



AWS
re:Start
LAB

Introduction to Amazon Aurora



WEEK 7





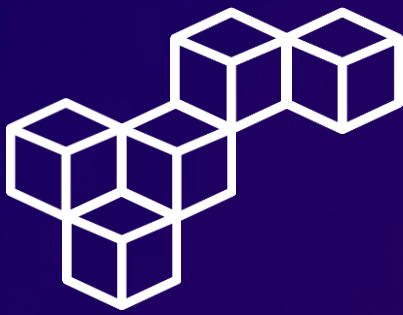
Overview

Aurora is a fully managed, MySQL-compatible, relational database engine that combines the performance and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. It delivers up to five times the performance of MySQL without requiring changes to most of your existing applications that use MySQL databases.

Amazon Aurora offers high-performance and scalable database solutions. Creating an Aurora instance allows efficient data management, while connecting to Amazon EC2 instances streamlines integration. Configuring EC2 to connect to Aurora ensures smooth data communication, enabling seamless querying for valuable insights and informed decision-making. Overall, Aurora's advanced features and architecture simplify database administration and enhance performance, making it a top choice for scalable and reliable data solutions.

Topics covered

- Create an Aurora instance
- Connect to a pre-created Amazon Elastic Compute Cloud (Amazon EC2) instance
- Configure the Amazon EC2 instance to connect to Aurora
- Query the Aurora instance

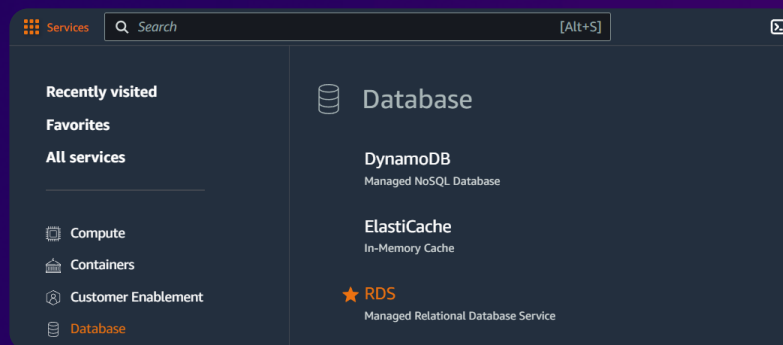


Task 1

Create an Aurora instance

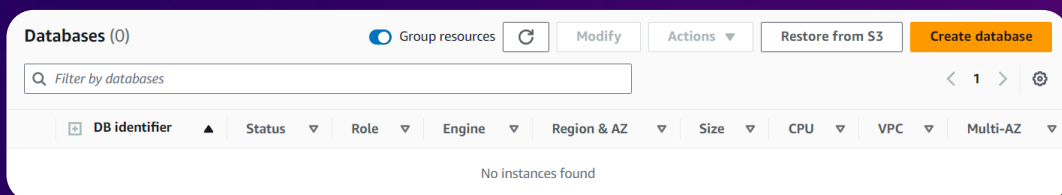
Step 1: Access the RDS database service

Open the AWS Management Console, and select RDS.



Step 2: Create database

Navigate to the Databases section, and select [Create database](#).





Task 1

Create an Aurora instance

Step 3: Choose a database creation method

In the **Choose a database creation method** section, choose [Standard create](#).

Choose a database creation method [Info](#)

☒ **Standard create**
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy create**
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


Step 4: Engine options

In the **Engine options** section, for Engine type, choose [Aurora \(MySQL Compatible\)](#), for Engine version, choose the default for major version 8.0.

Engine options

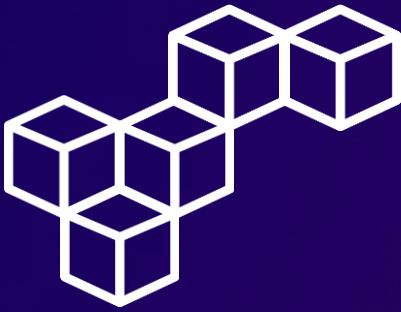
Engine type [Info](#)

☒ **Aurora (MySQL Compatible)**


☐ **Aurora (PostgreSQL Compatible)**


Engine Version

Aurora MySQL 3.04.2 (compatible with MySQL 8.0.28) - default for major version 8.0 [▼](#)



Task 1

Create an Aurora instance

Step 5: Templates

In the **Templates** section, choose [Dev/Test](#).

Templates
Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**
This instance is intended for development use outside of a production environment.

Step 6: Settings

In the **Settings** section, configure the following options.

Settings

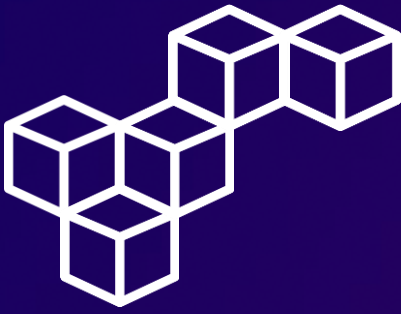
DB cluster identifier [Info](#)
Enter a name for your DB cluster.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

Master password [Info](#)

Confirm master password [Info](#)



Task 1

Create an Aurora instance

Step 7: Instance configuration

In the **Instance configuration** section, for DB instance class, configure the following settings.

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

☐ Standard classes (includes m classes)

☐ Memory optimized classes (includes r and x classes)

☒ Burstable classes (includes t classes)

db.t3.medium
2 vCPUs 4 GiB RAM Network: 2,085 Mbps

Step 8: Availability & durability

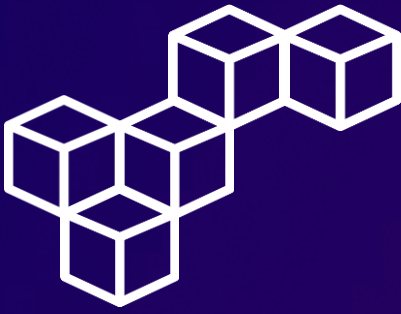
In the **Availability & durability** section, for Multi-AZ deployment, choose [Don't create an Aurora Replica](#).

Availability & durability

Multi-AZ deployment [Info](#)

☐ Create an Aurora Replica or Reader node in a different AZ
(recommended for scaled availability)
Creates an Aurora Replica for fast failover and high availability.

☒ Don't create an Aurora Replica



Task 1

Create an Aurora instance

Step 9: Connectivity

In the **Connectivity** section, configure the following options.

Connectivity [Info](#)

Virtual private cloud (VPC) [Info](#)
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

LabVPC (vpc-0ff3b06cbb91cf1ff)
2 Subnets, 2 Availability Zones

DB subnet group [Info](#)
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

dbsubnetgroup
2 Subnets, 2 Availability Zones

Public access [Info](#)

☐ Yes
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

☒ No
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

VPC security group (firewall) [Info](#)
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing
Choose existing VPC security groups

☐ Create new
Create new VPC security group

Existing VPC security groups

Choose one or more options

DBSecurityGroup X

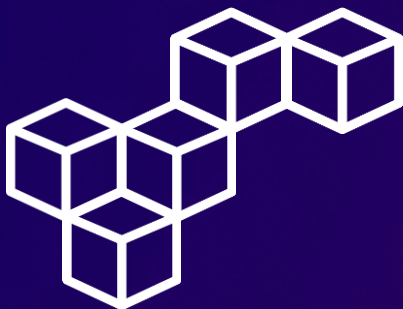
Step 10: Monitoring

In the **Monitoring** section, for Additional configuration, clear the check box for **Enable Enhanced Monitoring**.

Monitoring

Additional configuration
Enhanced Monitoring

☐ **Enable Enhanced Monitoring**
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.



Task 1

Create an Aurora instance

Step 11: Additional configuration

In the **Additional configuration** section, configure the following options.

▼ **Additional configuration**

Database options, encryption turned off, failover, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)

world

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

Encryption

☐ **Enable encryption**

Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

Maintenance

Auto minor version upgrade [Info](#)

☐ **Enable auto minor version upgrade**

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Step 12: Review database instance creation

Verify the availability of the [aurora-instance-1](#) database.

Databases (2)

Group resources

Refresh

Modify

Actions

Restore from S3

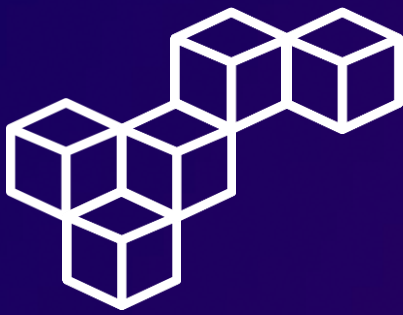
Create database

Filter by databases

< 1 >

Settings

<div><div></div></div>	DB identifier ▲	Status ▼	Role ▼	Engine ▼	Region & AZ ▼	Size ▼	VPC ▼
<div><div></div></div>	<div><div><div>aurora</div></div></div>	<div><div></div><div>Available</div></div>	Regional cluster	Aurora MySQL	us-west-2	1 instance	-
<div><div></div></div>	<div><div><div>aurora-instance-1</div></div></div>	<div><div></div><div>Available</div></div>	Writer instance	Aurora MySQL	us-west-2a	db.t3.medium	vpc-0ff3b06cbb91cf1ff

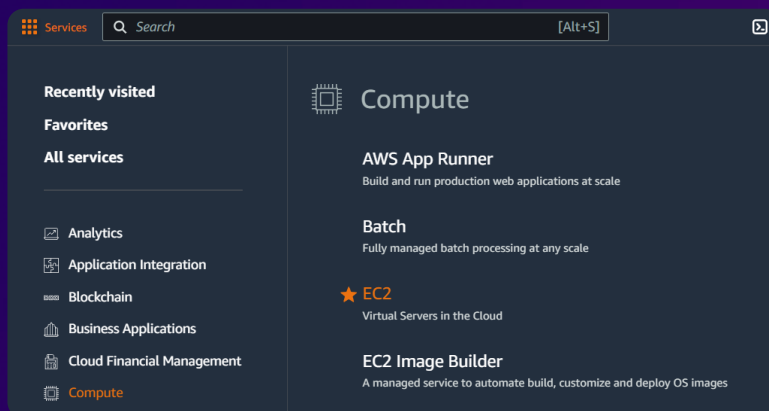


Task 2

Connect to an Amazon EC2 Linux instance

Step 1: Access the EC2 Management Console

In the AWS Management Console, select EC2.



Step 2: Review running EC2 instances

Navigate to the **Instances** section. The running **Command Host** EC2 instance is listed.

Instances (1) Info							
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>							
All states ▼							
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Status check	Availability Zone ▼	Public IPv4 ... ▼	Security group name ▼
<input type="checkbox"/>	Command Host	i-0031316fba9bbeae2	Running	2/2 checks passed	us-west-2a	35.93.177.207	EC2SecurityGroup



Task 2

Connect to an Amazon EC2 Linux instance

Step 3: Connect to the instance

Connect to the **Command Host** EC2 instance using [Session Manager](#).

Connect to instance [Info](#)

Connect to your instance i-00d6093ebb7333c21 (Command Host) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Session Manager usage:

- Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

Cancel

Connect

Step 4: Review connection

You have successfully connected to the Amazon EC2 instance named **Command Host**.

Session ID: user3195341=Cristhian_Becerra-0d9594b06f6a69107

Instance ID: i-0031316fba9bbeae2

```
sh-4.2$
```



Task 3

Configure the Amazon EC2 Linux instance to connect to Aurora

Step 1: Install the DB client

To install the MariaDB client, run the following command.

```
sh-4.2$ sudo yum install mariadb -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution
```

Step 2: Review the Aurora DB cluster

Go back to the RDS Management Console, navigate to the **Databases** section, and select the [aurora](#) DB cluster.

The screenshot shows the AWS RDS Management Console 'Databases (2)' page. It includes a search bar, a table of database instances, and a 'Create database' button. The table lists two instances: 'aurora' (Regional cluster) and 'aurora-instance-1' (Writer instance).

DB identifier	Status	Role	Engine	Region & AZ	Size	VPC
aurora	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	-
aurora-instance-1	Available	Writer instance	Aurora MySQL	us-west-2a	db.t3.medium	vpc-0ff3b06cbb91cf1ff







Task 3

Configure the Amazon EC2 Linux instance to connect to Aurora

Step 3: Review Endpoints

Choose the **Connectivity & Security** tab, and in the **Endpoints** section, copy the **Endpoint name** for the Writer instance.

Endpoints (2)					Actions	Create custom endpoint
<input type="text" value="Find resources"/>					< 1 >	⚙
Endpoint name	Status	Type	Port			
 aurora.cluster-c7ewsm2ky6s6.us-west-2.rds.amazonaws.com	 Available	Writer	3306			
 aurora.cluster-ro-c7ewsm2ky6s6.us-west-2.rds.amazonaws.com	 Available	Reader	3306			

Step 4: Connect to Aurora

To connect to the Aurora instance, run the following command.

```
sh-4.2$ mysql -u admin --password='admin123' -h aurora.cluster-c7ewsm2ky6s6.us-west-2.rds.amazonaws.com
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 104
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```



Task 4

Create a table and insert and query records

Step 1: Switch to the world database

To list the available databases and switch to the **world** database, run the following commands.

```
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| world |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> USE world;
Database changed
MySQL [world]> 
```

Step 2: Create a new table

To create a new table in the **world** database, run the following command.

```
MySQL [world]> CREATE TABLE `country` (
  -> `Code` CHAR(3) NOT NULL DEFAULT '',
  -> `Name` CHAR(52) NOT NULL DEFAULT '',
  -> `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') NOT NULL DEFAULT 'Asia',
  -> `Region` CHAR(26) NOT NULL DEFAULT '',
  -> `SurfaceArea` FLOAT(10,2) NOT NULL DEFAULT '0.00',
  -> `IndepYear` SMALLINT(6) DEFAULT NULL,
  -> `Population` INT(11) NOT NULL DEFAULT '0',
  -> `LifeExpectancy` FLOAT(3,1) DEFAULT NULL,
  -> `GNP` FLOAT(10,2) DEFAULT NULL,
  -> `GNPold` FLOAT(10,2) DEFAULT NULL,
  -> `LocalName` CHAR(45) NOT NULL DEFAULT '',
  -> `GovernmentForm` CHAR(45) NOT NULL DEFAULT '',
  -> `Capital` INT(11) DEFAULT NULL,
  -> `Code2` CHAR(2) NOT NULL DEFAULT '',
  -> PRIMARY KEY (`Code`)
  -> );
Query OK, 0 rows affected, 7 warnings (0.04 sec)
```



Task 4

Create a table and insert and query records

Step 3: Insert new records

To insert new records into the **country** table that you just created, run the following commands.

```
MySQL [world]> INSERT INTO `country` VALUES ('GAB','Gabon','Africa','Central Africa',267668.00,1960,1226000,50.1,5493.00,5279.00,'Le Gabon','Republic',902,'GA');
Query OK, 1 row affected (0.00 sec)

MySQL [world]> INSERT INTO `country` VALUES ('IRL','Ireland','Europe','British Islands',70273.00,1921,3775100,76.8,75921.00,73132.00,'Ireland/Eire','Republic',1447,'IE');
Query OK, 1 row affected (0.00 sec)

MySQL [world]> INSERT INTO `country` VALUES ('THA','Thailand','Asia','Southeast Asia',513115.00,1350,61399000,68.6,116416.00,153907.00,'Prathet Thai','Constitutional Monarchy',3320,'TH');
Query OK, 1 row affected (0.00 sec)

MySQL [world]> INSERT INTO `country` VALUES ('CRI','Costa Rica','North America','Central America',51100.00,1821,4023000,75.8,10226.00,9757.00,'Costa Rica','Republic',584,'CR');
Query OK, 1 row affected (0.00 sec)

MySQL [world]> INSERT INTO `country` VALUES ('AUS','Australia','Oceania','Australia and New Zealand',7741220.00,1901,18886000,79.8,351182.00,392911.00,'Australia','Constitutional Monarchy, Federation',135,'AU');
Query OK, 1 row affected (0.01 sec)

MySQL [world]> 
```

Step 4: Query the table

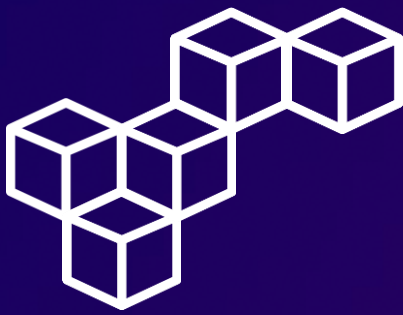
To query the table, run the following **SELECT** statement.

```
MySQL [world]> SELECT * FROM country WHERE GNP > 35000 and Population > 10000000;
```

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm	Capital	Code2
AUS	Australia	Oceania	Australia and New Zealand	7741220.00	1901	18886000	79.8	351182.00	392911.00	Australia	Constitutional Monarchy, Federation	135	AU
THA	Thailand	Asia	Southeast Asia	513115.00	1350	61399000	68.6	116416.00	153907.00	Prathet Thai	Constitutional Monarchy	3320	TH

```
2 rows in set (0.00 sec)

MySQL [world]> 
```



Conclusions

Amazon Aurora

Amazon Aurora offers high performance and scalability, making it an ideal choice for managing large datasets efficiently.

Create an Aurora instance

Creating an Aurora instance provides users with a robust and reliable database solution tailored to their specific needs.

Connect to an Amazon EC2 instance

Connecting to an Amazon EC2 instance enables seamless integration and utilization of Aurora's advanced features.

Configure the EC2 instance to connect to Aurora

Configuring the EC2 instance to connect to Aurora ensures smooth communication and data accessibility between applications and databases.

Query the Aurora instance

Querying the Aurora instance allows for extracting valuable insights and performing data analysis, supporting informed decision-making processes.



Cristhian Becerra



[cristhian-becerra-espinoza](#)



+51 951 634 354



cristhianbecerra99@gmail.com



Lima, Peru

