

How Backup and Restore works in AWS RDS



Amazon Relational Database Service (Amazon RDS)

Amazon RDS is a managed service which we can use to launch and manage relational databases on AWS. It makes it easy to set up, operate, and scale a relational database in the AWS cloud , also provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Amazon RDS supports 6 engines, PostgreSQL, MySQL, mariadb , oracledatabase , sqlserver and Amazon Aurora.

Here we're going to do the lab on MySQL.

Amazon RDS is an example for Relational database which stores data in a tabular format with rows and columns and uses sql query language to query the data. Each field in the table represents a data value , rows contain records and columns with attributes. It enables developers to define database's schema , relations and constraints between rows and columns. Highly structured and having more data accuracy and consistency.

Let's get started in the lab....!!!

Search for RDS from AWS console and click on create Database.

Then follow the steps shown below : select MySQL

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.

Engine options

Engine type [Info](#)

☐ Aurora (MySQL Compatible)



☐ Aurora (PostgreSQL Compatible)



☒ **MySQL**



☐ MariaDB



☐ PostgreSQL



☐ Oracle

ORACLE

☐ Microsoft SQL Server



Select Free-tier

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.



Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

[Info](#)

Provide Database Identifier name and Password'

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings


Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

☐ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

 If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.

[Learn more](#) 

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)


Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

Select Instance configuration as Burstable classes

Instance configuration

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

 **Amazon RDS Optimized Writes - new** [Info](#)
☐ Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)

- ☐ Standard classes (includes m classes)
- ☐ Memory optimized classes (includes r and x classes)
- ☒ Burstable classes (includes t classes)

db.t3.micro
 2 vCPUs 1 GiB RAM Network: 2,085 Mbps

☐ Include previous generation classes

Enable the public access from Connectivity

Public access [Info](#)

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

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

All the other settings are default and click on create database.

Once the newly created DB status showing as available , go to AWS console and search for AWS Backup.

RDS > Databases

Databases (1)										
DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
database-1	Available	Instance	MySQL Community	us-east-2b	db.t3.micro	4 Actions	2.81%	0 Connections	none	vpc-0901e48909247ddcb

Goto Protected resources from LHS and click on Create on-demand backup.

Select the Resource type as RDS and select the DB name which we created from Database name.


Here I'm selecting a retention period as 7 days, you can choose it on your choice. Rest all settings are default and click on Create on-demand backup.

[AWS Backup](#) > [Protected resources](#) > Create on-demand backup

Create on-demand backup [Info](#)

Settings

Resource type **Database name**


RDS database-1 

Backup window



☒ Create backup now
Starts within 1 hour.

☐ Customize backup window

Retention period [Info](#)
Tell AWS Backup how long to store your backups.

7 Days 

Backup vault [Info](#)
Specify the Backup vault this backup is organized in.

Default  [Create new Backup vault](#) 

IAM role [Info](#)
Specify the IAM role that AWS Backup will assume when creating and managing backups on your behalf.

☒ Default role
If the AWS Backup default role is not present, one will be created for you with the correct permissions.

☐ Choose an IAM role

We can refresh the Jobs page and check the backup status, here we can see the backup is completed:

[AWS Backup](#) > [Jobs](#)


Jobs

In Jobs, you can monitor the status and other details of backup, restore, and copy activity.

[Backup jobs](#) [Restore jobs](#) [Copy jobs](#)

Backup jobs (1) [Info](#)
Records of your scheduled or on-demand backups.

 [Stop backup job](#) [Create report](#) [Last 24 hours](#) 



	Backup job ID	Status	Resource name	Resource ID	Resource type	Creation time	Start by
<input type="radio"/>	08269e76-d16c-41a7-a9dc-0a5300a66900	Completed	database-1	database-1	RDS	August 31, 2023, 18:11:59 (UTC+05:30)	August 31, 2023, 19:11:59 (UTC+05:30)

Now we're going to delete the database from RDS : select the database → Actions → Delete

Delete database-1 instance?


Are you sure you want to Delete the **database-1** DB Instance?

☐ Create final snapshot
Determines whether a final DB Snapshot is created before the DB instance is deleted.

☒ I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.

To confirm deletion, type *delete me* into the field.

delete me

 We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.

Cancel Delete

Once the database is deleted , goto AWS Backup → Protected Resources → select the database and → click on Restore

[AWS Backup](#) > [Protected resources](#) > database-1

Resource details info

Create on-demand backup

Summary

Resource name database-1	Resource ID database-1	Resource type RDS	Last backup August 31, 2023, 18:11:59 (UTC+05:30)
-----------------------------	---------------------------	----------------------	--

Recovery points (1) info

↺ 1 ↻

Recovery point ID	Status	Backup type	Creation time
awsbackup-job-08269e76-d16c-41a7-a9dc-0a5300a66900	Completed	Snapshot	August 31, 2023, 18:12:22 (UTC+05:30)

It will take sometime to complete the process.

Select the DB instance class as db.t2.micro and rest all fields as default

[AWS Backup](#) > [Protected resources](#) > [database-1](#) > Restore backup

Restore backup [Info](#)

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.

Instance specifications

DB engine
Name of the database engine to be used for this instance

MySQL Community Edition ▼

License Model
License type associated with the database engine

general-public-license ▼

DB instance class
Contains the compute and memory capacity of the DB Instance.

db.t2.micro — 1 vCPU, 1 GiB RAM ▼

Storage type [Info](#)

General Purpose SSD (gp2) ▼

Select “ Do not create Standby instance “ option . The other option is only recommends for if we’re using RDS for production usage.

Availability and durability

Deployment options
The deployment options below are limited to those supported by the engine you selected above.

☐ 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.

☒ Do not create a standby instance

Enable public access from Network and Security settings and click on Restore backup.

Public accessibility [Info](#)

☒ Yes

EC2 instances and devices outside of the VPC hosting the DB instance will connect to the instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

☐ No

DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

All other settings are left as it is. We will get a restore job ID like below:

Restore jobs (1) Info						Refresh	Create report	Last 24 hours ▼
Records of your backup restoration.								
<input type="text" value="Filter restore jobs by job ID, status or resource ID"/>								
Restore job ID	Status	Resource ID	Resource type	Creation time	Recovery point ID			
68864300-B13F-C833-8695-018C2B4951D5	Running	-	RDS	August 31, 2023, 18:32:55 (UTC+05:30)	awsbackupjob-08269e76-d16c-41a7-a9dc-0a5300a66900			

Go to RDS page and select Databases there , we can see the newly created restore database there:

RDS > Databases											
Databases (1)											
<input type="text" value="Filter by databases"/>											
<input checked="" type="checkbox"/>	DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
<input type="radio"/>	db1-restore	Creating	Instance	MySQL Community	us-east-2a	db.t2.micro	-	-	-	none	vpc-0901e48909247ddcb

Once the restore is completed , go to RDS→ Databases → and delete db1-restore.

Then go to AWS Backup → Backup vaults , we could see the restoration is completed

Recovery points (1) Info									Refresh	Deselect all	Actions
<input type="text" value="Filter by resource type, recovery point ID, status, resource ID or source account ID"/>											
<input type="checkbox"/>	Recovery point ID	Status	Resource name	Resource ID	Resource type	Backup type	Creation time	Source acco			
<input type="checkbox"/>	awsbackupjob-08269e76-d16c-41a7-a9dc-0a5300a66900	Completed	database-1	database-1	RDS	Snapshot	August 31, 2023, 18:11:59 (UTC+05:30)	-			

This is how backup and restore works in AWS.

Enjoy your learning...!!!

