



# What is AWS RDS and how to set up a database?

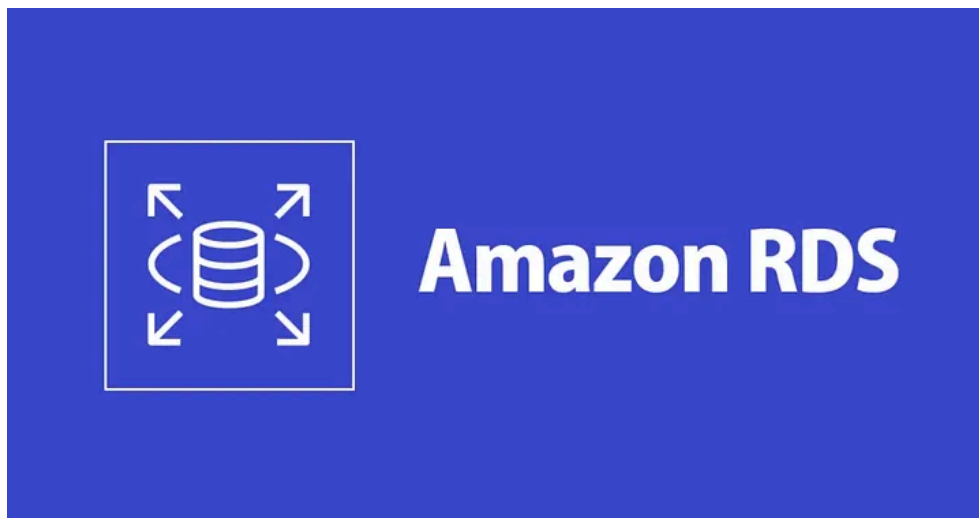


Kuldeepkumawat · Follow

5 min read · 20 hours ago



1



**Amazon RDS (Relational Database Service)** is a managed service provided by AWS that makes it easy to set up, operate, and scale a relational database in the cloud. AWS RDS automates time-consuming administrative tasks such as hardware provisioning, database setup, patching, backups, and scaling, allowing developers to focus on building applications.

AWS RDS supports various database engines:

- **Amazon Aurora** (compatible with MySQL and PostgreSQL)
- **MySQL**
- **PostgreSQL**
- **MariaDB**

- Oracle
- Microsoft SQL Server

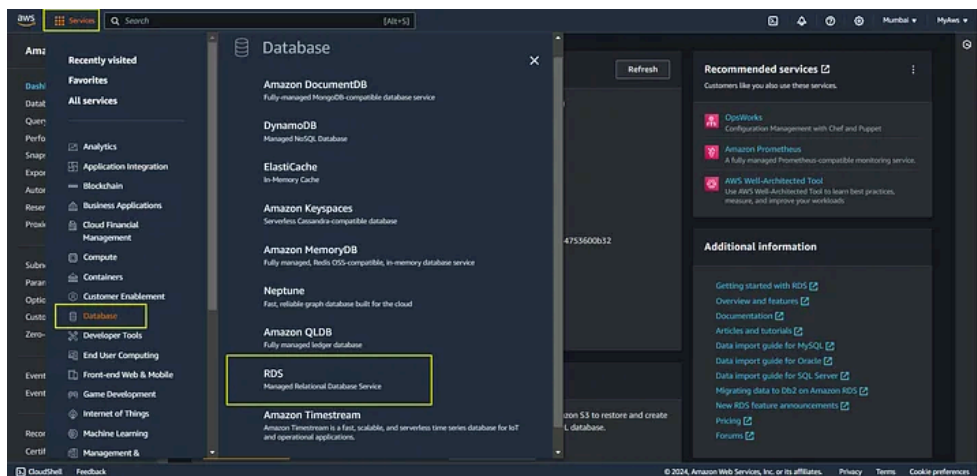
RDS offers benefits such as automated backups, easy scaling, high availability with multi-AZ deployments, and read replicas for improved read performance.

## How to Set Up a Database in AWS RDS

Here's a step-by-step guide on setting up a database in AWS RDS.

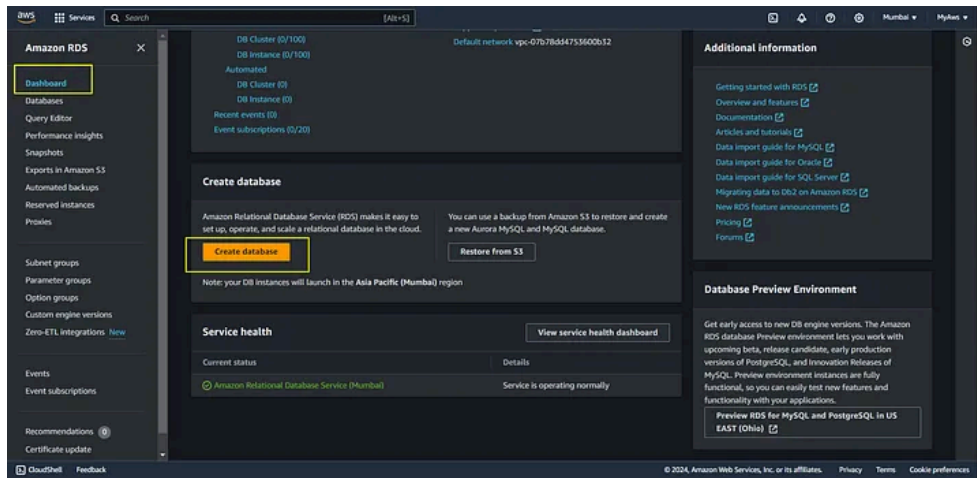
### Step 1: Sign in to the AWS Management Console

1. Go to the [AWS Management Console](#).
2. Search for RDS in the search bar and click on **Amazon RDS** under Services.



### Step 2: Launch an RDS Instance

1. On the RDS dashboard, click on **Create Database**.
2. Choose the **Standard Create** option for more control over settings.

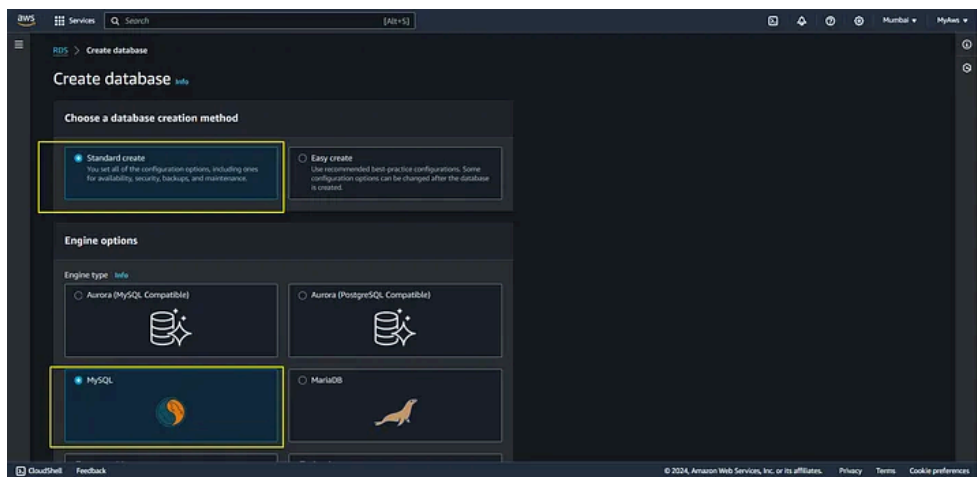


## Step 3: Select a Database Engine

1. Choose a database engine for your RDS instance. AWS RDS supports:

- Amazon Aurora
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

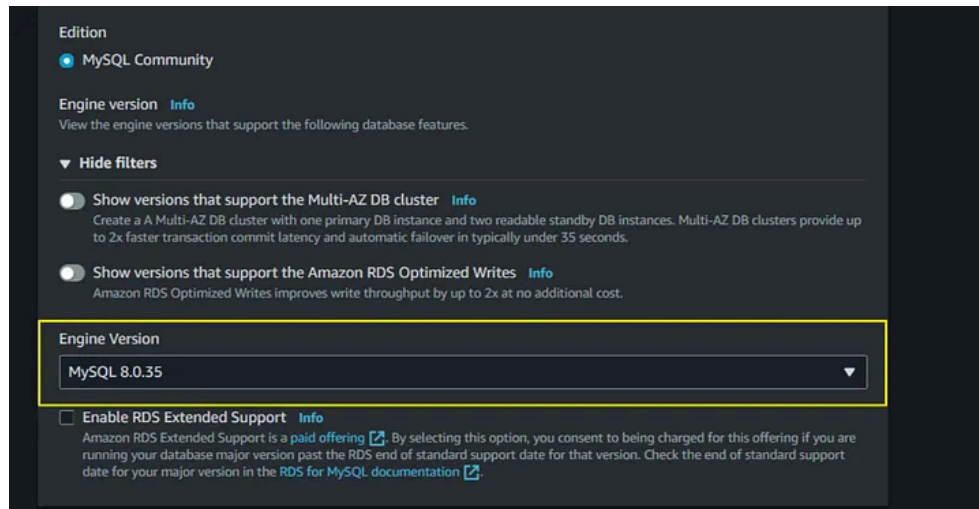
1. Select **MySQL** (for example) if you're familiar with it.



## Step 4: Choose a Database Version

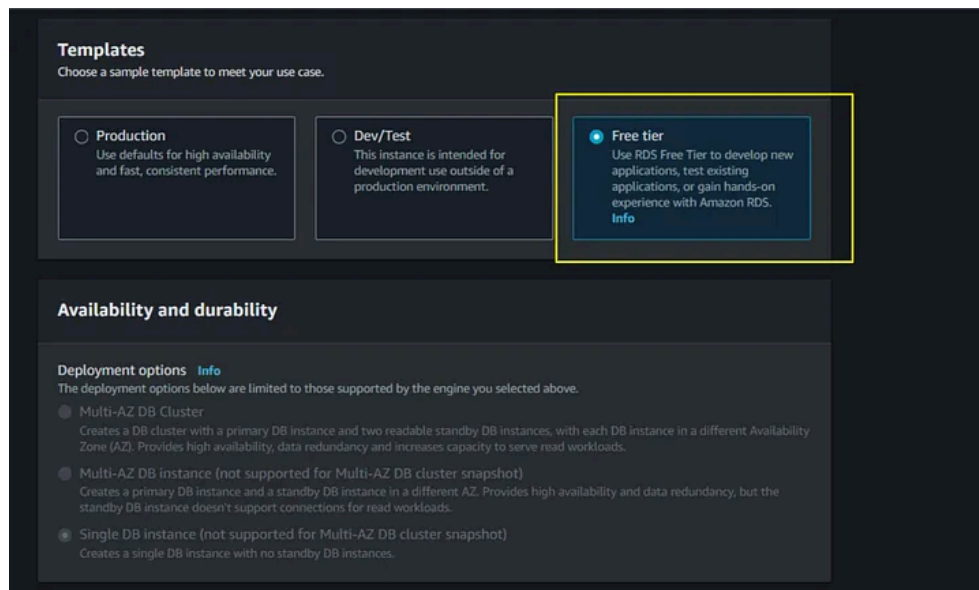
Select the database engine version you want to use. RDS will list the latest stable versions for each engine. For example, you can select **MySQL 8.0** if

you chose MySQL as your engine.



## Step 5: Select the Template

You can use it according to your use case I am using the free tier here.



## Step 6: Configure Settings

1. **DB Instance Identifier:** Give your database instance a unique name (e.g., `mydb-instance`).
2. **Master Username:** Set the master username (e.g., `admin`).
3. **Master Password:** Enter a secure password. You will need this to log in to the database.

**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.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

☐ **Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ **Self managed**  
Create your own password or have RDS create a password that you manage.

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

**Master password** [Info](#)  
  
**Password strength** Very strong  
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \* @

**Confirm master password** [Info](#)

## Step 7: Specify DB Instance Size

Choose your instance class depending on your expected workload:

1. **DB Instance Class:** Select the size of the instance based on your workload. For a small or development workload, you can choose `db.t3.micro` (if using the free tier).
2. **Storage:** Choose storage size (e.g., 20 GB) and type. AWS provides options like:
  - **General Purpose SSD** (default, cost-effective for most applications).
  - **Provisioned IOPS SSD** (for high-performance workloads).
1. You can also enable **Storage Auto-scaling** to automatically adjust storage size if needed.

**Storage**

**Storage type** [Info](#)  
Provisioned IOPS SSD (io2) storage volumes are now available.

**General Purpose SSD (gp3)**  
Performance scales independently from storage

**Allocated storage** [Info](#)  
20 GiB  
Minimum: 20 GiB, Maximum: 6,144 GiB

ⓘ After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. [Learn more](#)

▶ **Advanced settings**  
Baseline IOPS of 3,000 IOPS and storage throughput of 125 MiBps are included for allocated storage less than 400 GiB.

▼ **Storage autoscaling**

**Storage autoscaling** [Info](#)  
Provides dynamic scaling support for your database's storage based on your application's needs.

☒ **Enable storage autoscaling**  
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

**Maximum storage threshold** [Info](#)  
Charges will apply when your database autoscales to the specified threshold

1000 GiB  
Allocated storage value must be 22 GiB to 6,144 GiB

## Step 8: Configure Networking

1. **Virtual Private Cloud (VPC):** Select the VPC where the database will reside. Use the default VPC if you don't have a custom one.
2. **Subnet Group:** Leave the default subnet group unless you need custom networking.
3. **Public Access:** Choose **Yes** to make the database publicly accessible. This allows access from outside the VPC (e.g., from your local machine).
4. **VPC Security Groups:** Select an existing security group or create a new one that allows traffic to the database port (e.g., port 3306 for MySQL).

**Connectivity** Info

**Compute resource**  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☒ **Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

☐ **Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** Info  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

**Default VPC (vpc-07b78dd4753600b32)**  
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

**DB subnet group** Info  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

**default**

**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.

**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

launch-wizard-3 ✕

**Availability Zone** Info

No preference

## Monitoring:

- Enable **Enhanced Monitoring** to get detailed metrics on your database performance.

## Step 9: Review and Launch

1. Review all the configurations you have made.
2. Click **Create Database** to launch the RDS instance.
3. AWS will start provisioning the database, and the process may take a few minutes.

Services

Search

[Alt+S]

Tags - optional

A tag consists of a case-sensitive key-value pair.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Database authentication

Database authentication options [Info](#)

☒ Password authentication

Authenticates using database passwords.

☐ Password and IAM database authentication

Authenticates using the database password and user credentials through AWS IAM users and roles.

☐ Password and Kerberos authentication

Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Monitoring

☐ Enable Enhanced Monitoring

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

Additional configuration

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

Estimated Monthly costs

DB instance	15.33 USD
Storage	2.62 USD
Total	17.95 USD

This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel

Create database

Creating database mydb-instance

Your database might take a few minutes to launch. You can use settings from mydb-instance to simplify configuration of suggested database add-ons while we finish creating your DB for you.

View credential details

RDS > Databases

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (1)

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current activity
mydb-instance	Creating	Instance	MySQL Community	ap-south-1a	db.t4g.micro			

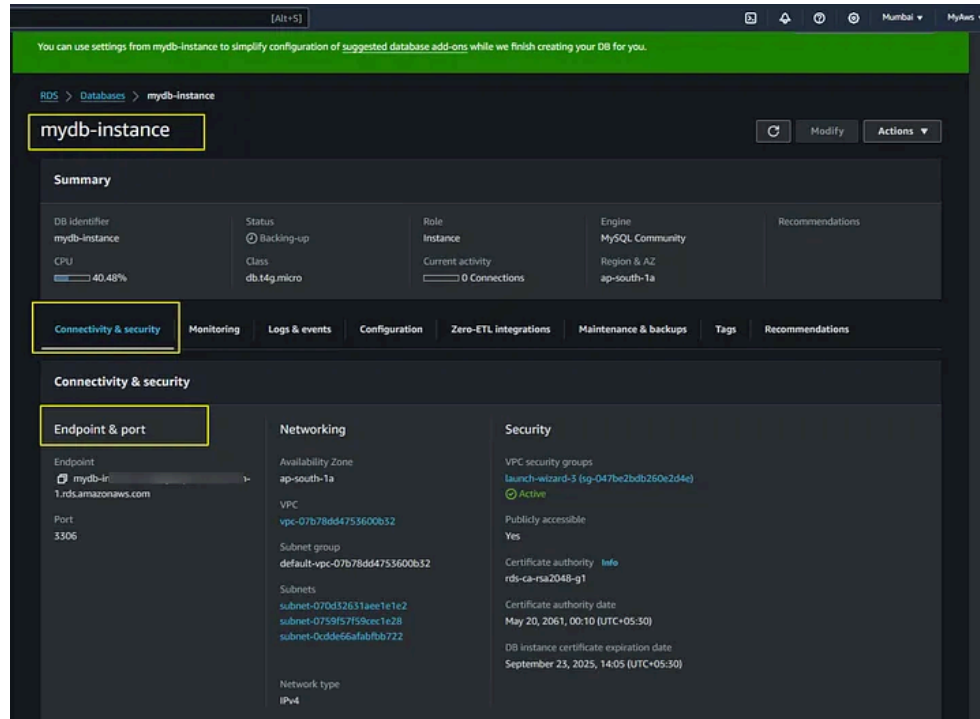


## Step 10: Connect to Your Database

Once your RDS instance is up and running, you can connect to the database:

### 1. Get Endpoint:

- Go to the RDS dashboard, click on **Databases**, and select your instance.
- Under the **Connectivity & Security** tab, copy the **Endpoint** (something like `mydb-instance.xyz123.us-west-2.rds.amazonaws.com`).



### 2. Connect Using a Client:

- Use a database client like **MySQL Workbench**, or **pgAdmin**, or connect via the command line.

For MySQL, use the following command:

```
mysql -h mydb-instance.xyz123.us-west-2.rds.amazonaws.com -u admin -p
```

```
root@ip-172-31-5-230:~# mysql -h mydb-instance.1u.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql>
```

🎯 We have successfully created the MySQL server and connected with the client command. 😊😊😊

## Step 11: Manage Your RDS Instance

- **Scaling:** You can scale your RDS instance up or down by changing the instance size in the console.
- **Backups and Restores:** AWS RDS manages backups automatically, but you can also create manual snapshots.
- **Monitoring:** Monitor your database's performance via CloudWatch and the RDS dashboard to ensure it runs optimally.

## Conclusion

AWS RDS simplifies database management by automating many of the heavy-lifting tasks such as backups, patching, and scaling. With a few steps, you can easily set up and manage a relational database, whether it's MySQL, PostgreSQL, or any other supported engine. This allows you to focus on building applications while AWS takes care of the underlying database infrastructure.

[Aws Rds Setup](#)

[Amazon Rds](#)

[Rds Mysql](#)

[Aws Rds Guide](#)

[AWS](#)