AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DAT302

Instant and fine-grained scaling with Amazon Aurora Serverless v2

Chayan Biswas
Sr. Manager, Product Management
Amazon Web Services

Anum Jang Sher
Sr. Product Manager
Amazon Web Services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Amazon Aurora Serverless v2 deep dive

Best practices

Getting started with Aurora Serverless v2

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora

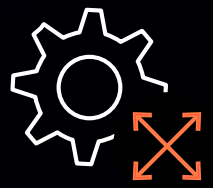
Aurora MySQL and Aurora PostgreSQL compatible relational database built for the cloud – performance and availability of commercial databases at 1/10th the cost



Performance and scalability

Several times faster than standard Aurora MySQL and Aurora PostgreSQL

15 read replicas



Availability and durability

Fault-tolerant self-healing storage

6 copies of data across 3 AZs

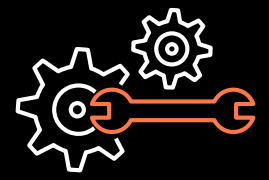
Single global database with cross-region replication



Highly secure

Network isolation

Encryption at rest/transit



Fully managed

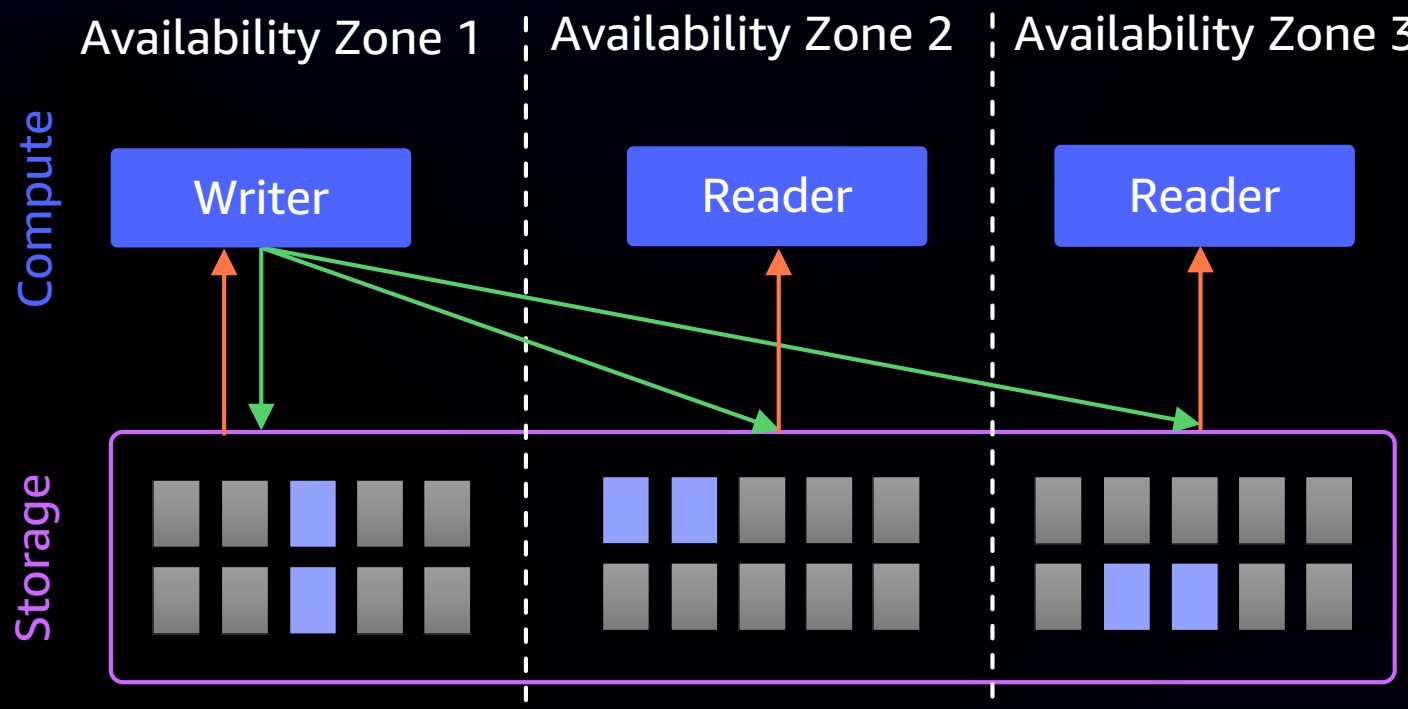
Managed by RDS: no hardware provisioning, software patching, setup, configuration, or backups

The fastest growing service in the history of AWS

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora architecture supports serverless



Purpose-built, log-structured, distributed storage designed for cloud databases

6 copies across 3 AZs for high availability, durability, and performance

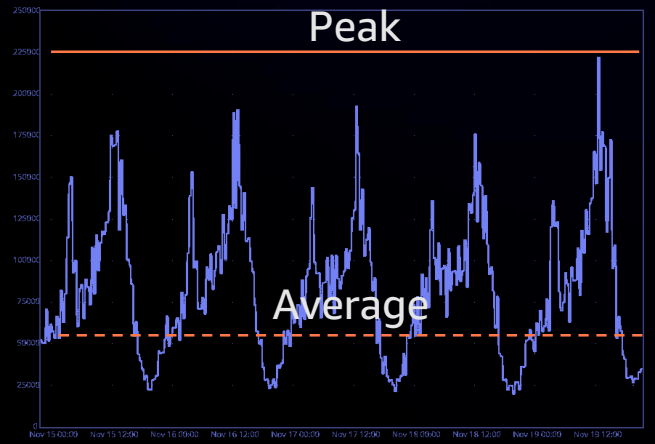
Separation of storage and compute:

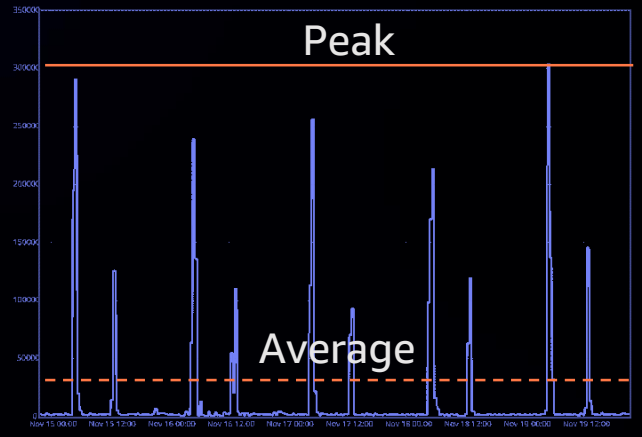
- Routine operations (e.g., backups) done without compute
- Up to 15 low-latency (<10 millisecond) read replicas
- **Compute scales independently**

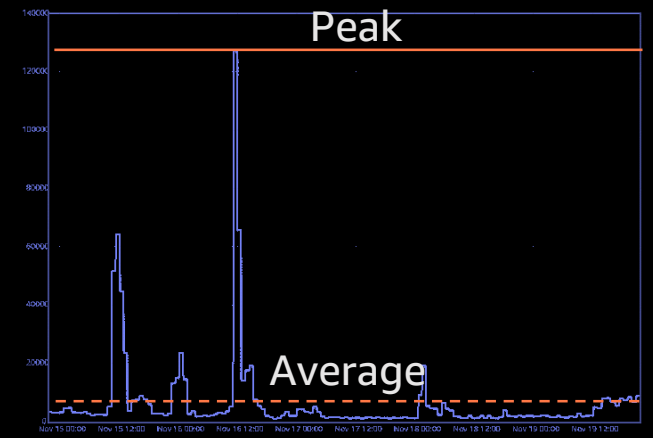
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Database capacity: Cost vs. management

Variable and unpredictable workloads







Insufficient capacity → Experience degradation

Provision for peak → Expensive

OR

Continuously monitor and scale → Difficult, requires experts, involves downtime

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Serverless v2 (preview)



On-demand and autoscaling configuration

Automatically scales capacity based on application needs

Simple **pay-per-use** pricing per second

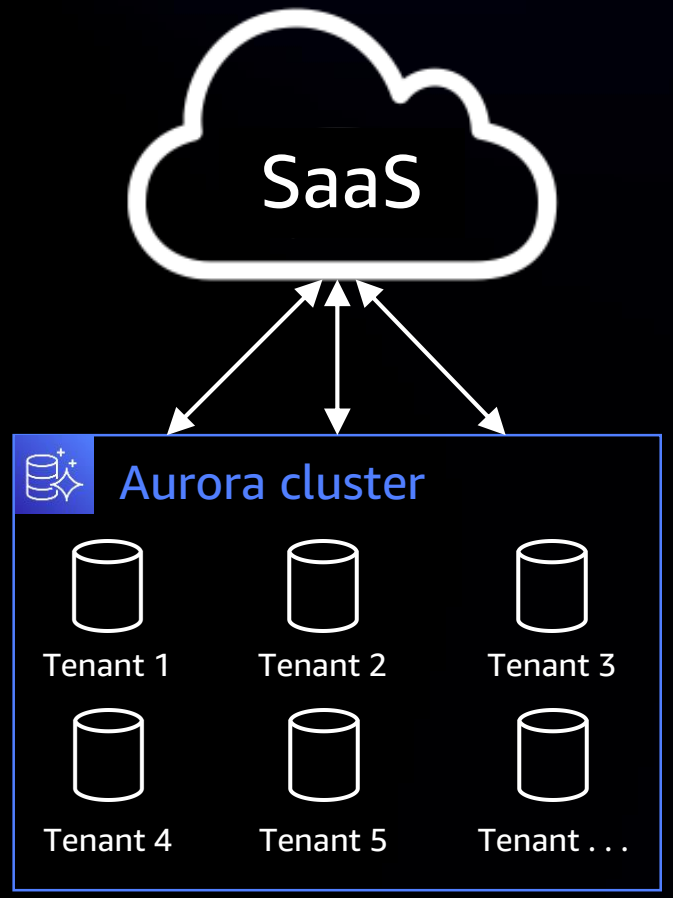
Next version scales instantly to support demanding applications

Worry-free database capacity management

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-tenant applications with per-tenant databases



Thousands of end users, each with their own database

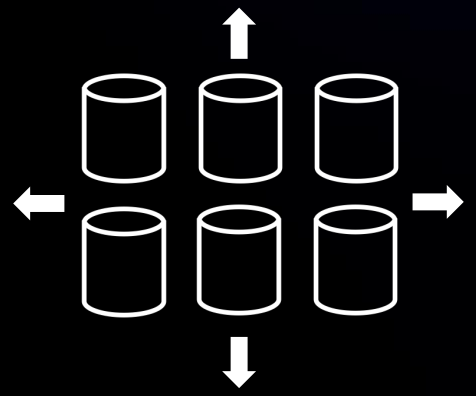
Colocate databases to improve utilization and cost efficiency

Tradeoff granularity of operational controls (e.g., backups and restores)

Requires continuous monitoring and shuffling due to busy workloads

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Several application backed by databases



100s to 1,000s of databases

Customers with 100s or 1,000s of applications

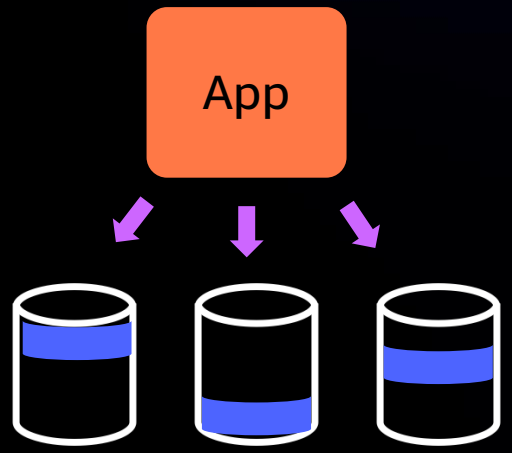
Each application backed by one or more databases

Evolving application requirements require database capacity adjustments

Managing capacity for a database fleet on a budget is daunting

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaled out databases



Applications needing write scalability split their database across multiple nodes for higher throughput

Predicting capacity for each node is hard and inefficient

Create too few nodes, you've to redistribute data taking downtime

Create too many nodes and pay for high costs as all nodes are not equally utilized

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Demanding workloads require . . .



Instant and non-disruptive capacity scaling



Fine-grained and predictable capacity adjustments

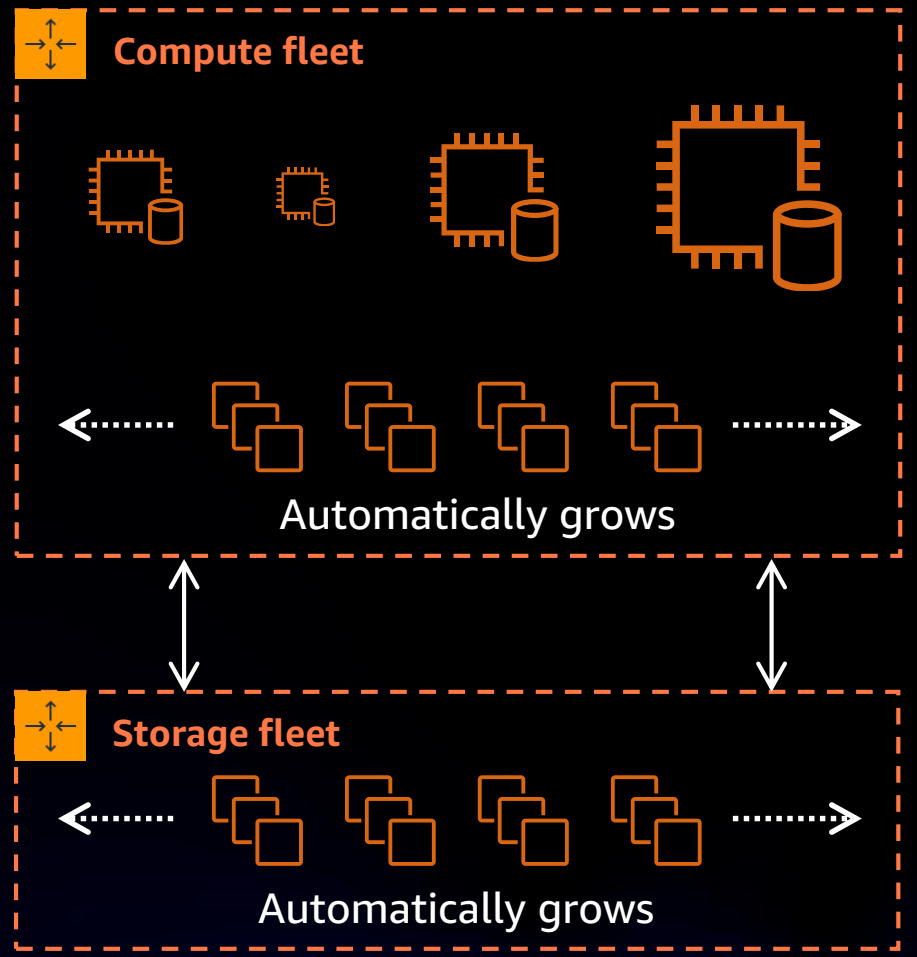


High availability, disaster recovery, and enhanced capabilities

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Instant, in-place scaling

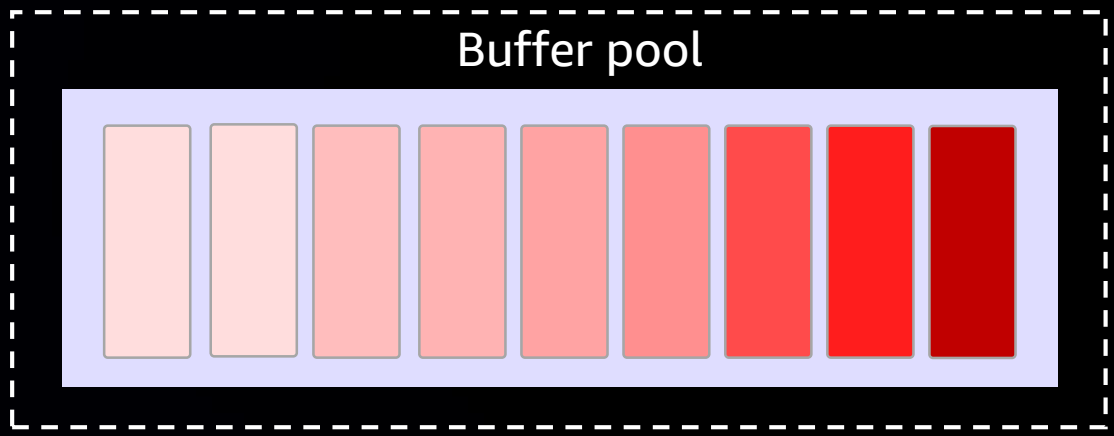
- Scales in place in under a second by adding more CPU and memory resources
- No impact due to scaling even when running hundreds of thousands of transactions
- Compute fleet continuously monitored and scaled horizontally for heat management
- Background movement of idling instances while preserving state (e.g., buffer pool, connections)
- Up to 15x faster scale downs



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

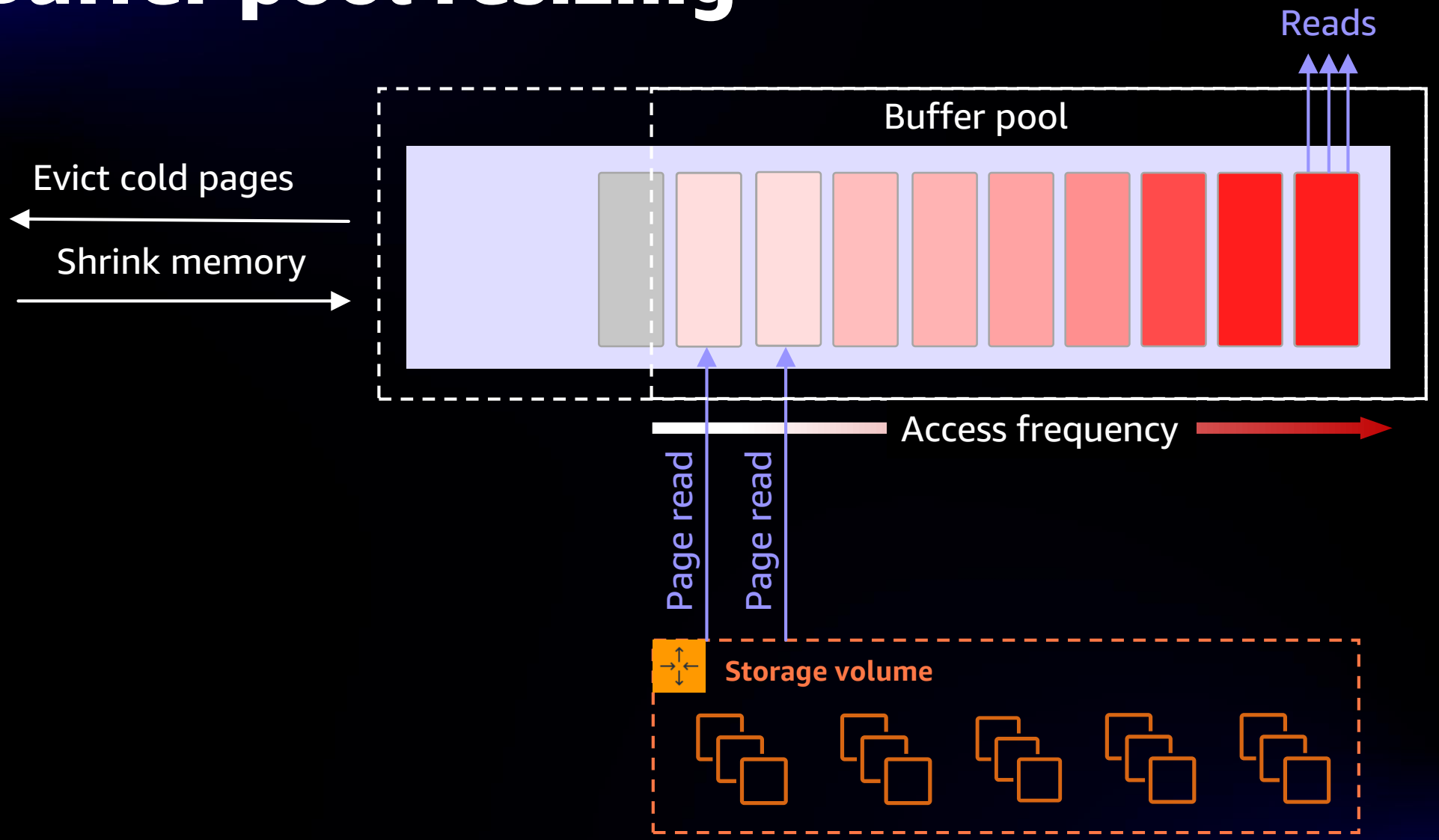
Buffer pool resizing

- Buffer pool size scaled along with database capacity
- Parameters automatically adjusted:
  - MySQL: innodb_buffer_pool_size
  - PostgreSQL: shared_buffers
- Memory allocation: 75% for buffer pool and 25% for heap
- Buffer pool scaled down through a combination of least frequently used (LFU) and least recently used (LRU) algorithms



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

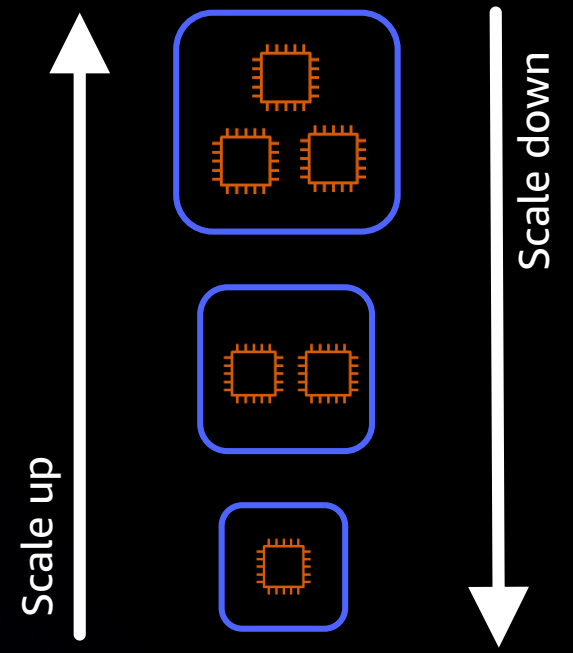
Buffer pool resizing



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Fine-grained capacity adjustments

- Aurora Serverless capacity is measured in Aurora Capacity Unit (ACU)
- 1 ACU comes with 2 GiB of memory
- CPU and networking similar to what is available in provisioned Aurora instances
- Starting capacity as small as 0.5 ACU (1 GiB)
- Fine-grained scaling with as little as 0.5 ACU increments

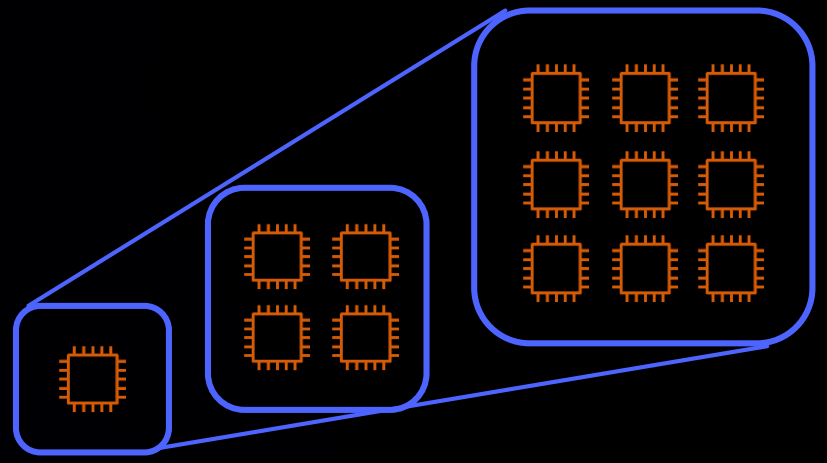


aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling factors

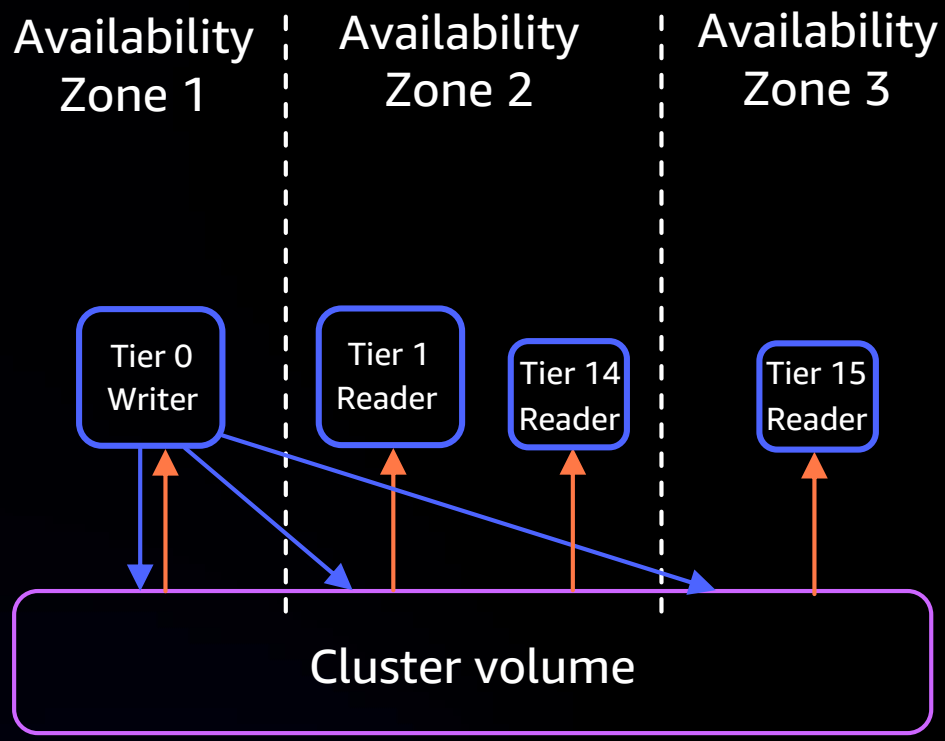
- Scale-up rate is predictable and proportional to current capacity – larger instances scale up faster
- CPU utilization of both foreground and background processes (e.g., purge or vacuum)
- Memory utilization of internal data structures (e.g., buffer pool)
- Network throughput is proportional to capacity; capacity is scaled to match network throughput needs



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

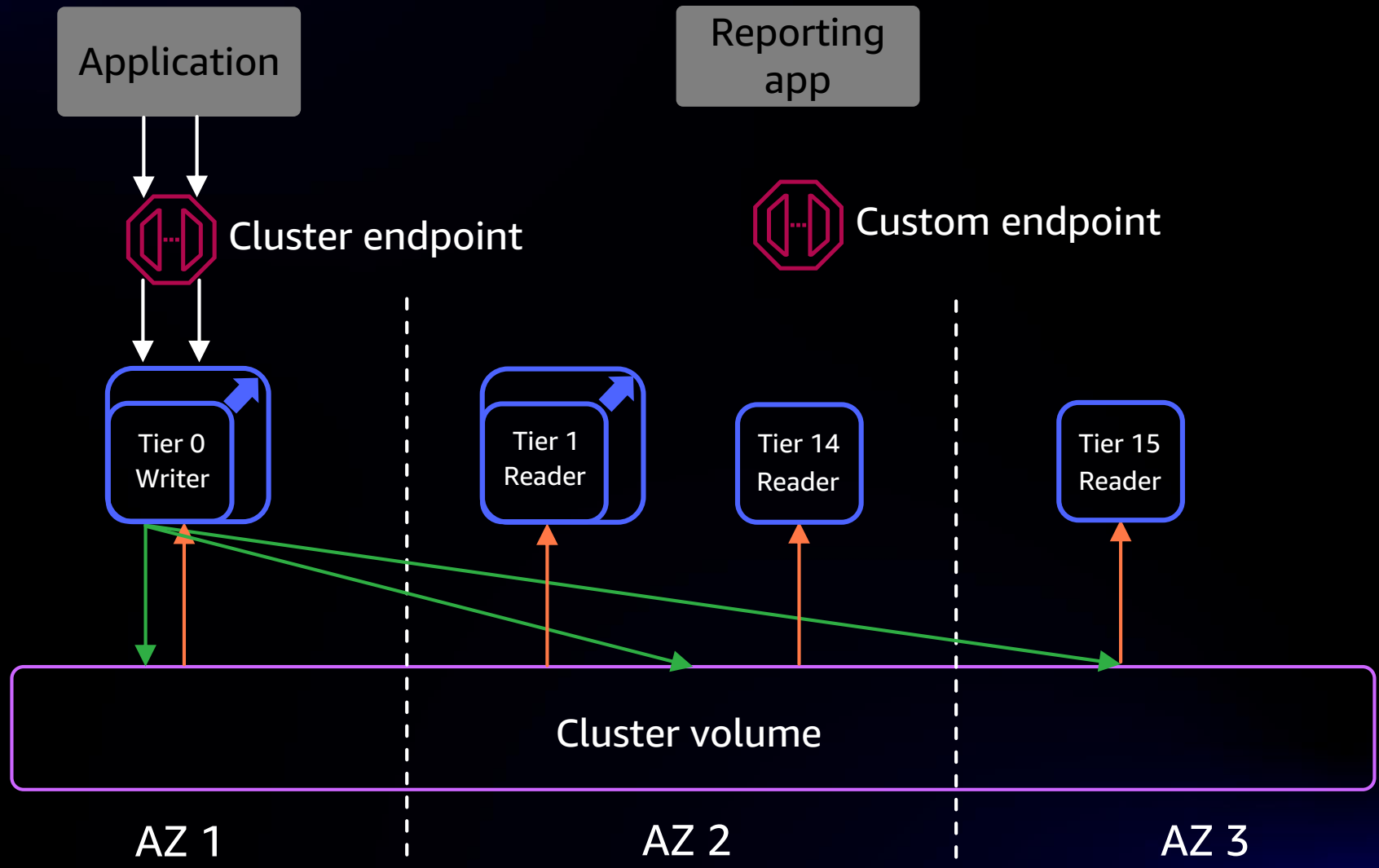
High availability and read scaling

- Up to 15 read replicas act as failover targets
- All instances inherit capacity configuration from the cluster
- Tier 0 and 1 read replicas match the size of the primary instance
- Deploy across separate AZs
- Multi-AZ Aurora clusters supported by 99.99% uptime SLA



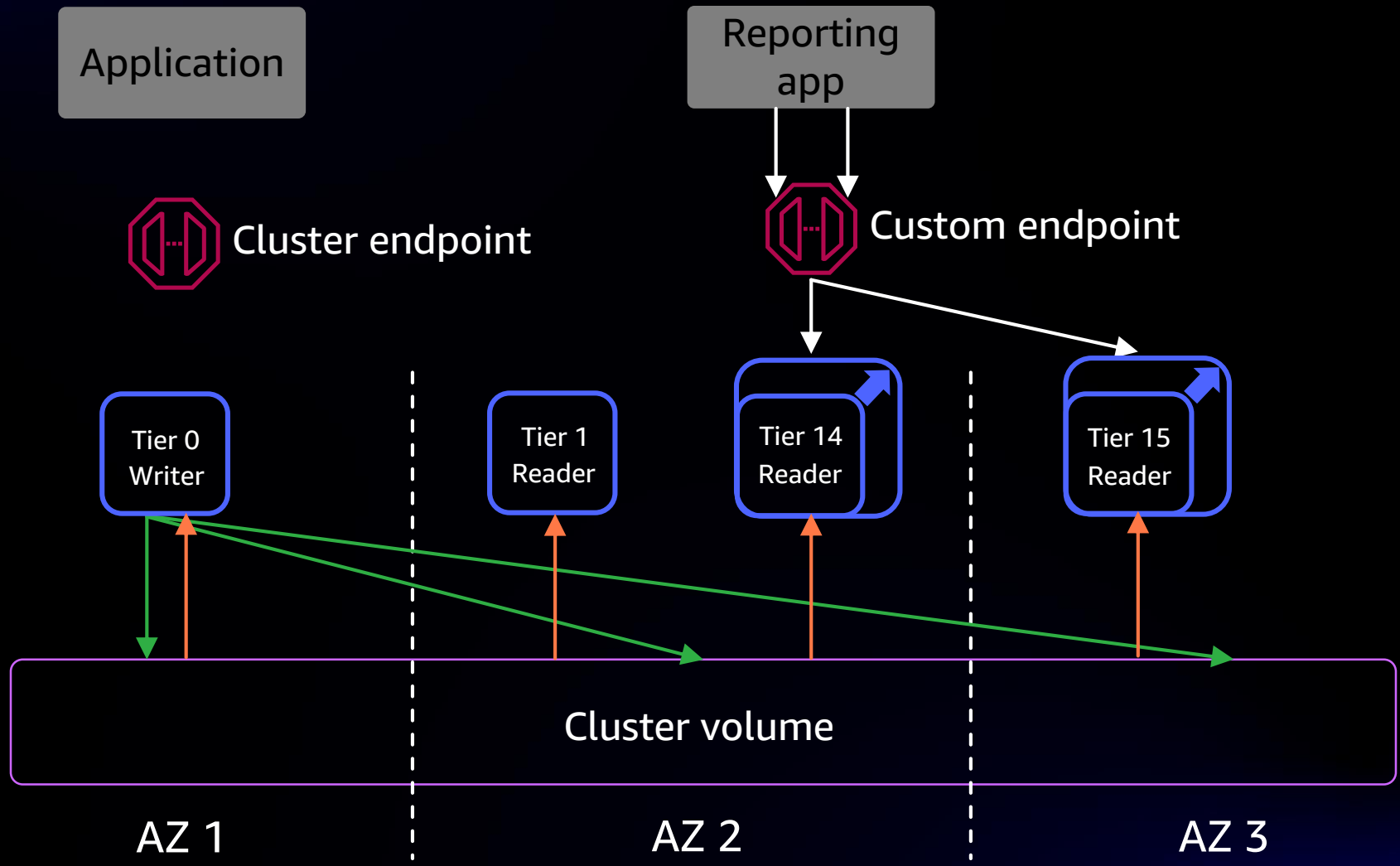
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-AZ high availability



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

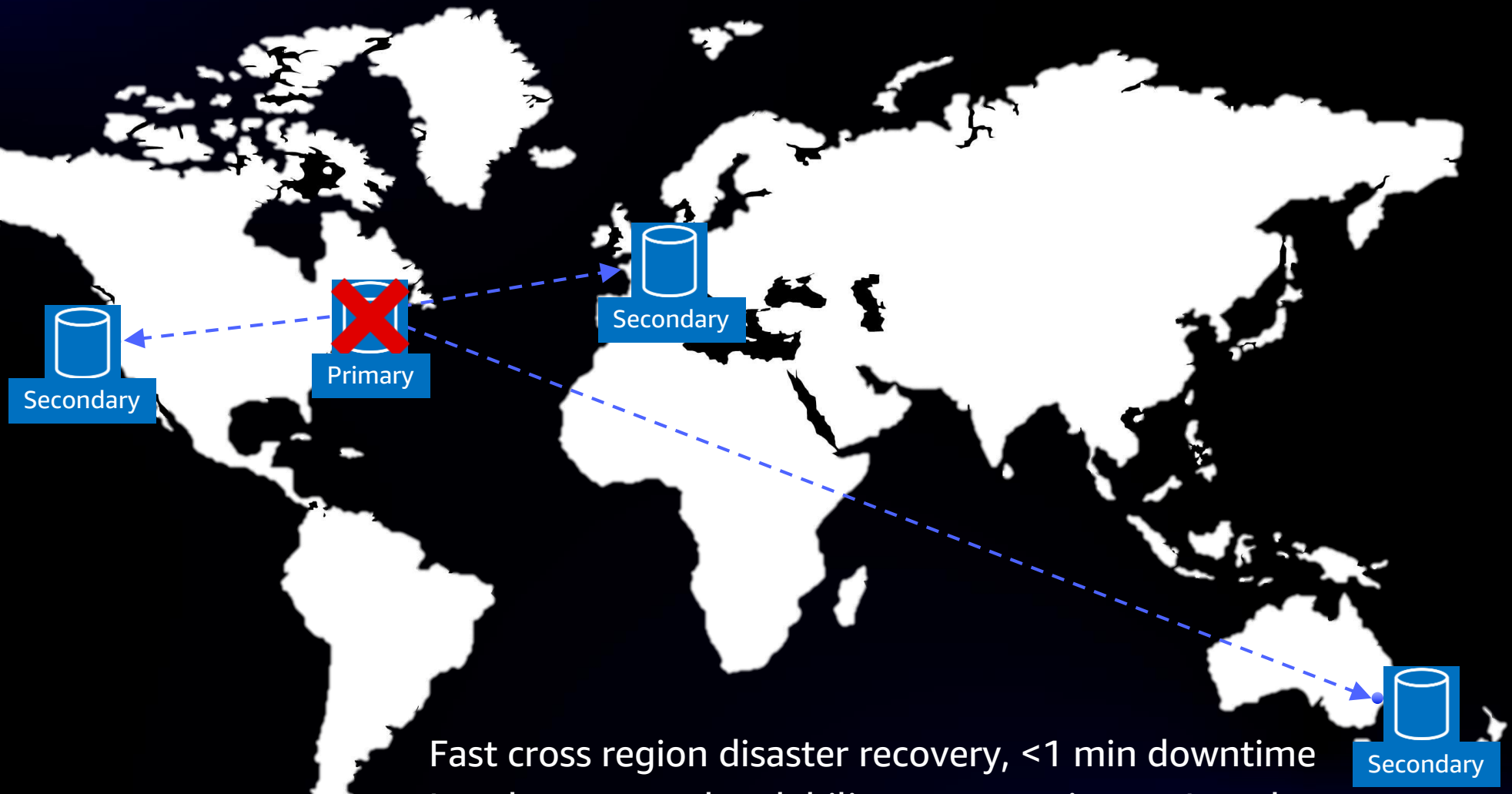
Read scalability



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora Global Database



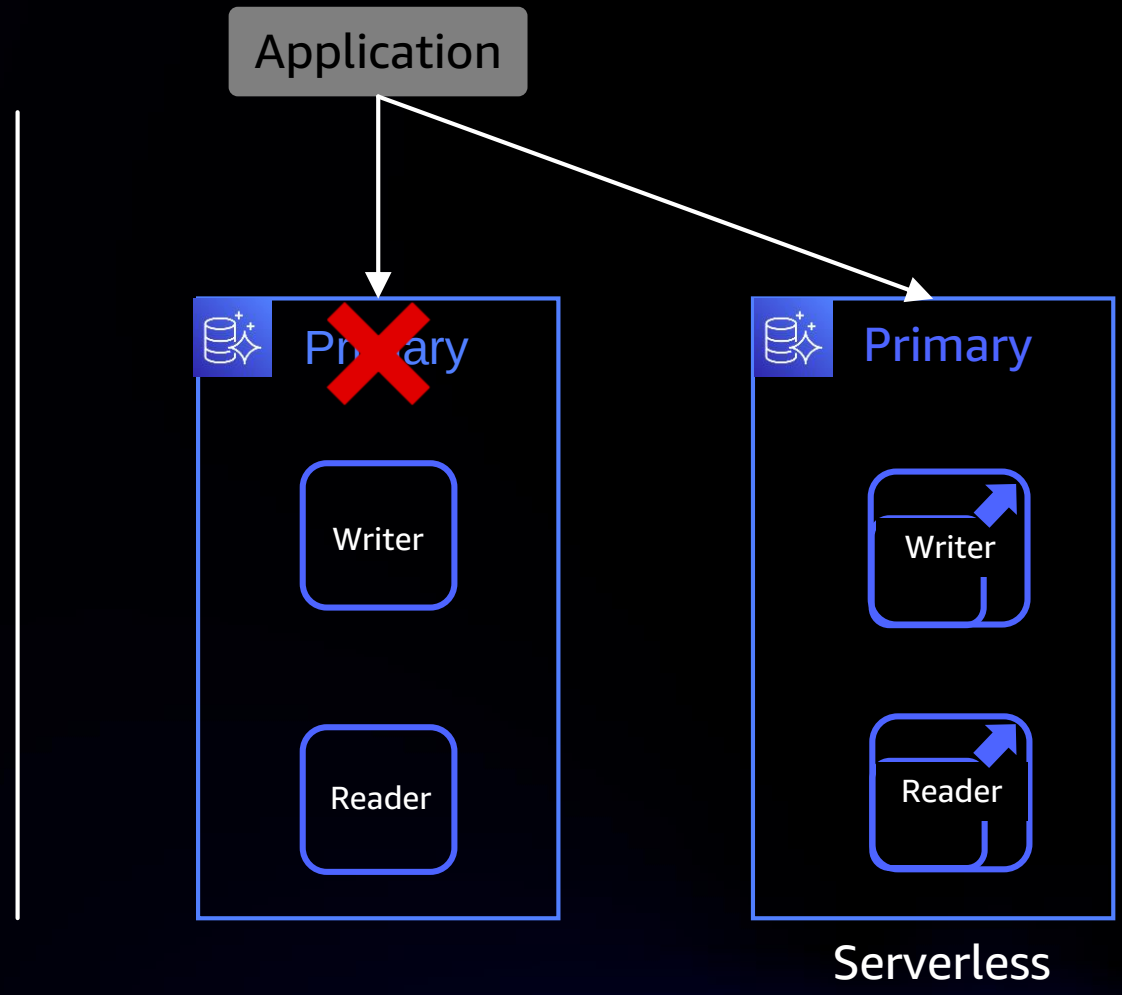
Fast cross region disaster recovery, <1 min downtime
Low latency read scalability across regions, <1 sec lag
Up to 5 secondary regions, 90 read replicas

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless global database

- Configure secondary regions with Aurora Serverless v2
- Pay for only minimal capacity when idling
- Writer and readers scale up after failover
- Scale secondary regions independently for reads closer to end users



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Supported major versions

Aurora MySQL 8.0

- Windowing functions
- Atomic and online DDL
- SKIP LOCKED and NOWAIT options
- Improved JSON functionality

Aurora PostgreSQL 13

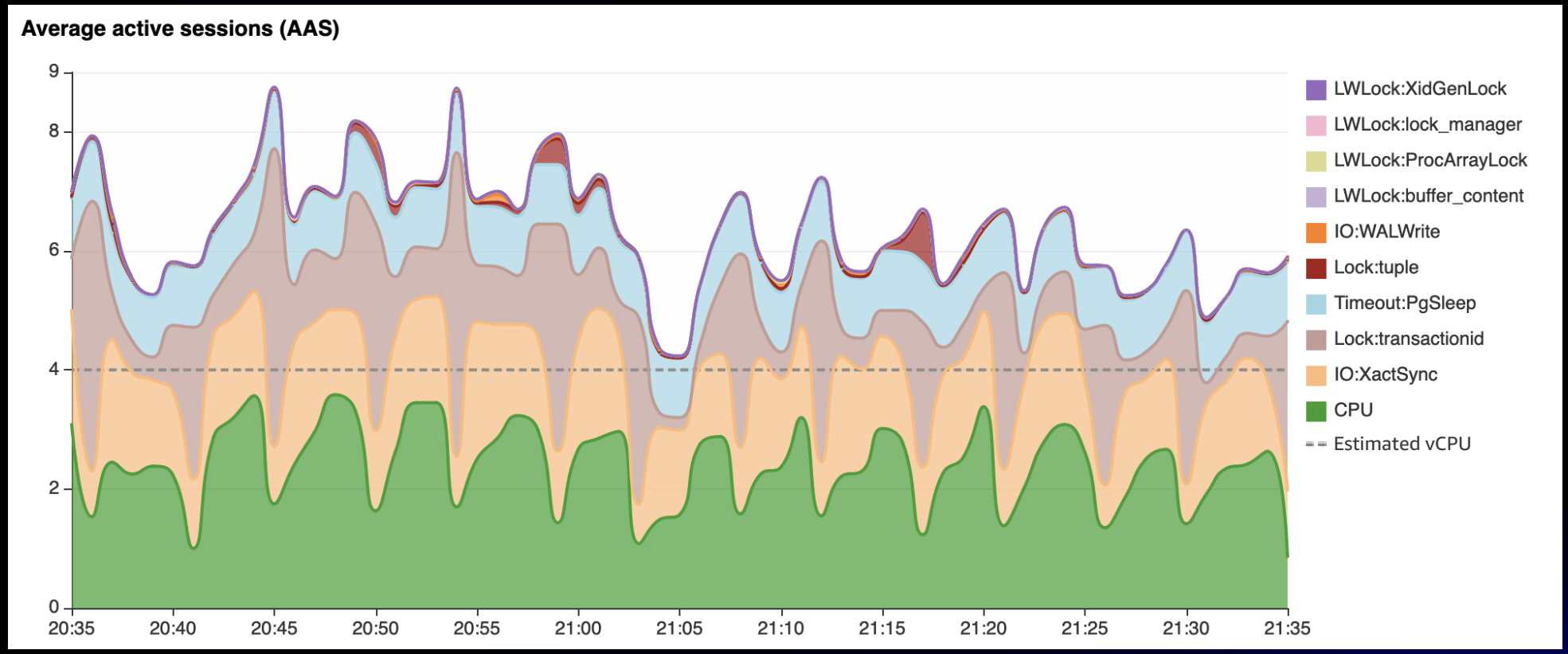
- Deduplication in B-tree indexes
- Improved partitioned table performance
- Incremental sorting for faster sorts
- Parallelized vacuum for indexes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS performance insights

MONITOR AND TUNE DATABASE PERFORMANCE

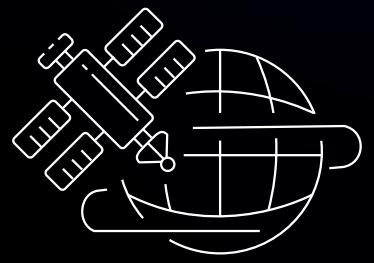
- Easy and powerful dashboard to visualize database load
- Investigate database performance issues, even if you're not a database expert



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS Proxy

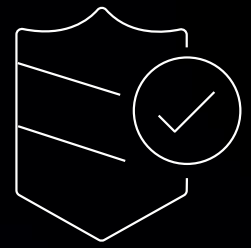
A FULLY MANAGED, HIGHLY AVAILABLE DATABASE PROXY FOR AMAZON RDS AND AMAZON AURORA



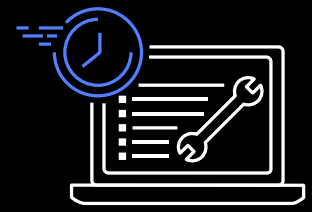
Pool and share DB connections for improved app scaling



Increase app availability and reduce DB failover times



Manage app data security with DB access controls



Fully managed DB proxy, compatible with your database

Amazon RDS Proxy will support Aurora Serverless v2, including mixed configurations with Aurora provisioned and serverless instances within a cluster

aws

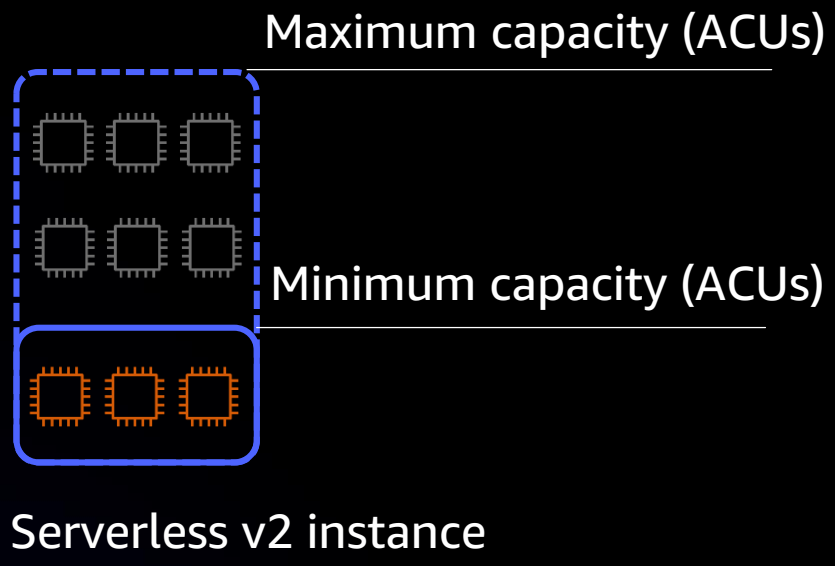
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Configuring capacity

- Minimum capacity determines the starting capacity of the instance
- Adjust minimum capacity for a suitable scaling rate
- Increase minimum capacity ahead of anticipated spike
- Maximum capacity is a budget control
- Configure maximum to support peak workload and features (e.g., performance insights)

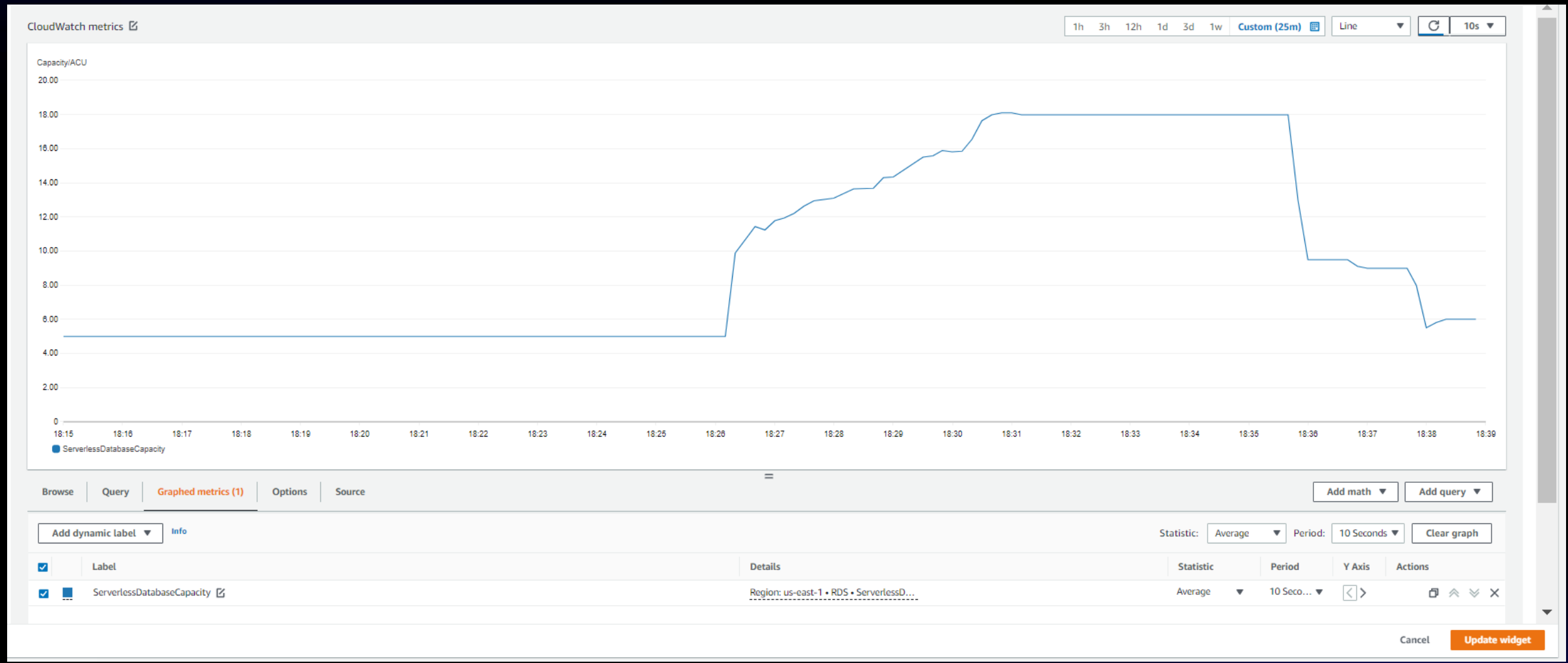


aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch metrics

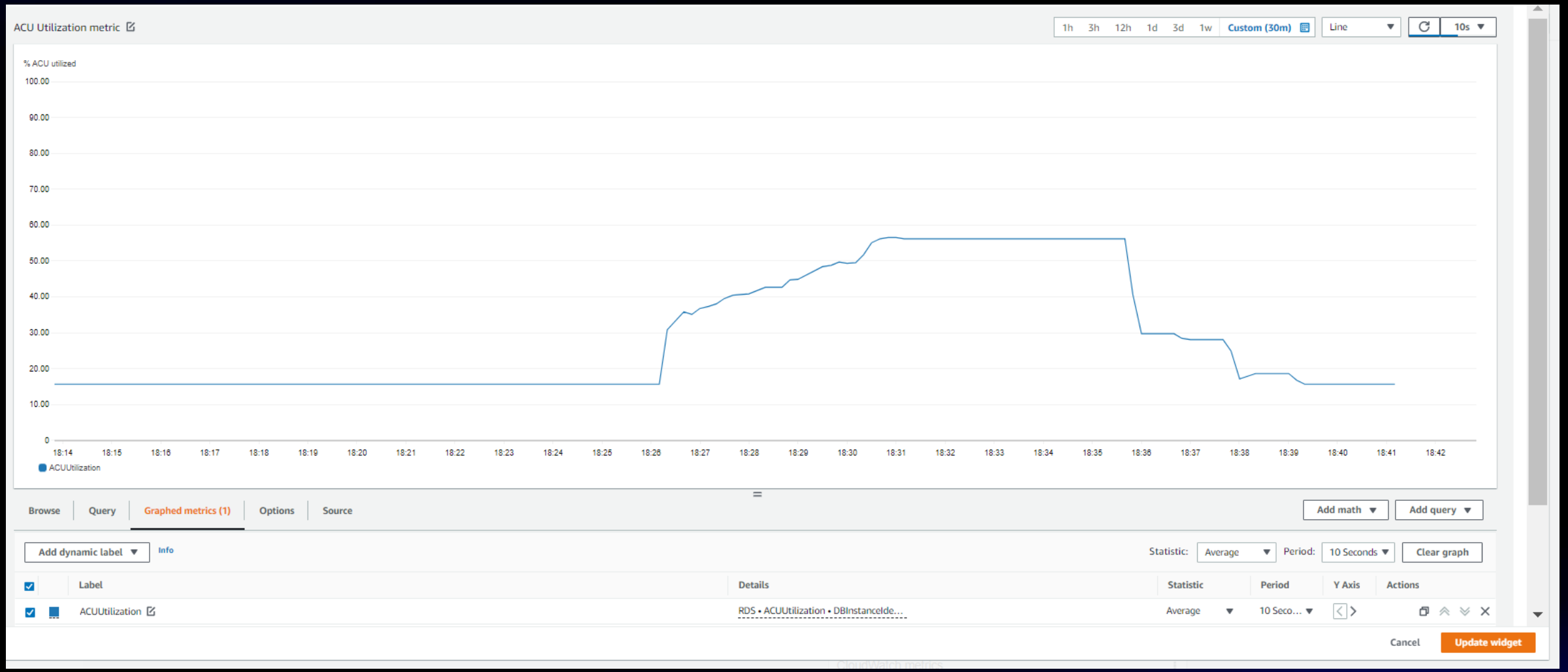
SERVERLESSDATABASECAPACITY (ACUS) – CURRENT CAPACITY OF THE INSTANCE IN ACUS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch metrics

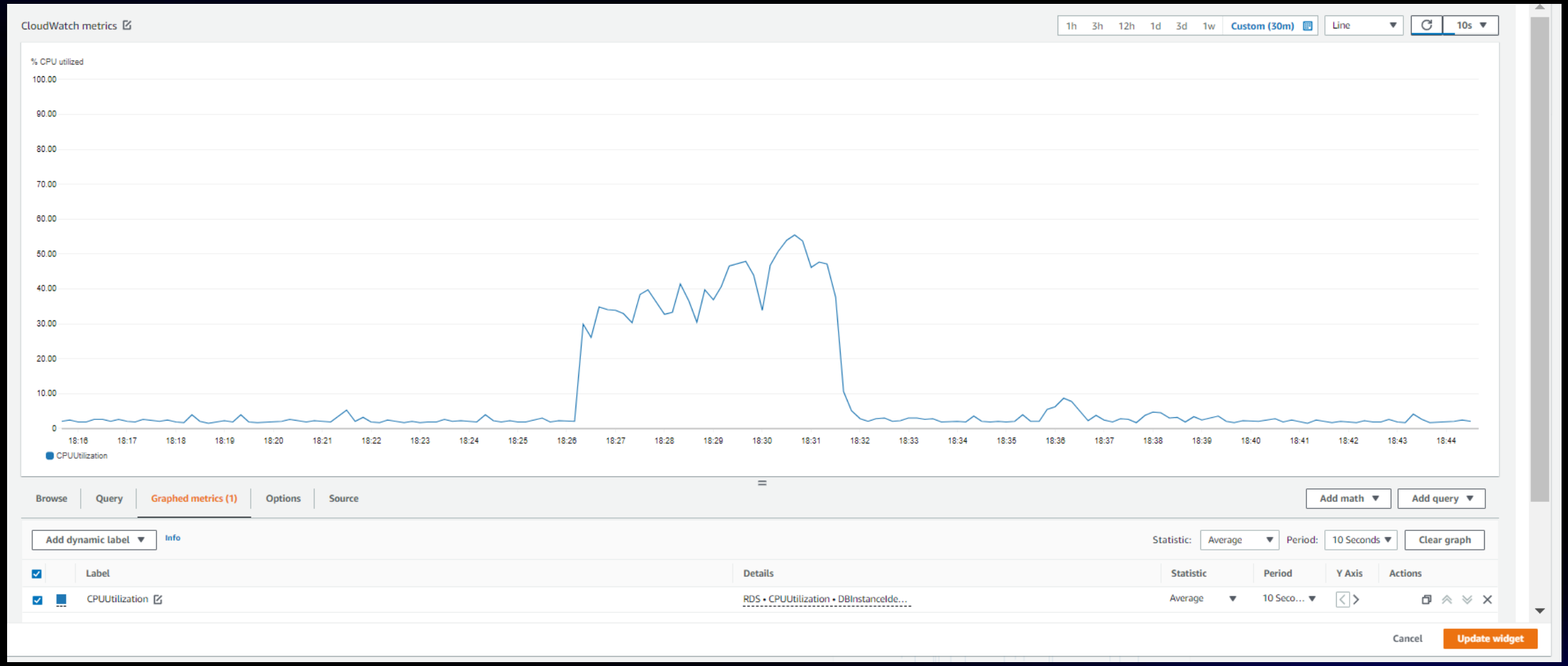
ACUUTILIZATION (%) – PERCENTAGE OF CURRENT CAPACITY OUT OF SPECIFIED MAXIMUM CAPACITY



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch metrics

CPUUTILIZATION (%)– PERCENTAGE OF CPU, BASED ON MAXIMUM ACU, CURRENTLY USED BY THE INSTANCE



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch metrics

FREEABLEMEMORY (GIB) – PERCENTAGE OF CURRENT CAPACITY OUT OF SPECIFIED MAXIMUM CAPACITY



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Parameter handling – Aurora MySQL

Parameters that change during scaling

Name	Values	Allowed values	Description
innodb_buffer_pool_size	{DBInstanceClassMemory*3/4}	335544320-9223372036854775807	The size in bytes of the memory buffer innodb uses to cache data and indexes of its tables
table_definition_cache	LEAST({DBInstanceClassMemory/393040}, 20000)	-1, 400-524288	The number of table definitions that can be stored in the definition cache
table_open_cache	LEAST({DBInstanceClassMemory/1179121}, 6000)	1-524288	The number of open tables for all threads. Increasing this value increases the number of file descriptors.

Parameters that change when max ACU configuration is changed

Name	Values	Allowed values	Description
max_connections	GREATEST({log(DBInstanceClassMemory/805306368)*45}, {log(DBInstanceClassMemory/8187281408)*1000})	1-16000	The number of simultaneous client connections allowed.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Parameter handling – Aurora PostgreSQL

Parameters that change during scaling

Name	Values	Allowed values	Description
shared_buffers	SUM({DBInstanceClassMemory/12038},-50003)	16-1073741823	(8kB) Sets the number of shared memory buffers used by the server.

Parameters that change when max ACU configuration is changed

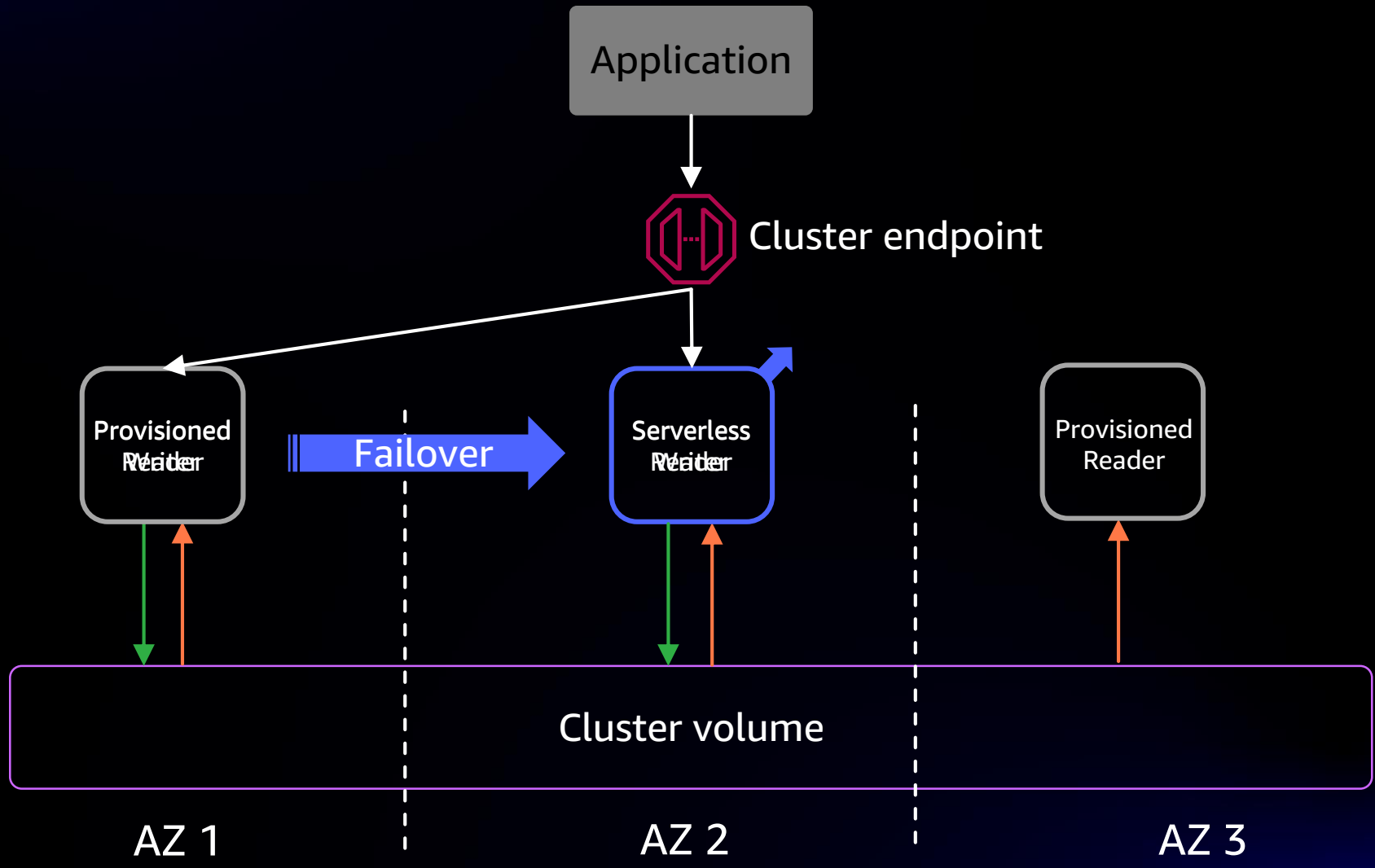
Name	Values	Allowed values	Description
effective_cache_size	SUM({DBInstanceClassMemory/12038},-50003)	1-2147483647	(8kB) Sets the planners assumption about the size of the disk cache.
autovacuum_max_workers	GREATEST({DBInstanceClassMemory/64371566592},3)	1-8388607	Sets the maximum number of simultaneously running autovacuum worker processes.
autovacuum_vacuum_cost_limit	GREATEST({log(DBInstanceClassMemory/21474836480)*600},200)	-1-10000	Vacuum cost amount available before napping, for autovacuum.
autovacuum_work_mem		-1-2147483647	(kB) Sets the maximum memory to be used by each autovacuum worker process.
max_connections	LEAST({DBInstanceClassMemory/9531392},5000)	6-262143	Sets the maximum number of concurrent connections.
max_worker_processes	GREATEST(${DBInstanceVCPU*2},8)	0-262143	Sets the maximum number of concurrent worker processes.
max_parallel_workers	GREATEST(${DBInstanceVCPU/2},8)	0-1024	Sets the maximum number of parallel workers than can be active at one time.
maintenance_work_mem	GREATEST({DBInstanceClassMemory/63963136*1024},65536)	1024-2147483647	(kB) Sets the maximum memory to be used for maintenance operations.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started

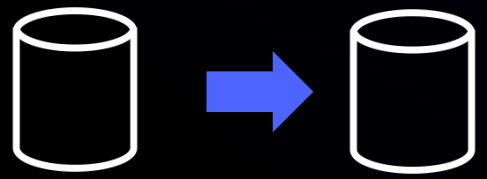
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora clusters in mixed configuration



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

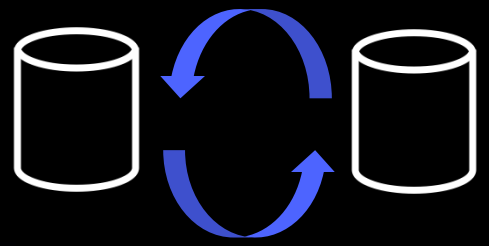
Additional choices for conversion



Cloning



Snapshot restore



Native replication

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Chayan Biswas
@cbbiswas

Anum Jang Sher
@anujangs

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.