



The logo for AWS re:Invent features the words "AWS" and "re:Invent" stacked vertically. "AWS" is in a smaller, sans-serif font above "re:Invent", which is in a larger, bold, sans-serif font. The entire logo is white against a red-to-purple gradient background.

DAT 435

# Oracle to Amazon Aurora Migration, Step by Step

David Yahalom  
CTO & Founder  
naya tech  
[www.naya-tech.com](http://www.naya-tech.com)

re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



# nayatech

## BigData & NoSQL



## Relational Databases



## Analytics & Data Science



## Cloud Database Migrations

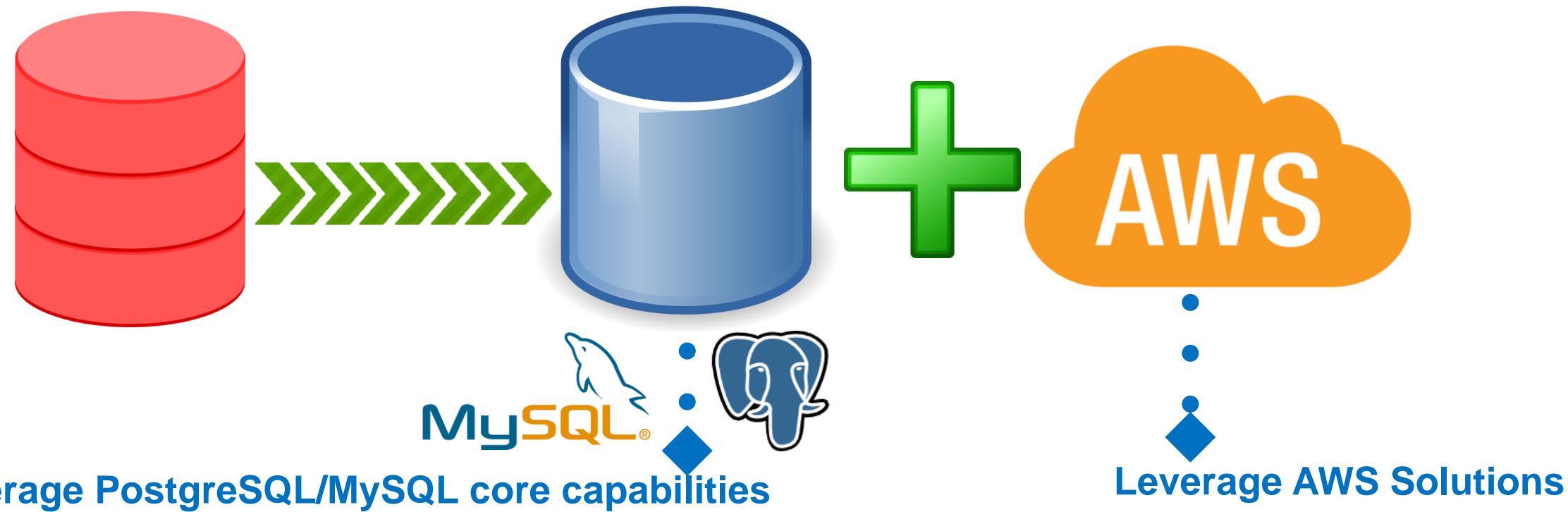
# How-To: Execute Database Migrations

# Traditional Databases



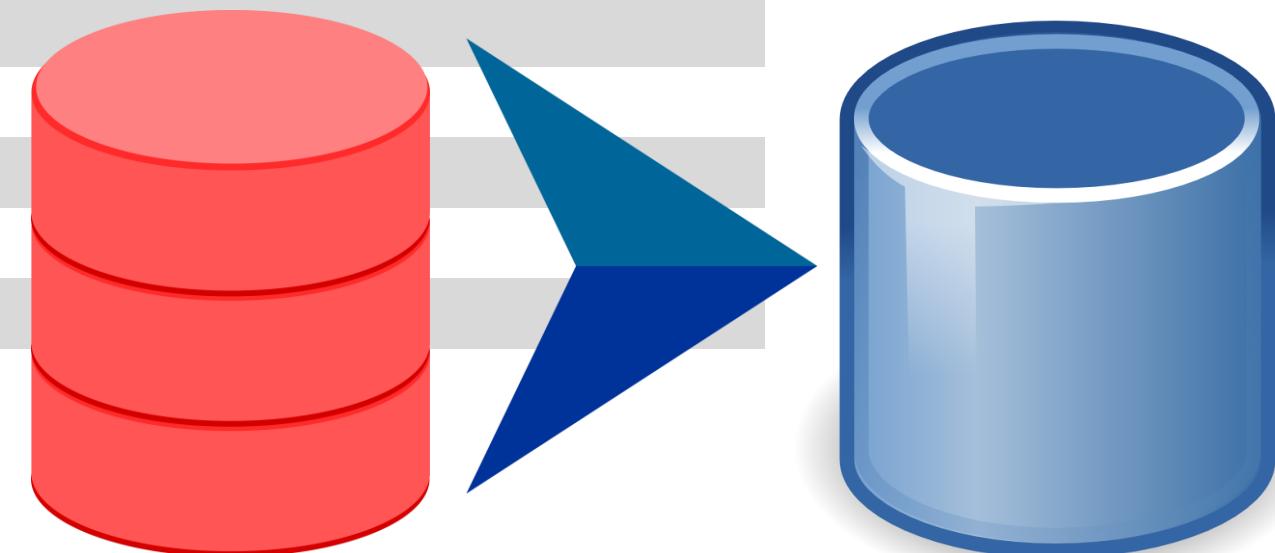
***(this is **not** a tombstone, it's a monolith!)***

# Leverage AWS Solutions where Possible

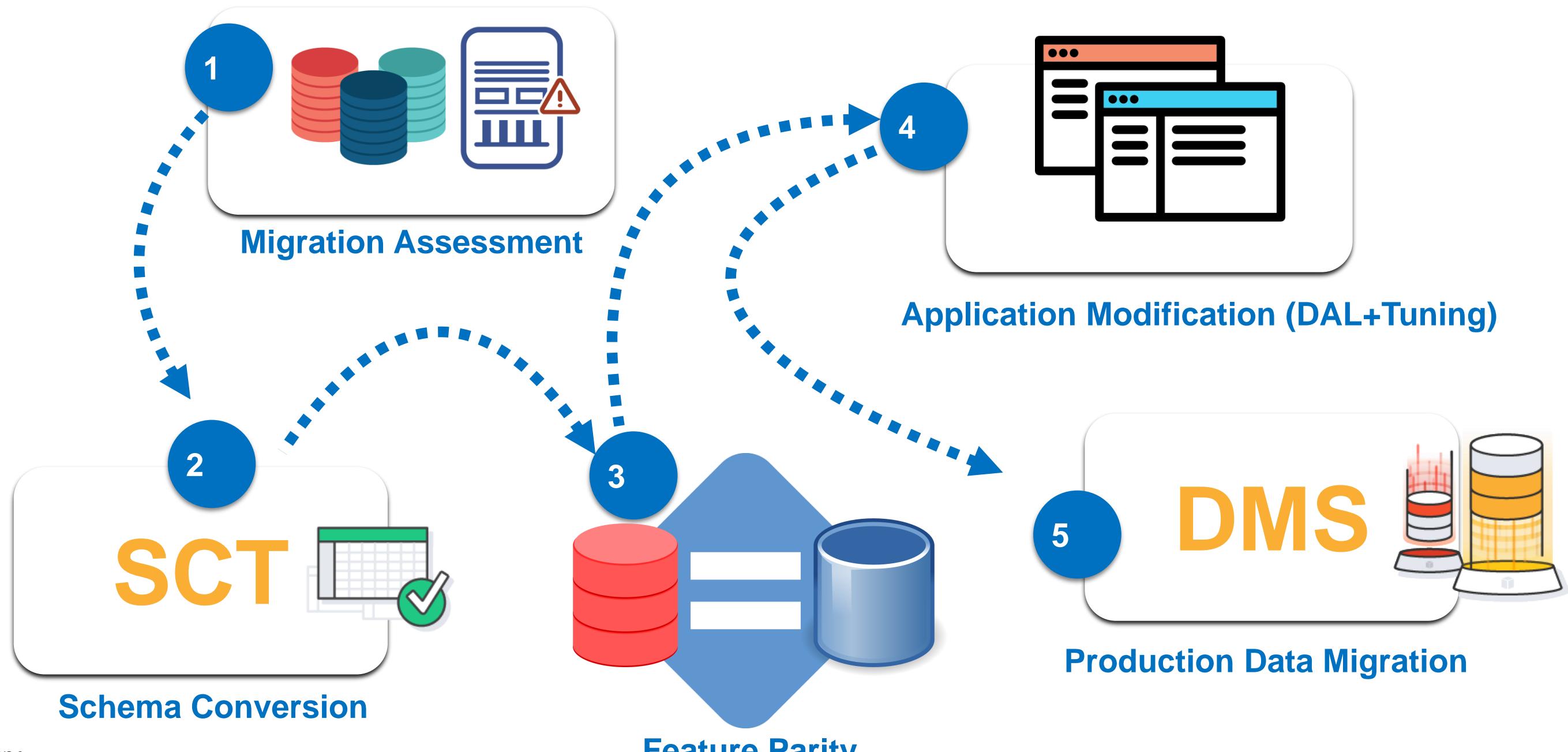


# The Steps for Database Migrations

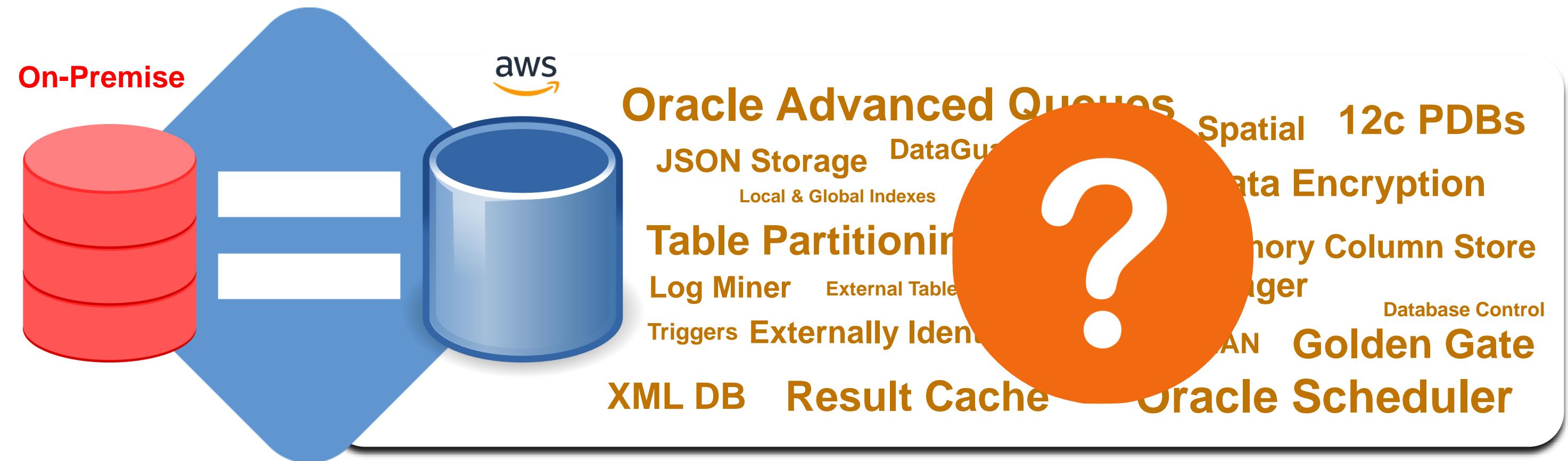
| Phase | Description   |
|-------|---|
| 1     | Choose databases for migration, based on licensing or functional requirements |
| 2     | Identify proprietary database features and establish target equivalents       |
| 3     | Schema assessment & automated conversion                                      |
| 4     | Manual fixes for problematic schema objects                                   |
| 5     | Modification of applications and data access layer                            |
| 6     | Data migration for QA   |
| 7     | Functional testing  |
| 8     | Performance tuning  |
| 9     | Integration & deployment  |
| 10    | Data migration for production   |
| 11    | Training on new technology stack  |
| 12    | Documentation   |
| 13    | Post production ongoing support   |



# The Steps for Database Migrations (Simplified)



# Un-Sticking Sticky Features



# Migration Playbooks



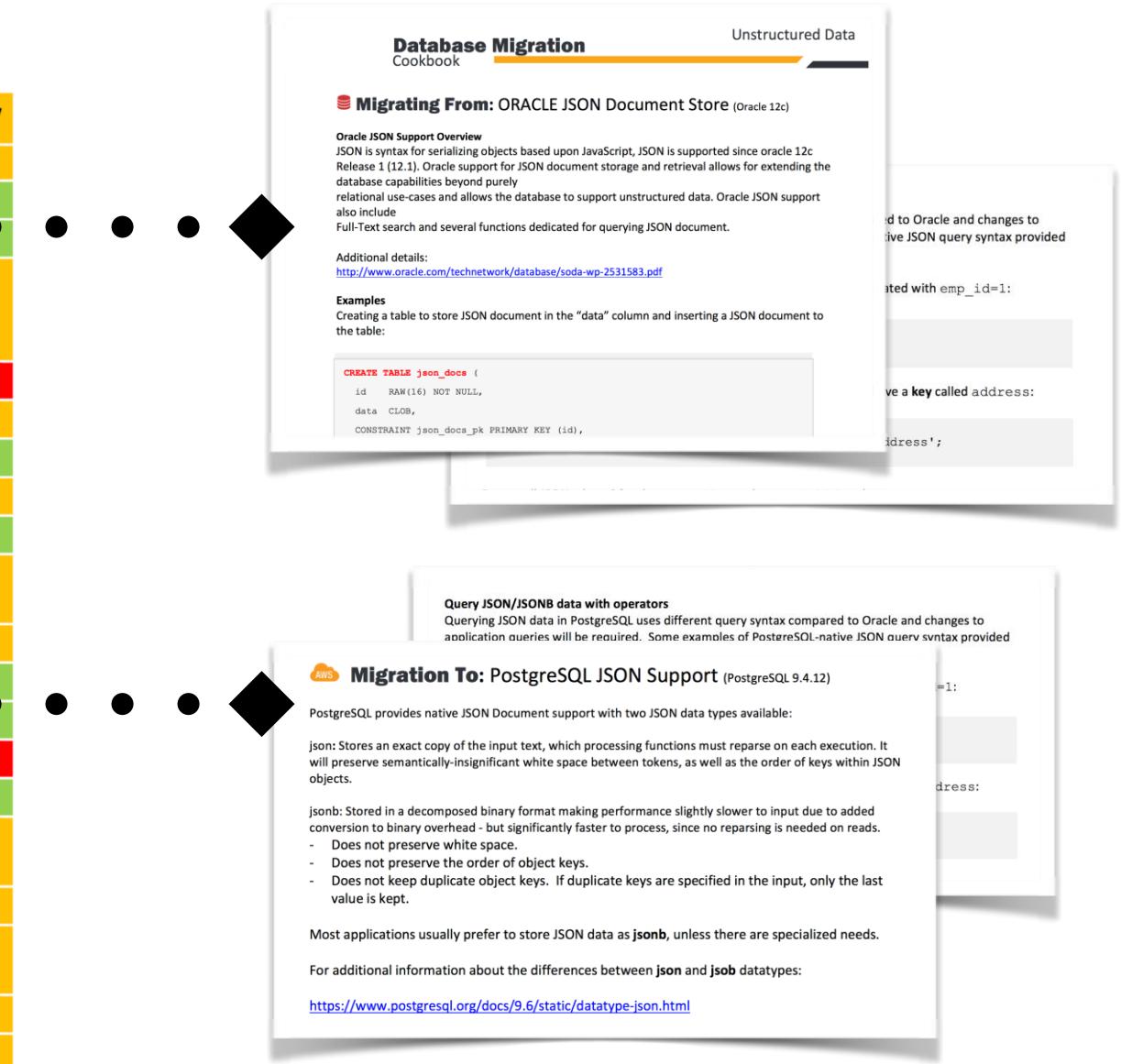
AWS  
re:Invent

<https://aws.amazon.com/dms/resources/>

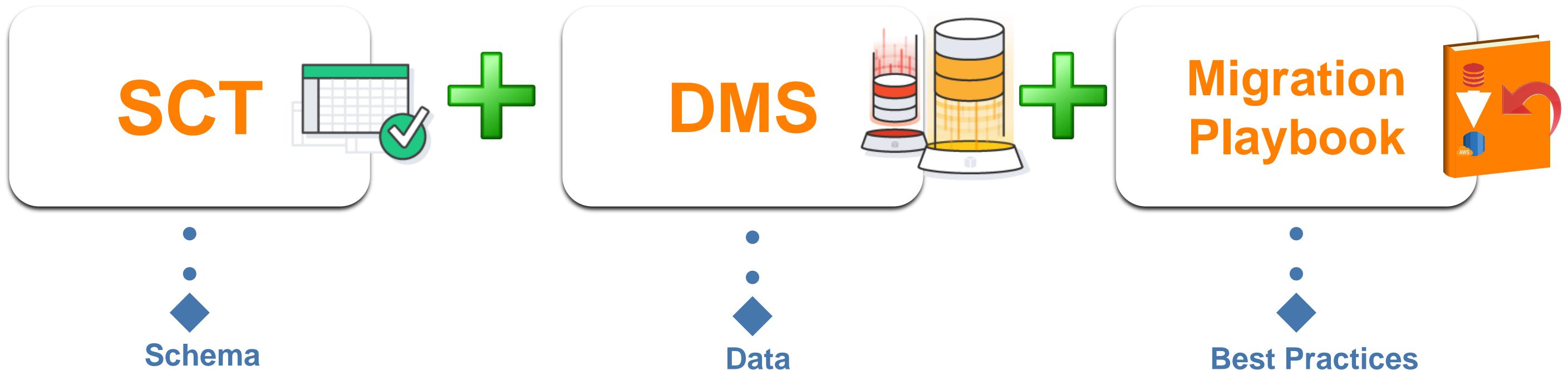
- Topic-by-topic overview of migrations best practices, blueprints and examples.
- How to migrate from proprietary feature and schema objects.
- Targets: Aurora MySQL and Aurora PostgreSQL with more coming soon!

# Migration Playbooks

|                      | Oracle Feature   | PostgreSQL Feature                           | Compatibility |
|----------------------|--|--|---------------|
| <a href="#">Link</a> | Index Organized Tables (IOTs)  | PostgreSQL “Cluster” Tables                  | Yes*          |
| <a href="#">Link</a> | Common Data Types  | Common Data Types                            | Yes           |
| <a href="#">Link</a> | Table Constraints  | Table Constraints                            | Yes           |
| <a href="#">Link</a> | Table Partitioning including:<br>RANGE, LIST, HASH, COMPOSITE,<br>Automatic LIST | Table Partitioning including:<br>RANGE, LIST | Yes*          |
| <a href="#">Link</a> | Exchange & Split Partitions  | N/A  | None          |
| <a href="#">Link</a> | Temporary Tables   | Temporary Tables                             | Yes*          |
| <a href="#">Link</a> | Unused Columns   | ALTER TABLE DROP COLUMN                      | Yes           |
| <a href="#">Link</a> | Virtual Columns  | Views and/or Function as a Column            | Yes*          |
| <a href="#">Link</a> | User Defined Types (UDTs)  | User Defined Types (UDTs)                    | Yes           |
| <a href="#">Link</a> | Read Only Tables & Table<br>Partitions   | Read Only Roles and/or Triggers              | Yes*          |
| <a href="#">Link</a> | Index Types  | Index Types                                  | Yes*          |
| <a href="#">Link</a> | B-Tree Indexes   | B-Tree Indexes                               | Yes           |
| <a href="#">Link</a> | Composite Indexes  | Multi-Column Indexes                         | Yes           |
| <a href="#">Link</a> | BITMAP Indexes   | BRIN Indexes                                 | Minimal       |
| <a href="#">Link</a> | Function-Based Indexes   | Expression Indexes                           | Yes           |
| <a href="#">Link</a> | Global and Local Partitioned<br>Indexes  | Partitioned Indexes                          | Yes*          |
| <a href="#">Link</a> | Identity Columns   | Serial Data Type                             | Yes*          |
| <a href="#">Link</a> | MVCC<br>(Table & Row Locks)  | MVCC<br>(Table & Row Locks)                  | Yes*          |
| <a href="#">Link</a> | Character Sets   | Encoding                                     | Yes*          |
| <a href="#">Link</a> | Transactional Model  | Transactional Model                          | Yes*          |



# Database Migrations Made Easy(er)

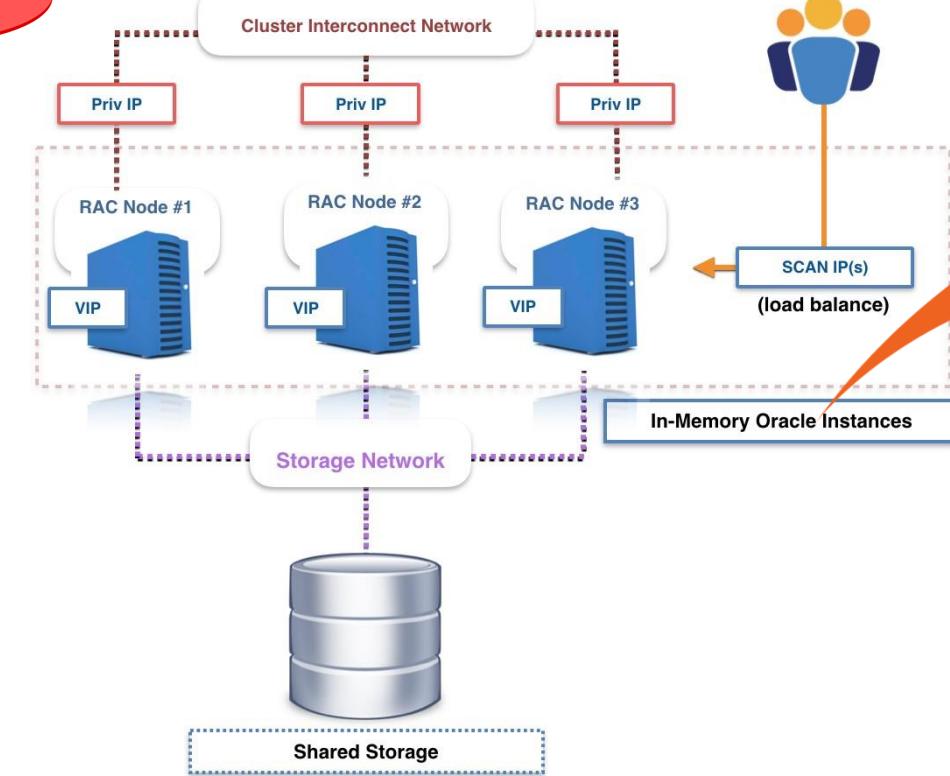


# How-To: Replace Proprietary Database Features with AWS Services

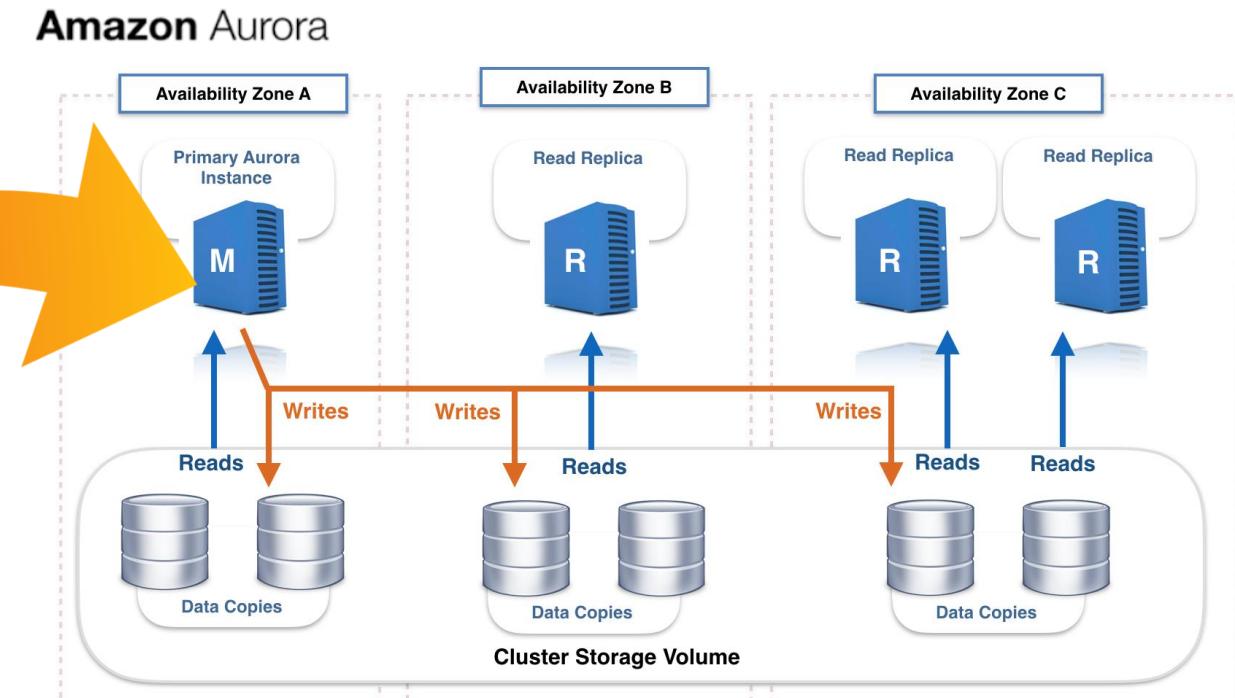
# Replace Proprietary Database Features with AWS Services



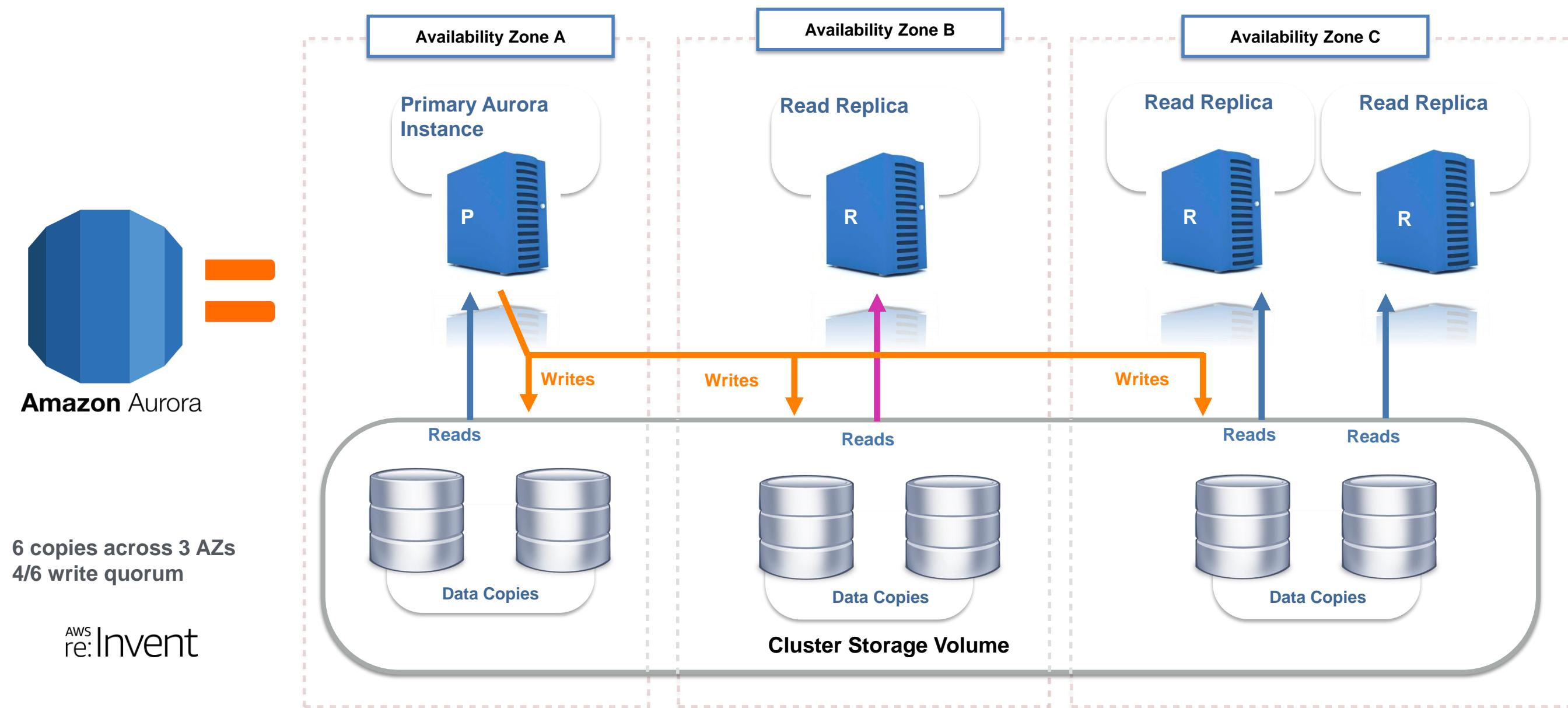
## Oracle RAC and/or ADG

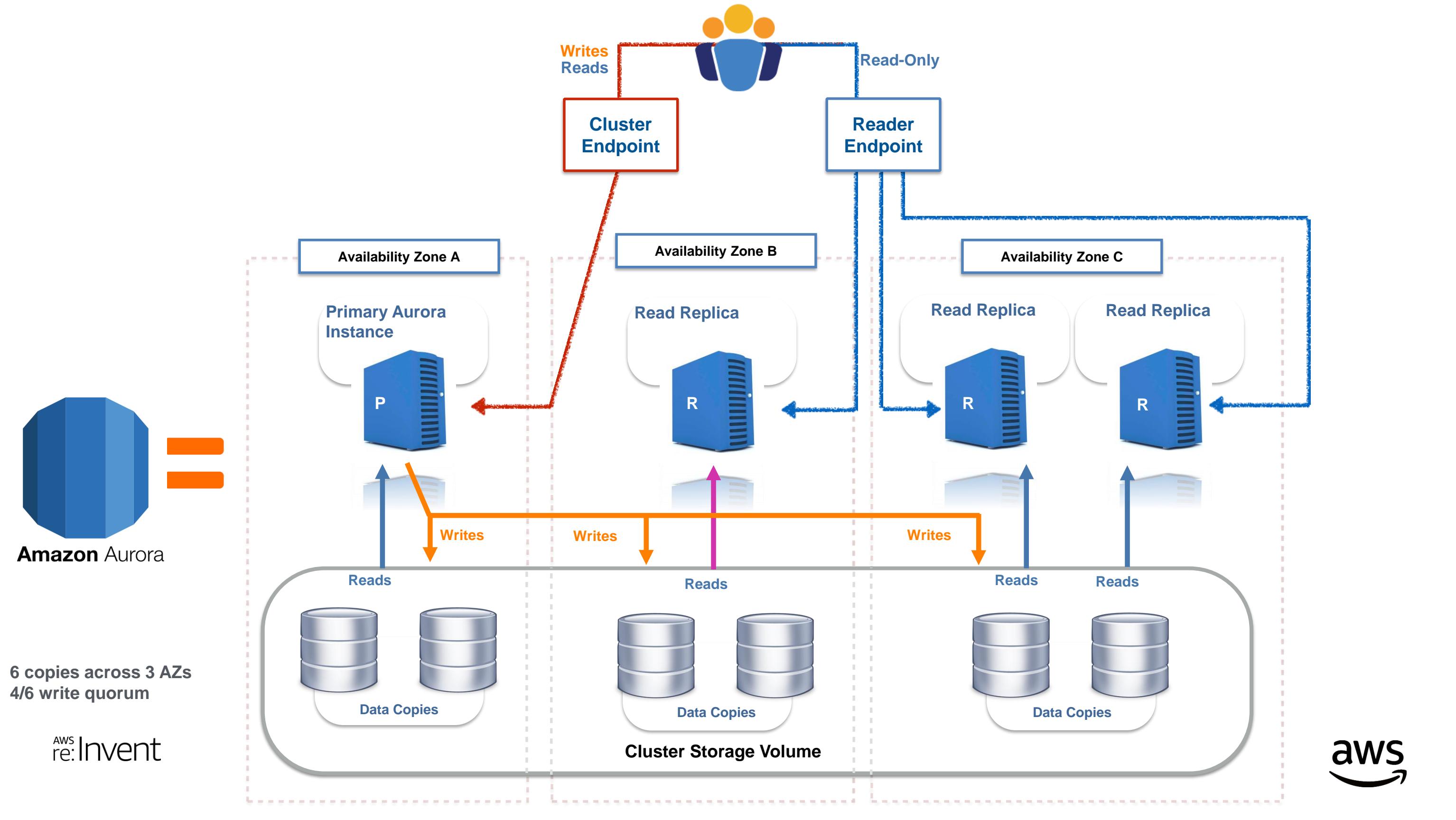


## Amazon Aurora / Aurora MM / GDB



# Replace Proprietary Database Features with AWS Services





# Replace Proprietary Database Features with AWS Services



Amazon Aurora

## Amazon Aurora / Aurora MM / GDB

```
aws rds describe-db-clusters --db-cluster-identifier aurora-mysql |  
find "DBInstanceIdentifier"
```

```
aws rds create-db-instance --db-instance-identifier aurora-mysql-3 -  
--db-cluster-identifier aurora-mysql --db-instance-class db.r4.large  
--engine aurora
```

# Replace Proprietary Database Features with AWS Services



Amazon Aurora

## Amazon Aurora: failover

```
aws rds describe-db-clusters --db-cluster-identifier aurora-mysql  
aws rds failover-db-cluster --db-cluster-identifier aurora-mysql --  
target-db-instance-identifier aurora-mysql-3
```

# Replace Proprietary Database Features with AWS Services



## Oracle Flashback Database

```
STARTUP MOUNT;
```

```
FLASHBACK DATABASE TO TIME"  
TO_DATE('09/20/05','MM/DD/YY'))"  
;
```

```
ALTER DATABASE OPEN READ ONLY;
```

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;  
ALTER DATABASE OPEN RESETLOGS;
```



## Amazon Aurora Backtrack

Amazon Aurora

```
aws rds backtrack-db-cluster --  
db-cluster-identifier aurora-  
mysql --backtrack-to 2018-03-  
19T10:00:00+00:00
```

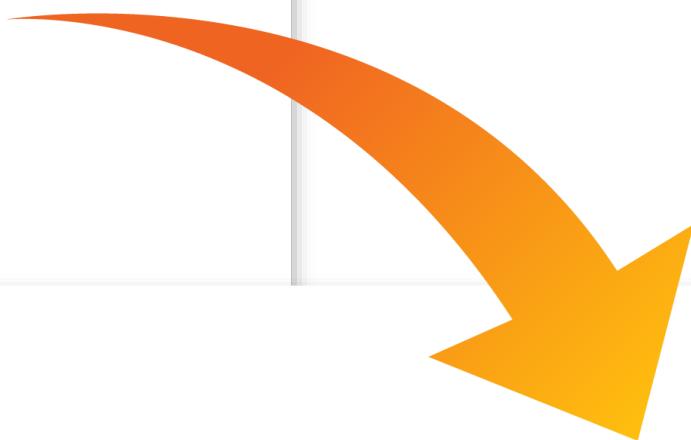
```
aws rds describe-db-cluster-  
backtracks --db-cluster-  
identifier aurora-mysql
```

# Replace Proprietary Database Features with AWS Services



## Oracle Advanced Queuing

```
DBMS_AQ.ENQUEUE ('msgqueue',
enqueue_opts, msgprops, my_msg,
msgid1);
```



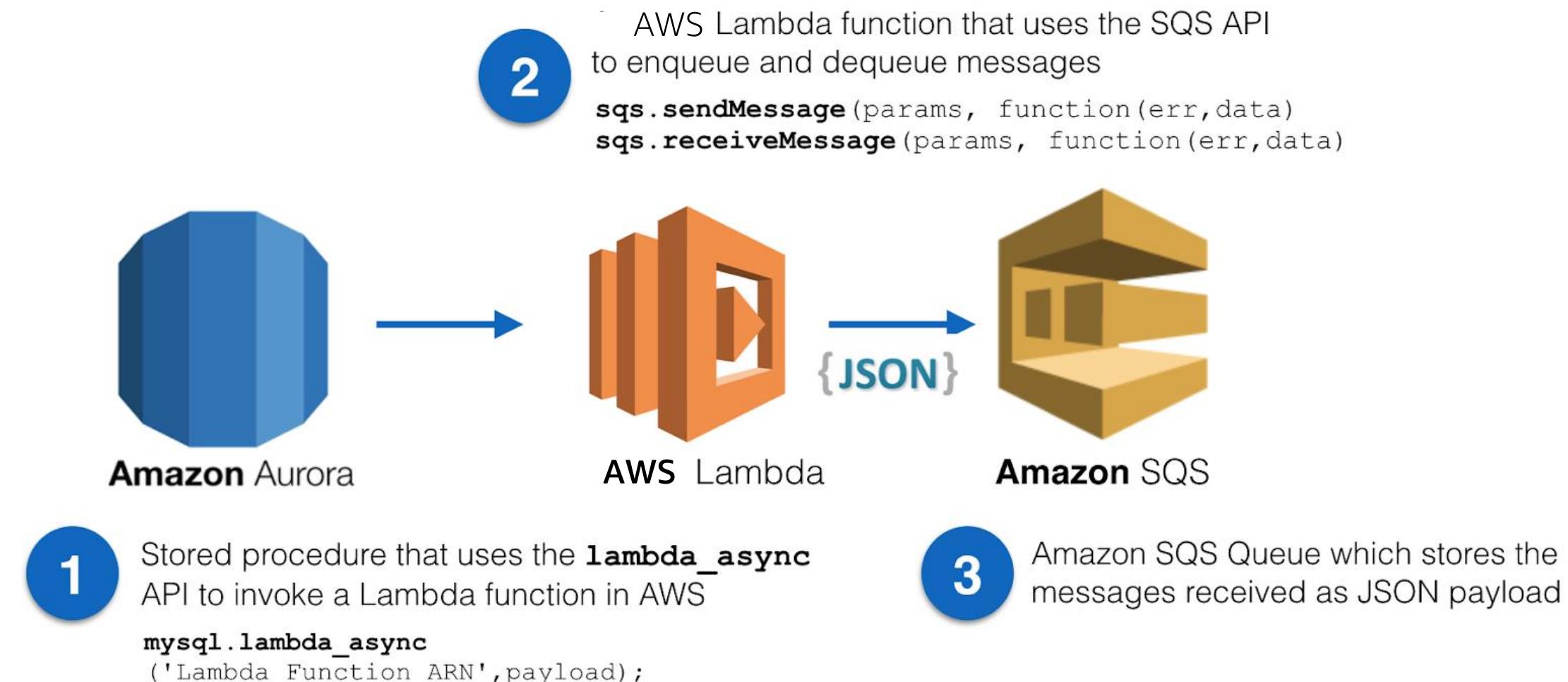
## AWS SQS

```
var sqs = new AWS.SQS...
var lambda = new AWS.Lambda...
```

```
CREATE PROCEDURE SQS_ENQUEUE...
CALL mysql.lambda_async(...)
```



# Replace Advanced Queuing with Amazon Simple Queue Service (Amazon SQS)





# Replace Advanced Queueing with Amazon SQS

## 1) Create an SQS Queue.

A screenshot of the AWS Services menu. The top navigation bar includes the AWS logo, a "Services" dropdown, a "Resource Groups" dropdown, and a search icon. A search bar is open, displaying the text "sq" and a list of results. The results include "Simple Queue Service" and "SQS Managed Message Queues" under the heading "sq", and other services like EC2, EC2 Container Service, Lightsail, Elastic Beanstalk, Lambda, and Batch below it. To the left of the search results, there is a vertical sidebar with links to History, Simple Queue Service, Console Home, Billing, AWS Glue, RDS, and IAM.

- Simple Queue Service
- SQS Managed Message Queues
- EC2
- EC2 Container Service
- Lightsail
- Elastic Beanstalk
- Lambda
- Batch



# Replace Advanced Queuing with Amazon SQS

Note the SQS queue URL:

A screenshot of the AWS SQS console. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and other account information. Below it, a table lists a single queue: 'best\_queue\_ever'. The queue is of type 'Standard', has 'N/A' for message retention, and was created on '2017-10-31 19:14:20 GMT-07:00'. In the main content area, a modal window titled '1 SQS Queue selected' shows the queue's details. The 'Details' tab is active. It displays the queue name 'best\_queue\_ever', its URL 'https://SQS.us-east-1.amazonaws.com/123456789012/best\_queue\_ever', and its ARN. To the right, various configuration parameters are listed: Default Visibility Timeout (30 seconds), Message Retention Period (4 days), Maximum Message Size (256 KB), Receive Message Wait Time (0 seconds), Messages Available (Visible) (0), Messages in Flight (Not Visible) (0), and Messages Delayed (0).

| Name            | Queue Type | Content-Based Deduplication | Messages Available | Messages in Flight | Created                       |
|-----------------|------------|-----------------------------|--------------------|--------------------|-------------------------------|
| best_queue_ever | Standard   | N/A                         | 0                  | 0                  | 2017-10-31 19:14:20 GMT-07:00 |

1 SQS Queue selected

Details Permissions Redrive Policy Monitoring Tags Encryption

Name: best\_queue\_ever  
URL: https://SQS.us-east-1.amazonaws.com/123456789012/best\_queue\_ever  
ARN:  
Created: 2017-10-31 19:14:20 GMT-07:00  
Last Updated: 2017-10-31 19:14:20 GMT-07:00  
Delivery Delay: 0 seconds  
Queue Type: Standard  
Content-Based Deduplication: N/A

Default Visibility Timeout: 30 seconds  
Message Retention Period: 4 days  
Maximum Message Size: 256 KB  
Receive Message Wait Time: 0 seconds  
Messages Available (Visible): 0  
Messages in Flight (Not Visible): 0  
Messages Delayed: 0

# Replace Advanced Queuing with Amazon SQS



Amazon SQS

## Configure the queue permissions:

The screenshot shows the AWS SQS console. In the top navigation bar, 'Services' and 'Resource Groups' are visible. Below the navigation, there are buttons for 'Create New Queue' and 'Queue Actions'. A search bar labeled 'Filter by Prefix:' with placeholder 'Enter Text...' is present. On the left, a table lists one queue: 'best\_queue\_ever' (Standard type). The 'Permissions' tab is selected in the bottom navigation bar. A modal window titled 'Add a Permission to best\_queue\_ever' is open. It contains fields for 'Effect' (set to 'Allow'), 'Principal' (set to 'aws account number(s)'), and 'Actions' (set to 'All SQS Actions (SQS:\*)'). Red arrows point from the text 'Bad permissions. Don't use for production!' to the 'Principal' field and the 'Actions' dropdown. At the bottom of the modal, there are 'Cancel' and 'Add Permission' buttons.

1 SQS Queue selected

Details Permissions Redrive Policy Monitoring Tags Er

Add a Permission Edit Policy Document (Advanced) What's an SQS Queue Access Policy?

© 2018, Amazon '

Cancel Add Permission

*(Bad permissions.  
Don't use for production!)*



AWS Lambda

# Replace Advanced Queuing with Amazon SQS

## 2) Create Lambda function:

The screenshot shows two side-by-side views of the AWS Lambda console. The left view is a search interface with a dropdown menu containing items like 'Lambda', 'CodeBuild', 'Lex', and 'Batch'. The right view is the 'Functions' list page, which displays five existing Lambda functions. A red arrow points to the 'Create function' button at the top right of this list page.

| Function name | Description | Runtime      | Code size | Last Modified |
|---------------|-------------|--------------|-----------|---------------|
| ○             |             | Python 2.7   | 169.8 kB  | last month    |
| ○             |             | Node.js 6.10 | 477 bytes | 19 hours ago  |
| ○             |             | Node.js 6.10 | 442 bytes | last month    |
| ○             |             | Node.js 6.10 | 469 bytes | 2 months ago  |
| ○             |             | Node.js 6.10 | 459 bytes | 2 months ago  |



AWS Lambda

# Replace Advanced Queuing with Amazon SQS

## 2) Create Lambda function which will replace **dbms\_aq.enqueue**

```
index.js

1 var QUEUE_URL = 'https://sns.us-east-1.amazonaws.com/270324613865/best_queue_ever';
2 var AWS = require('aws-sdk');
3 var sqs = new AWS.SQS({region : 'us-east-1'});
4
5 exports.handler = function(event, context) {
6     var params = {
7         MessageBody: JSON.stringify(event),
8         QueueUrl: QUEUE_URL
9     };
10    sqs.sendMessage(params, function(err,data){
11        if(err) {
12            console.log('error:', '** Failed to write to queue **' + err);
13            context.done('error', '** ERROR writing to SQS **');
14        // ERROR when trying to enqueue to SQS
15        }else{
16            console.log('data:',data.MessageId);
17            context.done(null,'');
18        // SUCCESS - message written to SQS
19        }
20    });
21}
22
```

- Configure the SQS queue URL, so that our Lambda Function can interact with the specific queue we created earlier.
- It loads the AWS SDK so that it can use the SQS API.
- Constructs an SQS object in the region where our SQS queue resides (using `new AWS.SQS`).
- Calls the SQS API with the `sqs.sendMessage` call and sends the JSON Payload to the SQS Queue.



# Replace Advanced Queuing with Amazon SQS



AWS Lambda

## Copy the ARN of the Lambda function

A screenshot of the AWS Lambda Functions console. The top navigation bar shows 'Lambda > Functions > sqs\_enqueue1'. Below the navigation is the function name 'sqs\_enqueue1'. To the right of the function name are several buttons: 'Qualifiers ▾' with a red arrow pointing to it from above, 'Actions ▾', 'DummyPayload ▾', and an orange 'Test' button. Further down, a green box contains the message 'Execution result: succeeded (logs)' with a checkmark icon, and a 'Details' link below it. The ARN of the function is visible in the top right corner of the main content area.

Lambda > Functions > sqs\_enqueue1

sqs\_enqueue1

Qualifiers ▾ ARN Actions ▾ DummyPayload ▾ Test

Execution result: succeeded (logs)

Details

ARN



AWS Lambda

# Replace Advanced Queuing with Amazon SQS

Create Lambda function which will replace `dbms_aq.dequeue`



The image shows a screenshot of a code editor with a tab labeled "index.js". The code is written in JavaScript and uses the AWS SDK for SQS. It defines a handler function that attempts to receive a message from a specific queue URL. If successful, it logs the message content and completes the execution. If there is an error, it logs the error message and completes with an error status.

```
1 var QUEUE_URL = 'https://sqs.us-east-1.amazonaws.com/270324613865/best_queue_ever';
2 var AWS = require('aws-sdk');
3 var sqs = new AWS.SQS({region : 'us-east-1'});
4
5 exports.handler = function(event, context) {
6     var params = {
7         QueueUrl: QUEUE_URL,
8         MaxNumberOfMessages: 1
9     };
10    sqs.receiveMessage(params, function(err,data){
11        if(err) {
12            console.log('error:', '** Failed to write to queue **' + err);
13            context.done('error', '** ERROR writing to SQS **');
14        // ERROR when trying to dequeue from SQS
15        }else{
16            console.log('data:',data);
17            context.done(null,'');
18        // SUCCESS - message read from SQS
19        }
20    });
21}
22
```

# Replace Advanced Queuing with Amazon SQS



## 3) Setup your Amazon Aurora MySQL Cluster

- Create a new **IAM Role**.
- Assign the **AWSLambdaFullAccess** policy to the role so that the IAM Role will be able to execute Lambda Function API calls.
- Created a new **custom DB Cluster Parameter Group**.
- Modify the **aws\_default\_lambda\_role** parameter with the value of the IAM Role ARN so that the Aurora MySQL instance will have permissions to invoke AWS Lambda function.
- Create a new Aurora MySQL Cluster, using the **custom Cluster Parameter Group**.
- Create a **stored procedure** invoking the Lambda function using the Lambda function ARN.

# Replace Advanced Queuing with Amazon SQS



Amazon Aurora

## Create IAM role:

A screenshot of the AWS Management Console. The top navigation bar shows 'Services' and 'Resource Groups'. The left sidebar includes links for 'History', 'Console Home', 'Simple Queue Service', 'IAM', and 'Lambda'. The main content area is titled 'iam' and shows the 'IAM' service with the sub-link 'Manage User Access and Encryption'. Other services like 'EC2' and 'EC2 Container Service' are also listed.

A screenshot of the AWS IAM 'Roles' page. The top navigation bar shows 'aws', 'Services', 'Resource Groups', and a search bar. The left sidebar has links for 'Dashboard', 'Groups', 'Users', 'Roles' (which is highlighted with a red arrow), 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. The main content area starts with a section about 'What are IAM roles?' followed by a list of entities that can use them. It then lists 'Additional resources' with links to 'IAM Roles FAQ', 'IAM Roles Documentation', 'Tutorial: Setting Up Cross Account Access', and 'Common Scenarios for Roles'. A red arrow points to the 'Create role' button at the bottom of the page.



# Replace Advanced Queuing with Amazon SQS

Amazon Aurora

AWS Services Resource Groups ▾

Create role

1 Trust 2 Permissions 3 Review

AWS service → Another AWS account Web identity Saml 2.0 federation

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

|                   |                       |                  |                 |
|-------------------|-----------------------|------------------|-----------------|
| API Gateway       | Data Pipeline         | IoT              | SWF             |
| Auto Scaling      | Directory Service     | Lambda           | Service Catalog |
| Batch             | DynamoDB              | Lex              | Storage Gateway |
| CloudFormation    | EC2                   | Machine Learning |                 |
| CloudHSM          | EC2 Container Service | OpsWorks         |                 |
| CloudWatch Events | EMR                   | RDS ←            |                 |
| CodeBuild         | Elastic Beanstalk     | Redshift         |                 |
| CodeDeploy        | Elastic Transcoder    | S3               |                 |
| Config            | Glue                  | SMS              |                 |
| DMS               | Greengrass            | SNS              |                 |

Select your use case

RDS ← Allows RDS to call AWS services on your behalf.

RDS Role for Enhanced Monitoring → Allows RDS to manage CloudWatch Logs resources for Enhanced Monitoring on your behalf.

\* Required Cancel Next: Permissions



Amazon Aurora

# Replace Advanced Queuing with Amazon SQS

## Summary

Delete role

Role ARN arn:aws:iam::270324613865:role/rds\_role

Role description Allows RDS to manage CloudHSM and Directory Service resources on your behalf. | [Edit](#)

Instance Profile ARNs

Path /

Creation time 2018-05-16 19:06 PST

Maximum CLI/API session duration 1 hour [Edit](#)

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Attach policies

Add inline policy

Policy name ▾

Policy type ▾

▶ AWSLambdaFullAccess

AWS managed policy

▶ AmazonRDSDirectoryServiceAccess

AWS managed policy

▶ RDSCloudHsmAuthorizationRole

▶ AWSLambdaExecute

# Replace Advanced Queuing with Amazon SQS



## Create custom Aurora Cluster Parameter Group:

The screenshot shows the AWS RDS Dashboard. In the left sidebar, under the 'Services' section, 'RDS' is selected. A search bar at the top of the main content area contains the text 'rds'. On the right, there is a table of Parameter Groups. At the top of this table, there is a blue button labeled 'Create Parameter Group' with a red arrow pointing to it from the left. Above the table, there is a search bar labeled 'Filter: Search Parameter Groups' with a red arrow pointing to it from the top. The table has columns for Name, Family, and Type. The 'Name' column lists various parameter groups, including 'default.aurora-postgresql1', 'default.aurora-postgresql9.6', 'default.aurora5.6', 'default.mysql5.6', 'default.mysql5.7', 'default.oracle-ee-12.1', and 'default.oracle-se-11.2'. The 'Family' column lists 'aurora-postgresql1', 'aurora-postgresql9.6', 'aurora5.6', 'mysql5.6', 'mysql5.7', 'oracle-ee-12.1', and 'oracle-se-11.2'. The 'Type' column shows 'DE' for all entries.

| Name                         | Family               | Type |
|------------------------------|----------------------|------|
| default.aurora-postgresql1   | aurora-postgresql1   | DE   |
| default.aurora-postgresql1   | aurora-postgresql1   | DE   |
| default.aurora-postgresql9.6 | aurora-postgresql9.6 | DE   |
| default.aurora-postgresql9.6 | aurora-postgresql9.6 | DE   |
| default.aurora5.6            | aurora5.6            | DE   |
| default.aurora5.6            | aurora5.6            | DE   |
| default.mysql5.6             | mysql5.6             | DE   |
| default.mysql5.7             | mysql5.7             | DE   |
| default.oracle-ee-12.1       | oracle-ee-12.1       | DE   |
| default.oracle-se-11.2       | oracle-se-11.2       | DE   |



# Replace Advanced Queuing with Amazon SQS

Amazon Aurora

## Edit the aws\_default\_lambda\_role parameter

The screenshot shows the AWS RDS Parameter Groups page for a group named "mysql-lambda". The left sidebar lists various RDS-related options, with "Parameter Groups" highlighted by a red arrow. The main content area shows a table of parameters. A specific row for "aws\_default\_lambda\_role" is selected, indicated by two red arrows pointing to the "Name" column and the value input field. The table includes columns for Name, Edit Values, Allowed Values, Is Modifiable, Source, Apply Type, Data Type, and Description. The "Data Type" for this parameter is listed as "string". At the top right of the table, there is a "Save Changes" button, which is also highlighted with a red arrow. The top navigation bar shows the AWS logo, Services dropdown, Resource Groups dropdown, and user information "naya tech N.". The top right corner of the page displays the text "Viewing 1 of 1 pa".

| Name                    | Edit Values          | Allowed Values | Is Modifiable | Source | Apply Type | Data Type | Description          |
|-------------------------|----------------------|----------------|---------------|--------|------------|-----------|----------------------|
| aws_default_lambda_role | <input type="text"/> |                | true          | user   | dynamic    | string    | Default IAM role ARN |



# Replace Advanced Queuing with Amazon SQS

Amazon Aurora

## Create the Amazon Aurora MySQL Cluster

The screenshot shows the AWS Management Console interface. On the left, the navigation bar includes links for History, Console Home, RDS, IAM, Simple Queue Service, and Lambda. The main search bar at the top has 'rds' typed into it. A dropdown menu is open, showing options like RDS (Managed Relational Data), Kinesis (Work with Real-Time Stream), Lightsail, and Elastic Beanstalk. To the right, a modal window titled 'Step 1: Select Engine' is displayed. It contains a checkbox for 'Free tier eligible only' and a note: 'To get started, choose a DB Engine below and click Select.' Below this are icons for various database engines: Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. The 'Amazon Aurora' section is highlighted with a blue background. To its right, a description of Amazon Aurora is provided, along with a bulleted list of features. A red arrow points to the 'Select' button next to the MySQL-compatible edition. At the bottom right of the modal is another 'Select' button for PostgreSQL and a 'Cancel' link.

Step 1: Select Engine

To get started, choose a DB Engine below and click Select.

Free tier eligible only

**Amazon Aurora**

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day.

- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL.
- Up to 64TB of auto-scaling SSD storage.
- 6-way replication across three Availability Zones.
- Up to 15 Read Replicas with sub-10ms replica lag.
- Automatic monitoring and failover in less than 30 seconds.

MySQL-compatible edition Select

PostgreSQL-compatible edition Select

Cancel



# Replace Advanced Queuing with Amazon SQS

Amazon Aurora

## Specify the custom Cluster Parameter Group created earlier

The screenshot shows the AWS RDS 'Configure Advanced Settings' page, Step 3: Configure Advanced Settings. The 'DB Cluster Parameter Group' dropdown is highlighted with a red arrow and contains the value 'mysql-lambda'. A tooltip to the right of the dropdown states: 'Parameter group associated with this instance's DB Cluster'.

Step 1: Select Engine  
Step 2: Specify DB Details  
Step 3: Configure Advanced Settings

Configure Advanced Settings

Network & Security

- VPC\*: Default VPC ( )
- Subnet Group: default
- Publicly Accessible: Yes
- Availability Zone: No Preference
- VPC Security Group(s): Create new Security Group

Database Options

- DB Cluster Identifier: [ ]
- Database Name: [ ]
- Database Port: 3306
- DB Parameter Group: default.aurora5.6
- DB Cluster Parameter Group: mysql-lambda ←
- Option Group: default:aurora-5-6
- Enable IAM DB Authentication: No Preference
- Enable Encryption: Yes
- Master Key: (default) aws/rds ←
- Description: Default master key that protects my RDS database volumes when no other key is defined
- Account: [ ]

Parameter group associated with this instance's DB Cluster

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



# Replace Advanced Queuing with Amazon SQS

## Assign IAM role to cluster

The screenshot shows the AWS RDS Dashboard. On the left sidebar, 'Clusters' is selected. In the main content area, there is a table with three rows. The first two rows have empty checkboxes. The third row has a checked checkbox and is highlighted with a red arrow pointing to it. To the right of the table is a 'Manage IAM Roles' button, also highlighted with a red arrow. A modal window titled 'Manage IAM Roles' is open on the right side of the screen, showing a table of current IAM roles for the selected cluster.

| Role      | Status      | Action |
|-----------|-------------|--------|
| rds_role1 | not applied | Delete |



# Replace Advanced Queuing with Amazon SQS

## Invoke the Lambda Function

**CALL**

```
mysql.lambda_async('arn:aws:lambda:us-  
east-1:270324613865:function:sqs_enqueue',  
'{"subject" : "Aurora Lambda test!",  
"message" : "If this works, I can write to  
SQS!"}');
```



# Replace Advanced Queuing with Amazon SQS

Invoke the Lambda Function from a stored procedure

```
CREATE PROCEDURE `dbms_aq.enqueue` (IN payload TEXT)
BEGIN
CALL mysql.lambda_async('arn:aws:lambda:us-east-
1:270324613865:function:sqs_enqueue', payload);
END
```

```
CALL `dbms_aq.enqueue` ('{"subject" : "Aurora Lambda test!", "message" : "If
this works, I can write to SQS!"}')
```



AWS Glue



Amazon Redshift



AWS Encryption



AWS Identity and Access Management (IAM)

A (very) incomplete list...  
• • • • • • • • • • • • • • • • •



And many more!

# Thank you!



Please complete the session  
survey in the mobile app.