

# Amazon Relational Database Service (RDS)

- Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
- The basic building block of Amazon RDS is the DB instance. A DB instance is an isolated database environment in the AWS Cloud.
- You can create and modify a DB instance by using the AWS Command Line Interface, the Amazon RDS API, or the AWS Management Console.
- Each DB instance runs a DB engine. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines.

## **DB instance storage comes in three types:**

**Magnetic:** 3TB and 1000 IOPS.

**General Purpose (SSD):** Storage Range 20GB-64TB for all. Sql 20Gb-16TB

**Provisioned IOPS (PIOPS).** 100GB-64TB. 1000 to 80,000 IOPS.

db.m4.xlarge 4cpu 16GB RAM

db.t3.large 2cpu 8GB RAM

db.t2.large 2cpu 8GB RAM

- **Amazon RDS manages backups, software patching, automatic failure detection, and recovery.**
- **Amazon RDS uses Network Time Protocol (NTP) to synchronize the time on DB Instances.**
- You can run your DB instance in several Availability Zones, an option called a Multi-AZ deployment.
- In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone.
- The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- The identifier is used as part of the DNS hostname allocated to your instance by RDS. Mostly DB name will be identifier.

For example, if you specify db1 as the DB instance identifier, then DNS endpoint will be db1.123456789012.us-east-1.rds.amazonaws.com, where 123456789012 is the fixed identifier for a specific region for your account.

- Amazon RDS creates a master user account for your DB instance as part of the creation process.
- master user has permissions to create databases and to perform create, delete, select, update, and insert operations
- You must set the master user password when you create a DB instance. You can also change the master user password and manage users using standard SQL commands.

## **Configuring the Processor:**

Number of CPU cores

Threads per core

you can configure the number of CPU cores and threads per core only for Oracle DB instances.

metrics are useful for monitoring storage for your DB instance:

1. IOPS: The number of I/O operations completed each second.

2. Latency: The elapsed time between the submission of an I/O request and its completion.
3. Throughput – The number of bytes each second that are transferred to or from disk.
4. Queue Depth – The number of I/O requests in the queue waiting to be serviced.. These has to be sent to device when it is free.

**\*\*\* Once your DB instance is available, you can access its endpoint via the DB instance description.** Using this endpoint you can construct the connection string required to connect directly with your DB instance. By default, customers are allowed to have up to a total of 40 Amazon RDS DB instances.

### To create a DB instance

1. In the navigation pane, choose **Databases**.
2. Choose **Create database**.
3. In **Choose a database creation method**, select **Standard Create**.
4. In **Engine options**, choose the engine type: MariaDB, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL. **Microsoft SQL Server** is shown here.

## Create database


### 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](#)


☐ Amazon Aurora  


☐ MySQL  


☐ MariaDB  


☐ PostgreSQL  


☐ Oracle  


☒ **Microsoft SQL Server**  


Edition

☒ **SQL Server Express Edition**  
 Affordable database management system that supports database sizes up to 10 GB.

☐ **SQL Server Web Edition**  
 In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.

☐ **SQL Server Standard Edition**  
 Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.

☐ **SQL Server Enterprise Edition**  
 Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

Version [Info](#)

5. For **Edition**, if you're using Oracle or SQL Server choose the DB engine edition that you want to use. MySQL has only one option for the edition, and MariaDB and PostgreSQL have none.
6. For **Version**, choose the engine version.
7. **Database features select:**

- One writer and multiple readers
- One writer and multiple readers
- Multiple writers
- Serverless

8. In **Templates**, choose the template that matches your use case. If you choose **Production**, the following are preselected in a later step:

- **Multi-AZ** failover option
- **Provisioned IOPS** storage option
- **Enable deletion protection** option

9. To enter your master password, do the following:

- In the **Settings** section, open **Credential Settings**.
- Clear the **Auto generate a password** check box.
- (Optional) Change the **Master username** value and enter the same password in **Master password** and **Confirm password**.

By default, the new DB instance uses an automatically generated password for the master user.

10. For the remaining sections, specify your DB instance settings. For information about each setting,

11. Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.

To connect to the DB instance as the master user, use the user name and password that appear.

12. For **Databases**, choose the name of the new DB instance.

#### For Linux, macOS, or Unix:

**\$ aws rds create-db-instance**

```
--engine sqlserver-se \
--db-instance-identifier mymsftsqlserver \
--allocated-storage 250 \
--db-instance-class db.m1.large \
--db-security-groups mydbsecuritygroup \
--db-subnet-group mydbsubnetgroup \
--master-username masterawsuser \
--master-user-password masteruserpassword \
--backup-retention-period 3
```

#### For Windows:

**\$ aws rds create-db-instance ^**

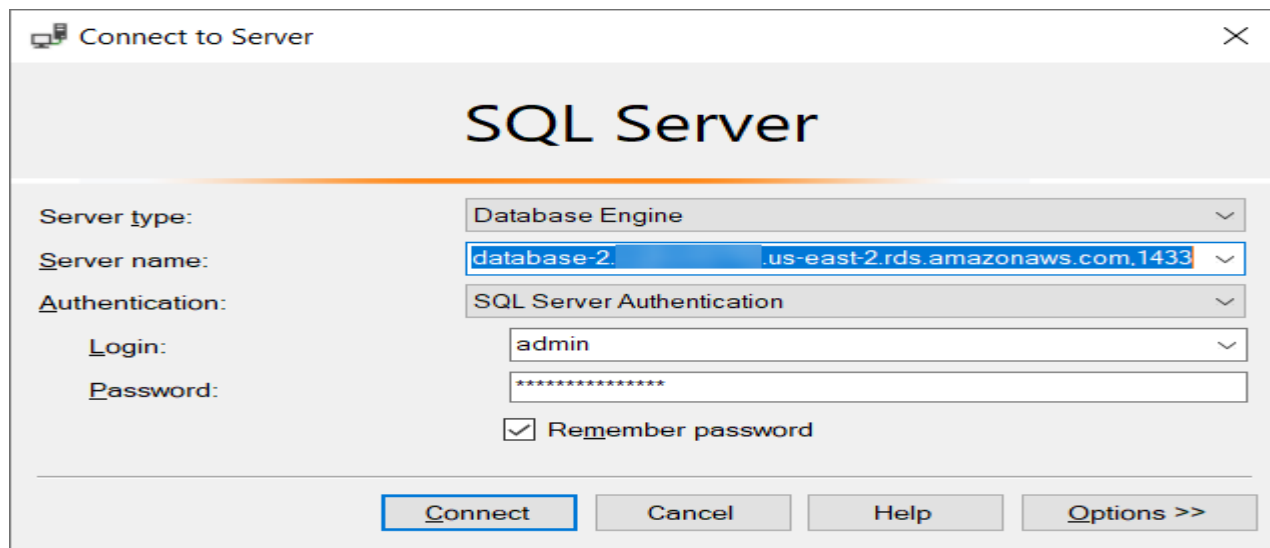
```
--engine sqlserver-se ^
--db-instance-identifier mydbinstance ^
--allocated-storage 250 ^
--db-instance-class db.m1.large ^
--db-security-groups mydbsecuritygroup ^
--db-subnet-group mydbsubnetgroup ^
--master-username masterawsuser ^
--master-user-password masteruserpassword ^
--backup-retention-period 3
```

#### **To connect to a DB instance using SSMS**

- In the upper-right corner of the Amazon RDS console, choose the AWS Region of your DB instance.
- Find the Domain Name System (DNS) name and port number for your DB instance:
  - Open the RDS console and choose **Databases** to display a list of your DB instances.

- b. Choose the SQL Server DB instance name to display its details.
- c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.
3. Start SQL Server Management Studio.

The **Connect to Server** dialog box appears.



4. Provide the information for your DB instance:
  - a. For **Server type**, choose **Database Engine**.
  - b. For **Server name**, enter the DNS name and port number of your DB instance, separated by a comma.

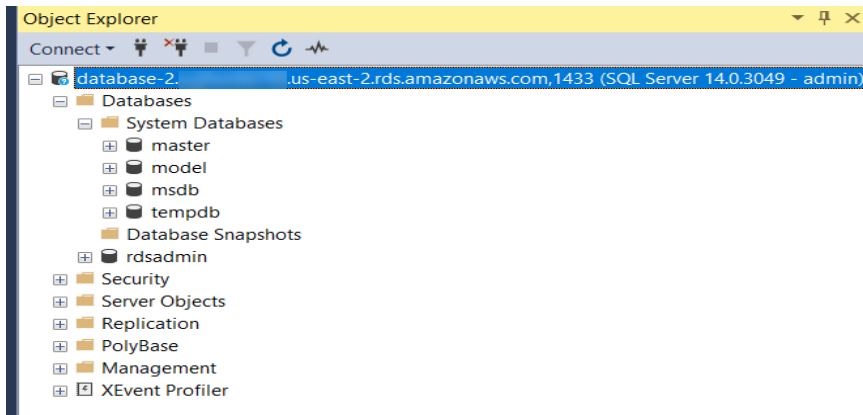
For example, your server name should look like the following.

database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433

- c. For **Authentication**, choose **SQL Server Authentication**.
- d. For **Login**, enter the master user name for your DB instance.
- e. For **Password**, enter the password for your DB instance.
5. Choose **Connect**.

After a few moments, SSMS connects to your DB instance. If you can't connect to your DB instance, see [Security Group Considerations](#) and [Troubleshooting Connections to Your SQL Server DB Instance](#).

6. Your SQL Server DB instance comes with SQL Server's standard built-in system databases (master, model, msdb, and tempdb). To explore the system databases, do the following:
  - a. In SSMS, on the **View** menu, choose **Object Explorer**.
  - b. Expand your DB instance, expand **Databases**, and then expand **System Databases**.



7. Your SQL Server DB instance also comes with a database named `rdsadmin`. Amazon RDS uses this database to store the objects that it uses to manage your database. The `rdsadmin` database also includes stored procedures that you can run to perform advanced tasks.
8. You can now start creating your own databases and running queries against your DB instance and databases as usual. To run a test query against your DB instance, do the following:
  - a. In SSMS, on the **File** menu point to **New** and then choose **Query with Current Connection**.
  - b. Enter the following SQL query.

```
select @@VERSION
```
  - c. Run the query. SSMS returns the SQL Server version of your Amazon RDS DB instance.

## **Amazon DynamoDB (NoSQL database)**

**Amazon DynamoDB is a key-value and document database** that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable database with built-in security

DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

DynamoDB is serverless with no servers to provision, patch, or manage and no software to install, maintain, or operate.