# AWS RDS(Relational Database Service)

# Relational Databases

* In a relational database, data is stored in a way that relates it to other pieces of data. An example of a relational database might be the coffee shop’s inventory management system. Each record in the database would include data for a single item, such as product name, size, price, and so on.
* Relational databases use structured query language (SQL) to store and query data. This approach allows data to be stored in an easily understandable, consistent, and scalable way. For example, the coffee shop owners can write an SQL query to identify all the customers whose most frequently purchased drink is a medium latte.

# What is Amazon RDS?

Amazon Relational Database Service (Amazon RDS) is a managed database service provided by Amazon Web Services (AWS). Amazon RDS was launched in 2009. Amazon RDS is a managed service that automates tasks such as hardware provisioning, database setup, patching, and backups. With these capabilities, you can spend less time completing administrative tasks and more time using data to innovate your applications. With RDS, you can set up a database instance in minutes. With Amazon RDS, you can choose from several popular relational database engines, including:

1. MySQL
2. PostgreSQL
3. MariaDB
4. Oracle Database
5. Microsoft SQL Server
6. Amazon Aurora (a MySQL and PostgreSQL-compatible database built for the cloud with enhanced performance and availability)

Amazon RDS handles routine database tasks such as provisioning, patching, backup, recovery, and scaling, allowing you to focus on building your applications instead of managing database infrastructure. RDS takes care of:

* **Provisioning:** No more worrying about server setup and configuration.
* **Patching:** RDS automatically applies critical security patches to your database.
* **Backups & Recovery:** Automated backups ensure easy recovery in case of any issues.
* **Scaling:** Scale your database storage and compute resources effortlessly to meet changing demands.

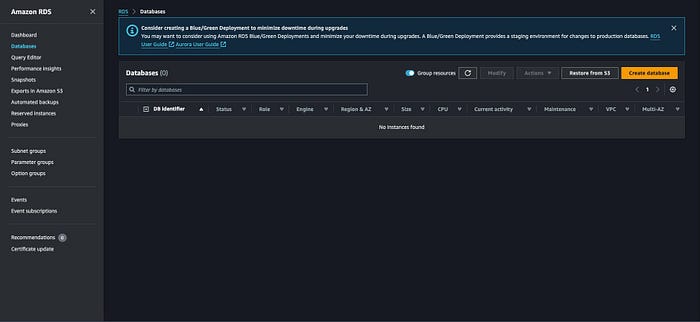
# Key Features of Amazon RDS

* **Automated Database Management:** Amazon RDS automates routine database tasks such as provisioning, patching, backups, and recovery, allowing developers to focus on building applications rather than managing infrastructure.
* **High Availability and Fault Tolerance:** RDS enables deployment across multiple Availability Zones (AZs), ensuring high availability and durability of database instances. In the event of AZ failures, RDS seamlessly redirects traffic to standby replicas, minimizing downtime and ensuring business continuity.
* **Scalability:** With Amazon RDS, scaling database resources is effortless. Whether it’s vertical scaling by adjusting instance size or horizontal scaling through adding replicas, RDS provides flexible options to accommodate fluctuating workloads and evolving business needs.
* **Security and Compliance:** RDS prioritizes data security with features such as encryption at rest and in transit, network isolation using Virtual Private Cloud (VPC), and IAM database authentication. These measures ensure data confidentiality, integrity, and regulatory compliance.
* **Monitoring and Performance Optimization:** RDS offers comprehensive monitoring capabilities through Amazon CloudWatch, allowing users to track database performance metrics and set up alarms for critical events. Additionally, RDS supports database logs for auditing, troubleshooting, and performance optimization.

While EBS (Elastic Block Store) and EFS (Elastic File System) offer storage solutions for cloud environments, they aren’t specifically designed for database management. That’s why RDS stands out for Relational Database Management.

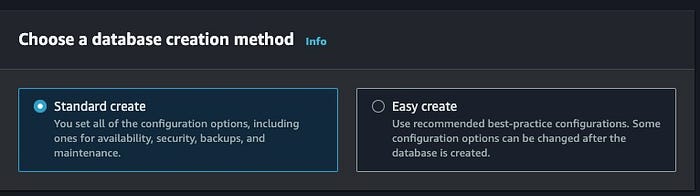
**NOTE : RDB is the managed the database in AWS provider has the handle the data base**

# Launching a MySQL RDS instance

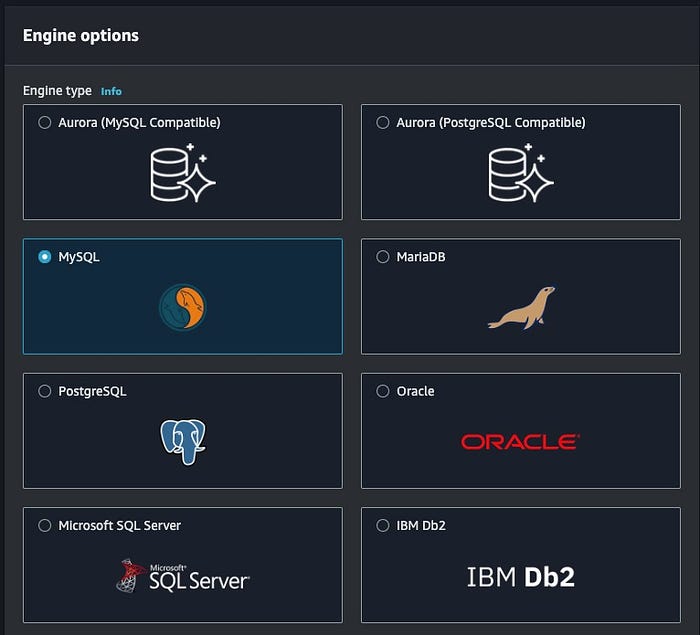


[RDS dashboard](https://console.aws.amazon.com/rds/)

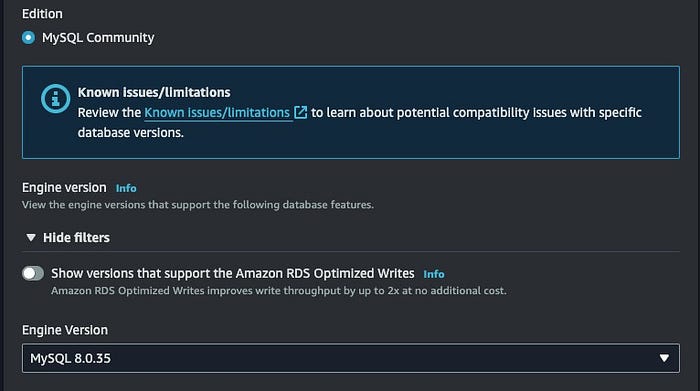
1. Click**Create database** button on [RDS dashboard](https://console.aws.amazon.com/rds/).



2. Choose one of create methods under **Choose a database creation method**. Easy Create will offer you a list of predefined settings. This usually works fine, and Amazon’s tutorial uses this method as well, but I’m choosing **Standard create** to give instructions about the options in detail.



Engine options — 1

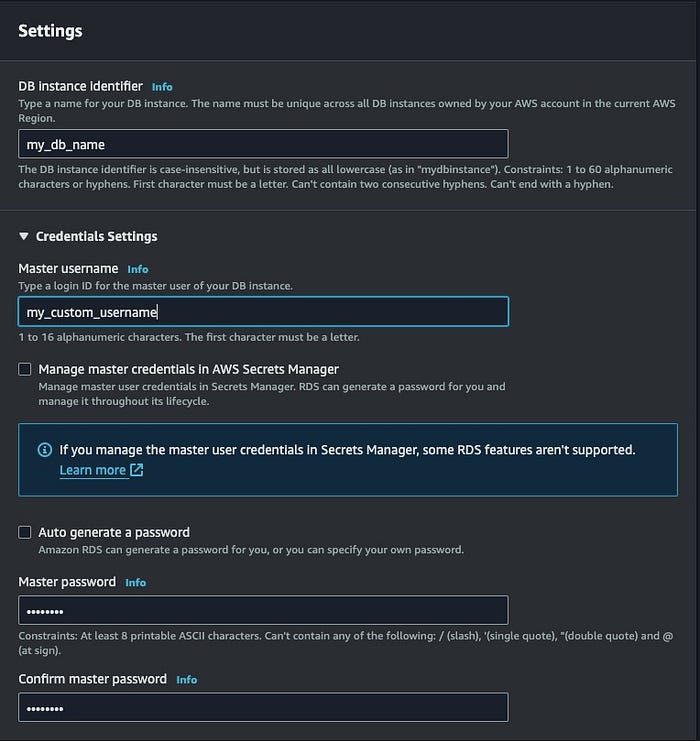


Engine options — 2

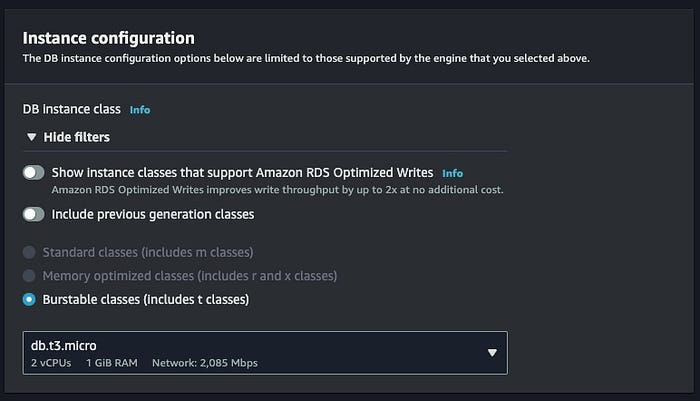
3. Choose MySQL and select a desired version under **Engine options**.



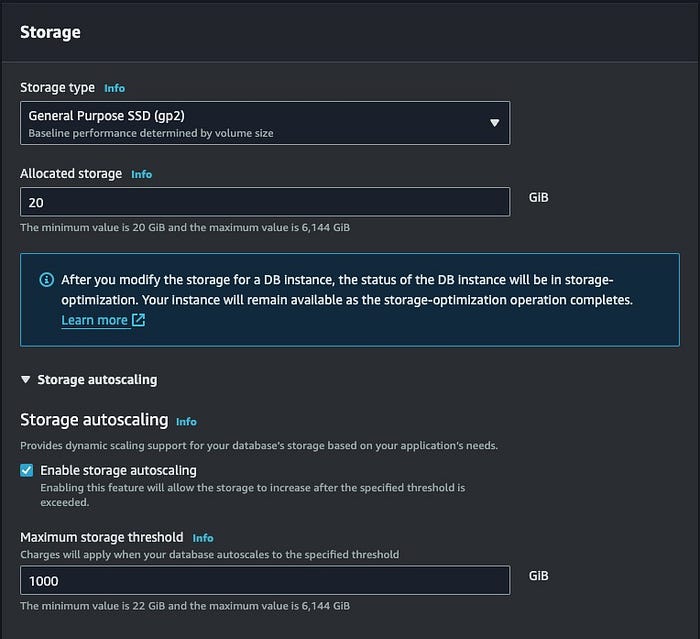
4. Select one of Production, Dev/Test, and Free tier under **Templates**. All of them differs in terms of performance and stability. If you are creating an instance for production server(the server that actually provides service to your customers), select **Production**. Select **Dev/Test** if you are creating an instance for your development or test server(A server being used while developing your product). For this tutorial, we are just making an instance to study. So, let’s choose**Free tier**.



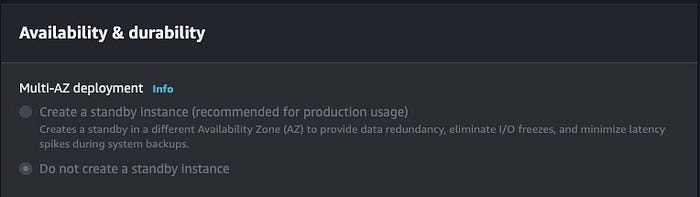
5. Under Settings, you can determine **DB instance identifier**(instance name), **Master username**(username for the database), and **Master password**(password for the database). You have to determine your password cautiously, though it’s obvious. There are some assaulters, who cracks the username and password by brute force search, access the cracked instances, encrypt the data inside, and threats you to transact some money to decrypt the data back. Because they use brute force search, which is try-and-error-kind search method, if you choose passwords like 12345678, and use default username, which is “admin”, it won’t take long for you to receive the threatening email.



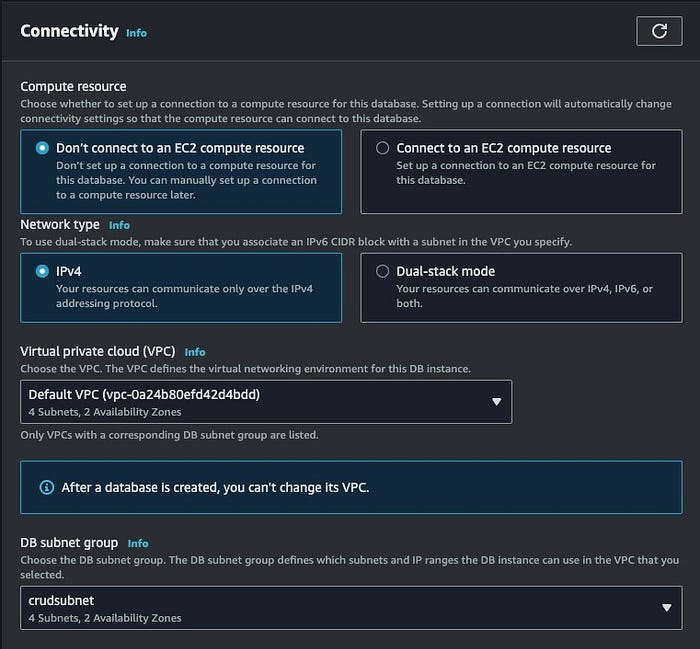
6. You can choose instance class under the **Instance configuration**. Unfortunately, the Free tier template supports Burstable classes only. You can choose the other classes if you choose Production or Dev/Test option for Template option.



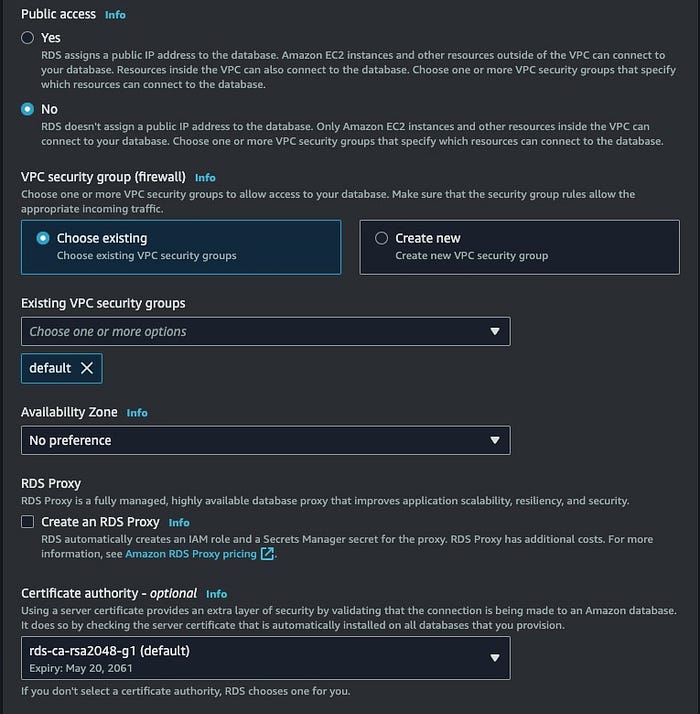
7. You can choose instance storage and the size of it under **Storage**. Depends on your Template and **storage type**options, the options under Storage vary. You can also set your storage to automatically scale its storage size by expanding **Storage autoscaling**.



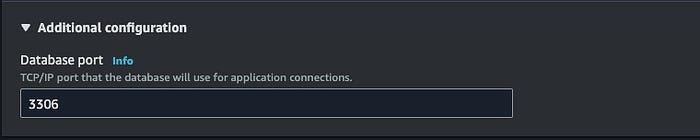
8. Free tier database instance does not offer **Availability & duration** options. Multi-AZ deployment is about if you are going to create replicas of your database in different availability zone. The replicas created can be used to troubleshoot, reduce latency, and data distribution(Distributed GIS).



Connectivity — 1

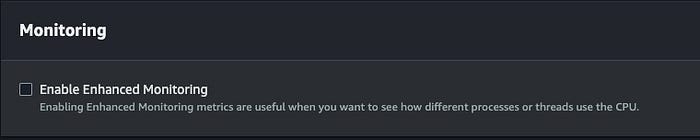


Connectivity — 2

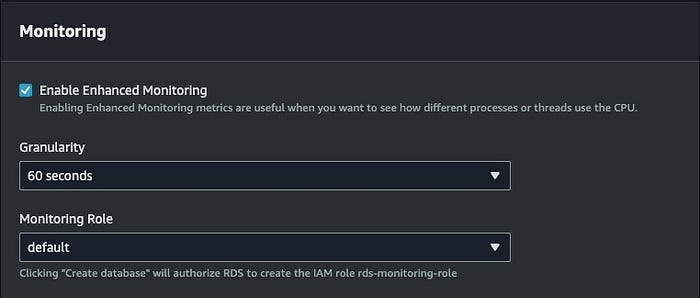


Connectivity — 3

9. Under **Connectivity**, you can choose whether you are going to connect this instance with an existing EC2 instance, so that it shares the same security group with the EC2 instance and live in the same VPC, or connect to another VPC(new/existing) with different subset group and security group. Also, you can determine your port number for this instance’s database engine under **Database port**, which appears when you expand **Additional configuration of Connectivity**. For this tutorial, let’s assume we are connecting this to an existing EC2 resource. Then you will see **EC2 instance** with dropbox, which you can choose an existing EC2 instance. After that, you can choose one of IPv4 and Dual-stack mode for **Network type**. You can set up other options as well, but you can just leave them with default values for now.

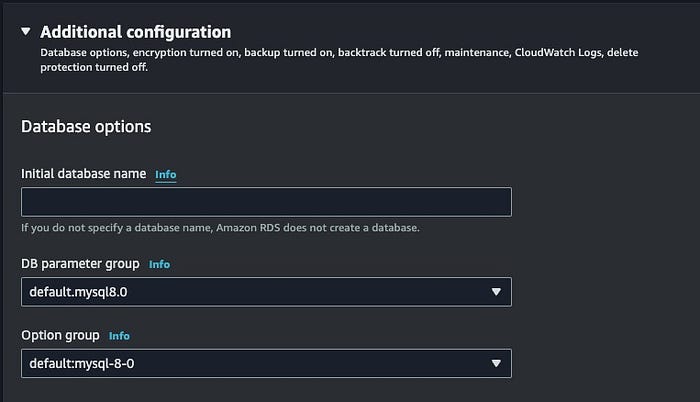


Monitoring — 1

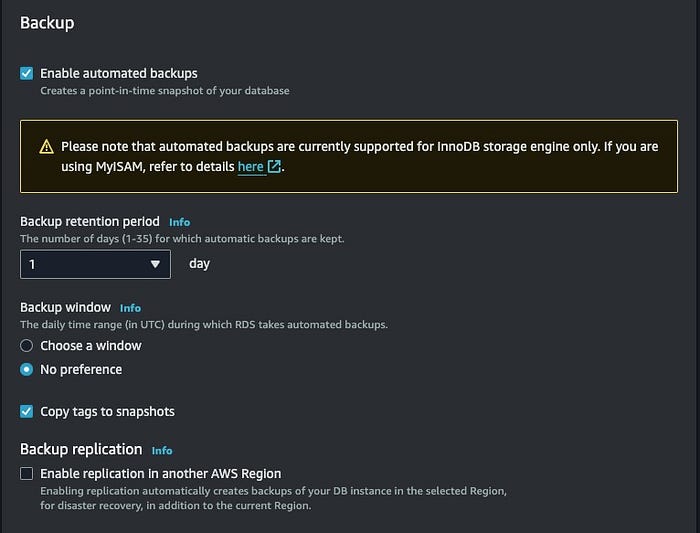


Monitoring — 2

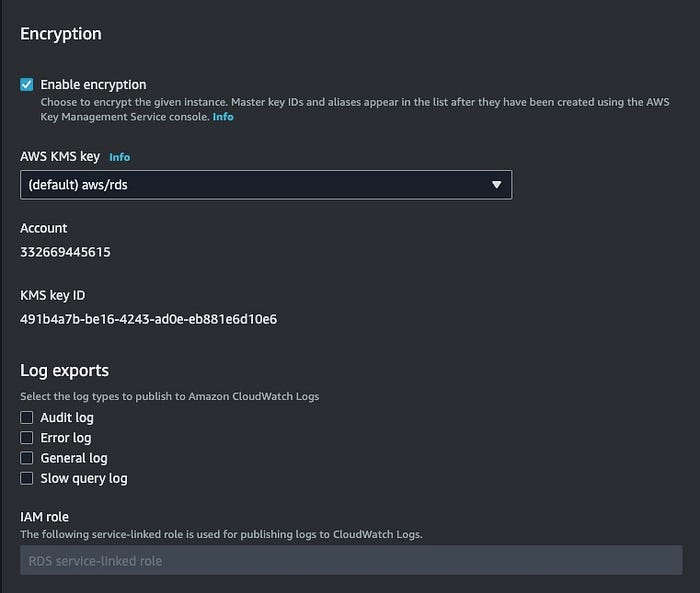
10. Under **Monitoring**, you can choose whether you are going to monitor your instance, and if so, you can choose the frequency of the logs, and the role of the monitoring.



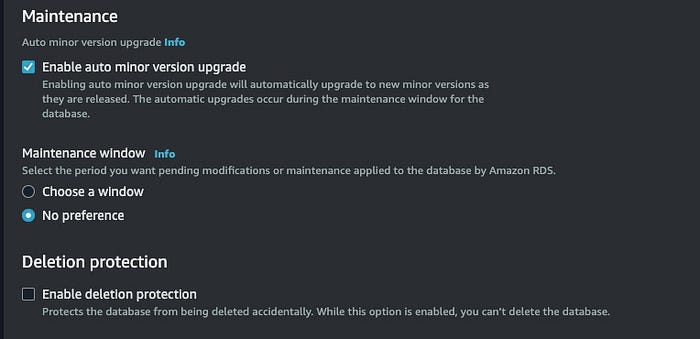
Additional configuration — Database options



Additional configuration — Backup

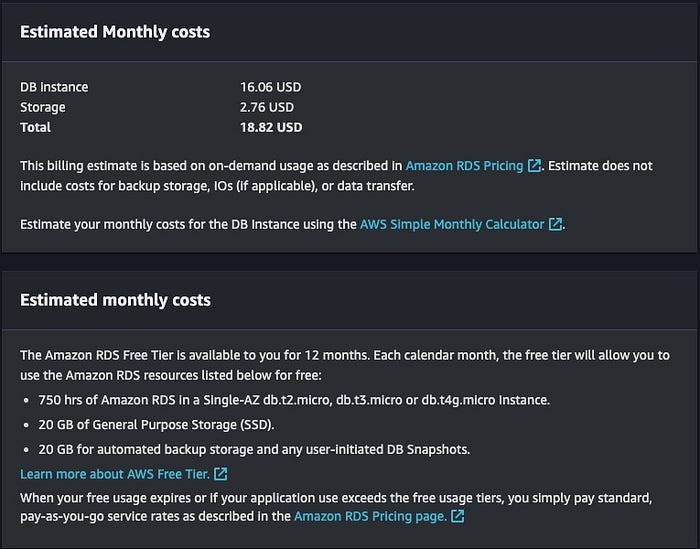


Additional configuration — Encryption, Log exports



Additional configuration — Maintenance

11. Under **Additional configuration**, you can determine **Database options**, **Backup**, **Encryption**, **Maintenance**, and **Deletion protections.**For now, let’s leave them as they are.



Estimated Monthly costs

At the end of the window, Amazon gives you Estimated Monthly costs. However, this number may not be the actual cost Amazon charges you because it is measured as much as you use. It can be less or more.

# ****Connect to your DB instance!****

* To connect to your DB instance, first we have to access our EC2 instance. The tutorial for accessing EC2 instance is in [my other post](https://medium.com/aws-tip/aws-ec2-345486d00409).
* Assuming you are already in your EC2 instance, install mysql first. Depend on the AMI that your instance uses, the shell script for mysql installation differs. To see the list of shell scripts, click [here](https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/installing.html).
* After installation, connect to the MySQL DB instance by passing the following shell script.

$ mysql -h ENDPOINT -P 3306 USERNAME -p  
$ PASSWORD(invisible)