

## SUMMARY:

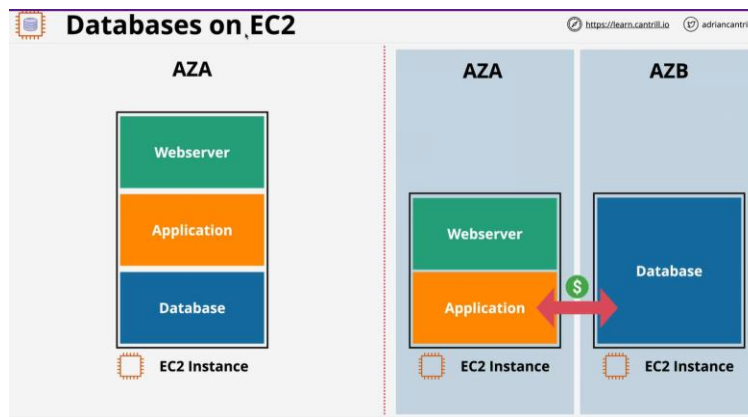
This Lab covers RDS – Migrating from monolithic, MariaDB, Multi-AZ, into Aurora, into Aurora Serverless

### Contents

1. Migrating the Wordpress Monolith to a dedicated EC2 DB ..... 1
2. Migrating MariaDB database into RDS ..... 4
3. MultiAZ & Using a snapshot to recover from data corruption ..... 8
4. Migrating Wordpress onto Aurora..... 10
5. Migrating to Aurora Serverless ..... 12

## 1. Migrating the Wordpress Monolith to a dedicated EC2 DB

The outcome of this lesson is to understand the process for performing a simple database migration manually. Taking a monolith architecture and splitting the DB out to an EC2:



Create wordpress and db stack:

The screenshot shows the AWS Management Console with a table of EC2 instances. The instance 'A4L-Wordpress' is selected, and its details are shown below. A green box highlights the 'Public IPv4 address' field, which contains the IP address '52.87.142.196' and a link to 'open address'. The instance is in a 'Running' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
A4L-Wordpress	i-0475389799d724ee9	Running	t2.micro	2/2 checks ...	No alarms +	us-east-1a
A4L-DB-Wordpress	i-077d821298f6b920e	Running	t2.micro	2/2 checks ...	No alarms +	us-east-1a

Instance: i-0475389799d724ee9 (A4L-Wordpress)

Details | Security | Networking | Storage | Status Checks | Monitoring | Tags

▼ Instance summary Info

Instance ID  
i-0475389799d724ee9 (A4L-Wordpress)

Instance state  
Running

Public IPv4 address  
52.87.142.196 | open address

Private IPv4 addresses  
10.16.51.23

Public IPv4 DNS  
ec2-52-87-142-196.compute-1.amazonaws.com | open address

Private IPv4 DNS  
ip-10-16-51-23.ec2.internal

Use public IP to see the wordpress site, configure and add blogpost with images from demo file:

The screenshot shows the WordPress installation screen. The URL bar at the top shows '52.87.142.196/wp-admin/install.php'. The page has a 'Welcome' message and a section titled 'Information needed'. It asks for the following information:

- Site Title: The Best Cats
- Username: admin
- Password: 4n1m4ls4L1f3 (Medium strength)
- Your Email: test@test.com
- Search engine visibility: Discourage search engines from indexing this site (unchecked)

At the bottom, there is a button labeled 'Install WordPress'.

Login into DB and configure with private IP:

```
GNU nano 2.9.8 wp-config.php
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'a4lwordpress' );

/** MySQL database username */
define( 'DB_USER', 'a4lwordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', '4n1m4ls4L1f3' );

/** MySQL hostname */
define( 'DB_HOST', '10.16.60.102' );

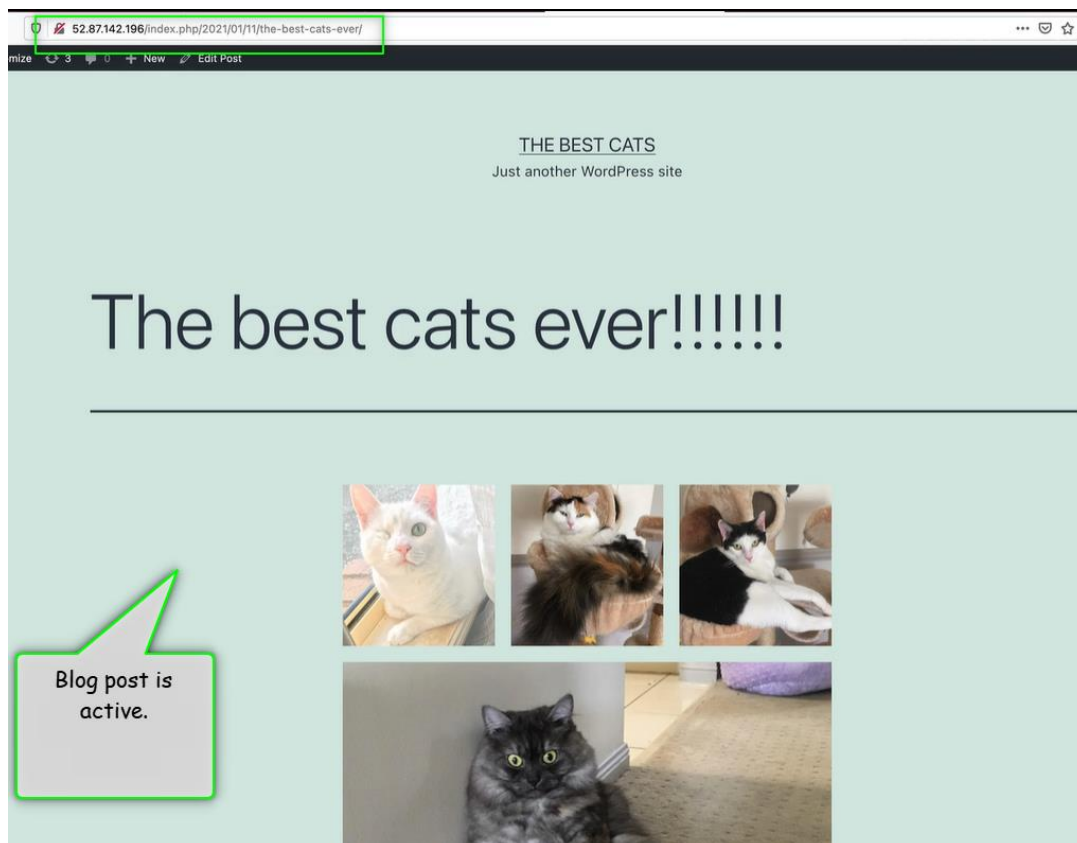
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 */

[ec2-user@ip-10-16-51-23 html]$
```

Check blog post:



## 2. Migrating MariaDB database into RDS

In this [DEMO] Lesson you will create a MySQL RDS instance and migrate the Wordpress Database from the self-managed MariaDB server running on EC2 into this RDS instance.

Due to the time required to work with RDS please ensure you have at least 1 hour to work through this demo.

Commands for demo:

```
# Backup of Source Database
mysqldump -h PRIVATEIPOFMARIADBINSTANCE -u a4lwordpress -p a4lwordpress > a4lwordpress.sql

# Restore to Destination Database
mysql -h CNAMEOFRDSINSTANCE -u a4lwordpress -p a4lwordpress < a4lwordpress.sql

# Change WP Config
cd /var/www/html
sudo nano wp-config.php

replace
/** MySQL hostname */
define('DB_HOST', 'PRIVATEIPOFMARIADBINSTANCE');

with
/** MySQL hostname */
define('DB_HOST', 'REPLACEME_WITH_RDSINSTANCEENDPOINTADDRESS');
```

Do same steps in 1 to get wordpress stack, except create DB subnet group:

**Amazon RDS** X

- Dashboard
- Databases
- Query Editor
- Performance Insights
- Snapshots
- Automated backups
- Reserved instances
- Proxies
- Subnet groups**
- Parameter groups
- Option groups
- Custom Availability Zones
- Events
- Event subscriptions
- Recommendations
- Certificate update

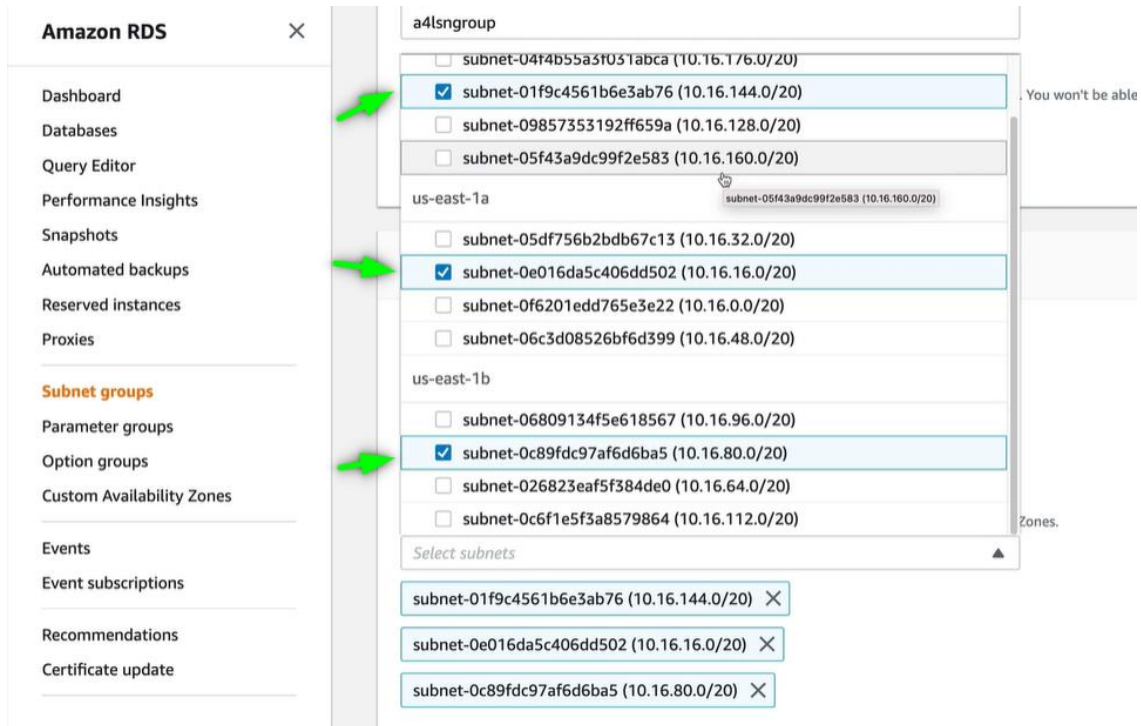
**Name**  
You won't be able to modify the name after your subnet group has been created.  
a4lsngroup  
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

**Description**  
a4lsngroup

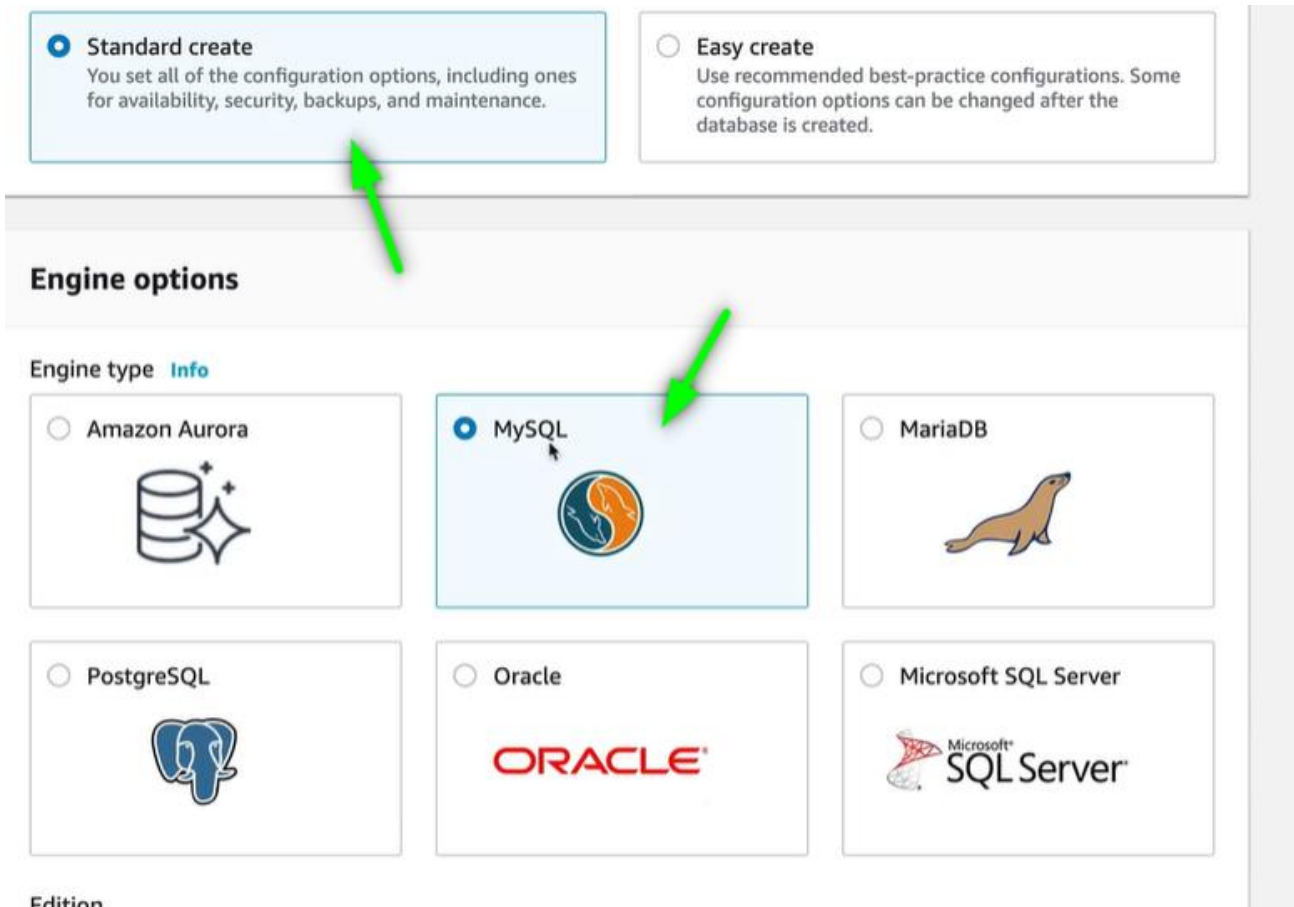
**VPC**  
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.  
a4l-vpc1 (vpc-0ead26fa520f91867)

**Subnets**  
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.  
Select subnets

Selected subnets: us-east-1a, us-east-1b, us-east-1c



Create MySQL dvb and configure:



Public access [Info](#)

☐ Yes  
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☒ No  
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

☐ Choose existing  
Choose existing VPC security groups

☒ Create new  
Create new VPC security group

New VPC security group name

A4LVPC-RDS-SG

Availability Zone [Info](#)

No preference

No preference

us-east-1a

us-east-1b

us-east-1c

### ▼ Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

#### Database options

Initial database name [Info](#)

a4lwordpress

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

default.mysql5.7

Option group [Info](#)

default:mysql-5-7

#### Backup

Creates a point-in-time snapshot of your database

☒ Enable automatic backups

Enabling backups will automatically create backups of your database during a certain time window.



The screenshot shows the AWS RDS console. At the top, the 'Databases' section is active. A table lists databases, with 'a4lwordpress' highlighted by a green arrow. Below this, the 'a4lwordpress' database details are shown. A green callout box points to the 'Summary' section, stating 'DB is up with correct configs'. The 'Summary' section displays: DB identifier 'a4lwordpress', Role 'Instance', CPU usage at 1.80%, Current activity at 0 Connections, Status 'Available' (with a green checkmark), Engine 'MySQL Community', Class 'db.t2.micro', and Region & AZ 'us-east-1c'. Below the summary, the 'Connectivity & security' tab is selected. It shows three columns: 'Endpoint & port' (Endpoint: 'a4lwordpress.c1v6gpj9pezi.us-east-1.rds.amazonaws.com', Port: '3306'), 'Networking' (Availability zone: 'us-east-1c', VPC: 'a4l-vpc1 (vpc-0ead26fa520f91867)', Subnet group: 'a4lsngroup', Subnets: 'a4l-subnet1'), and 'Security' (VPC security groups: 'A4LVPC-RDS-SG (sg-0d86c499de2a064a2) (active)', Public accessibility: 'No', Certificate authority: 'rds-ca-2019').

**Databases** Group resources Refresh Modify Actions Restore from S3 Create database

Filter databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CF
<a href="#">a4lwordpress</a>	Instance	MySQL Community	-	db.t2.micro	Creating	-

### a4lwordpress Modify

#### Summary

DB identifier a4lwordpress	CPU 1.80%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1c

**Connectivity & security** **Monitoring** Logs & events Configuration Maintenance & backups Tags

#### Connectivity & security

<b>Endpoint &amp; port</b> Endpoint a4lwordpress.c1v6gpj9pezi.us-east-1.rds.amazonaws.com Port 3306	<b>Networking</b> Availability zone us-east-1c VPC a4l-vpc1 (vpc-0ead26fa520f91867) Subnet group a4lsngroup Subnets a4l-subnet1	<b>Security</b> VPC security groups A4LVPC-RDS-SG (sg-0d86c499de2a064a2) (active) Public accessibility No Certificate authority rds-ca-2019
-----------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

Connect to wordpress config and point DB connection to the new RDS using the Endpoint address:

```

GNU nano 2.9.8                               wp-config.php
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'a4lwordpress' );

/** MySQL database username */
define( 'DB_USER', 'a4lwordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', '4n1m4ls4L1f3' );

/** MySQL hostname */
define( 'DB_HOST', 'a4lwordpress.c1v6gpj9pezi.us-east-1.rds.amazonaws.com' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication Unique Keys and Salts.
 */
* Change these to different unique phrases!
* You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
* You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
*/

[ec2-user@ip-10-16-60-130 html]$

```

### 3. MultiAZ & Using a snapshot to recover from data corruption

In this [DEMO] lesson you will gain experience how snapshots and restores can be used to recover from data corruption issues.

Use stack from previous, create snapshot:

RDS > Databases > Take snapshot

## Take DB snapshot

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

**Settings**

To take a snapshot of this DB instance you must provide a name for the snapshot.

**DB instance**  
The unique key that identifies a DB instance. This parameter isn't case-sensitive.

rd6vig3mqkejz

**Snapshot name**  
The identifier for the DB snapshot.

a4lwordpress-with-cat-post-mysql-5-7

Cancel Take snapshot



## Snapshots

Manual snapshots (1)									
<div> <div>Manual</div> <div>System</div> <div>Shared with me</div> <div>Public</div> <div>Backup service</div> <div>Exports in Amazon S3</div> </div>									
<div> <div>Filter manual snapshots</div> <div> <div>Actions</div> <div>Take snapshot</div> </div> </div>									
	Status	Progress	Engine	VPC	Snapshot type	Storage	Storage Type	Zone	
3 pm UTC	Available	Completed	MySQL Community	vpc-08dff5af01a2c4c99	manual	20 GiB	General Purpose (SSD)	us-east-1c	

Configure DB with multi-AZ:

## Availability &amp; durability

Multi-AZ deployment [Info](#)

- ☐ Do not create a standby instance
- ☒ Create a standby instance (recommended for production usage)  
 Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

enables multi-AZ

## Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.

Attribute	Current value	New value
Multi-AZ deployment	No	Yes



## Potential performance impact

You may experience a significant performance impact when converting this database instance to Multi-AZ configuration. This impact will be more noticeable on database instances with large amounts of storage and write-intensive workloads.

## Scheduling of modifications

When to apply modifications

- ☐ Apply during the next scheduled maintenance window  
 Current maintenance window: January 15, 2021 17:47 - 18:17 UTC+10
- ☒ Apply immediately  
 The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

Cancel

Back

Modify DB instance

Restoring done here:

RDS > Databases > Restore to point in time

## Launch DB Instance

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB security group and DB parameter groups.

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

#### 4. Migrating Wordpress onto Aurora

In this [DEMO] lesson you will migrate an RDS Snapshot created earlier in the course into an Aurora Cluster and verify access by provisioning the Wordpress EC2 instance.

The lesson also introduces the next architectural problem which the course will resolve ... local media storage.

Use Stack to create Aurora cluster:

CloudFormation > Stacks > Create stack

### Quick create stack

**Template**

Template URL  
[https://learn-cantrill-labs.s3.amazonaws.com/awscoursedemos/0017-aws-associate-rds-migrating-to-aurora/A4L\\_WORDPRESS\\_AND\\_AURORA.yaml](https://learn-cantrill-labs.s3.amazonaws.com/awscoursedemos/0017-aws-associate-rds-migrating-to-aurora/A4L_WORDPRESS_AND_AURORA.yaml)

Stack description  
 Animals4Life base VPC Template + Bootstrapped Wordpress + Aurora DB Cluster (2.07.1) Instance

**Stack name**

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Use ARN from previous:

Search for services, features, marketplace products, and docs [Option+S]

RDS > Snapshots > a4lwordpress-with-cat-post-mysql-5-7

## a4lwordpress-with-cat-post-mysql-5-7

**Details**

ARN  
[arn:aws:rds:us-east-1:945690336440:snapshot:a4lwordpress-with-cat-post-mysql-5-7](#)

**DatabaseRestoreSnapshot**  
The snapshot name to restore from - Leave Blank for a new DB, USE ARN for a MIGRATION of a non aurora snapshot or name for aurora

arn:aws:rds:us-east-1:945690336440:snapshot:a4lwordpress-with-cat-post-mysql-5-7

**LatestAmild**  
AMI for Wordpress Instance (default is latest AmaLinux2)

/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2

**Capabilities**

**i The following resource(s) require capabilities: [AWS::IAM::Role]**

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more](#)

☐ I acknowledge that AWS CloudFormation might create IAM resources.

Cancel Create change set Create stack

Aurora Cluster:

RDS > Databases

**Databases** ☒ Group resources

Filter databases

	DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
<input checked="" type="radio"/>	aurora-database-juq9p5f6835k	Regional	Aurora MySQL	us-east-1	2 instances	Available	-
<input type="radio"/>	aa2pagwi6e0sw4	Writer	Aurora MySQL	us-east-1c	db.t3.small	Available	
<input type="radio"/>	aatkmg3xmg98x7	Reader	Aurora MySQL	us-east-1a	db.t3.small	Available	

Take aurora snapshot:

RDS > Databases > Take snapshot

## Take DB snapshot

**Settings**  
To take a snapshot of this DB cluster you must provide a name for the snapshot.

**Cluster**  
The unique key that identifies a DB cluster. This parameter isn't case-sensitive.

aurora-dbcluster-juq9p5f6835k

**Snapshot name**  
The identifier for the DB snapshot.

a4lwordpress-aurora-with-cat-post

Cancel Take snapshot

Confirm both snapshots are present – MySQL + Aurora:

**Manual snapshots (2)**

Filter manual snapshots

Snapshot name	DB instance or cluster
a4lwordpress-aurora-with-cat-post	aurora-dbcluster-juq9p5f6835k
a4lwordpress-with-cat-post-mysql-5-7	rd6vig3mqkejz

## 5. Migrating to Aurora Serverless

In this [DEMO] lesson you will experience how to migrate an Aurora provisioned snapshot into an Aurora serverless cluster. In addition you will see how an aurora serverless cluster can scale down to 0 ACU and pause - meaning the cluster costs will be for storage only. You will experience how when incoming load reaches the serverless cluster it will unpause and allocate ACU to begin servicing requests.

Use same stack, point to Aurora snapshot – restore to a serverless:

RDS > Snapshots > Restore snapshot

## Restore snapshot

You are creating a new DB Instance from a source DB Instance at a specified time. This new DB Instance will have the default DB Security group and DB Parameter groups.

### DB specifications

Engine

Amazon Aurora with MySQL compatibility ▼

Capacity type [Info](#)

☐ Provisioned  
You provision and manage the server instance sizes.

☒ Serverless  
You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

Version

Aurora (MySQL 5.7) 2.07.1 ▼

To see more versions, modify the capacity types. [Info](#)

Establish DB instance identifier and connectivity:

DB Instance identifier [Info](#)

a4lwordpress-serverless

### Connectivity

Virtual private cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB cluster.

a4l-vpc1 (vpc-0b0d7069d1530bf42) ▼

Only VPCs with a corresponding DB subnet group are listed.

[i](#) After a database is created, you can't change the VPC selection.

Explore serverless options:

### auroraserverless-dbcluster-1a3t0gk56mlb0

Modify Actions

#### Summary

DB cluster id auroraserverless-dbcluster-1a3t0gk56mlb0	CPU -	Info Available	Current capacity 2 capacity units
Role Serverless	Current activity	Engine Aurora MySQL	Region & AZ us-east-1

Big advantage is pay model changes to 'pay what you use'

### Database

#### Configuration

Resource id  
cluster-54CEQRA7ZH32BYU7ZOKY37GYU

Amazon Resource Name (ARN)  
arn:aws:rds:us-east-1:945690336440:cluster:aurora-serverless-dbcluster-1a3t0gk56mlb0

DB name  
a4lwordpress

DB cluster parameter group  
default:aurora-mysql5.7

Deletion protection  
Disabled

#### Capacity settings

Minimum Aurora capacity unit  
1 capacity unit

Maximum Aurora capacity unit  
2 capacity units

Pause compute capacity after consecutive minutes of inactivity  
5 minutes

Force scaling the capacity to the specified values when the timeout is reached  
Not Enabled

#### Availability

IAM db authentication  
Not Enabled

Master username  
a4lwordpress

Master password  
\*\*\*\*\*

#### Encryption

Encryption  
Enabled

