Understanding AWS RDS

Aurora Capabilities

Percona Live Online
May 2021

Ronald Bradford - http://ronaldbradford.com
Overview

- What is Aurora?
  - Features & Capabilities
- Why consider Aurora?
- The various Aurora HA Setups
- Upsizing / Failover Example
- Aurora specific internals for MySQL architects & admins
- Other Aurora Features and Functionality
About Myself

- 20+ years MySQL experience in architecture and operations
- 15 years conference speaking
- Published author of 4 MySQL books

- Lead Data Architect/Engineer at Lifion by ADP

http://ronaldbradford.com
What is AWS RDS Aurora?

- Amazon Web Services (AWS)
- Relational Database Service (RDS)
  - MySQL/MariaDB/Postgresql/Oracle/SQL Server
- Aurora
  - MySQL and Postgres wire-compatible database built specifically for the AWS cloud

https://aws.amazon.com/rds/aurora
Aurora Features & Capabilities

- AWS managed RDBMS option
- Distributed cloud native architecture
- MySQL/Postgresql wire compatible
- A different transactional storage engine
- A different replication approach (read-free replicas)
- HA/Clustering/failover built-in by default
Aurora Features & Capabilities (2)

- Single writer/multiple readers
  - can support multi-master
- Decoupled compute/storage infrastructure
- Highly durable/redundant storage via quorum
- Log based architecture
- Improved recovery capabilities
- Fast DDL
Aurora Improved Availability, Backup & Recovery

- Fast recovery capabilities (log append design)
- Database cloning
- Snapshot restore
- Backtrack
- Zero Downtime Patching (ZDP)


Aurora Cluster Architecture Features

A cluster has:

- Data in 3 Availability Zones (AZ)
- 2 copies per AZ
- 4 of 6 need for Quorum
- Route 53 Cluster & Instance Endpoints
  - Writer, Reader, Custom (Cluster), Instance
- Automatic Instance failover
- Replica Autoscaling
Aurora Cluster - Single Instance

- **Cluster**
  - Storage in 3 AZs
  - Writer endpoint
  - Reader endpoint

- **Single instance**
  - In 1 AZ
  - Endpoint
  - Easily add additional instances

... (Diagram)
Aurora Cluster - Multiple Instances

- Cluster
- Writer endpoint
  - Primary
- Reader endpoint
  - Load balanced across non primary instance(s)
- Multiple instance(s)
  - AZs of choice
- Promotion Tiers
  ... (Diagram)
Cluster with Multiple Instances

AWS Region

Availability Zone 1

Availability Zone 2

Availability Zone 3

VPC

Primary

Replica Tier 0

Replica Tier 1

Cluster Volume

Writes

Reads

Reads

Reads

Cluster with Multiple Instances
Aurora Cluster - Multi-Master

- DB Instances are read & write
  - `--engine-mode multimaster`

Limitations

- Snapshots / ZDP / Load Balancing / Backtrack / Performance Insights
- Binary Logging
- Certain Datatypes
- Foreign Key CASCADE
- no fast DDL

Multiple Aurora Clusters (1)

- Same region option
- Uses MySQL binary log replication
  - Needs to be enabled
  - GTID not support > 5.7
- Blue/Green deployments
- Shorter downtime upgrades

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Replication.MySQL.html
Two separate clusters with binlog replication
Multiple Aurora Clusters Considerations

Source

mysql> CALL mysql.rds_show_configuration;
mysql> CALL mysql.rds_set_configuration('binlog retention hours', 144);
mysql> CREATE USER 'repl_user'@'<domain_name>' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'<domain_name>';
mysql> GRANT USAGE ON *.* TO 'repl_user'@'<domain_name>' REQUIRE SSL;

Target

# Get position from snapshot restore
$ aws rds describe-events

mysql> CALL mysql.rds_set_external_master (host_name, host_port, replication_user_name, replication_user_password,
   mysql_binary_log_file_name, mysql_binary_log_file_location,
   ssl_encryption);
mysql> CALL mysql.rds_start_replication;
mysql> SHOW SLAVE STATUS;
# Get position from snapshot restore

```bash
$ aws rds describe-events

{
  "Events": [
    {
      "EventCategories": [],
      "SourceType": "db-instance",
      "Date": "2016-10-28T19:43:46.862Z",
      "Message": "Binlog position from crash recovery is mysql-bin-changelog.000003 4278",
      "SourceIdentifier": "sample-restored-instance"
    }
  ]
}
```
Multiple Aurora Clusters (2)

- Cross-region read replica
  - Support local read latency
- Improved DR
  - Failover not failback
- Region migration path
- Requires binary log replication
- Incurs cross-region transfer costs $$$

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Replication.CrossRegion.html
Aurora Global Cluster

- One primary region
  - Up to 5 read-only secondary regions
- Uses Aurora storage for replication
  - Lag < 1 second
- RPO = 0
- Blocks writes before failover
- Read-only cluster supports write-forwarding capabilities
Maintenance Situations
Aurora Upgrades

- In-place upgrades (e.g. 2.09.1 to 2.09.2)
  - Whole process 5-10 minutes
  - DNS loss 10-20 seconds
  - ZDP (yet to see this work)
- Minor version (e.g. 2.07.3 to 2.09.2)
  - Very similar to in-place
- Major version (e.g. 2.09.2 to ?.?)
  - Yet to attempt

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Updates.MajorVersionUpgrade.html
Aurora Upsizing / Failover

- Instances can be different instance types
  - Read Endpoint moves to Writer during upsize
- Controlled failover
  - Writer endpoint moves to new promoted instance
  - What was writer becomes a reader
- DNS connectivity loss 10-20 seconds
Aurora Upsizing / Failover Commands

```
CLUSTER_ID="demo"
INSTANCE_ID="${CLUSTER_ID}-0"
aws rds describe-db-instances --db-instance-identifier ${INSTANCE_ID} | jq -r '.DBInstances[] | [.DBInstanceIdentifier, .DBInstanceClass, .DBInstanceStatus]'
[ "demo-0", "db.r5.large", "available" ]

aws rds modify-db-instance --db-instance-identifier ${INSTANCE_ID} --db-instance-class db.r5.4xlarge --apply-immediately

aws rds describe-db-instances --db-instance-identifier ${INSTANCE_ID} | jq -r '.DBInstances[] | [.DBInstanceIdentifier, .DBInstanceClass, .DBInstanceStatus]'
[ "demo-0", "db.r5.large", "modifying" ]

aws rds wait db-instance-available --db-instance-identifier ${INSTANCE_ID}
aws rds describe-db-instances --db-instance-identifier ${INSTANCE_ID} | jq -r '.DBInstances[] | [.DBInstanceIdentifier, .DBInstanceClass, .DBInstanceStatus]'
[ "demo-0", "db.r5.4xlarge", "available" ]

# Failover
aws rds describe-db-clusters --db-cluster-identifier ${CLUSTER_ID} | jq ".DBClusters[].DBClusterMembers"

aws rds failover-db-cluster --db-cluster-identifier ${CLUSTER_ID}

aws rds describe-db-clusters --db-cluster-identifier ${CLUSTER_ID} | jq ".DBClusters[].DBClusterMembers"
```
Aurora Upsizing / Failover Monitoring

```
# Endpoints
CLUSTER_ID="demo"
INSTANCE_ID="${CLUSTER_ID}-0"
aws rds describe-db-clusters --db-cluster-identifier ${CLUSTER_ID} | jq '.DBClusters[].DBClusterMembers'

# Cluster Status
while : ; do date ; aws rds describe-db-instances --db-instance-identifier ${INSTANCE_ID} | jq -r '.DBInstances[] | [.DBInstanceIdentifier, .DBInstanceClass, .DBInstanceStatus]'; sleep 5; done

# Instance endpoint availability (goes down during upsize)
MYSQL_HOST=$(aws rds describe-db-instances --db-instance-identifier ${INSTANCE_ID} | jq -r '.DBInstances[0].Endpoint.Address');
echo $MYSQL_HOST
while : ; do [ -n "${MYSQL_PASSWD}" ] && date; time mysql -h ${MYSQL_HOST} -u${MYSQL_USER} -p${MYSQL_PASSWD} -An --connect-timeout=1 -e "SELECT NOW(),@@aurora_server_id, variable_value from information_schema.global_status where variable_name='uptime'"; sleep 1; done

# Cluster reader endpoint (fails over for new connections)
MYSQL_HOST=$(aws rds describe-db-clusters --db-cluster-identifier ${CLUSTER_ID} | jq -r '.DBClusters[0].ReaderEndpoint'); echo $MYSQL_HOST
while : ; do [ -n "${MYSQL_PASSWD}" ] && date; time mysql -h ${MYSQL_HOST} -u${MYSQL_USER} -p${MYSQL_PASSWD} -An --connect-timeout=1 -e "SELECT NOW(),@@aurora_server_id, variable_value from information_schema.global_status where variable_name='uptime'"; sleep 1; done
```
## Aurora Upsizing / Failover Timing Example

<table>
<thead>
<tr>
<th>Status</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>status=available</td>
<td>17:30:01 EDT 2021</td>
<td>18:05:12 EDT 2021</td>
</tr>
<tr>
<td>status=modifying</td>
<td>17:30:02 EDT 2021</td>
<td>18:05:19 EDT 2021</td>
</tr>
<tr>
<td>Reads flip to writer endpoint</td>
<td>17:32:48 UTC 2021</td>
<td>18:07:10 EDT 2021</td>
</tr>
<tr>
<td>Lose reader access</td>
<td>17:33:13 EDT 2021</td>
<td>18:07:42 EDT 2021</td>
</tr>
<tr>
<td>Accessible reader instance</td>
<td>17:37:33 EDT 2021</td>
<td>18:12:42 EDT 2021</td>
</tr>
<tr>
<td>status=modifying</td>
<td>17:40:35 EDT 2021</td>
<td>18:14:46 EDT 2021</td>
</tr>
<tr>
<td>status=storage-optimization</td>
<td>17:41:40 EDT 2021</td>
<td>N/A</td>
</tr>
<tr>
<td>status=available</td>
<td>17:53:53 EDT 2021</td>
<td>18:16:15 EDT 2021</td>
</tr>
</tbody>
</table>
Aurora Upsizing / Failover Graphs (CPU example)
Other Topics (for another time)
Additional RDS/Aurora Capabilities

- IAM Authentication for users
- Aurora Query Cache
- Aurora Parallel Query
- Aurora Monitoring
- DMS source & target
  - Replicate to/from RDS to RDS/Redshift/Kinesis etc
- Database Activity Streams
  - CDC to Kinesis
- Aurora specific tuning (binlog)
- RDS Proxy
- Autoscaling (ASG) read replicas
- ...

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/MonitoringAurora.html
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/DBActivityStreams.html
https://aws.amazon.com/rds/proxy/
Aurora Serverless

- For development & integration non 24x7 environments
- Cost versus performance benefits
- V1
- V2 (preview)

https://aws.amazon.com/rds/aurora/serverless/
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless-2.how-it-works.html
SHOW VOLUME STATUS;

ALTER SYSTEM CRASH [ INSTANCE | DISPATCHER | NODE ];

ALTER SYSTEM SIMULATE percentage_of_failure PERCENT READ REPLICA FAILURE

[ TO ALL | TO "replica name" ]

FOR INTERVAL quantity { YEAR | QUARTER | MONTH | WEEK | DAY | HOUR | MINUTE | SECOND };

ALTER SYSTEM SIMULATE percentage_of_failure PERCENT DISK FAILURE

[ IN DISK index | NODE index ]

FOR INTERVAL quantity { YEAR | QUARTER | MONTH | WEEK | DAY | HOUR | MINUTE | SECOND };

ALTER SYSTEM SIMULATE percentage_of_failure PERCENT DISK CONGESTION

BETWEEN minimum AND maximum MILLISECONDS

[ IN DISK index | NODE index ]

FOR INTERVAL quantity { YEAR | QUARTER | MONTH | WEEK | DAY | HOUR | MINUTE | SECOND };
Aurora under the hood

Quorums


Conclusion
Conclusion

- Managed services helps less resourced teams
- Monitoring cost is important
- Review performance between native/ec2/rds/aurora MySQL installations
- With managed services, some existing actions are limited/restricted
- HA infrastructure/ failover / upgrades are built-in capabilities

Slides:
http://ronaldbradford.com/blog/understanding-aws-rds-aurora-capabilities-2021-05-13/
Thank You

https://perconalive.com
Aurora Metadata (5.7)
RDS/Aurora Metadata

- Variables (35)
- Status (101)
- INFORMATION_SCHEMA tables/views (18)
- mysql tables (19)
- MySQL Procedures (31)
- MySQL Privileges (5)
- Other
<table>
<thead>
<tr>
<th>Variable_name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>aurora_backtrace_compare_metrics_file</td>
<td>backtrace_compare_metrics.txt</td>
</tr>
<tr>
<td>aurora_backtrace_dedupe_string_filename</td>
<td>backtrace_dedupe_strings.txt</td>
</tr>
<tr>
<td>aurora_backtrace_file_name</td>
<td>mysqld-backtrace.txt</td>
</tr>
<tr>
<td>aurora_binlog_reserved_event_bytes</td>
<td>1024</td>
</tr>
<tr>
<td>aurora_binlog_use_large_read_buffer</td>
<td>ON</td>
</tr>
<tr>
<td>aurora_das_persistence_threads</td>
<td>4</td>
</tr>
<tr>
<td>aurora_disable_hash_join</td>
<td>ON</td>
</tr>
<tr>
<td>aurora_disable_sync_tmz</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_enable_repl_bin_log_filtering</td>
<td>ON</td>
</tr>
<tr>
<td>aurora_enable_replica_log_compression</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_enable_staggered_replica_restart</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_enable_zdr</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_fwd_master_idle_timeout</td>
<td>60</td>
</tr>
<tr>
<td>aurora_fwd_master_max_connections_pct</td>
<td>10</td>
</tr>
<tr>
<td>aurora_globaldb_rpo_in_seconds</td>
<td>4294967295</td>
</tr>
<tr>
<td>aurora_globaldb_rpo_wait_timeout</td>
<td>60</td>
</tr>
<tr>
<td>aurora_is_57_supported</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_is_mm_supported</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_lab_mode</td>
<td>demo-0</td>
</tr>
<tr>
<td>aurora_load_from_s3_role</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_max_alter_table_log_entries</td>
<td>1000</td>
</tr>
<tr>
<td>aurora_max_connections_limit</td>
<td>16000</td>
</tr>
<tr>
<td>aurora_parallel_query</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_performance_schema_sql_info_max_size</td>
<td>4096</td>
</tr>
<tr>
<td>aurora_pq_force</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_read_replica_read_committed</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_replica_read_consistency</td>
<td></td>
</tr>
<tr>
<td>aurora_select_into_s3_encryption_default</td>
<td>OFF</td>
</tr>
<tr>
<td>aurora_select_into_s3_role</td>
<td></td>
</tr>
<tr>
<td>aurora_server_id</td>
<td>demo-0</td>
</tr>
<tr>
<td>aurora_table_open_cache</td>
<td>0</td>
</tr>
<tr>
<td>aurora_use_key_prefetch</td>
<td>ON</td>
</tr>
<tr>
<td>aurora_version</td>
<td>2.09.2</td>
</tr>
<tr>
<td>aurora_zdr_timeout_on_replica_fall_behind</td>
<td>60</td>
</tr>
</tbody>
</table>

35 rows in set (0.00 sec)
### RDS/Aurora Metadata Status

```sql
MySQL [(none)]> SHOW GLOBAL STATUS LIKE 'aurora%';
+--------------------------------------------------------------+---------------------+
| Variable_name                                              | Value               |
+--------------------------------------------------------------+---------------------+
| AuroraDb_commits                                          | 2713838             |
| AuroraDb_commit_latency                                  | 14240360666         |
| AuroraDb_ddl_stmt_duration                                 | 6813933496          |
| AuroraDb_select_stmt_duration                              | 447117597784         |
| AuroraDb_insert_stmt_duration                              | 149184545636         |
| AuroraDb_update_stmt_duration                              | 1160788797           |
| AuroraDb_delete_stmt_duration                              | 2157752888           |
| AuroraDb_lockmgr_bitmaps0_in_use                           | 0                   |
| AuroraDb_lockmgr_bitmaps1_in_use                           | 0                   |
| AuroraDb_lockmgr_bitmaps_mem_used                          | 6506512              |
| AuroraDb_thread_deadlocks                                  | 0                   |
| Aurora_available_alter_table_log_entries                   | 0                   |
| Aurora_dict_sys_mem_size                                   | 132210210            |
| Aurora_fast_insert_cache_hits                              | 9                   |
| Aurora_fast_insert_cache_misses                            | 9                   |
| Aurora_fwd_master_ddl_stmt_count                           | 0                   |
| Aurora_fwd_master_ddl_stmt_duration                        | 0                   |
| Aurora_fwd_master_ddl_stmt_duration_timeout                 | 0                   |
| Aurora_fwd_master_ddl_stmt_session_limit                   | 0                   |
| Aurora_fwd_master_ddl_stmt_session_timeout                 | 0                   |
| Aurora_fwd_master_open_sessions                            | 0                   |
| Aurora_fwd_master_select_stmt_count                        | 0                   |
| Aurora_fwd_master_select_stmt_duration                     | 0                   |
+--------------------------------------------------------------+---------------------+
101 rows in set (0.00 sec)
```
### RDS/Aurora I_S tables

```sql
MySQL [information_schema]> SELECT table_name, table_type FROM information_schema.tables WHERE table_name LIKE 'RDS%' OR table_name LIKE 'aurora%';

+-----------------------------------------+-------------+
<table>
<thead>
<tr>
<th>table_name</th>
<th>table_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDS_METRICS_META</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>RDS_METRICS_GAUGE</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>RDS_METRICS_COUNTER</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>RDS_BUFFER_STAT_AT_STARTUP</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>RDS_PROCESSLIST</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>RDS_CONTROL_PERFORMANCE_INSIGHTS_STATUS</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>AURORA_FORWARDING_PROCESSLIST</td>
<td>SYSTEM VIEW</td>
</tr>
<tr>
<td>aurora_s3_load_history</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_configuration</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_global_status_history</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_global_status_history_old</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_history</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_replication_status</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_sysinfo</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_events_threads_waits_current</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_events_threads_waits_lock_current</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>rds_processlist</td>
<td>BASE TABLE</td>
</tr>
<tr>
<td>Rds_Cluster_Info</td>
<td>BASE TABLE</td>
</tr>
</tbody>
</table>
+-----------------------------------------+-------------+
```

18 rows in set (0.11 sec)
### RDS/Aurora mysql tables

<table>
<thead>
<tr>
<th>Tables in mysql</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aurora_s3_load_history</td>
<td>rds_global_status_history</td>
</tr>
<tr>
<td>bin_log_md_table</td>
<td>rds_global_status_history_old</td>
</tr>
<tr>
<td>bin_log_table</td>
<td>rds_history</td>
</tr>
<tr>
<td>ddi_log_md_table</td>
<td>rds_replication_status</td>
</tr>
<tr>
<td>ddi_log_table</td>
<td>rds_sysinfo</td>
</tr>
<tr>
<td>general_log_backup</td>
<td>ro_replica_status</td>
</tr>
<tr>
<td>metadata_md_table</td>
<td>slow_log_backup</td>
</tr>
<tr>
<td>metadata_table</td>
<td></td>
</tr>
<tr>
<td>rds_configuration</td>
<td></td>
</tr>
</tbody>
</table>

50 rows in set (0.00 sec)
# RDS/Aurora Procedures

MySQL [information_schema]>

```sql
SELECT name FROM mysql.proc WHERE name like 'rds%';
```

<table>
<thead>
<tr>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>rds_collect_global_status_history</td>
</tr>
<tr>
<td>rds_disable_gsh_collector</td>
</tr>
<tr>
<td>rds_disable_gsh_rotation</td>
</tr>
<tr>
<td>rds_enable_gsh_collector</td>
</tr>
<tr>
<td>rds_enable_gsh_rotation</td>
</tr>
<tr>
<td>rds_external_master</td>
</tr>
<tr>
<td>rds_import_binlog_ssl_material</td>
</tr>
<tr>
<td>rds_innodb_buffer_pool_dump_now</td>
</tr>
<tr>
<td>rds_innodb_buffer_pool_load_abort</td>
</tr>
<tr>
<td>rds_innodb_buffer_pool_load_now</td>
</tr>
<tr>
<td>rds_kill</td>
</tr>
<tr>
<td>rds_kill_query</td>
</tr>
<tr>
<td>rds_next_master_log</td>
</tr>
<tr>
<td>rds_remove_binlog_ssl_material</td>
</tr>
<tr>
<td>rds_reset_external_master</td>
</tr>
<tr>
<td>rds_rotate_general_log</td>
</tr>
</tbody>
</table>

RDS/Aurora Privileges

MySQL [information_schema]> SHOW PRIVILEGES;
+----------------------------------------+----------------------------------------+----------------------------------------------------------+
<table>
<thead>
<tr>
<th>Privilege</th>
<th>Context</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load from S3</td>
<td>AWS S3 Access</td>
<td>To load files from AWS S3</td>
</tr>
<tr>
<td>Select into S3</td>
<td>AWS S3 Access</td>
<td>To query data from table(s) and store the result into file(s) on AWS S3</td>
</tr>
<tr>
<td>Invoke Lambda</td>
<td>AWS Lambda Access</td>
<td>To invoke Lambda from within the database</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INVOKE SAGEMAKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INVOKE COMPREHEND</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RDS/Aurora Other
AWS Region

Availability Zone 1

Availability Zone 2

Availability Zone 3

VPC

Primary

Reads

Writes

Replica Tier 0

Reads

Replica Tier 1

Reads

Cluster Volume

Amazon Aurora

Cluster

Amazon Aurora

Cluster

Amazon Aurora

Cluster

Amazon Aurora

Cluster

Amazon Aurora

Cluster