Amazon Relational Database Service

Getting Started Guide

API Version 2013-05-15

Amazon Relational Database Service: Getting Started Guide

Get Started with Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. You can get started with Amazon RDS by following the tasks shown in the following diagram. You'll primarily use the AWS Management Console, a point-and-click web-based interface.

This guide walks you through creating and connecting to your first Amazon RDS DB Instance.

Sign Up for Amazon RDS

To use Amazon RDS, you need an AWS account. If you don't already have one, you'll be prompted to create one when you sign up for Amazon RDS.

To sign up for Amazon RDS

1. Go to http://aws.amazon.com/rds and click **Sign Up for Amazon RDS**.
2. Follow the on-screen instructions.

   Once you've signed up for Amazon RDS, you'll be able to begin the process of creating your own DB Instance. Jump to Authorize Access: Create a DB Security Group (p. 3).

Authorize Access: Create a DB Security Group

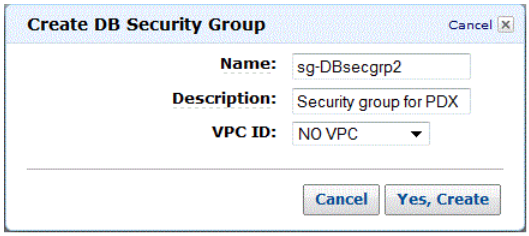
Your first task is to set up a DB Security Group that controls what IP addresses or EC2 instances have access to your DB Instance. You will use this security group when you create a DB Instance. Once created, you can reuse this security group or create new security groups for specific DB Instances.

Creating a DB Security Group

To create a DB Security group for this example, you enter CIDR (Classless Inter-Domain Routing) notation to specify either a single IP address or a range of IP addresses that you will allow to connect to your DB Instance. Since these IP addresses will be allowed access, it is important that you grant access to the correct IP addresses.

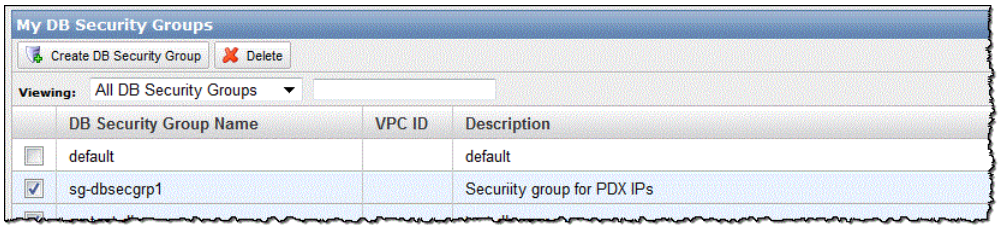
To create a new DB Security Group

1. In the left column of the AWS Management Console, click **DB Security Groups**.
2. In the **My DB Security Groups** page, click the **Create DB Security Group** button.



3. In the **Create DB Security Group** dialog box, type the name of the security group and a brief description. If you are using a Virtual Private Cloud (VPC), enter the ID of the VPC instance. Click

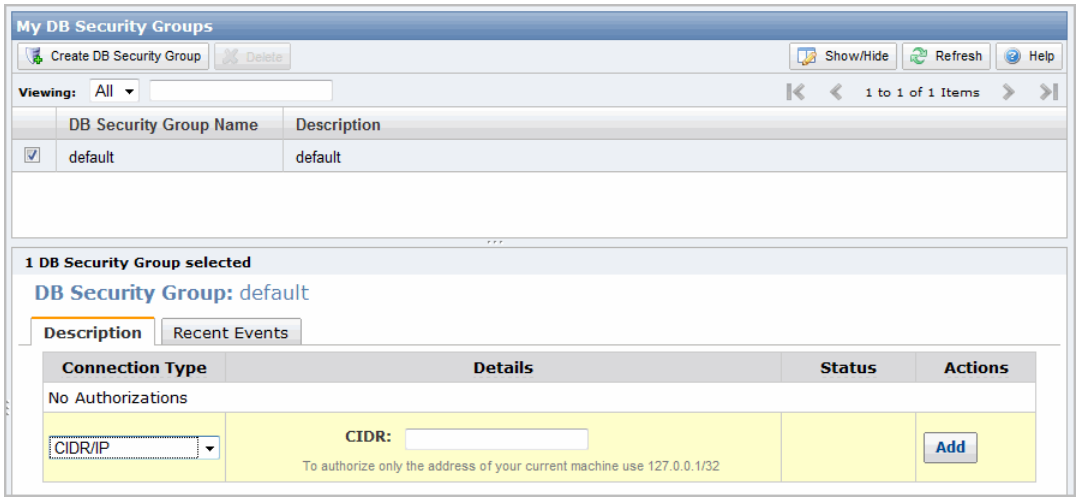
Yes, Create to close the dialog box and show the **My DB Security Groups** page of the AWS Management Console.



4. On the **My DB Security Groups** page, the DB Security Group you created is selected. On the **Description** tab at the bottom of the window, select *CIDR/IP* from the **Connection Type** drop-down list. Type your CIDR range into the **CIDR** text box, and click the **Add** button.

   Note

   The IP address you enter should be the public-facing address or range of addresses of the computers that will be accessing the DB Instance. If you are behind a firewall, the IP addresses could be a limited set of addresses that the firewall exposes. To help you determine your current IP address, the CIDR range for your current IP address appears on the page just below the **CIDR** text box. Due to how firewalls work, this value may not be the publically visible IP address you need to provide in the CIDR textbox. For information about the IP addresses you should include in the security group, consult with your network administrator.



You will use the name of the DB Security Group in the next step when you launch your DB Instance. Jump to Launch a DB Instance (p. 5).

Launch a DB Instance

Now that you have signed up for Amazon RDS and created a DB Security Group, you're ready to launch a DB Instance using the AWS Management Console.

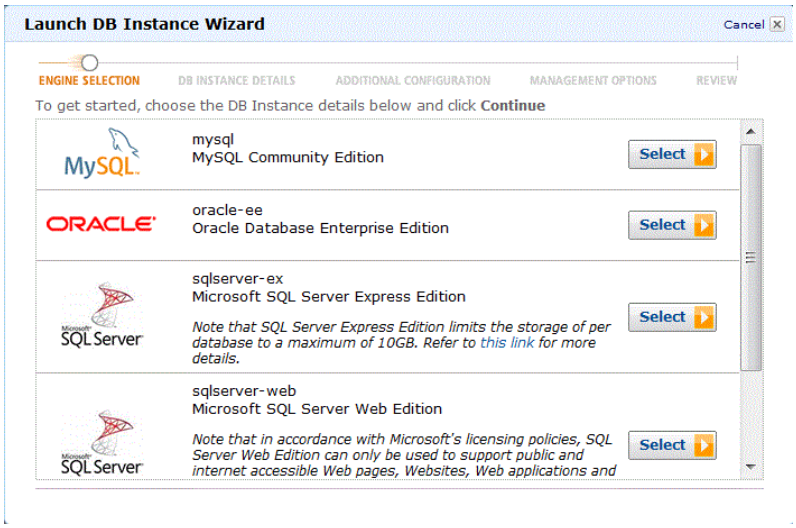
> **Important**
> The DB Instance you're about to launch will be live (and not running in a sandbox). You will incur the standard Amazon RDS usage fees for the instance until you terminate it. The total charges will be minimal if you complete the exercise described here in one sitting and terminate your DB Instance when you are finished. For more information about Amazon RDS usage rates, go to the Amazon RDS product page.

The quickest way to create a new DB Instance is by using the AWS Console. Once you select the DB Engine you want to use, the process for creating the DB Instance is specific to the DB Engine.

To launch 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 left column of the AWS Console, select the region in which you want to create the DB Instance. This should be the same region as the region in which you created your DB Security Group in the previous step.
3. Click **Launch DB Instance** to start the Launch DB Instance Wizard.

   The wizard opens on the **Engine Selection** page.



4. Click the **Select** button for the DB Engine that you want to install in the DB Instance you are launching.

   Note
   To use the suggested values in this tutorial for a SQL Server DB Engine, select Microsoft SQL Server Express Edition.

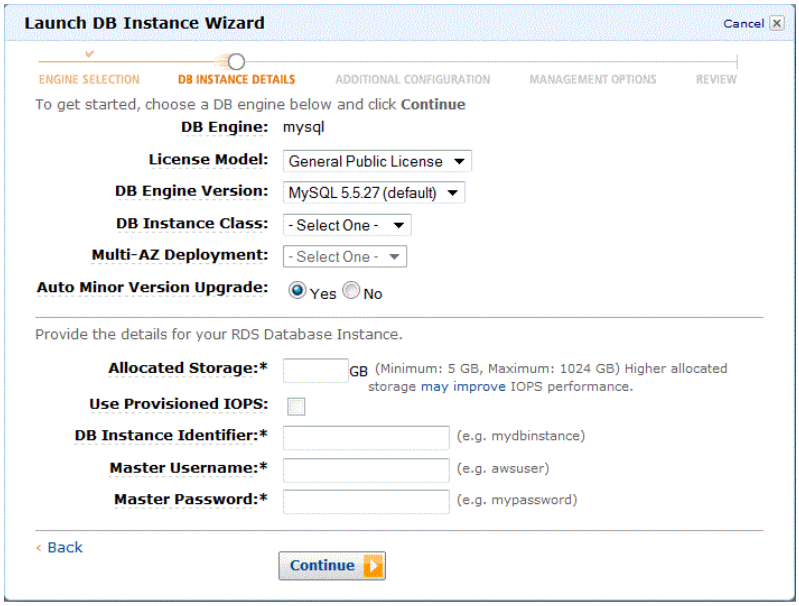
5. To continue, select the link that corresponds to the DB Engine you chose:

   - Launching a MySQL DB Instance (p. 6)
   - Launching an Oracle DB Instance (p. 11)
   - Launching a Microsoft SQL Server DB Instance (p. 16)

Launching a MySQL DB Instance

To launch a MySQL DB Instance

Once you have selected MySQL as your DB Engine, the wizard displays the **DB Instance Details** page for MySQL. The most important parameters you set here are for the DB Instance Class in the **DB Instance Class** drop-down list and the **Allocated Storage** text box. The DB Instance class defines the CPU and memory capacity of your DB Instance, which can impact processing speed and responsiveness. The allocated storage value determines how much storage is allocated for this DB Instance. Note that these two values are used to calculate the cost of your DB Instance.

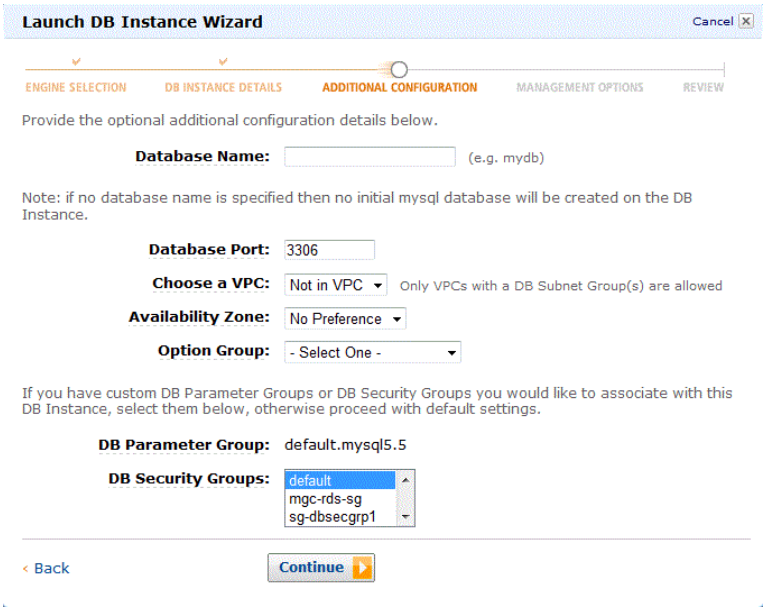


1. On the **DB Instance Details** page, specify your DB Instance information as shown in the following table, then click **Continue**.

For this parameter...	...Do this:
License Model	Select the default, **`General-Public-License`**, to use the general license agreement for MySQL.
DB Engine Version	Select **`5.5.20`** to use the default version of MySQL. Note that RDS supports additional versions of MySQL.
DB Instance Class	Select **`db.m1.small`** to select a configuration that equates to 1.7 GB memory, 1 ECU (1 virtual core with 1 ECU), 64-bit platform, and moderate I/O capacity. for more information about the capacity for all the DB Instance class options, see Amazon Relational Database Service Features.
Multi-AZ Deployment	Select **`No`** to not request that your database be made available in multiple availability zones. For more information about multiple availability zones, see the RDS documentation.
Auto Minor Version Upgrade	Select **`Yes`** to enable your DB Instance to receive minor DB Engine version upgrades automatically when they become available.
Allocated Storage	Type **`5`** to allocate 5 GB of storage for your database. In some cases, allocating a higher amount of storage for your DB Instance than the size of your database can improve I/O performance. For more information about storage allocation, see Amazon Relational Database Service Features.

For this parameter...	...Do this:
Use Provisioned IOPS	Leave the check box unselected. This option turns on Provisioned IOPS (I/O operations per second), a high-performance storage option in RDS that is optimized for I/O-intensive, transactional (OLTP) database workloads. For more information about high performance storage, see Provisioned IOPS.
DB Instance Identifier	Type a name for the DB Instance that is unique for your account in the region you selected. You may chose to add some intelligence to the name such as including the region and DB Engine you selected, for example **`west2-mysql-instance1`**.
Master User Name	Type a name using alphanumeric characters that you will use as the master user name to log on to your DB Instance with all database privileges.
Master User Password	Type a password that contains from 8 to 16 printable ASCII characters (excluding /,", and @) for your master user password.

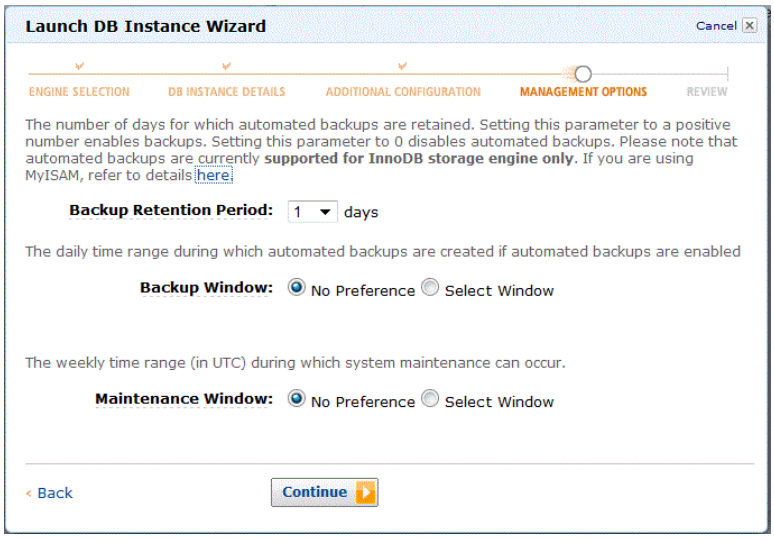
2. On the **Additional Configuration** page, you provide additional information that RDS needs to launch the DB Instance for MySQL. Specify your DB Instance information as shown in the following table, then click Continue.



For this parameter...	...Do this:
Database Name	Type a name for your database of up to 8 alpha-numeric characters. If you do not provide a name, Amazon RDS will not create a database on the DB Instance you are creating.

For this parameter...	...Do this:
Database Port	Leave the default value of **3306** unless you have a specific port you want to access the database through. MySQL installations default to port 3306.
Choose a VPC	Leave the default value of **Not in VPC** unless you are creating this DB Instance in a Virtual Private Cloud with a DB Subnet Group.
Availability Zone	Leave the default of **No Preference** unless you want to specify a particular Availability Zone. If you selected **Yes** for the Multi-AZ Deployment parameter on the previous page, you will not have any options here.
Option Group	Select the default value of **default:mysql-5-5** since this option group is used with the MySQL version you selected on the previous page, in this case version 5.5.20.
DB Parameter Group	Leave the default value of **default.mysql5.5** unless you created your own DB Parameter group.
DB Security Groups	Select the security group that you created in the Authorize Access step of this guide.

3. On the **Management Options** page, you can specify backup and maintenance options for your DB Instance. For this example, accept the default values, and then click **Continue**. Note that setting the **Backup Retention Period** to zero disables automatic backups.



In addition, Federated Storage Engine is currently not supported by Amazon RDS for MySQL.

Note

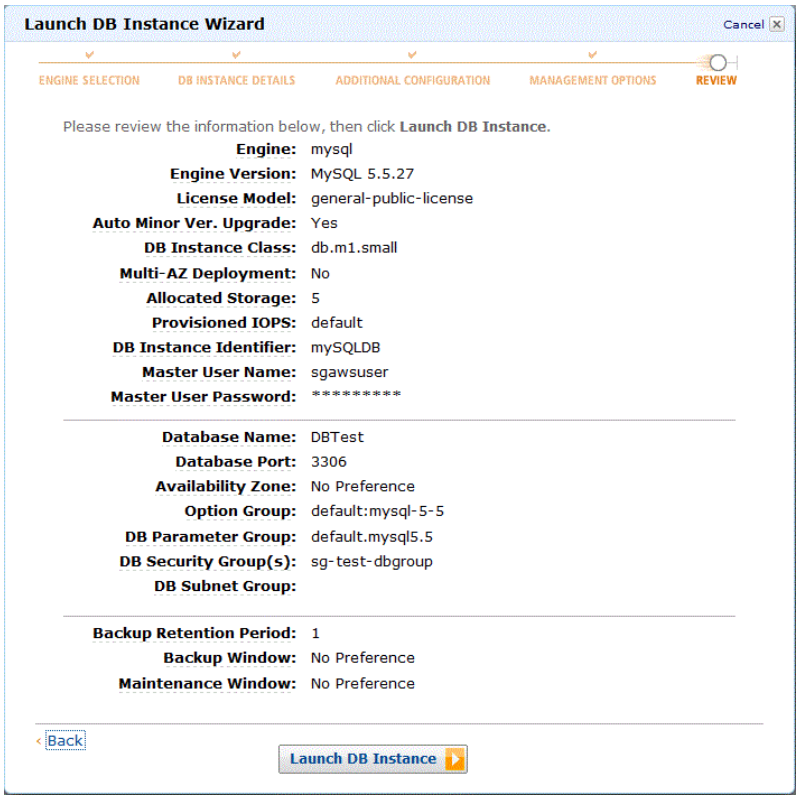
The Point-In-Time-Restore and Snapshot Restore features of Amazon RDS for MySQL require a crash recoverable storage engine and are supported for the InnoDB storage engine only. While MySQL supports multiple storage engines with varying capabilities, not all of them are optimized for crash recovery and data durability. For example, the MyISAM storage engine does not support reliable crash recovery and may result in lost or corrupt data when MySQL is restarted after a crash, preventing Point-In-Time-Restore or Snapshot restore from working as intended.

If you would like to convert existing MyISAM tables to InnoDB tables, you can use the alter table command (e.g., alter table TABLE_NAME engine=innodb;). Note that MyISAM and InnoDB have different strengths and weaknesses, so you should fully evaluate the impact of making this switch on your applications before doing so.

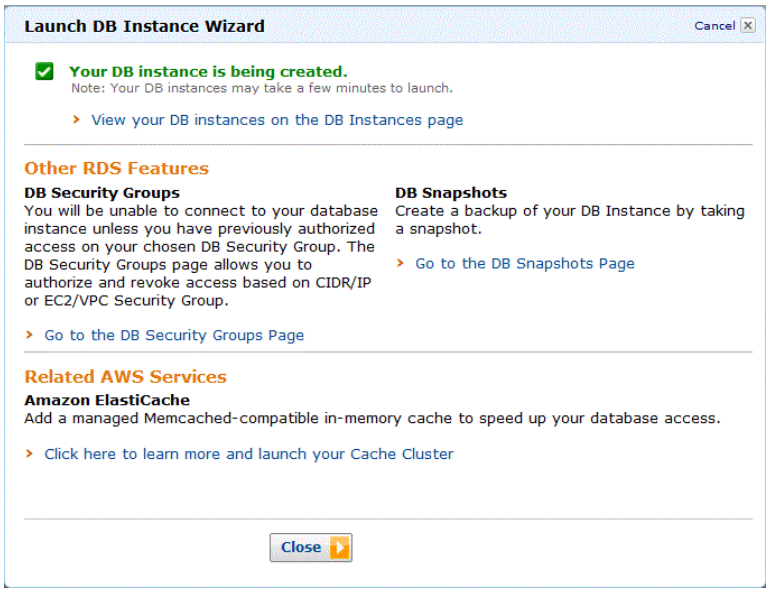
4. On the **Review** page, review the options for your DB Instance:

   If you need to correct any options, click **Back** to return to previous pages and make corrections. You can also modify a DB Instance from the AWS Console after you have launched a DB Instance.

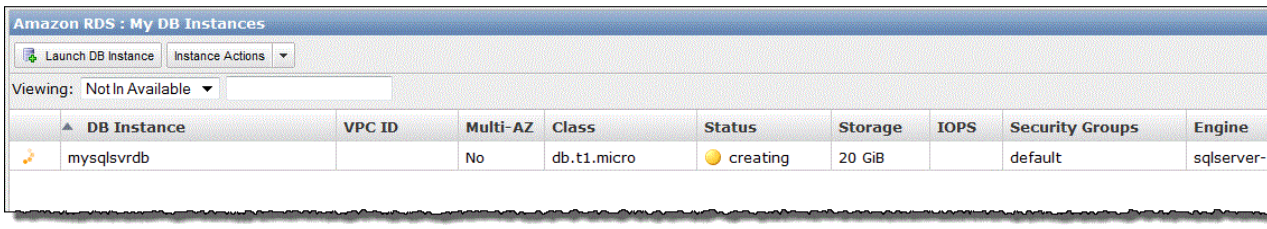
   If all your options are entered correctly, click the **Launch DB Instance** button to launch your new DB Instance.



5. On the dialog box that indicates that your DB Instance is being created, click the **Close** button.



6. On the AWS Management Console, the new DB Instance appears in the list of DB Instances. The DB Instance will have a status of **creating** until the DB Instance is created and ready for use. Depending on the DB Instance class and store allocated, it could take several minutes for the new instance to be created.

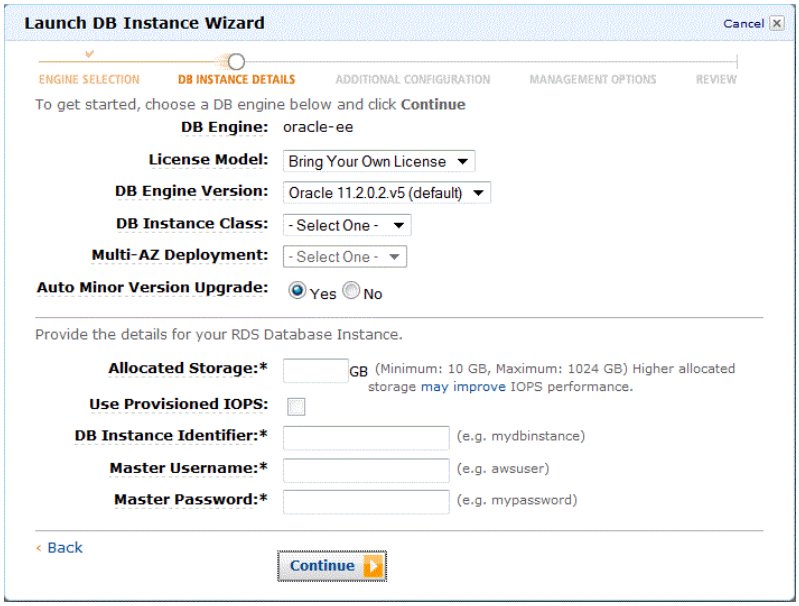


Once your DB instance changes to the **available** state, you can connect to it. For more information, see Connect to Your DB Instance (p. 21).

Launching an Oracle DB Instance

To launch an Oracle DB Instance

Once you have selected Oracle as your DB Engine, the wizard displays the **DB Instance Details** page for Oracle. The most important parameters you set here are for the DB Instance Class in the **DB Instance Class** drop-down list and the **Allocated Storage** textbox. The DB Instance class defines the CPU and memory capacity of your DB Instance, which can impact processing speed and responsiveness. The allocated storage value determines how much storage is allocated for this DB Instance. Note that these two values are used to calculate the cost of your DB Instance.

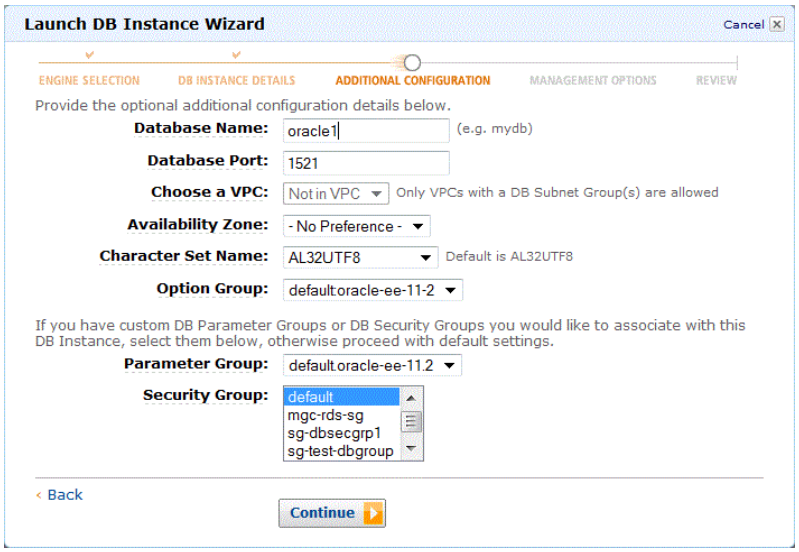


1. On the **DB Instance Details** page, specify your DB Instance information as shown in the following table, then click **Continue**.

For this parameter...	...Do this:
License Model	Keep the default, **`Bring Your Own License`**, to provide your own license for using Oracle.
DB Engine Version	Select **`11.2.0.2.v3`** to use the default version of Oracle.
DB Instance Class	Select **`db.m1.small`** to select a configuration that equates to 1.7 GB memory, 1 ECU (1 virtual core with 1 ECU), 64-bit platform, and moderate I/O capacity. For more information about the capacity for all the DB Instance class options, see Amazon Relational Database Service Features.
Multi-AZ Deployment	Select **`No`** to not request that your database be made available in multiple availability zones. For more information about multiple availability zones, see the RDS documentation.
Auto Minor Version Upgrade	Select **`Yes`** to enable your DB Instance to receive minor DB Engine version upgrades automatically when they become available.
Allocated Storage	Type **`10`** to allocate 10 GB of storage for your database. In some cases, allocating a higher amount of storage for your DB Instance than the size of your database can improve I/O performance. For more information about storage allocation, see Amazon Relational Database Service Features.

For this parameter...	...Do this:
Use Provisioned IOPS	Leave the check box unselected. This option turns on Provisioned IOPS (I/O operations per second), a high-performance storage option in RDS that is optimized for I/O-intensive, transactional (OLTP) database workloads. For more information about high performance storage, see Provisioned IOPS.
DB Instance Identifier	Type a name for the DB Instance that is unique for your account in the region you selected. You may choose to add some intelligence to the name such as including the region and DB Engine you selected, for example **`west2-oracle1`**.
Master User Name	Type a name that you will use as the master user name to log on to your DB Instance with all database privileges.
Master User Password	Type a password that contains from 8 to 30 printable ASCII characters (excluding /,", and @) for your master user password.

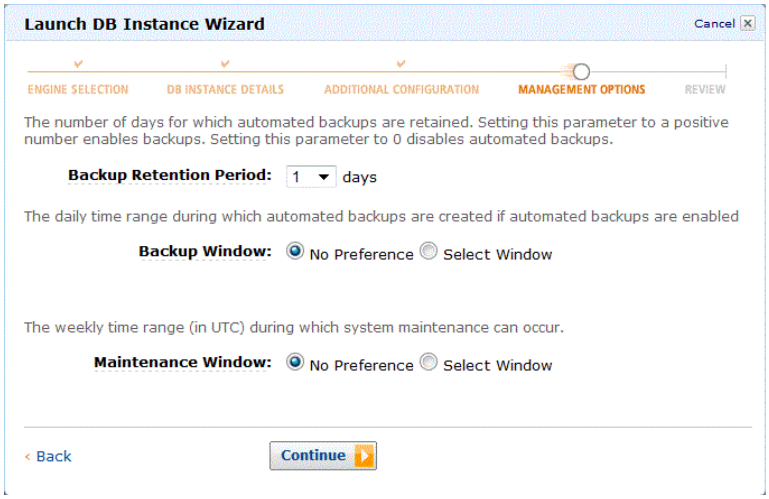
2. On the **Additional Configuration** page, you provide additional information that RDS needs to launch the DB Instance for Oracle. Specify your DB Instance information as shown in the following table, then click **Continue**.



For this parameter...	...Do this:
Database Name	Type a name for your database that begins with a letter and contains up to 8 alpha-numeric characters. If you do not provide a name, Amazon RDS will not create a database on the DB Instance you are creating.
Database Port	Leave the default value of **1521** unless you have a specific port you want to access the database through. Oracle installations default to port 1521.

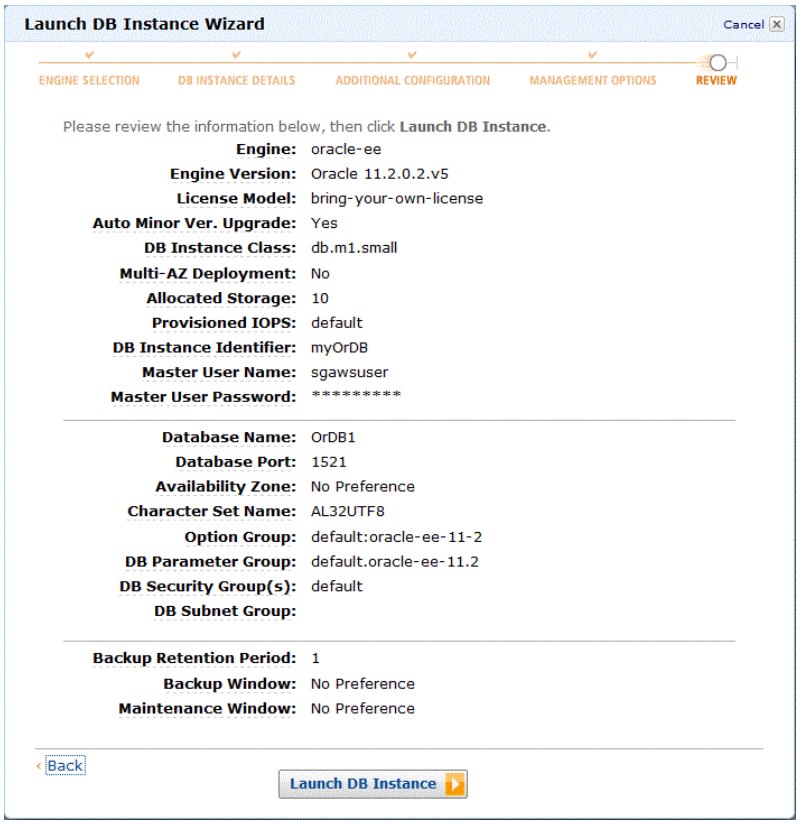
For this parameter...	...Do this:
Choose a VPC	Leave the default value of **`Not in VPC`** unless you are creating this DB Instance in a Virtual Private Cloud with a DB Subnet Group.
Availability Zone	Leave the default of **`No Preference`** unless you want to specify a particular Availability Zone. If you selected **Yes** for the Multi-AZ Deployment parameter on the previous page, you will not have any options here.
Character Set Name	Select the default value of **`AL32UTF8`** for the Unicode 5.0 UTF-8 Universal character set. Note that you cannot change the character set after the DB Instance is created.
Option Group	Select the default value of **`default:oracle-ee-11-2`** unless you have created an option group you want to use instead.
DB Parameter Group	Leave the default value of **`default.oracle-ee-11.2`** unless you have created your own DB Parameter group you want to use.
DB Security Groups	Select the security group you created in the Authorize Access step of this guide.

3. On the **Management Options** page, you can specify backup and maintenance options for your DB Instance. For this example, accept the default values, and then click **Continue**. Note that setting the **Backup Retention Period** to zero disables automatic backups.

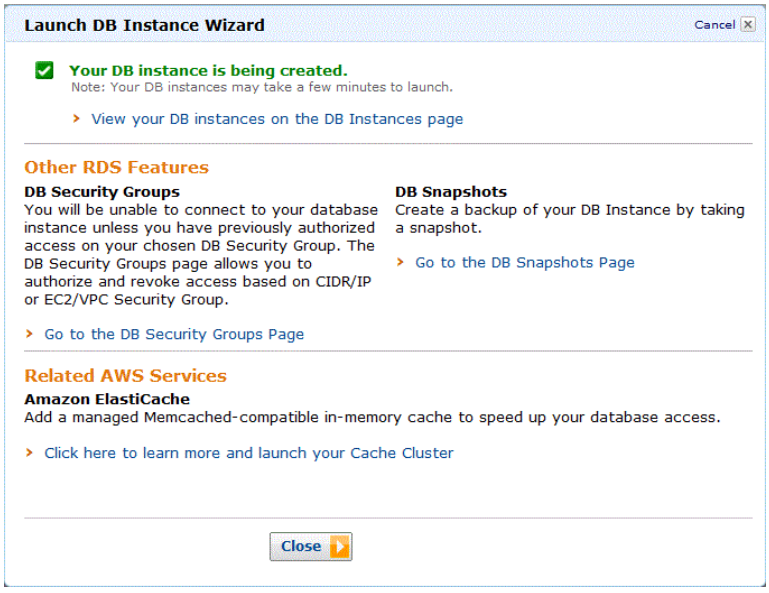


4. On the **Review** page, review the options for your DB Instance.

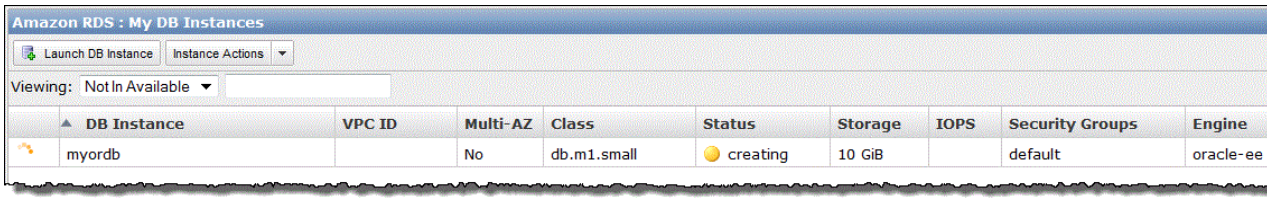
   If you need to correct any options, click **Back** to return to previous panels and make corrections. You can also modify a DB Instance from the AWS Console after you have launched a DB Instance. If all your options are entered correctly, click the **Launch DB Instance** button to launch your new DB Instance.



5. On the dialog box that indicates that your DB Instance is being created, click the **Close** button.



6. On the AWS Management Console, the new DB Instance appears in the list of DB Instances. The DB Instance will have a status of **creating** until the DB Instance is created and ready for use. Depending on the DB Instance class and store allocated, it could take several minutes for the new instance to be created.

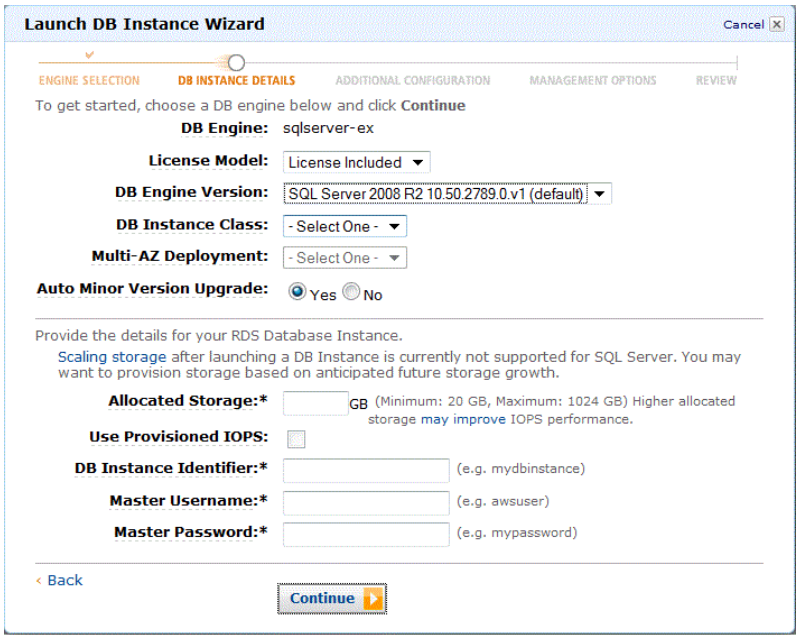


Once your DB instance changes to the **available** state, you can connect to it. Jump to Connect to Your DB Instance (p. 21).

Launching a Microsoft SQL Server DB Instance

To launch a Microsoft SQL Server DB Instance

Once you have selected SQL Server as your DB Engine, the wizard displays the **DB Instance Details** page for Microsoft SQL Server. The most important parameters you set here are for the DB Instance Class in the **DB Instance Class** drop-down list and the **Allocated Storage** textbox. The DB Instance class defines the CPU and memory capacity of your DB Instance, which can impact processing speed and responsiveness. The allocated storage value determines how much storage is allocated for this DB Instance. Note that these two values are used to calculate the cost of your DB Instance.

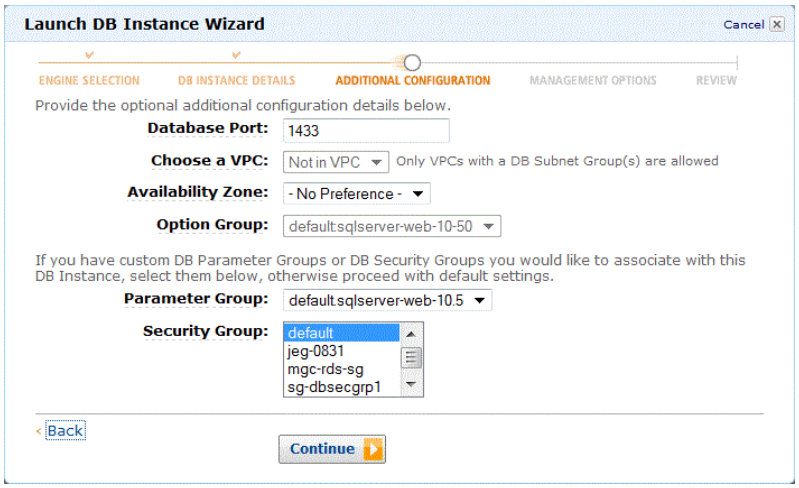


1. On the **DB Instance Details** page, specify your DB Instance information as shown in the following table, then click **Continue**.

For this parameter...	...Do this:
License Model	Keep the default, **`License Included`**, to use the general license agreement for Microsoft SQL Server.
DB Engine Version	Select **`SQL Server 2008 R2 10.50.2789.0.v1 (default)`** to use the default version of SQL Server.

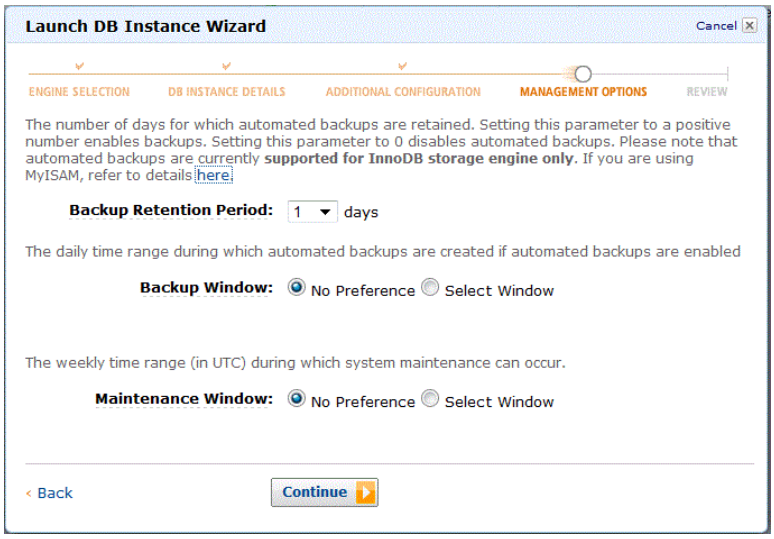
For this parameter...	...Do this:
DB Instance Class	Select **`db.t1.micro`** to select a configuration that equates to 630 MB memory, up to 2 ECUs (for short, periodic bursts), 64-bit platform, and low I/O capacity. For more information about the capacity for all the DB Instance class options, see RDS Features.
Multi-AZ Deployment	Select **`No`** to not request that your database be made available in multiple availability zones. For more information about multiple availability zones, see the RDS documentation.
Auto Minor Version Upgrade	Select **`Yes`** to enable your DB Instance to receive minor DB Engine version upgrades automatically when they become available.
Allocated Storage	Type **`20`** to allocate 20 GB of storage for your database. In some cases, allocating a higher amount of storage for your DB Instance than the size of your database can improve I/O performance. For more information about storage allocation, see Amazon Relational Database Service Features.
Use Provisioned IOPS	Leave the check box unselected. This option turns on Provisioned IOPS (I/O operations per second), a high-performance storage option in RDS that is optimized for I/O-intensive, transactional (OLTP) database workloads. For more information about high performance storage, see Provisioned IOPS.
DB Instance Identifier	Type a name for the DB Instance of 15 alphanumeric characters or less that is unique for your account in the region you selected. You may chose to add some intelligence to the name such as including the region and DB Engine you selected, such as **`west2-sqlsvr-1`**.
Master User Name	Type a name that you will use as the master username to log on to your DB Instance with all database privileges.
Master User Password	Type a password that contains from 8 to 128 printable ASCII characters (excluding /,", and @) for your master user password.

2. On the **Additional Configuration** page, you provide additional information that RDS needs to launch the DB Instance for Microsoft SQL Server. Specify your DB Instance information as shown in the following table, then click **Continue**.



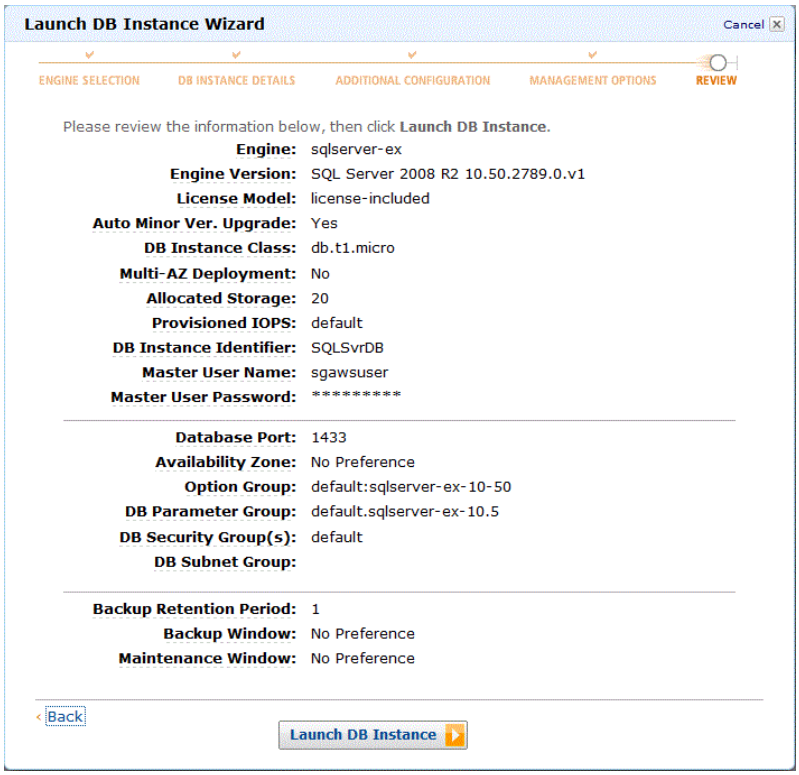
For this parameter...	...Do this:
Database Port	Leave the default value of **1433** unless you have a specific port you want to access the database through. SQL Server installations default to port 1433.
Availability Zone	Leave the default of **No Preference** unless you want to specify a particular Availability Zone. If you selected **Yes** for the Multi-AZ Deployment parameter on the previous page, you will not have any options here.
Choose a VPC	Leave the default value of **Not in VPC** unless you are creating this DB Instance in a Virtual Private Cloud with a DB Subnet Group.
DB Parameter Group	Use the default value unless you have created your own DB Parameter group.
DB Security Groups	Select the security group you created in the Authorize Access step of this guide.

3. On the **Management Options** page, you can specify backup and maintenance options for your DB Instance. For this example, accept the default values, and then click **Continue**. Note that setting the **Backup Retention Period** to zero disables automatic backups.

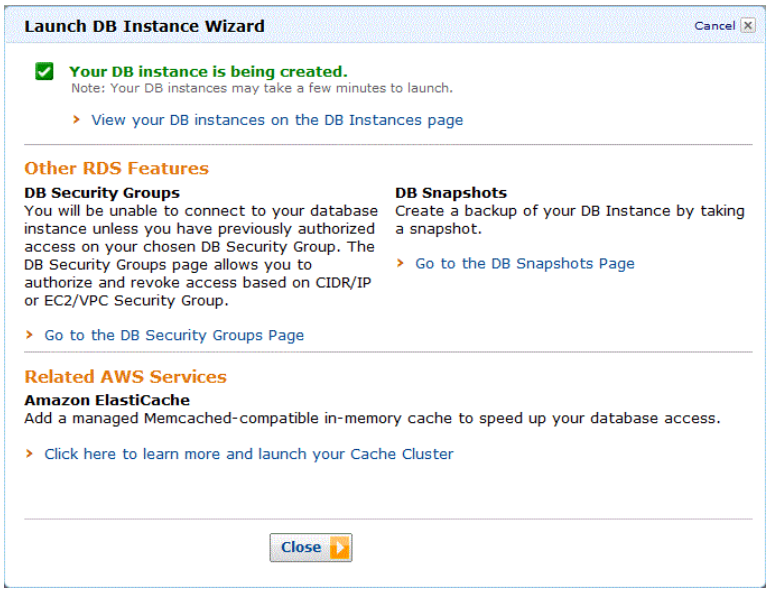


4. On the **Review** page, review the options for your DB Instance. panel appears.

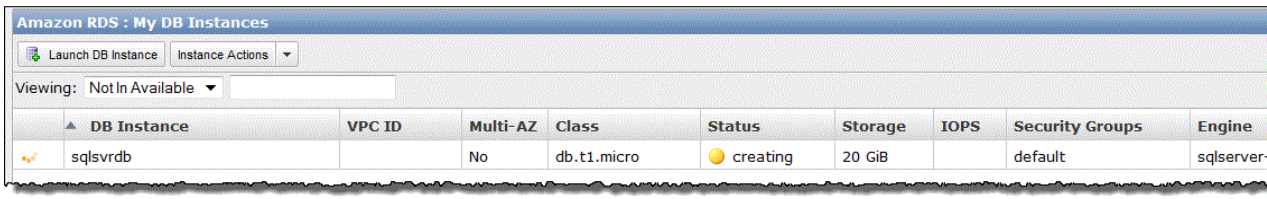
   If you need to correct any options, click the **Back** to return to previous panels and make corrections. You can also modify a DB Instance from the AWS Console after you have launched a DB Instance. If all your options are entered correctly, click the **Launch DB Instance** button to launch your new DB Instance.



5. On the dialog box that indicates that your DB Instance is being created, click the **Close** button.



6. On the AWS Management Console, the new DB Instance appears in the list of DB Instances. The DB Instance will have a status of **creating** until the DB Instance is created and ready for use. Depending on the DB Instance class and store allocated, it could take several minutes for the new instance to be created.



Once your DB instance changes to the **available** state, you can connect to it. For more information, see Connect to Your DB Instance (p. 21).

Connect to Your DB Instance

After you've authorized access to the DB Instance by creating a DB Security Group and your DB Instance is in the **`available`** state, you can connect to the DB Instance. The Amazon Relational Database Service takes care of the infrastructure management of your database instances. When a DB Instance is created, you can connect to it with any tools for the database engine that the instance supports.

> **Note**
> You must install any third-party database tools that you want to use with your Amazon RDS DB Instances; Amazon RDS does not provide or install any third-party tools or libraries.

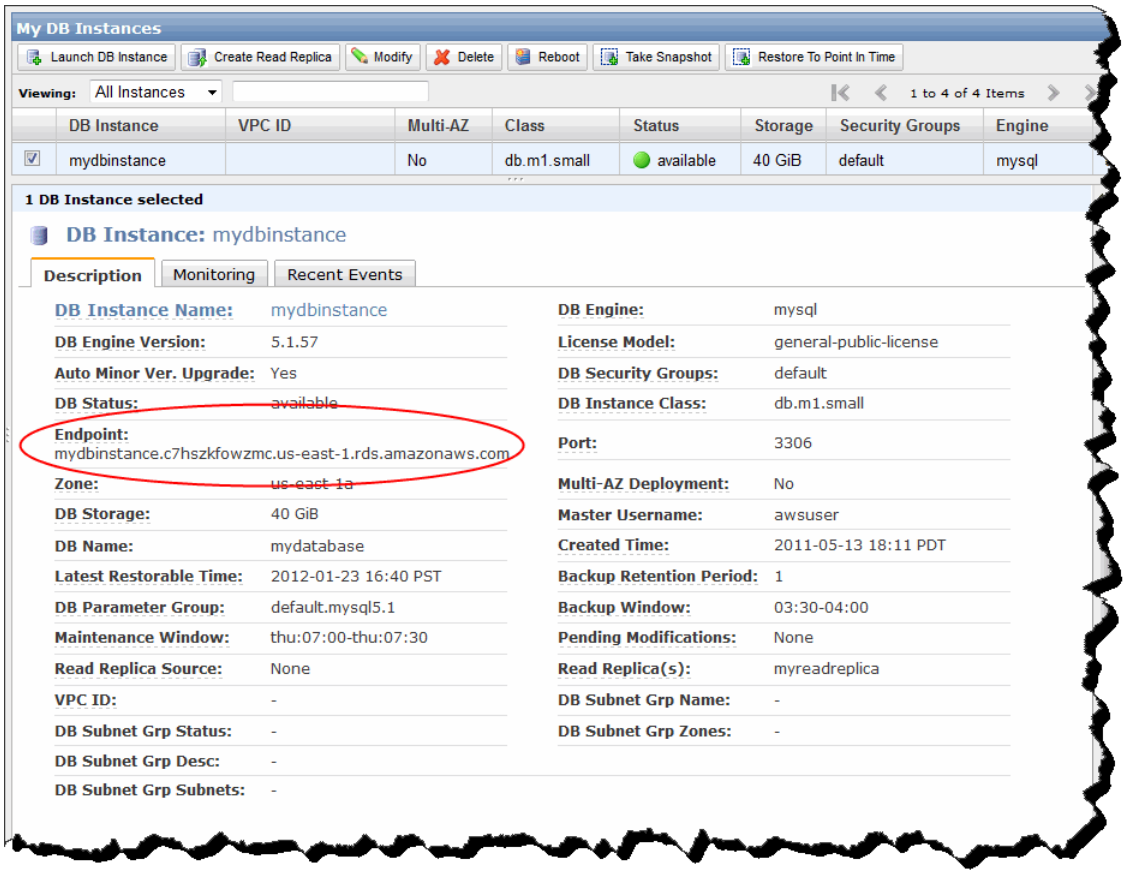
In the following example, you use the MySQL command line tools to connect to the DB Instance you just created. The procedure is similar if you are connecting to an Oracle or Microsoft SQL Server DB Instance. For more information about using other DB engines, see the following topics in the *Amazon Relational Database Service User Guide*:

- Connecting to a DB Instance Running the Oracle Database Engine.
- Connecting to a DB Instance Running the Microsoft SQL Server Database Engine.

To connect to a DB Instance using the MySQL command line client

1. On the **My DB Instances** page of the AWS management Console, select the check box next to the DB Instance named "mydbinstance."
2. On the **Description** tab of the lower panel, note the endpoint of the DB Instance to use in the next step.

   > **Note**
   > The endpoint for your DB Instance isn't available until your DB Instance is in the **available** state.



3. Open a command prompt and enter the following command; make sure to use the endpoint of the DB Instance you created.

```
PROMPT> mysql -h mydbinstance.c7hszkfowzmc.us-east-1.rds.amazonaws.com -P
3306 -u mymasteruser -p
```

You will see output similar to the following.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 350
Server version: 5.1.50-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Note

If you are having trouble connecting, you may be having a problem with your firewall configuration. Contact your network security administrator to verify that you can connect to an external port on 3306.

Once you're finished with your new DB Instance, make sure to terminate it to avoid incurring further usage charges.

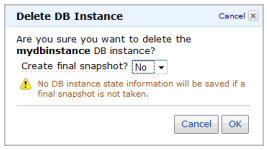
Terminate Your DB Instance

As soon as your DB Instance becomes available, you're billed for each hour or partial hour that you keep the DB Instance running (even if the DB Instance is idle). Once you've decided that you no longer need the DB Instance, you can terminate it.

To terminate your DB Instance

1. In the AWS Management Console, locate the DB Instance in your list of DB Instances on the **My DB Instances** page.
2. Select the check box next to the DB Instance, and then click **Delete** button at the top of the **My DB Instances** page.

   The **Delete DB Instance** window appears.



3. Select **No** in the **Create final snapshot?** drop-down list.

   If this weren't an exercise, you might create a final snapshot before you deleted the DB Instance so that you could restore the DB Instance later.

   Note
   Creating a final snapshot incurs additional storage fees.

4. Click the **OK** button.
   Amazon RDS begins terminating the instance. As soon as the DB Instance status changes to `deleted`, you stop incurring charges for that DB Instance.

Congratulations! You successfully launched, authorized access to, connected to, and terminated a DB Instance. For more information about Amazon RDS and how to continue, see Where Do I Go from Here? (p. 26).

Where Do I Go from Here?

Topics

Amazon RDS is a rich service offering many things we haven't covered in this guide, such as controlling automatic backup and maintenance schedules, security features, and more. This section provides links to additional resources, which will help you deepen your understanding and use of Amazon RDS.

AWS Account and Security Credentials

So far you signed up for the service, got an AWS account and security credentials, and then completed a short exercise covering the essential product functions. Now that you're finished with the exercise, we recommend that you check with an administrator or coworker in your organization to determine if he or she already has an AWS account and security credentials for you to use in future interactions with AWS.

If you're an account owner or administrator and want to know more about AWS Identity and Access Management, go to the product description at http://aws.amazon.com/iam or to the technical documentation at Using IAM.

Other Ways to Access Amazon RDS

This guide has shown you how to launch and terminate a DB Instance using the AWS Management Console. You can continue using Amazon RDS through the console, or try one of the other interfaces.

Continue Using the Console

The AWS Management Console includes many other functions besides just launching and terminating DB Instances. To learn more about how to use Amazon RDS through the console, consult the online Help to assist you (just click the **Help** button in the console) or go to the Amazon Relational Database Service User Guide.

Use the Command Line Interface

For information on using Amazon RDS's Java-based command line interface, go to the Amazon Relational Database Service User Guide. These command line tools are a fast way to execute all of the Amazon RDS functions without coding to the API or using a library.

Use an Existing Library

If you prefer to use Amazon RDS through a programmatic interface, there are libraries and resources available for the following languages:

- Java
- PHP
- Ruby
- Windows and .NET

For libraries and sample code in all languages, go to the Amazon RDS Sample Code & Libraries.

Code Directly to the Web Service API

If you want to write code directly to the Amazon RDS web service APIs, go to the Amazon Relational Database Service Developer Guide. The guide describes how to create and authenticate API requests, and how to use Amazon RDS through the APIs. For a complete description of all the API actions, go to the Amazon Relational Database Service API Reference.

Amazon RDS Resources

The table below lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Relational Database Service User Guide	The User Guide provides conceptual information about Amazon RDS and describes how to use Amazon RDS features using the AWS Management Console and command line tools.
Amazon Relational Database Service API Reference	The API Reference contains a comprehensive description of all Amazon RDS Query APIs and data types.
Amazon Relational Database Service Command Line Interface Reference	The Command Line Tools Reference contains a comprehensive description of all the command line tools and their options.
Amazon RDS Technical FAQ	The FAQ covers the top 20 questions developers have asked about this product.
Release notes	The release notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.

Resource	Description
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
Amazon RDS product information	The primary web page for information about Amazon RDS.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon RDS by completing our Getting Started Survey.

Thank you.

Document History

This document history is associated with the 2013-05-15 release of RDS. This guide was last updated on 28 August 2013.

The following table describes the important changes since the last release of the *Amazon Relational Database Service Getting Started Guide*.

Amazon Relational Database Service is often referred to within this guide as "Amazon RDS" or simply "RDS"; all copyrights and legal protections still apply.

Change	Description	Date
Oracle and SQL Server support	Expanded information on launching and connecting to Oracle and SQL Server DB Instances. Expanded information on instance parameters. Changed sequence of actions, creating DB Security group before launching DB Instance so DB Security group can be used when creating DB Instance.	In this release.