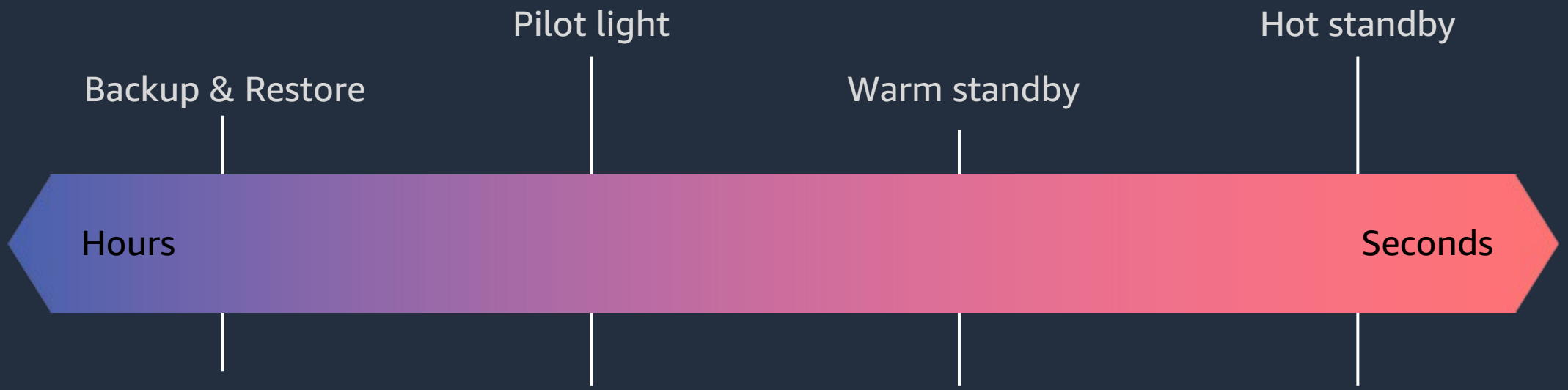
- Incremental daily automated backup (cluster level), retain up to 35 days
- WAL backup every 5 mins to support PiTR
- Manual snapshot backup
- Copy snapshots to another region and/or share snapshots with other AWS accounts
- Native pg_dump / pg_restore
- Integration w/ AWS Backup for lifecycle management, ransomware protection

Continuous backup with point-in-time restore



- Backup is performed by storage nodes, without performance or availability impact
- Storage segments are backed up to S3 at storage node level
- Log records streamed to Amazon S3 continuously
- At restore, appropriate segment snapshots and log records are retrieved by storage nodes to bring up the volume state at the requested point in time
  - **No need for traditional log replay between “backup point ” and “restore point”**

Aurora : DR options



Backup & Restore	Pilot light	Warm standby	Hot standby
RPO/RTO: Hours	RPO/RTO: 10s of Minute	RPO/RTO: Minutes	RPO/RTO: 10s of Second
• Cross Region snapshot copy • Cost $ • Restore in DR Region	• Cross region replica • Cost $$ • Global DB with head-less secondary	• Global DB • Cost $$$ • Small compute or Serverless v2	• At-par Global DB in secondary region(s) • Cost $$$$ • Compute sizes are same as primary

Scaling and Optimization

Scaling and optimization

SQL Server

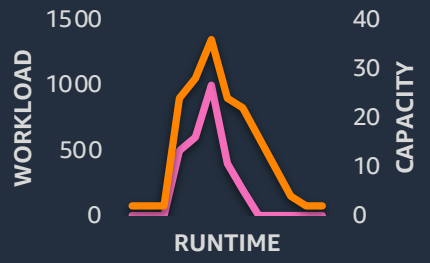
- Scalable shared databases
- AlwaysOn readable secondaries
- Replication

Aurora PostgreSQL

- Compute scale up via higher db instance class
- Up to 15 reader nodes per region, up to 5 regions
- Auto storage scale up/down on-demand
- Aurora auto scaling of read replicas
- Aurora Serverless v2
- Aurora Global Database readable cluster
- Self-managed logical replication

Amazon Aurora Serverless v2

A serverless, auto-scaling configuration for Amazon Aurora that now supports even the most demanding applications and database workloads



Scale instantly, from hundreds to hundreds-of-thousands of transactions, in a fraction of a second

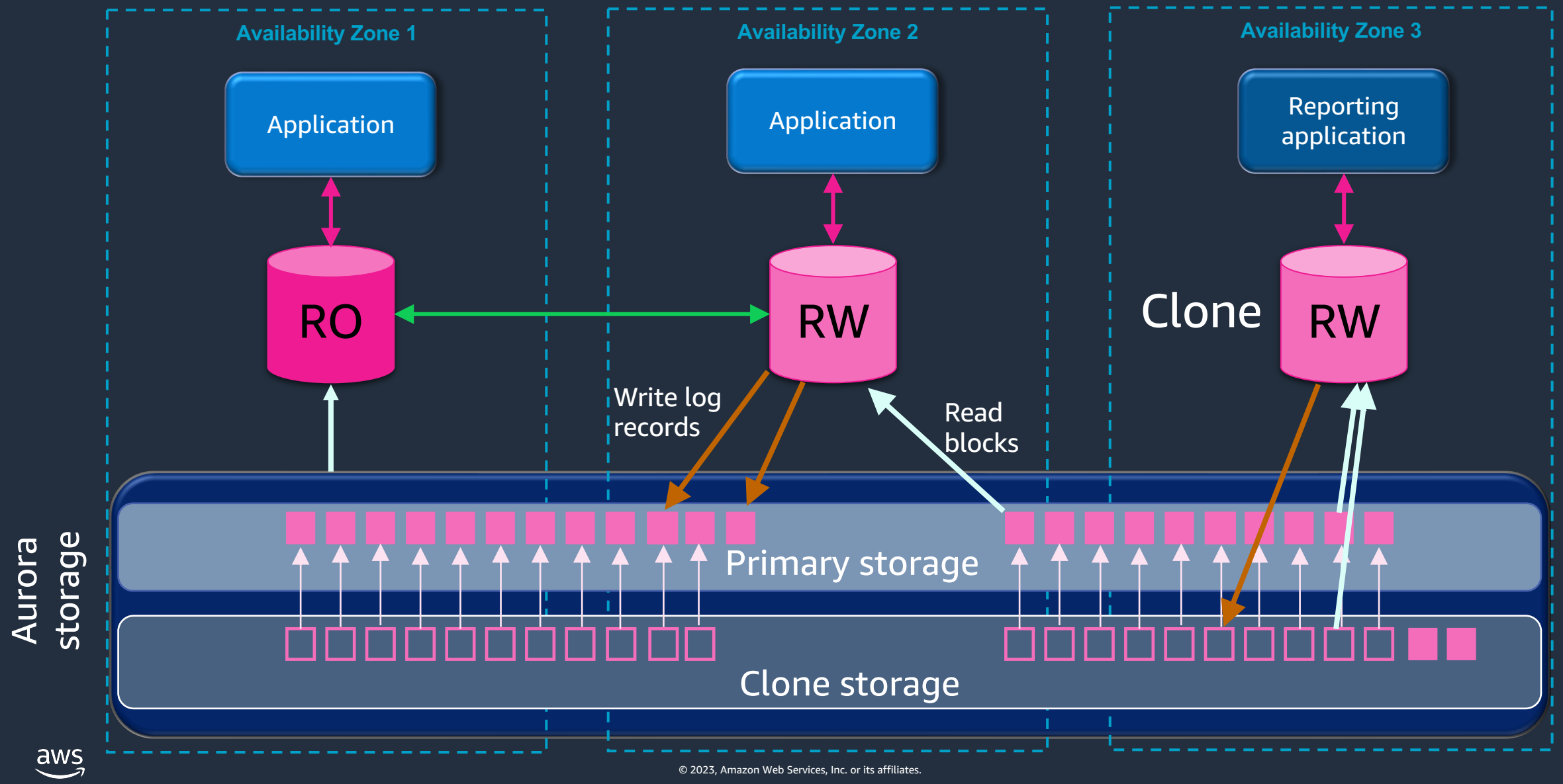
Scale in fine-grained increments to provide just the right amount of database capacity

Full breadth of Aurora capabilities, including Multi-AZ, Global Database

Up to 90% cost savings when compared to provisioning for peak load

aws

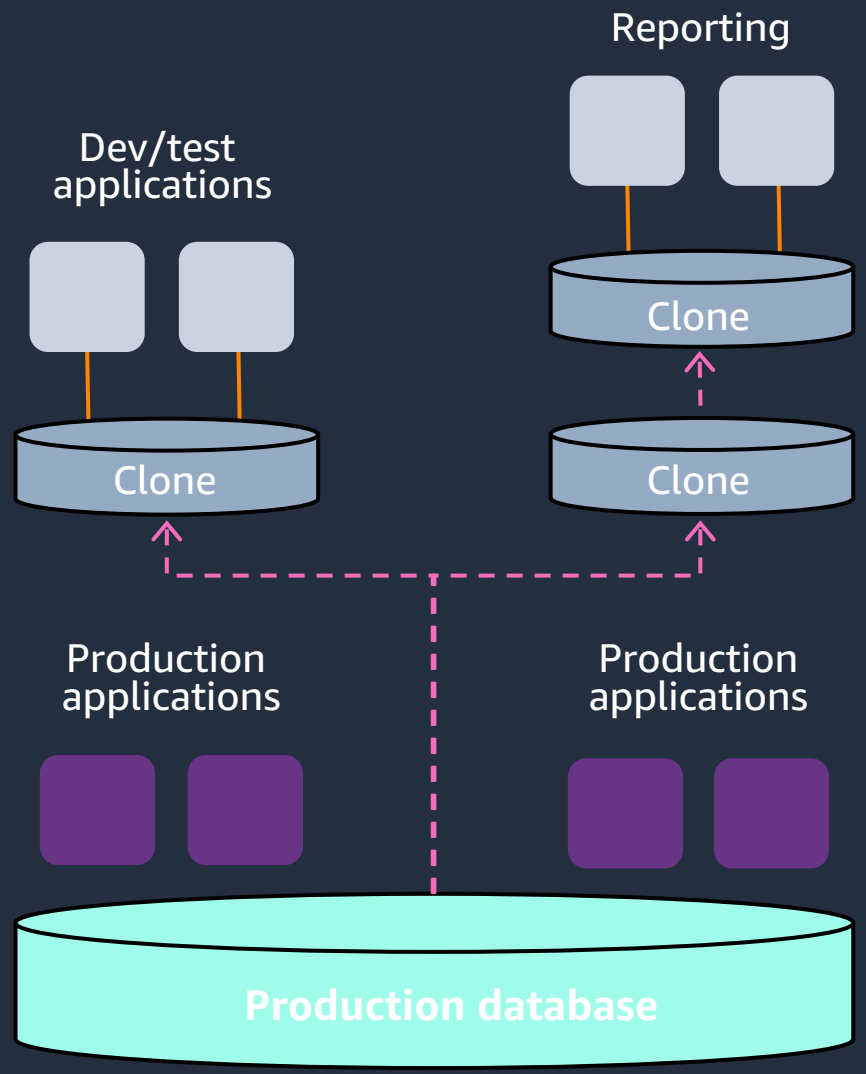
Fast clones



Using fast database clones

- **Use cases:**
- Test changes in pre-prod on relevant data sets
- Reorganize a DB with minimal impact (clone + CDC)
- Save a point in time snapshot for reporting/analytics with no impact on prod/OLTP
- Failback for maintenance activities

- **Using clones:**
- Up to 15 clones from same source
- 2 step process: clone cluster + add instances (optional)
- Clone starts with same DB engine version as source
- Clone across AWS accounts with resource sharing



Security

Authentication

SQL Server

- SQL Server authentication
- Windows authentication
- SQL Server + Windows authentication

Aurora PostgreSQL

- PostgreSQL native authentication
- IAM authentication
- Kerberos authentication
- Integrated with AWS Secrets Manager for credential management and rotation of native credentials

Encryption

SQL Server

- TDE
- Always Encrypted
- Encryption in transit – TLS

Aurora PostgreSQL

- AES256-based storage encryption, incl. backups, snapshots and metadata (Performance Insights, CloudWatch Logs publishing, Enhanced Monitoring)
- Key management using AWS KMS
- Encryption in transit – TLS
- Pgcrypto extension

Performance Monitoring

Performance monitoring tools

SQL Server

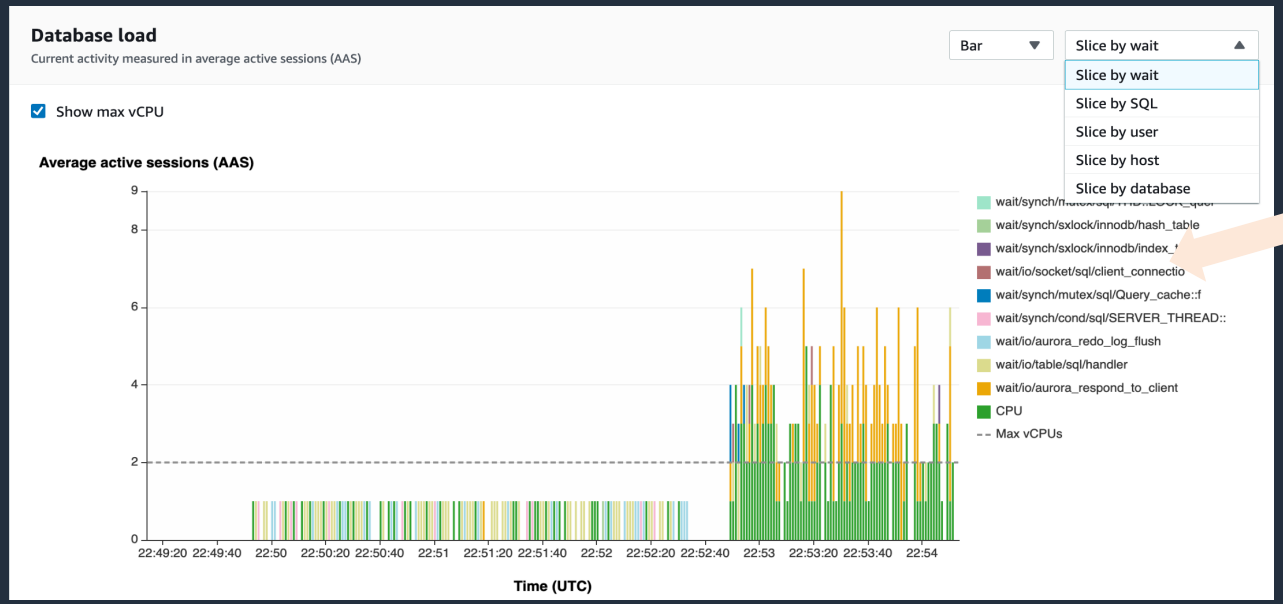
- Windows Perfmon
- SQL Server DMVs
- Extended Events

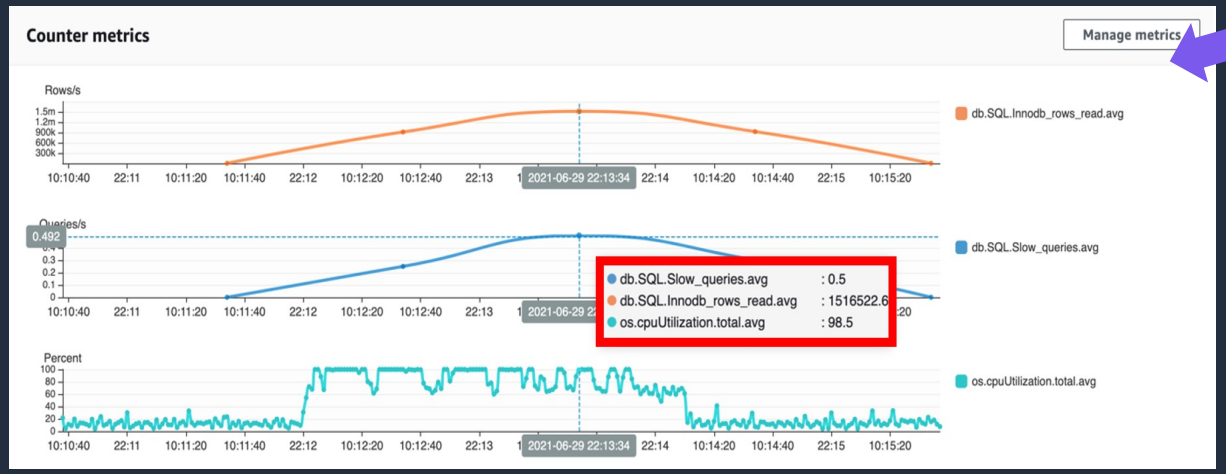
Aurora PostgreSQL

- CloudWatch (metrics, events, logs)
- Enhanced Monitoring
- Performance Insights
- PostgreSQL Statistics views
- PostgreSQL logging options

Performance Insights

- Easy and powerful dashboard showing load on your database
- Uses Average Active Session (AAS) as a load aggregation method over time
- Helps you identify source of bottlenecks: top SQL queries, wait statistics, DB engine counters
- Adjustable time frame (hour, day week, month)
- 7 days of performance data history free – perfect for developers; up to 2 years of long term retention for production use cases





Migration Options

Migration options

1

Aurora PostgreSQL

Using AWS Database Migration Service (DMS) and Schema Conversion Tool (SCT) for conversion and data migration

+

porting application T-SQL code to PL/pgSQL

2

Babelfish for Aurora PostgreSQL

Using Babelfish Compass to assess compatibility

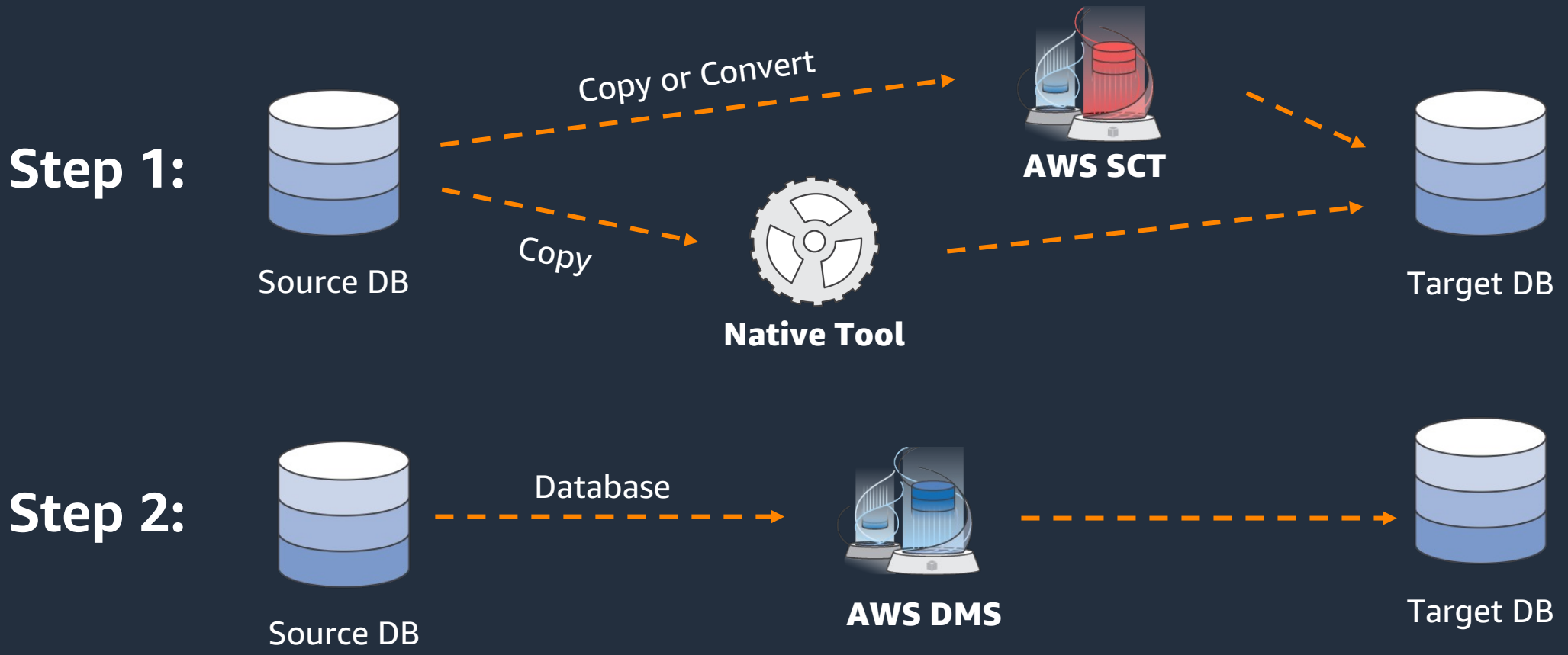
Using AWS Database Migration Service (DMS) w/ Babelfish as target endpoint for data migration

+

Little to no application T-SQL code changes

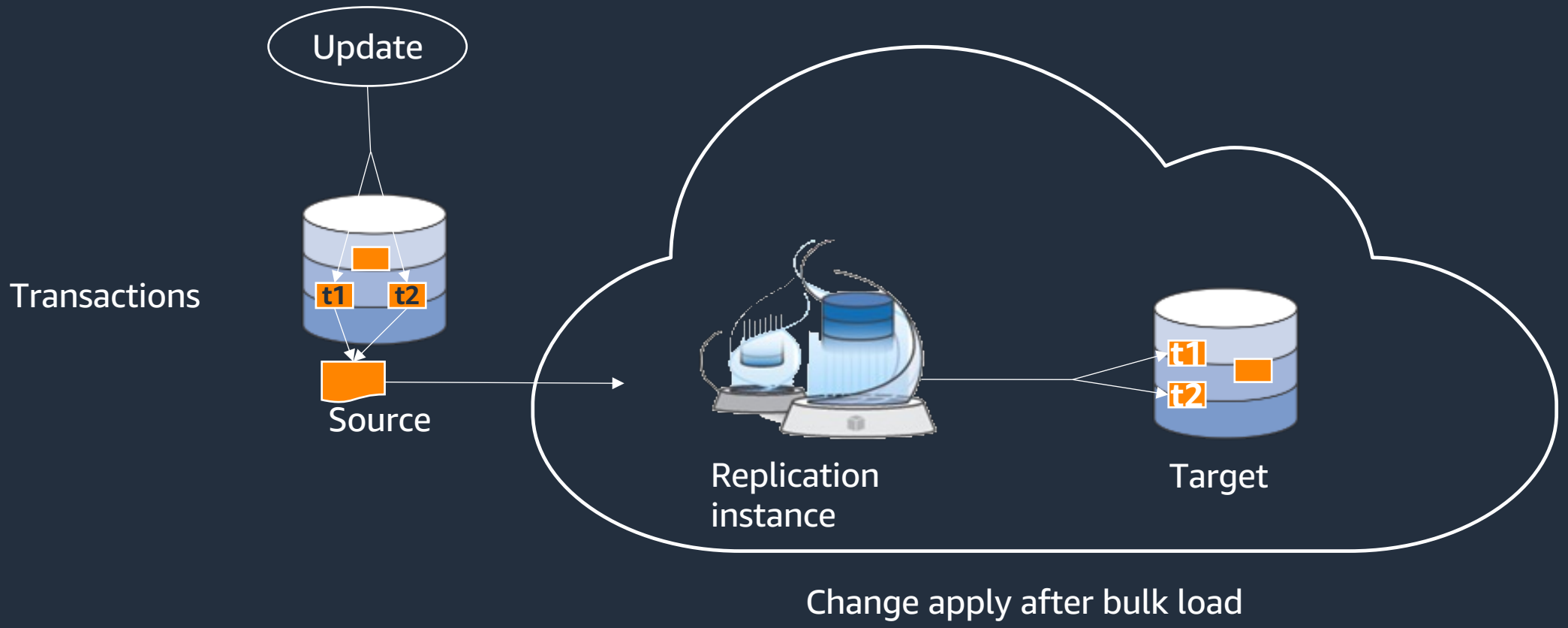
How does DMS work? (1)

First, convert schemas and perform initial data loads

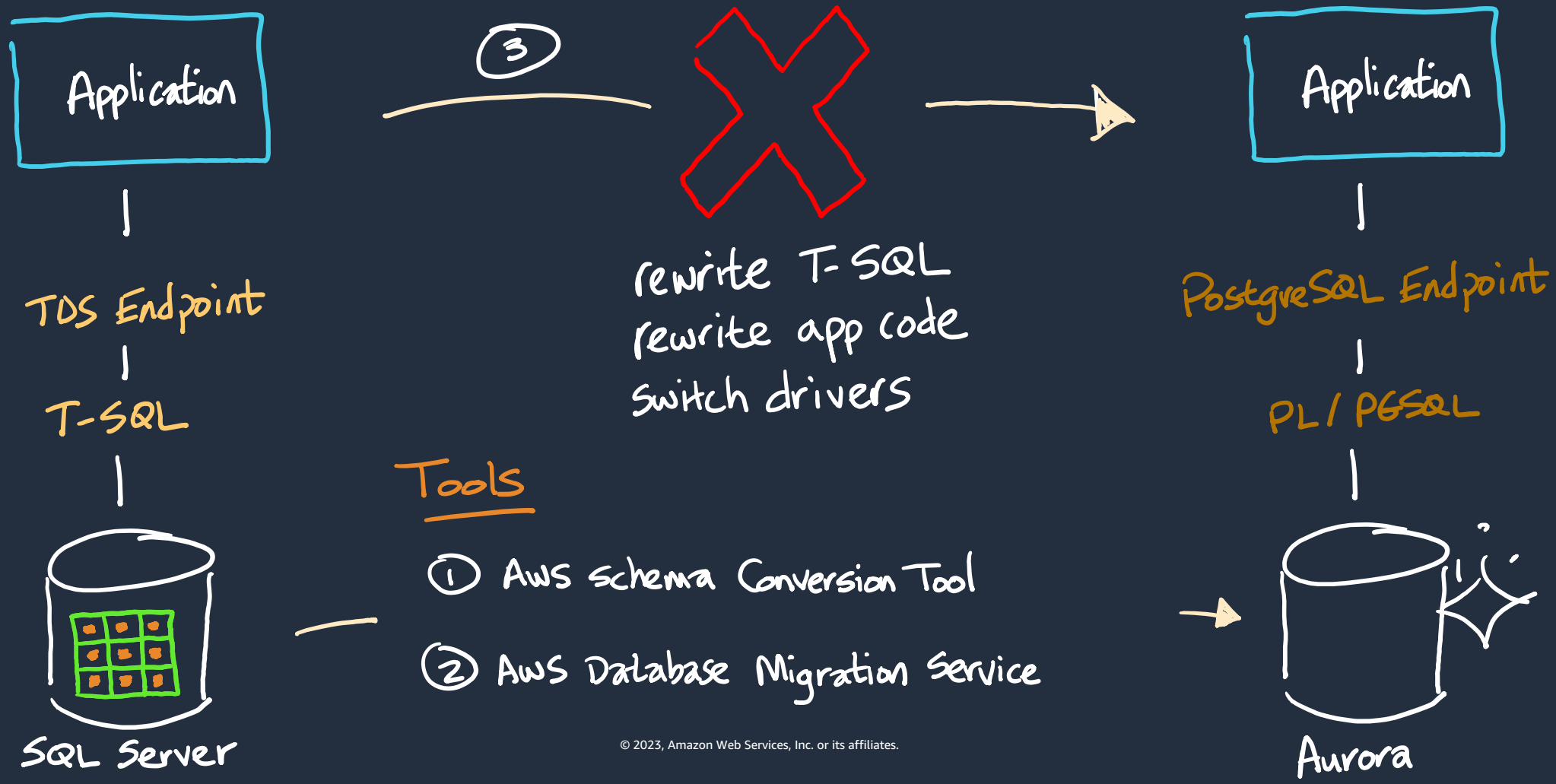


How does DMS work? (2)

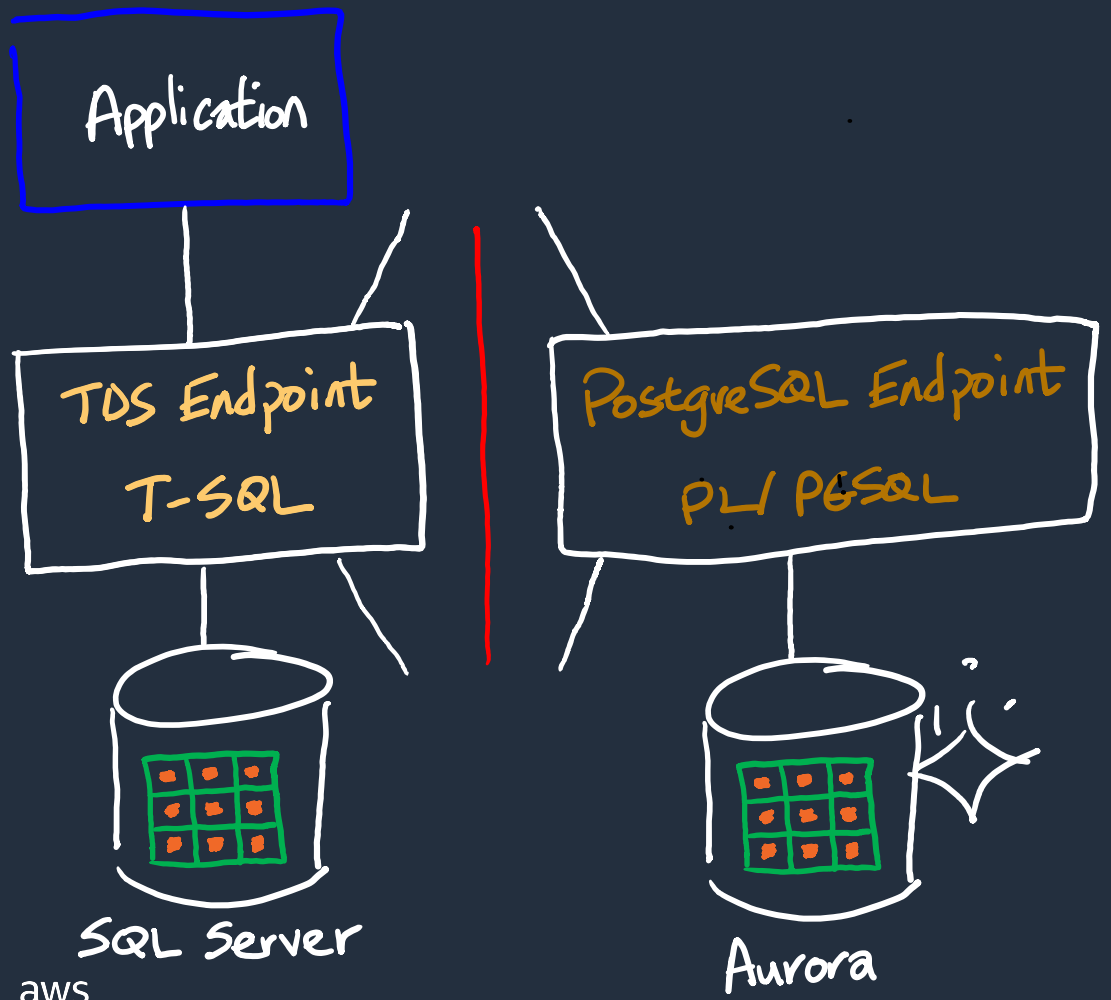
Change data capture (CDC) and apply



Challenges in migrating from commercial to open source



Imagine if you could . . .



1. Legacy application code remains written for SQL Server
2. Client drivers do not need to be changed
3. New application code written directly to PostgreSQL

aws

Introducing Babelfish for Aurora PostgreSQL

Run SQL Server applications on PostgreSQL with little to no code changes

Keep existing queries

Translation layer enables Aurora PostgreSQL to understand Microsoft SQL Server's proprietary T-SQL

Accelerate migrations

Lower risk and complete migrations faster, saving you months to years of work

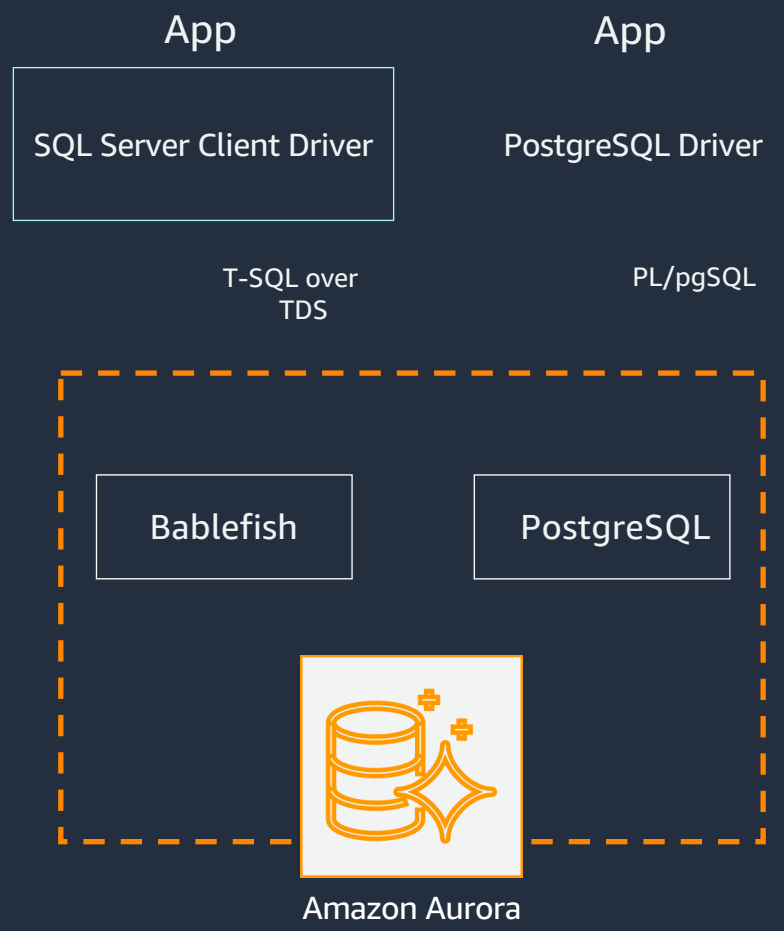
Freedom to innovate

Run T-SQL code side-by-side with new open source functionality and continue developing with familiar tools

Bablefish for Aurora PostgreSQL

How it works

- Adds an endpoint to PostgreSQL that understands the SQL server wire protocol Tabular Data Streams (TDS), and commonly used T-SQL programming language constructs used by SQL Server.
- Provides support for T-SQL including:
    - SQL dialect
    - Cursors
    - Catalog views
    - Data types
    - Triggers
    - Stored procedures
    - Functions



Thank you!