Creating an Amazon RDS DB instance

The basic building block of Amazon RDS is the DB instance, where you create your databases. You choose the engine-specific characteristics of the DB instance when you create it. You also choose the storage capacity, CPU, memory, and so on, of the AWS instance on which the database server runs.

Console

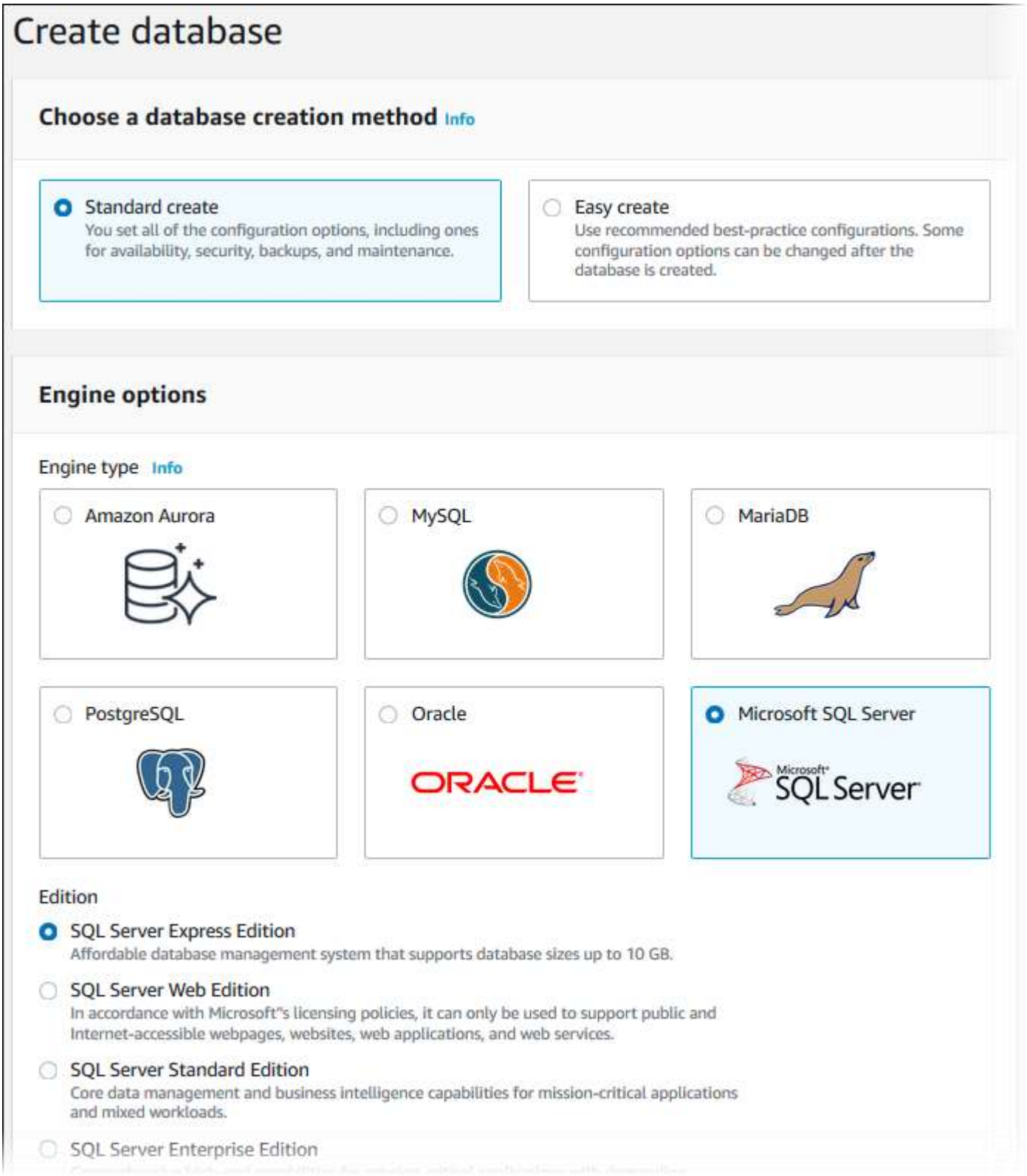
You can create a DB instance by using the AWS Management Console with **Easy create** enabled or not enabled. With **Easy create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy create** uses the default setting for other configuration options. With **Easy create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

Note

In the following procedure, **Standard create** is enabled, and **Easy create** isn't enabled. This procedure uses Microsoft SQL Server as an example.

To create a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**.
5. In **Choose a database creation method**, select **Standard Create**.
6. In **Engine options**, choose the engine type: MariaDB, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL. **Microsoft SQL Server** is shown here.



7. 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.
8. For **Version**, choose the engine version.
9. 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

  AWS recommend these features for any production environment.

10. To enter your master password, do the following:
    a. In the **Settings** section, open **Credential Settings**.
    b. If you want to specify a password, clear the **Auto generate a password** check box if it is selected.
    c. (Optional) Change the **Master username** value.
    d. Enter the same password in **Master password** and **Confirm password**.
11. For the remaining sections, specify your DB instance settings. For information about each setting, see Settings for DB instances.
12. Choose **Create database**.

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

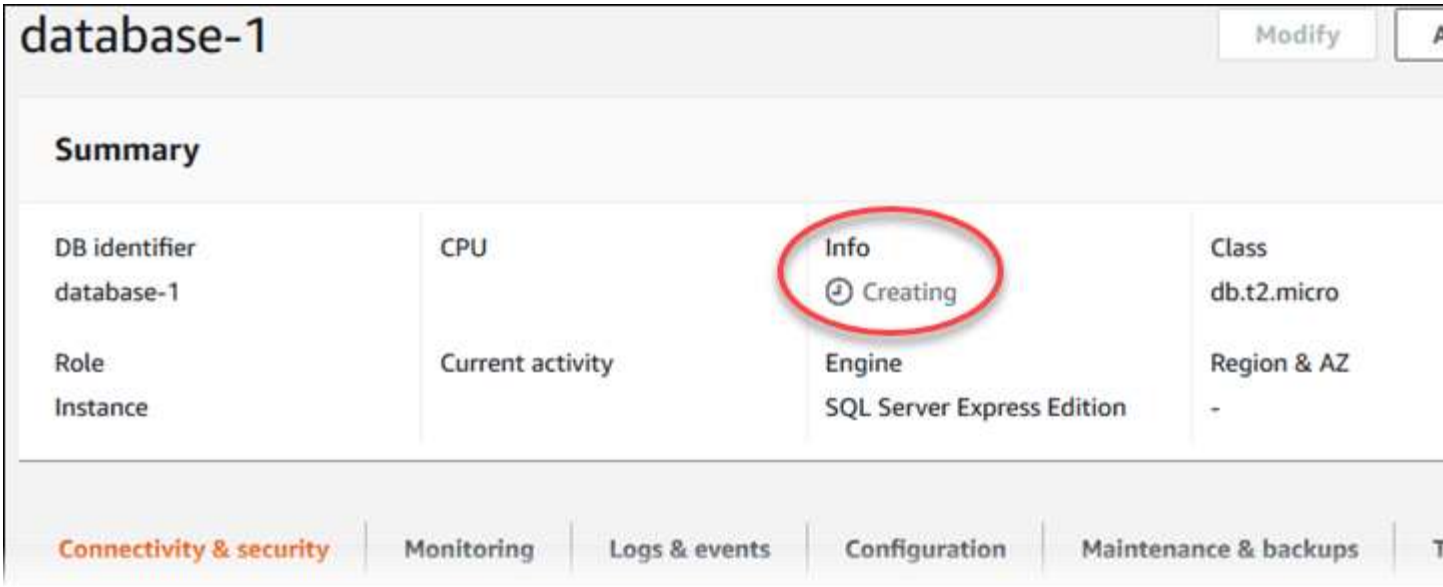
    To view the master user name and password for the DB instance, choose **View credential details**.

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

    Important

    You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, modify the DB instance to do so. For more information about modifying a DB instance, see Modifying an Amazon RDS DB instance.
13. For **Databases**, choose the name of the new DB instance.

    On the RDS console, the details for the new DB instance appear. The DB instance has a status of **Creating** until the DB instance is created and ready for use. When the state changes to **Available**, you can connect to the DB instance. Depending on the DB instance class and storage allocated, it can take several minutes for the new instance to be available.



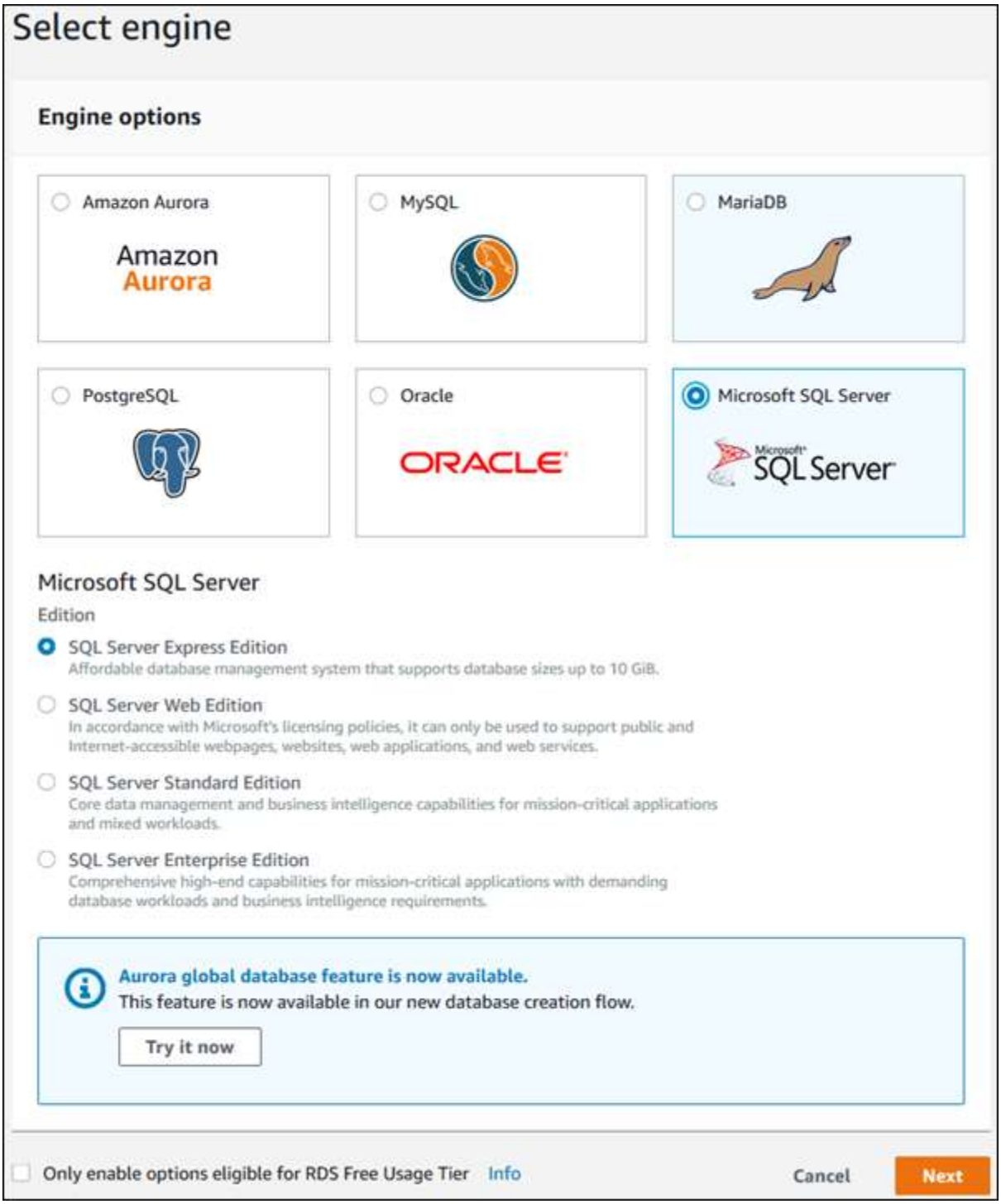
Original console example

You can create a DB instance with the original AWS Management Console. In the below Example we use Microsoft SQL Server.

To launch a SQL Server DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.

   If the navigation pane is closed, choose the menu icon at the top left to open it.
4. Choose **Create database** to open the **Select engine** page.
5. Choose the **Microsoft SQL Server** icon.



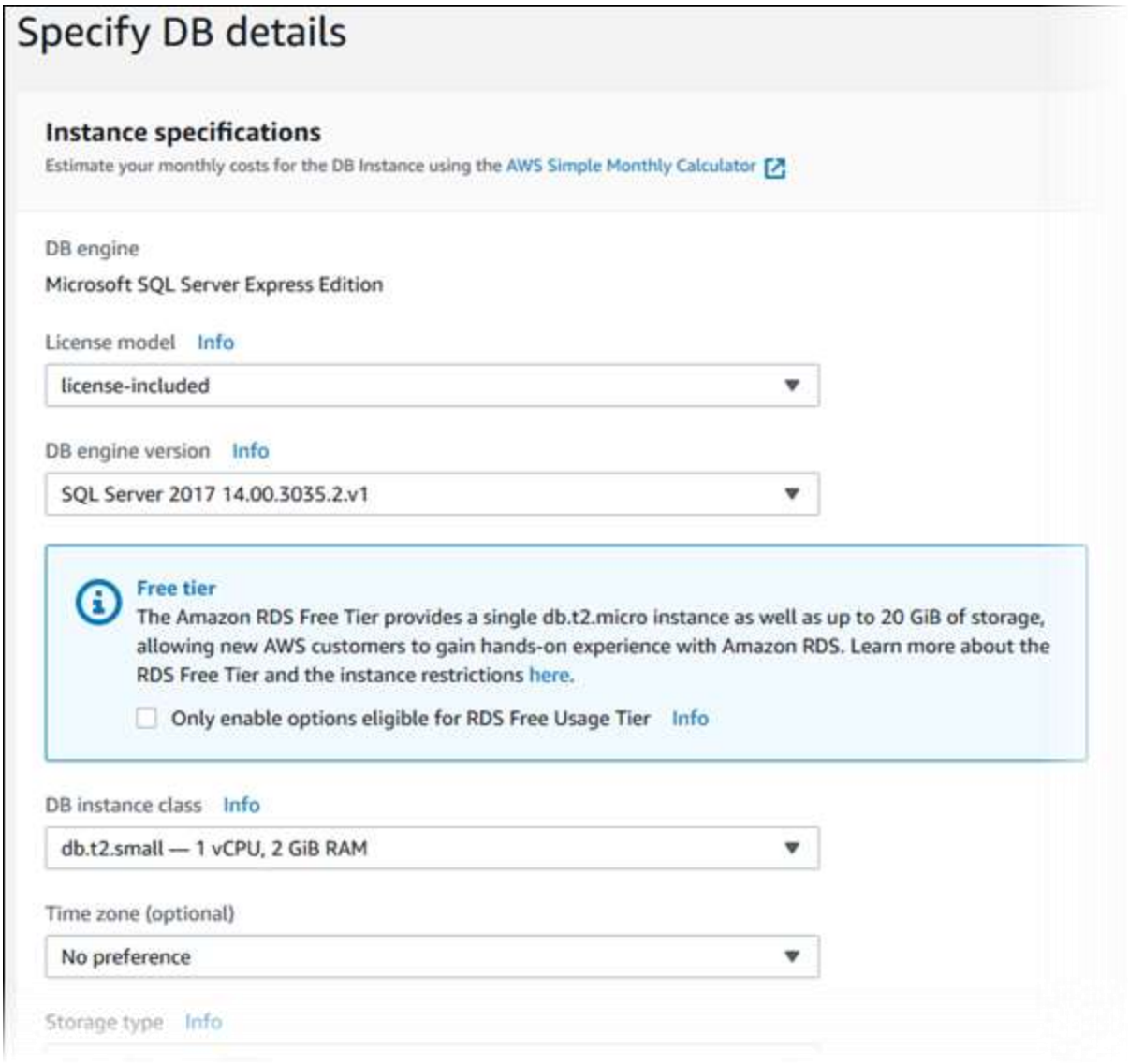
6. Choose the SQL Server DB engine edition that you want to use. The SQL Server editions that are available vary by AWS Region.
7. For some editions, the **Use Case** step asks if you are planning to use the DB instance you are creating for production. If you are, choose **Production**. If you choose **Production**, the following are all preselected in a later step:

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

AWS RDS recommend these features for any production environment.

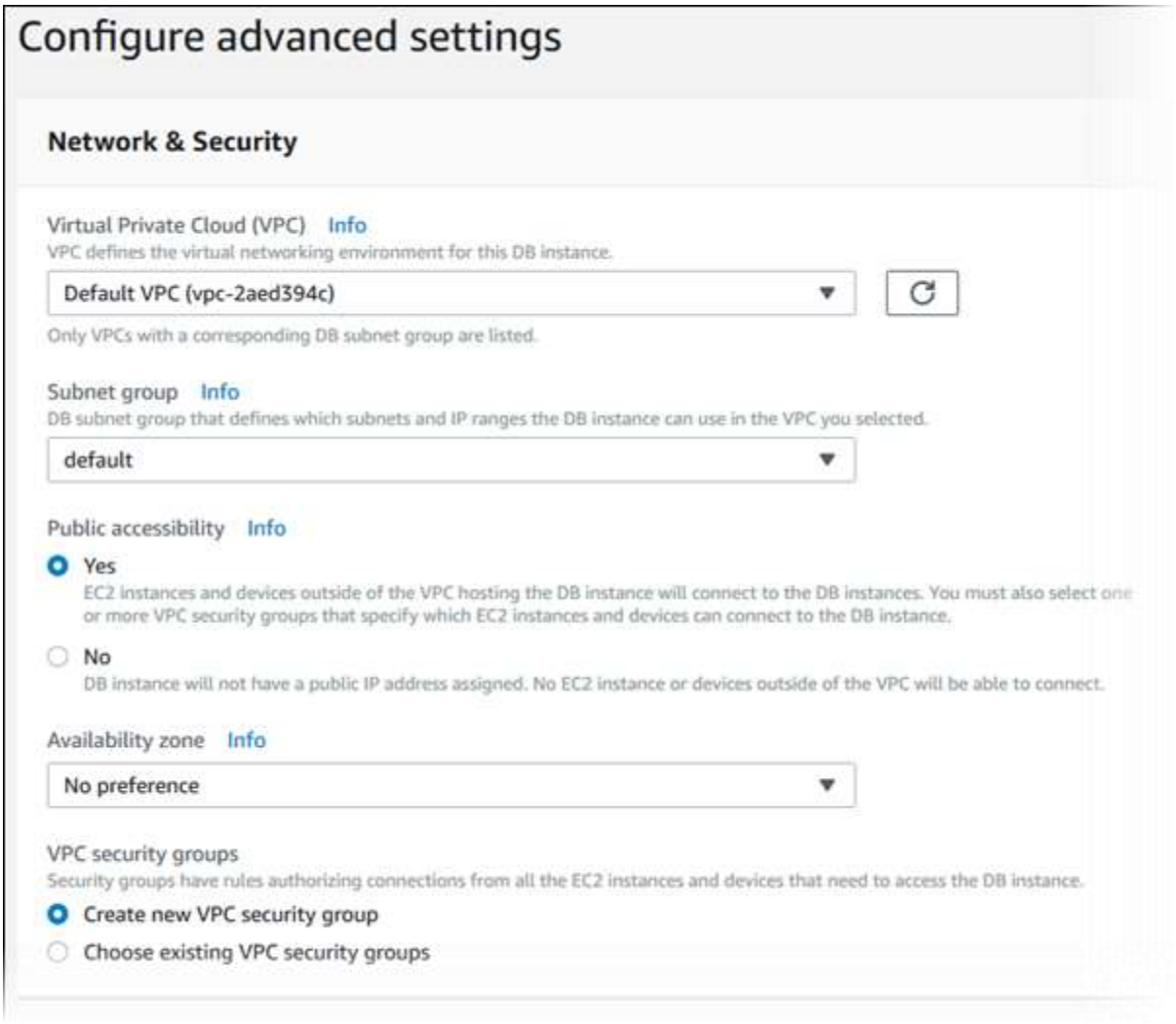
8. Choose **Next** to continue. The **Specify DB Details** page appears.

   On the **Specify DB Details** page, specify your DB instance information. For information about each setting, see Settings for DB instances.



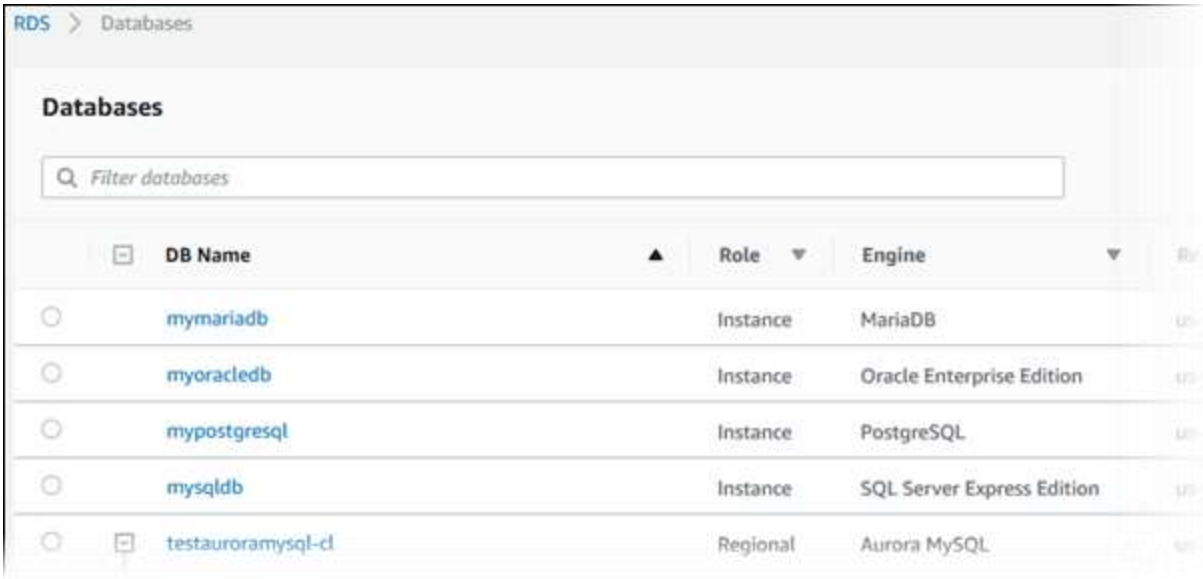
9. Choose **Next** to continue. The **Configure Advanced Settings** page appears.

   On the **Configure Advanced Settings** page, provide additional information that Amazon RDS needs to launch the DB instance. For information about each setting, see Settings for DB instances.



10. Choose **Launch DB Instance**.
11. On the final page of the wizard, choose **Close**.

    On the RDS console, the new DB instance appears in the list of DB instances. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.


RDS
Databases
Databases
Filter databases
DB Name
Role
Engine
mymariadb
Instance
MariaDB
myoracledb
Instance
Oracle Enterprise Edition
mypostgresql
Instance
PostgreSQL
mysqldb
Instance
SQL Server Express Edition
testauroramysql-cl
Regional
Aurora MySQL

Thank You !!!