

Hands-on tutorials

Create and Connect to a Microsoft SQL Server Database with Amazon RDS



Create and Connect to a Microsoft SQL Server Database with Amazon RDS: Hands-on tutorials

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

| | |
|--|----------|
| Create and Connect to a Microsoft SQL Server Database with Amazon RDS | i |
| Overview | 1 |
| What you will accomplish | 1 |
| Prerequisites | 2 |
| Implementation | 2 |
| Conclusion | 21 |

Create and Connect to a Microsoft SQL Server Database with Amazon RDS

| | |
|---|--|
| AWS experience | Beginner |
| Time to complete | 25 minutes |
| Cost to complete | \$0.005 per hour* |
| <p>*You will only incur charges if you select In-use Public IPv4 Address.</p> | |
| Requires | <ul style="list-style-type: none">AWS account <div><p>Note Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.</p></div> <ul style="list-style-type: none">Recommended browser: The latest version of Chrome or Firefox |
| Last updated | November 17, 2022 |

Overview

In this tutorial, you will learn how to create a Microsoft SQL Server database instance, connect to the database, and delete the DB instance. We will do this using [Amazon Relational Database Service \(Amazon RDS\)](#).

What you will accomplish

In this tutorial, you will:

- Create a Microsoft SQL Server database instance
- Connect to the database
- Delete the database instance

Prerequisites

Before starting this tutorial, you will need:

- **An AWS account:** If you don't already have an account, follow the [Setting Up Your AWS Environment](#) tutorial for a quick overview.

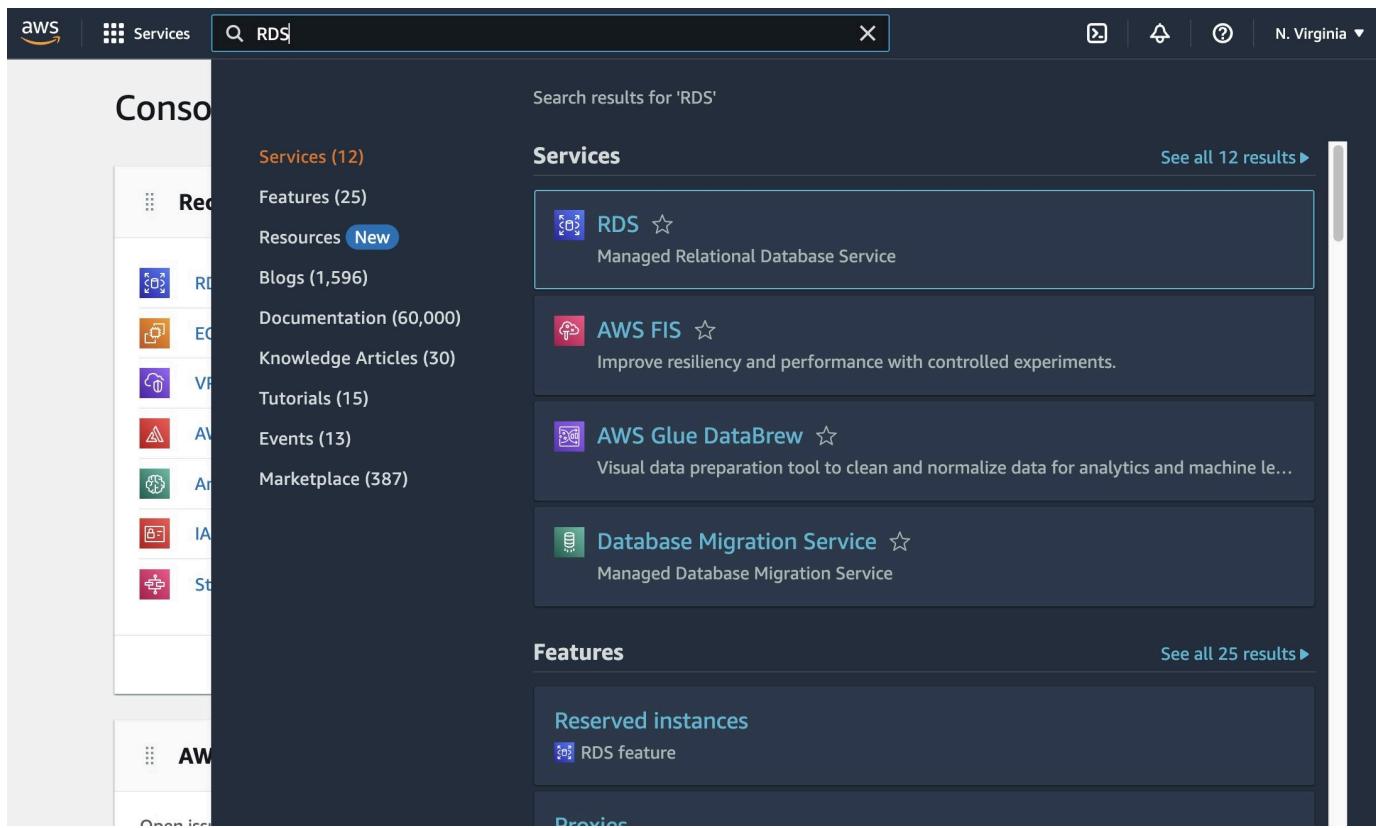
Implementation

Step 1: Create a Microsoft SQL Server DB instance

In this step, we will use Amazon RDS to create a Microsoft SQL Server DB instance with db.t3.small DB instance class, 20 GB of storage, and automated backups enabled with a retention period of one day.

1. Open the Amazon RDS console

Open the [AWS Management Console](#) in a new browser window, so you can keep this step-by-step guide open. In the console, enter **RDS** in the search bar and select **RDS** from the search results.

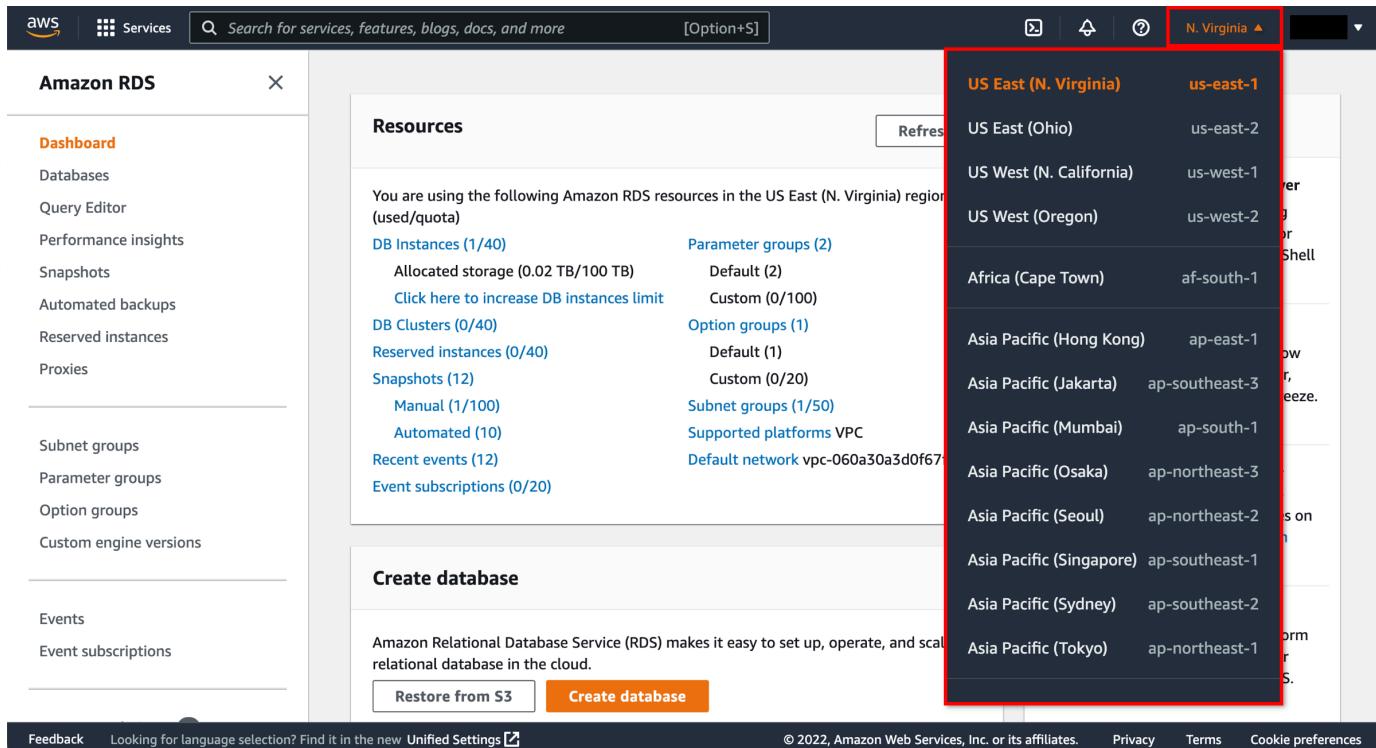


2. Choose a Region

In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.

Note

AWS Cloud resources are housed in highly available data center facilities in different areas of the world. Each Region contains multiple distinct locations called Availability Zones. You have the ability to choose which Region to host your Amazon RDS activity in.

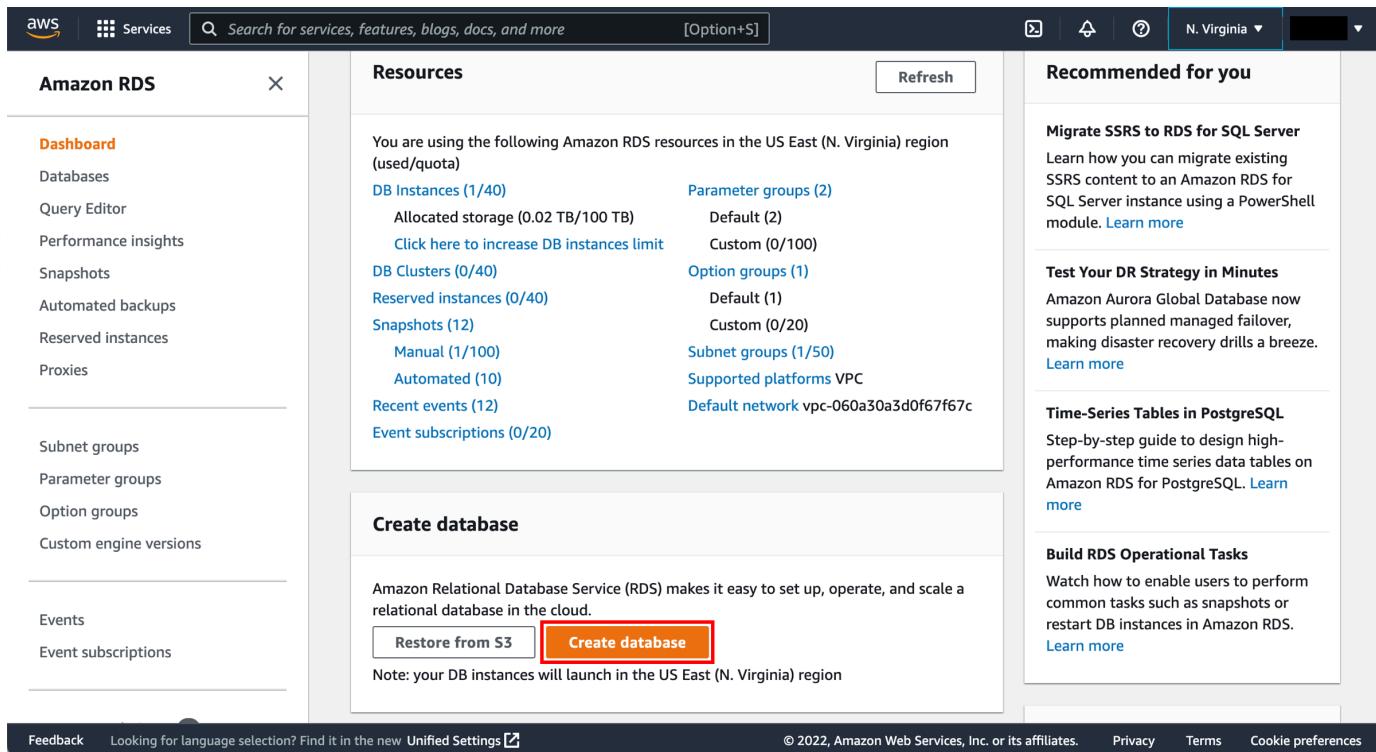


The screenshot shows the AWS RDS Dashboard. On the left, a sidebar lists various RDS management options. The main area displays 'Resources' for the 'US East (N. Virginia)' region. It shows 1/40 DB Instances, 0/40 DB Clusters, 0/40 Reserved instances, 12 Snapshots, 12 Recent events, and 0/20 Event subscriptions. To the right, a list of AWS regions is shown, with 'N. Virginia' highlighted in red. The regions listed are:

| Region | Region Code |
|--------------------------|----------------|
| US East (N. Virginia) | us-east-1 |
| US East (Ohio) | us-east-2 |
| US West (N. California) | us-west-1 |
| US West (Oregon) | us-west-2 |
| Africa (Cape Town) | af-south-1 |
| Asia Pacific (Hong Kong) | ap-east-1 |
| Asia Pacific (Jakarta) | ap-southeast-3 |
| Asia Pacific (Mumbai) | ap-south-1 |
| Asia Pacific (Osaka) | ap-northeast-3 |
| Asia Pacific (Seoul) | ap-northeast-2 |
| Asia Pacific (Singapore) | ap-southeast-1 |
| Asia Pacific (Sydney) | ap-southeast-2 |
| Asia Pacific (Tokyo) | ap-northeast-1 |

3. Create a database

In the **Create database** section, choose **Create database**.



The screenshot shows the AWS RDS Dashboard. The 'Create database' section is highlighted with a red box. The 'Recommended for you' sidebar on the right also has a red box around its title. The sidebar contains four items:

- Migrate SSRS to RDS for SQL Server**: Learn how you can migrate existing SSRS content to an Amazon RDS for SQL Server instance using a PowerShell module. [Learn more](#)
- Test Your DR Strategy in Minutes**: Amazon Aurora Global Database now supports planned managed failover, making disaster recovery drills a breeze. [Learn more](#)
- Time-Series Tables in PostgreSQL**: Step-by-step guide to design high-performance time series data tables on Amazon RDS for PostgreSQL. [Learn more](#)
- Build RDS Operational Tasks**: Watch how to enable users to perform common tasks such as snapshots or restart DB instances in Amazon RDS. [Learn more](#)

4. Choose engine options

You now have options to select your engine. For this tutorial, choose the Microsoft SQL Server icon. In the Edition section, select SQL Server Express Edition. Leave the default values for License and Version.

Engine options

Engine type [Info](#)

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle



Microsoft SQL Server



Database management type [Info](#)

Amazon RDS

RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.

Amazon RDS Custom

RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.

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.

License

license-included

Version

SQL Server 2019 15.00.4198.2.v1



5. Configure basic settings

You will now configure your DB instance. Enter the configuration settings listed below:

Settings:

- **DB instance identifier:** Enter a name for the DB instance that is unique for your account in the Region that you selected. For this tutorial, enter **myrdstest**.
- **Master username:** Enter a username that you will use to log in to your DB instance. We will use **masterUsername** in this example.
- **Master password:** Enter a password that contains from 8 to 41 printable ASCII characters (excluding /, ", and @) for your master user password.
- **Confirm password:** Re-enter your 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. First character must be a letter.

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 password [Info](#)

Instance configuration

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

DB instance class [Info](#)

Burstable classes (includes t classes)

2 vCPUs 2 GiB RAM Network: 2,085 Mbps



Include previous generation classes

Storage

Storage type [Info](#)

Implementation

Baseline performance determined by volume size



Allocated storage

6. Configure instance specifications

Now configure your instance specifications.

Instance specifications:

- **DB instance class:** Select the default, **db.t3.small — 2 vCPUs, 2 GiB RAM**. This equates to 2 GB memory and 2 vCPUs. To see a list of supported instance classes, see [Amazon RDS Pricing](#).
- **Storage type:** Select **General Purpose SSD (gp2)**. For more information about storage, see [Storage for Amazon RDS](#).
- **Allocated storage:** Select the default of 20 to allocate 20 GB of storage for your database. You can scale up to a maximum of 16 TB with Amazon RDS for SQL Server.
- **Option group:** Leave the default value. Amazon RDS uses option groups to enable and configure additional features. For more information, see [Working with Option Groups](#).
- **Enable storage autoscaling:** If your workload is cyclical or unpredictable, you would enable storage autoscaling to enable Amazon RDS to automatically scale up your storage when needed. This option does not apply to this tutorial.

The screenshot shows the 'Instance configuration' section of the AWS RDS console. The 'DB instance class' is set to 'Burstable classes (includes t classes)' with 'db.t3.small' selected, which has 2 vCPUs, 2 GiB RAM, and a Network speed of 2,085 Mbps. The 'Allocated storage' is set to 20 GiB. The 'Storage type' is 'General Purpose SSD (gp2)'. Under 'Storage autoscaling', the 'Enable storage autoscaling' checkbox is unchecked. The 'Storage' section also includes a note about provisioning less than 100 GiB of SSD storage for high throughput workloads.

7. Configure network settings

You are now on the **Connectivity** section, where you can provide information that Amazon RDS needs to launch the Microsoft SQL Server DB instance. See the following list for the example settings for your DB instance.

Connectivity

- **Network type:** Keep the default **IPv4**.
- **Virtual Private Cloud (VPC):** Select **Default VPC**. For more information about VPC, see [Amazon RDS and Amazon Virtual Private Cloud \(VPC\)](#).

Additional connectivity configurations

- **Subnet group:** Choose the **default** subnet group. For more information about subnet groups, see [Working with DB Subnet Groups](#).
- **Public access:** Choose **Yes**. This will allocate an IP address for your database instance so that you can directly connect to the database from your own device.

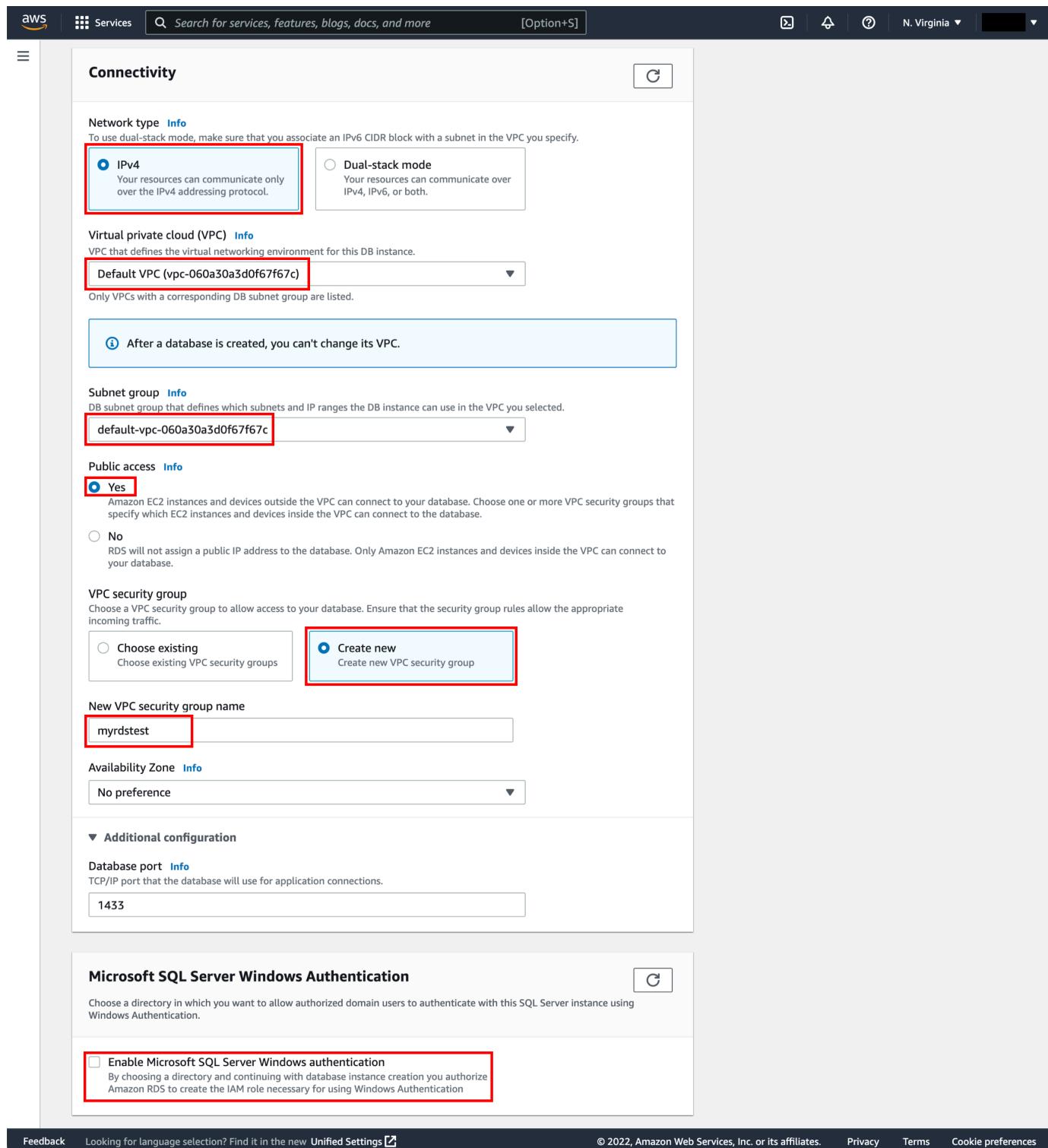
 **Note**

You will incur charges of \$0.005 per hour.

- **VPC security groups:** Select **Create new VPC security group**. This will create a security group that will allow connection from the IP address of the device that you are currently using to the database created.
- **New VPC security group name:** For this tutorial, enter **myrdstest**.
- **Availability zone:** Choose **No preference**. See [Regions and Availability Zones](#) for more details.
- **Port:** Leave the default value of 1433.

Microsoft SQL Server Windows Authentication

- **Directory:** Leave this option disabled.



The screenshot shows the AWS RDS 'Create a DB instance' wizard, Step 8: Configure additional options. The 'Microsoft SQL Server Windows Authentication' section is highlighted with a red box around the 'Enable Microsoft SQL Server Windows authentication' checkbox.

Microsoft SQL Server Windows Authentication

Choose a directory in which you want to allow authorized domain users to authenticate with this SQL Server instance using Windows Authentication.

Enable Microsoft SQL Server Windows authentication
By choosing a directory and continuing with database instance creation you authorize Amazon RDS to create the IAM role necessary for using Windows Authentication

8. Configure additional options

In the **Additional configurations** section:

Database options

- **DB parameter group:** Leave the **default value**. For more information, see [Working with DB Parameter Groups](#).
- **Option group:** Leave the **default value**. Amazon RDS uses option groups to enable and configure additional features. For more information, see [Working with Option Groups](#).

Backup

- **Backup retention period:** You can choose the number of days to retain the backup you take. For this tutorial, set this value to **1 day**.
- **Backup window:** Use the default of **No preference**.

Performance Insights

For this tutorial, do not select **Turn on performance insights**. When this option is enabled, you will receive advanced database performance-monitoring features that make it easy to diagnose and solve performance challenges on Amazon RDS databases.

Monitoring

- **Enhanced monitoring:** Use the default of **Enable Enhanced monitoring**. Enabling Enhanced monitoring will give you metrics in real time for the operating system (OS) that your DB instance runs on. For more information, see [Viewing DB Instance Metrics](#).

Maintenance

- **Auto minor version upgrade:** Select **Enable auto minor version upgrade** to receive automatic updates when they become available.
- **Maintenance window:** Select **No preference**.

Deletion protection

Do not select **Enable deletion protection** for this tutorial. When this option is enabled, you're prevented from accidentally deleting the database.

Additional configuration
Database options, backup turned on, backtrack turned off, Performance Insights turned off, Enhanced Monitoring turned on, maintenance, CloudWatch Logs, delete protection turned off.

Database options

DB parameter group [Info](#)
default.sqlserver-ex-15.0

Option group [Info](#)
default:sqlserver-ex-15-00

Time zone
No preference

Collation [Info](#)

Backup

Enable automated backups
Creates a point-in-time snapshot of your database

Backup retention period [Info](#)
The number of days for which automated backups are retained. You can choose a number from 1 to 35.
1 day

Backup window [Info](#)
The daily time range (in UTC) during which automated backups occur.
 Choose window
 No preference

Copy tags to snapshots

Performance Insights [Info](#)

Turn on Performance Insights [Info](#)

Monitoring

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

Granularity
60 seconds

Monitoring Role
default

Clicking "Create database" will authorize RDS to create the IAM role rds-monitoring-role

Log exports
Select the log types to publish to Amazon CloudWatch Logs
 Error log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.
RDS service-linked role

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.
 Choose window
 No preference

Deletion protection

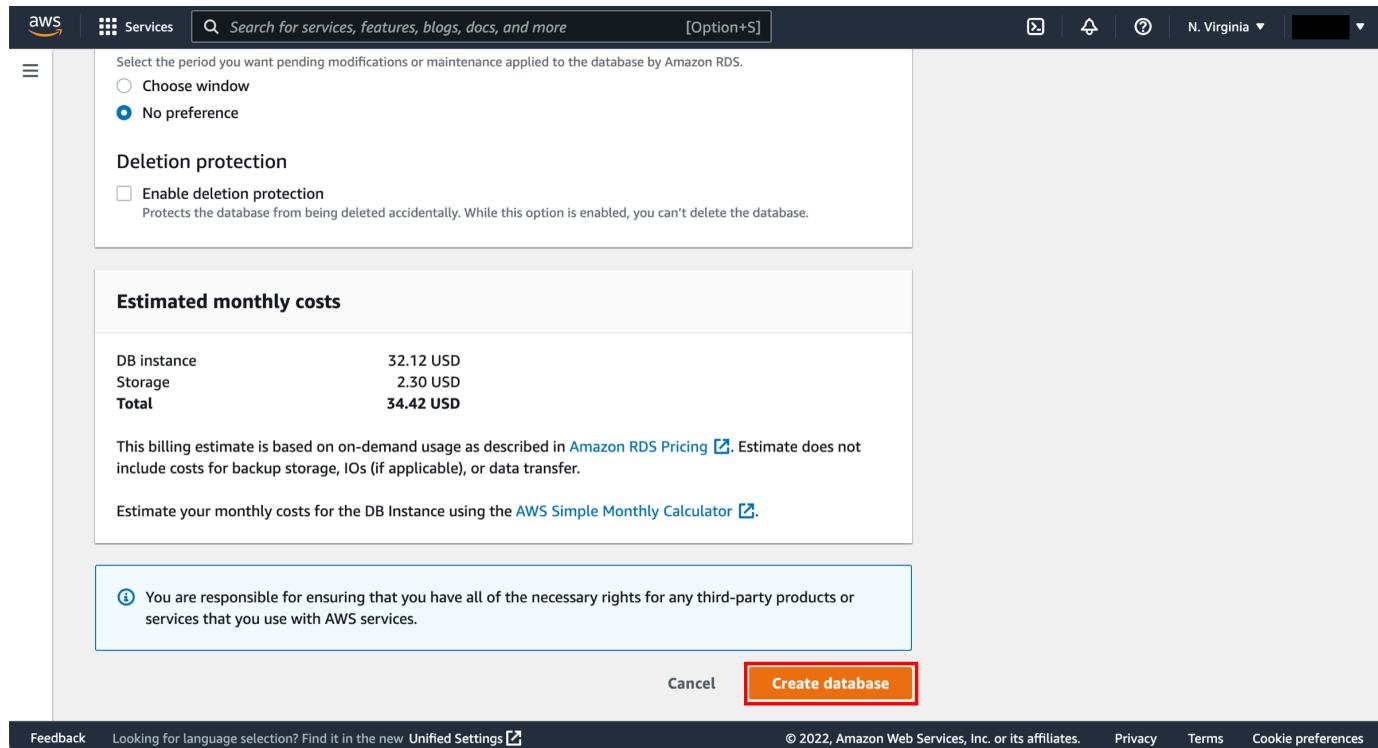
Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Implementation

9. Review costs

At the bottom of the creation wizard, AWS will show you estimated monthly costs for your Amazon RDS database. If you are still eligible for the [Amazon RDS Free Tier](#), you will see a note that the database will be free to you for up to 12 months.

Choose the **Create database** button to create your database.



10. Monitor database creation

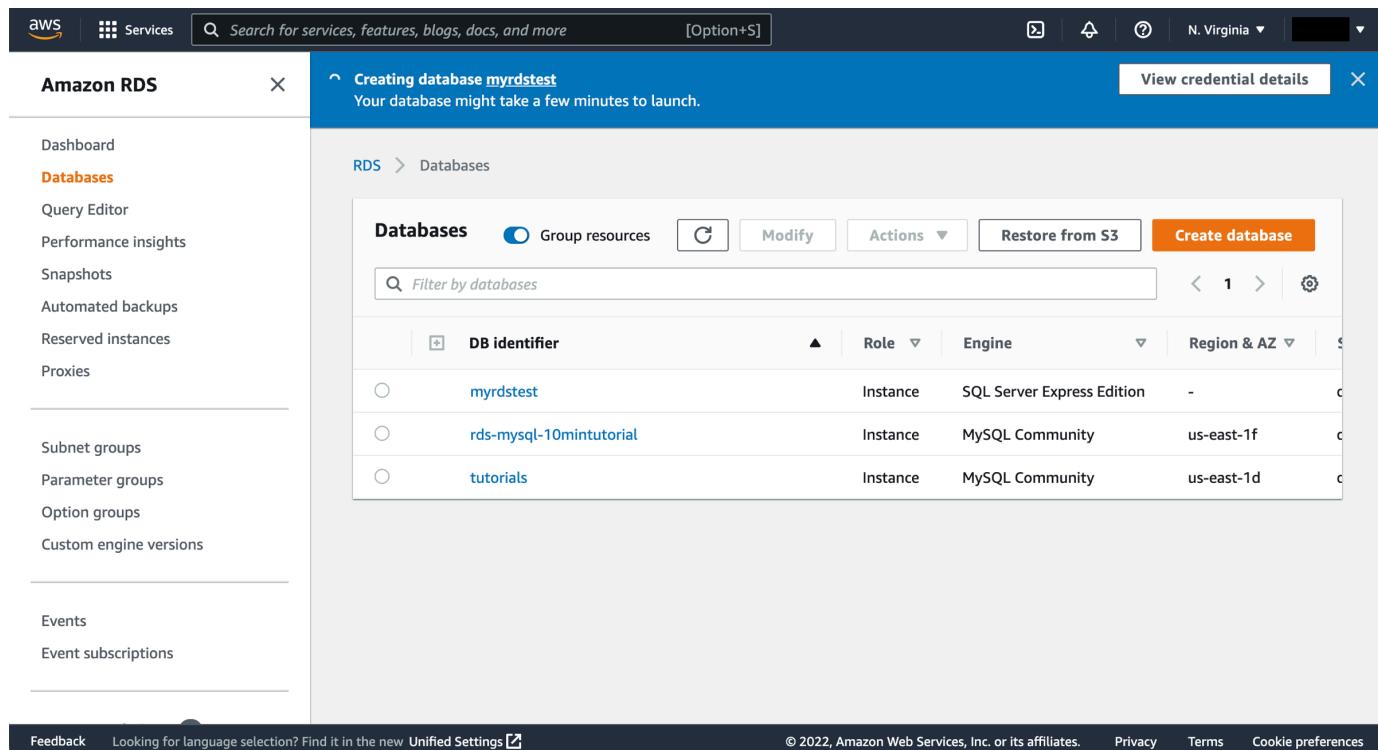
Your DB Instance is now being created. Choose **View Your DB Instances**.

Note

Depending on the DB instance class and storage allocated, it could take several minutes for the new DB instance to become available.

The new DB instance appears in the list of DB instances on the Amazon RDS console. The DB instance will have a status of creating until the DB instance is created and ready for use. When the state changes to available, you can connect to a database on the DB instance.

Feel free to move on to the next step as you wait for the DB instance to become available.



The screenshot shows the AWS RDS console. On the left, a sidebar lists various RDS features: Dashboard, Databases (which is selected and highlighted in orange), Query Editor, Performance insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, and Event subscriptions. The main content area shows a confirmation message: "Creating database myrdstest" and "Your database might take a few minutes to launch." Below this, the "Databases" list is displayed. The table has columns: DB identifier, Role, Engine, and Region & AZ. It contains three entries: "myrdstest" (Instance, SQL Server Express Edition, Region: -), "rds-mysql-10mintutorial" (Instance, MySQL Community, Region: us-east-1f), and "tutorials" (Instance, MySQL Community, Region: us-east-1d). The "Create database" button is visible at the top right of the list table.

Step 2: Download a SQL client

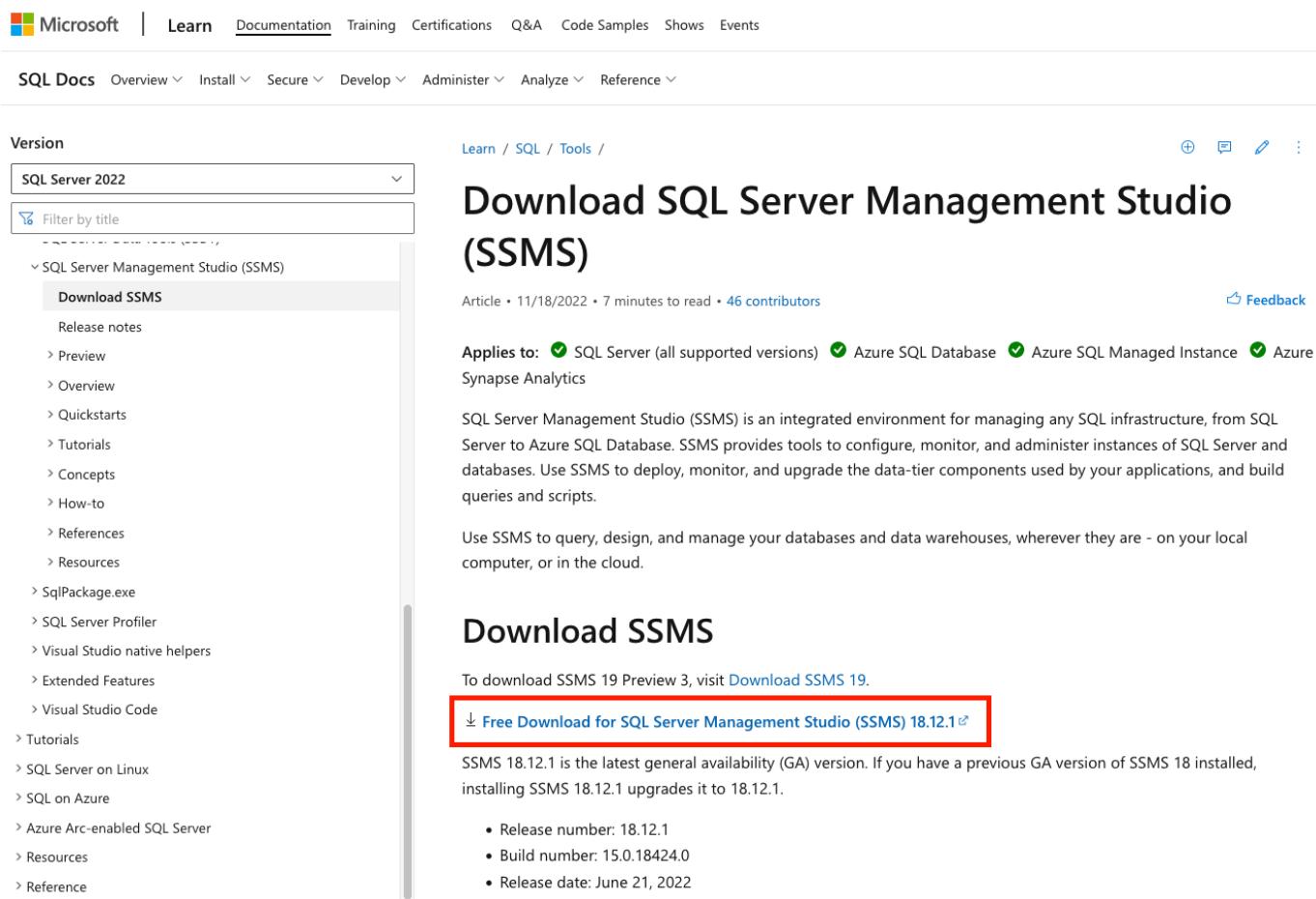
Once the database instance creation is complete and the status changes to available, you can connect to a database on the DB instance using any standard SQL client. In this step, we will download Microsoft SQL Server Management Studio, a popular client for SQL Server.

- Download SQL Server Management Studio

Go to the SQL Documentation, under which you will find SQL tools. Look for [SQL Server Management Studio \(SSMS\)](#) and download the latest version.

Note

Remember to download the SQL client to the same device from which you created the RDS DB Instance. The security group your database is placed in is configured to allow connection only from the device from which you created the DB instance.



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) download page on Microsoft Docs. The page is titled "Download SQL Server Management Studio (SSMS)". It features a sidebar on the left with a "Version" dropdown set to "SQL Server 2022" and a "Filter by title" input field. The main content area includes a "Feedback" link and a "Applies to" section with checkboxes for SQL Server, Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. The main text describes SSMS as an integrated environment for managing SQL infrastructure. Below this, a section titled "Download SSMS" lists various download links and resources. A red box highlights the "Free Download for SQL Server Management Studio (SSMS) 18.12.1" link, which is described as the latest general availability (GA) version. The page also mentions that upgrading from SSMS 18 to 18.12.1 is possible.

Step 3: Connect to the Microsoft SQL Server database

In this step, you will connect to the database you created using SQL Server Management Studio.

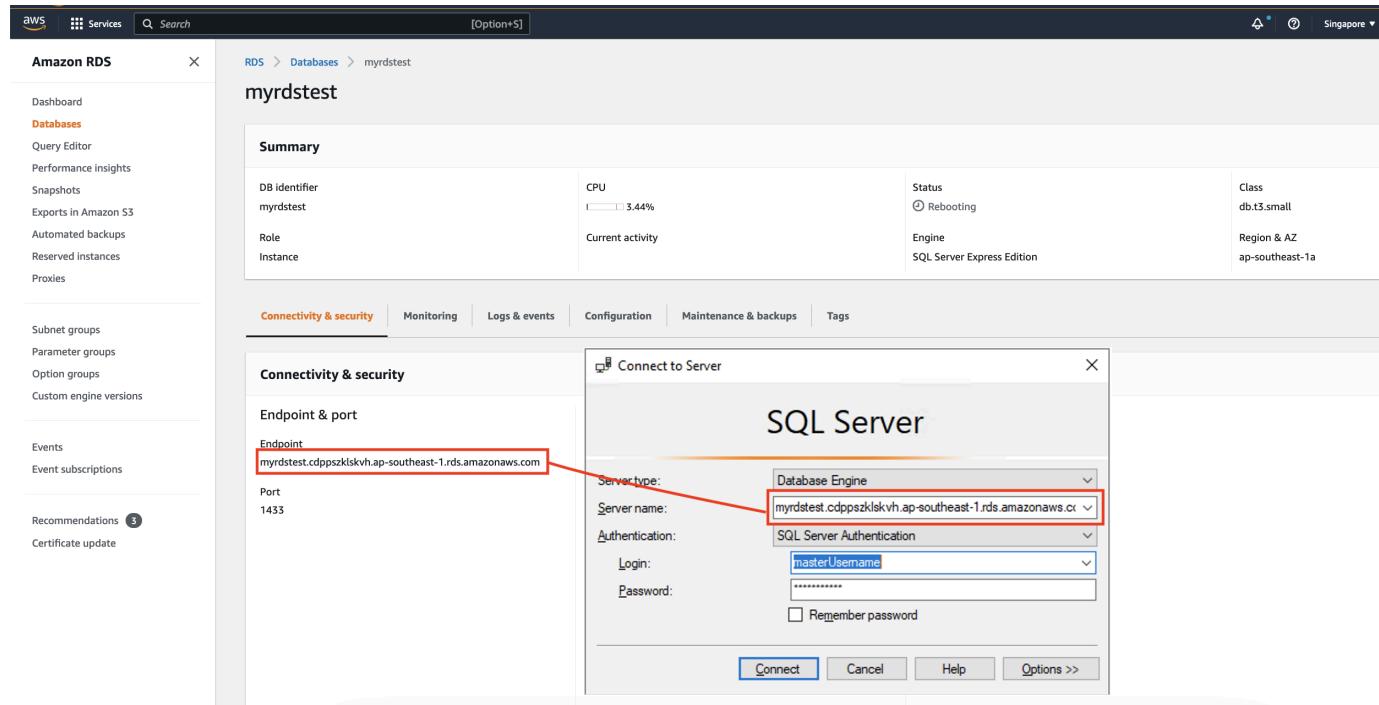
1. Configure SSMS connection settings

Once you have completed your download, install and open the program. A dialog box appears. Enter the following:

- **Server type:** Select Database Engine
- **Hostname:** Copy and paste the hostname from the Amazon RDS console as shown in the screenshot to the right. Afterwards, change the colon between the DNS and port number to a comma. For example, your server name should look like **sample-instance.cg034hpkmmjt.us-east-1.rds.amazonaws.com,1433**

- **Username:** Type in the username you created for the Amazon RDS database. Our example is **masterUsername**.
- **Password:** Enter the password you used while creating the Amazon RDS database.

Choose Connect.



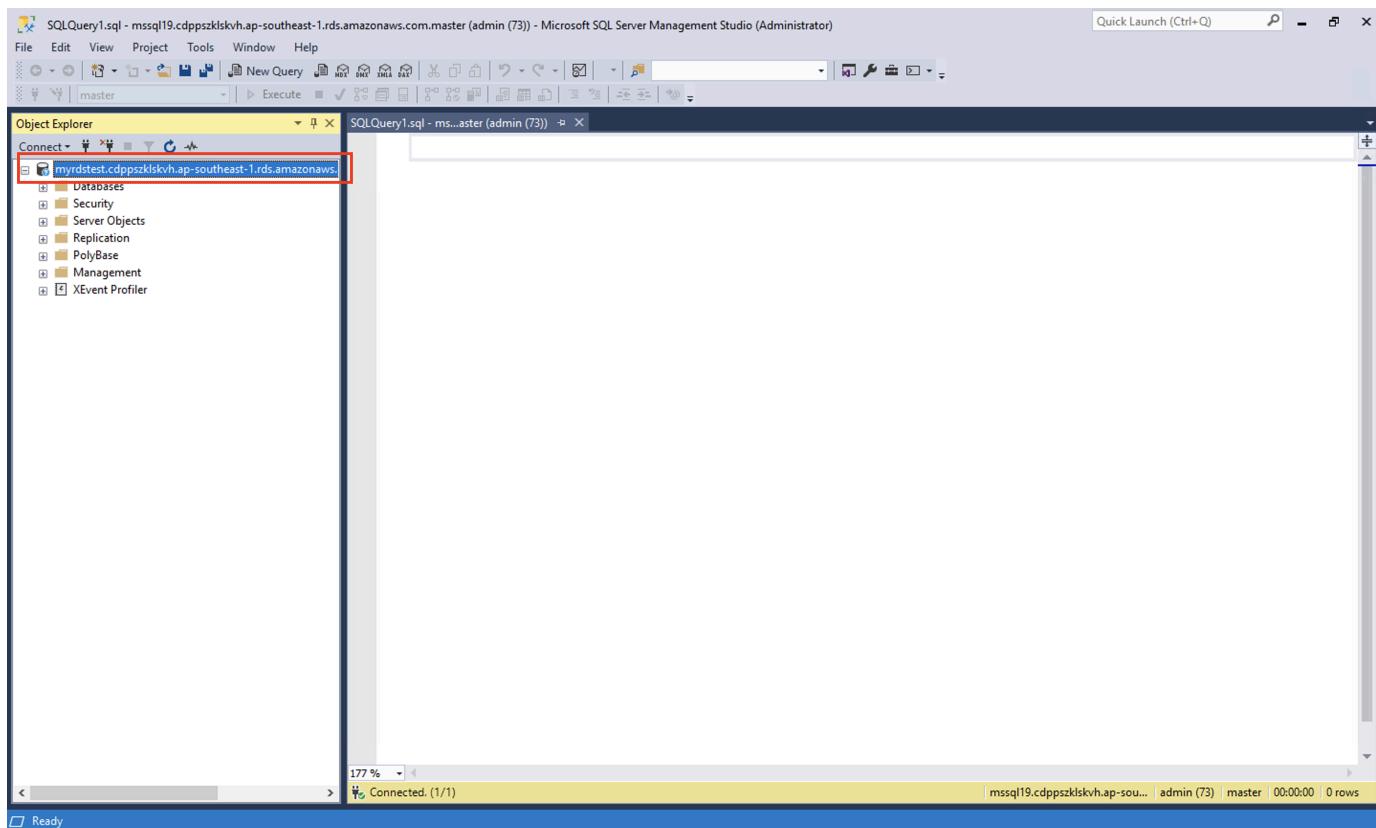
The screenshot shows the AWS RDS console with the 'Databases' section selected. A database named 'myrdstest' is selected. On the left, the 'Connectivity & security' tab is active. A 'Connect to Server' dialog box is overlaid on the page, specifically for a 'SQL Server' instance. The dialog fields are as follows:

- Server type:** Database Engine
- Server name:** myrdstest.cdppszklskvh.ap-southeast-1.rds.amazonaws.com (highlighted with a red box)
- Authentication:** SQL Server Authentication
- Login:** masterUsername (highlighted with a red box)
- Password:** (redacted)
- Remember password:**

At the bottom of the dialog are buttons for **Connect**, **Cancel**, **Help**, and **Options >**.

2. Verify database connection

You are now connected to the database. In the SQL Server Management Studio, you will see various schema objects available in the database. Now you can create tables, insert data, and run queries.

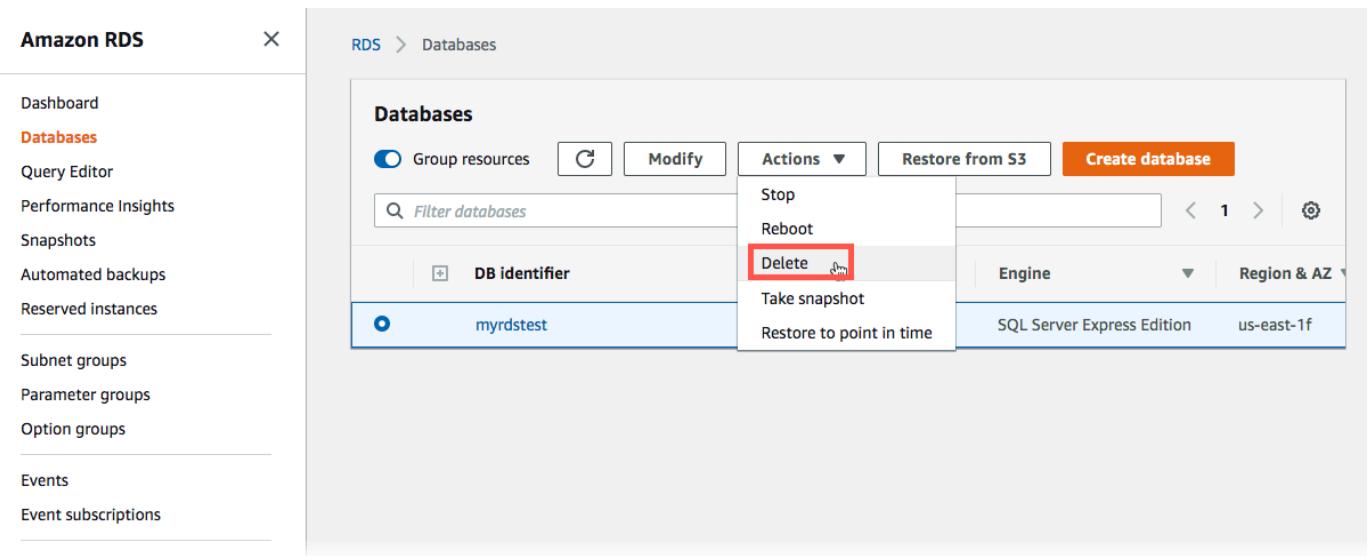


(Optional) Delete the DB instance

You can easily delete the Microsoft SQL Server DB instance from the Amazon RDS console. It is a best practice to delete instances that you are no longer using so that you don't keep getting charged for them.

1. Delete the instance

Go back to the Amazon RDS console. Select **Databases**, choose the instance that you want to delete, and then select **Delete** from the **Actions** dropdown menu.



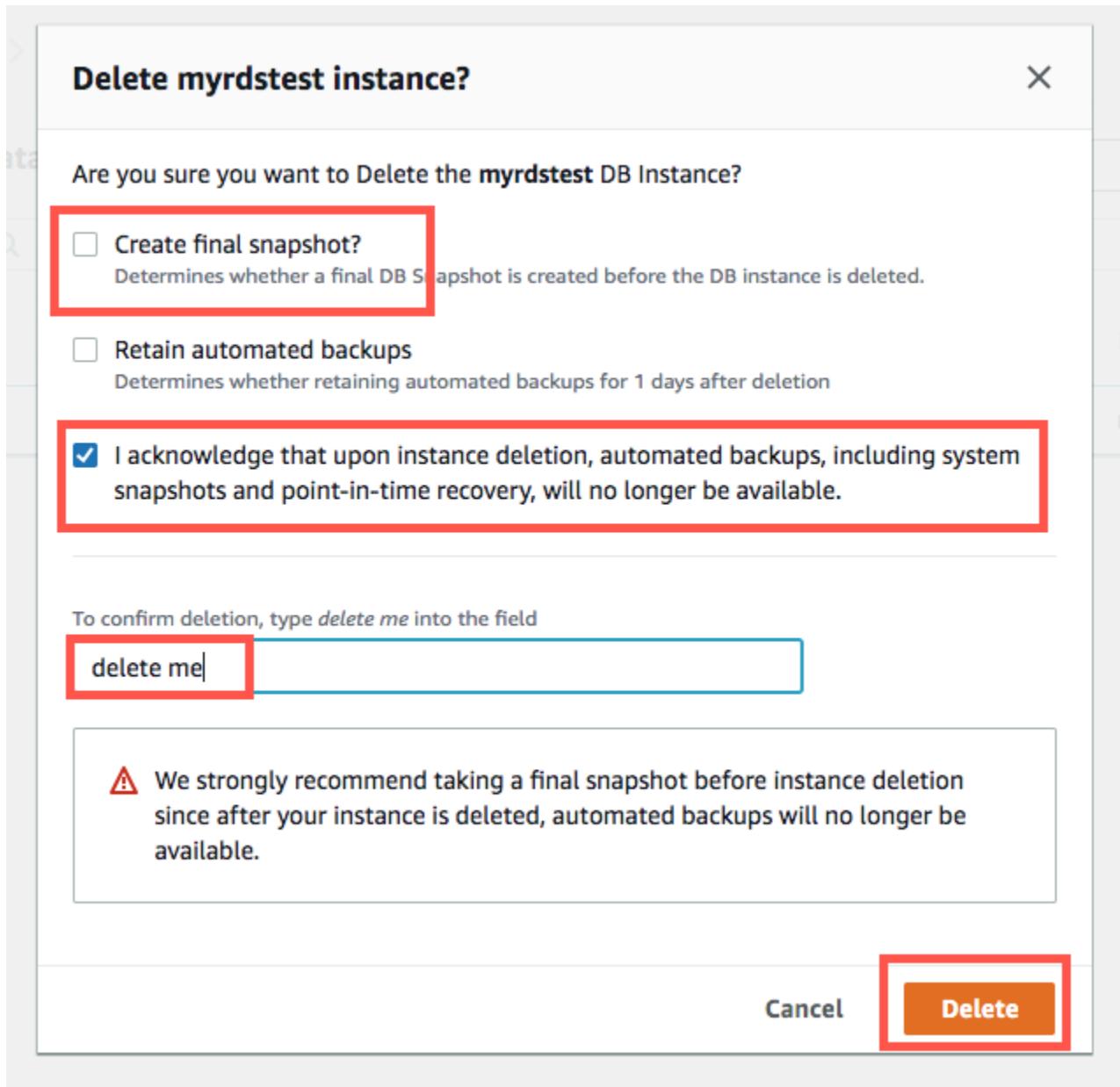
The screenshot shows the 'Databases' section of the Amazon RDS console. On the left, a sidebar lists various RDS services: Dashboard, Databases (which is selected and highlighted in orange), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, and Event subscriptions. The main content area is titled 'Databases' and shows a table with one row for 'myrdstest'. The table includes columns for 'Actions', 'DB identifier', 'Engine', and 'Region & AZ'. The 'Actions' column contains buttons for 'Stop', 'Reboot', 'Delete', 'Take snapshot', and 'Restore to point in time'. The 'Delete' button is highlighted with a red box. The 'myrdstest' row also shows 'SQL Server Express Edition' as the engine and 'us-east-1f' as the region and AZ.

2. Confirm deletion

You are asked to create a final snapshot and to confirm the deletion. For our example, do not create a final snapshot, acknowledge that you want to delete the instance, and then choose **Delete**.

Note

Deleting your DB Instance may take a few minutes



Conclusion

Congratulations! You have created, connected to, and deleted a Microsoft SQL Server database instance with [Amazon RDS](#). Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.