Migrate Your Application Database to Amazon RDS

Introduction

If you would like to migrate your Bitnami application's database to [Amazon Relational Database Service (RDS)](http://aws.amazon.com/rds/) , this guide will walk you through the necessary steps. The example application used in this guide is WordPress, although the steps will work generically for any Bitnami application stack.

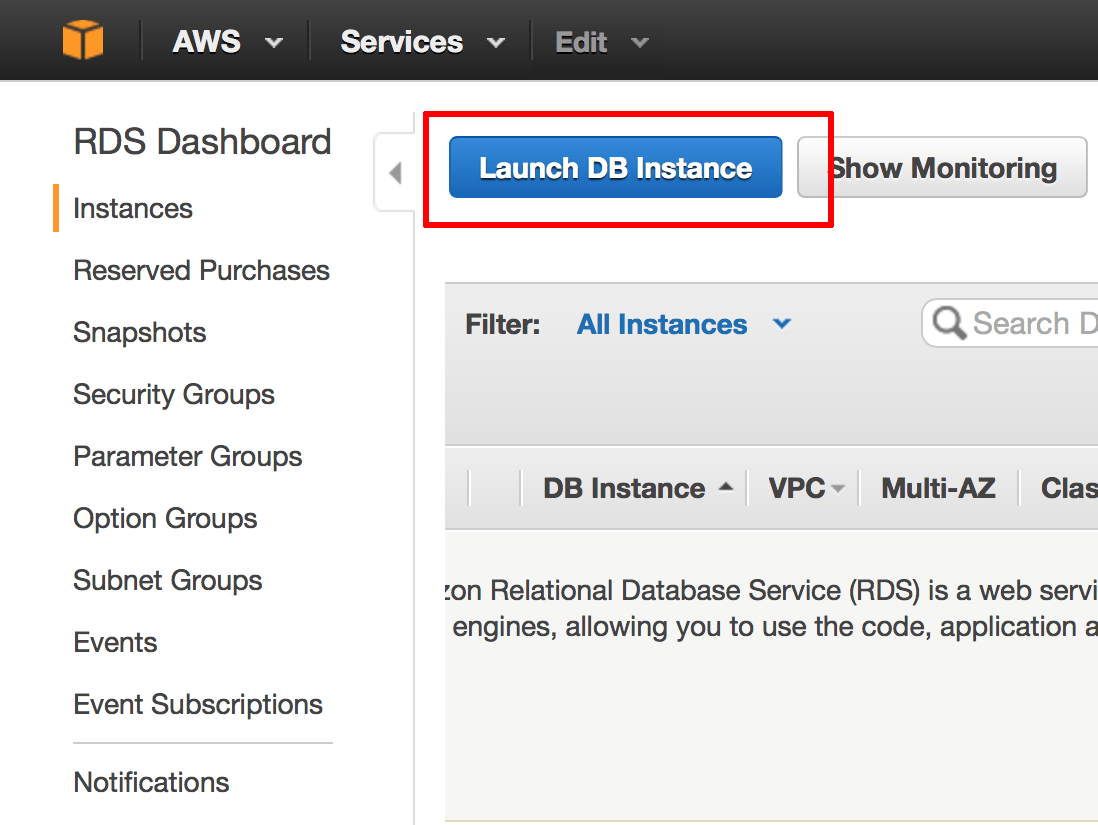
|  |
| --- |
| NOTE: Before following the steps in this guide, ensure that you have an Amazon Web Services account, and that the server running your Bitnami application stack has a public IP address. |

Step 1: Create a new database on Amazon RDS

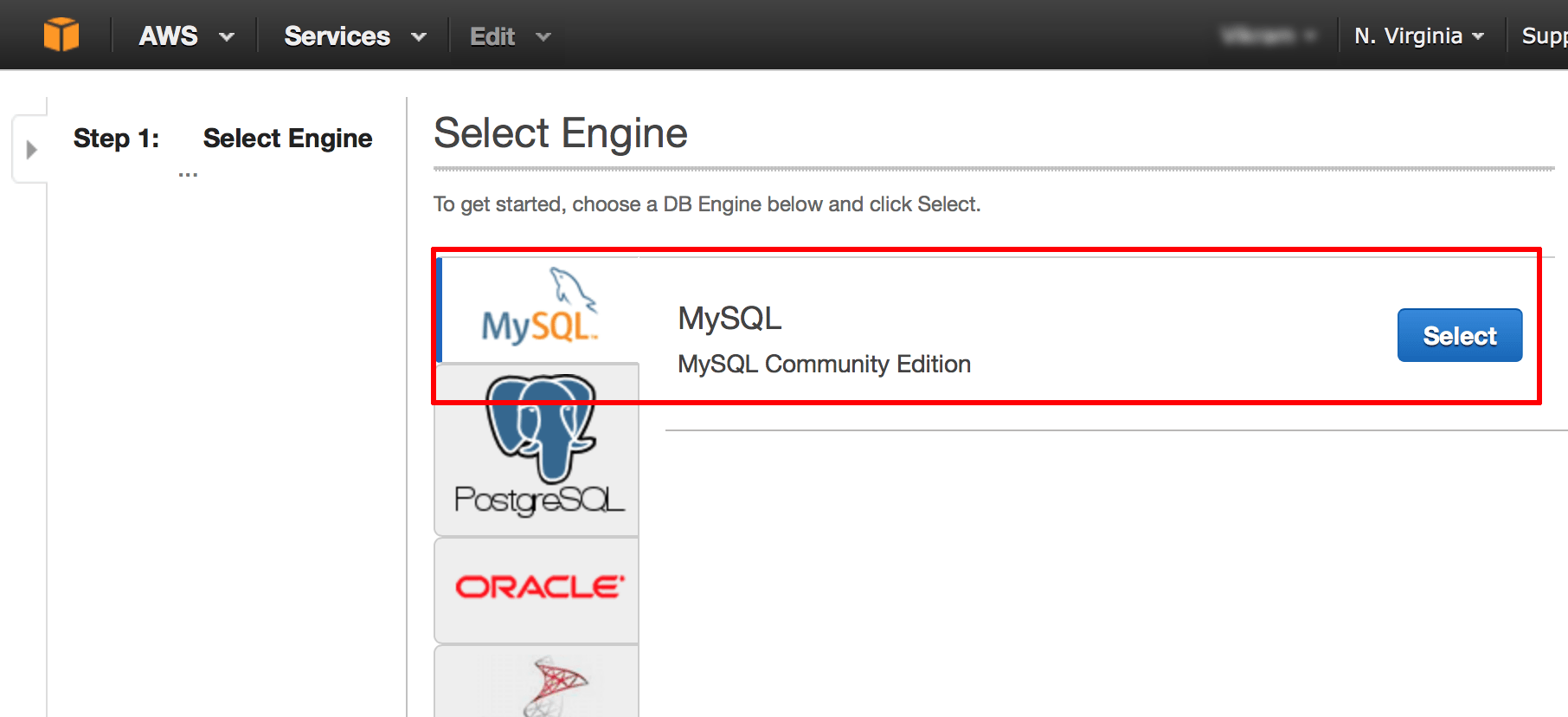
Begin by creating a new database on Amazon RDS, as described below:

Log in to the [Amazon RDS dashboard](http://aws.amazon.com/rds/) .

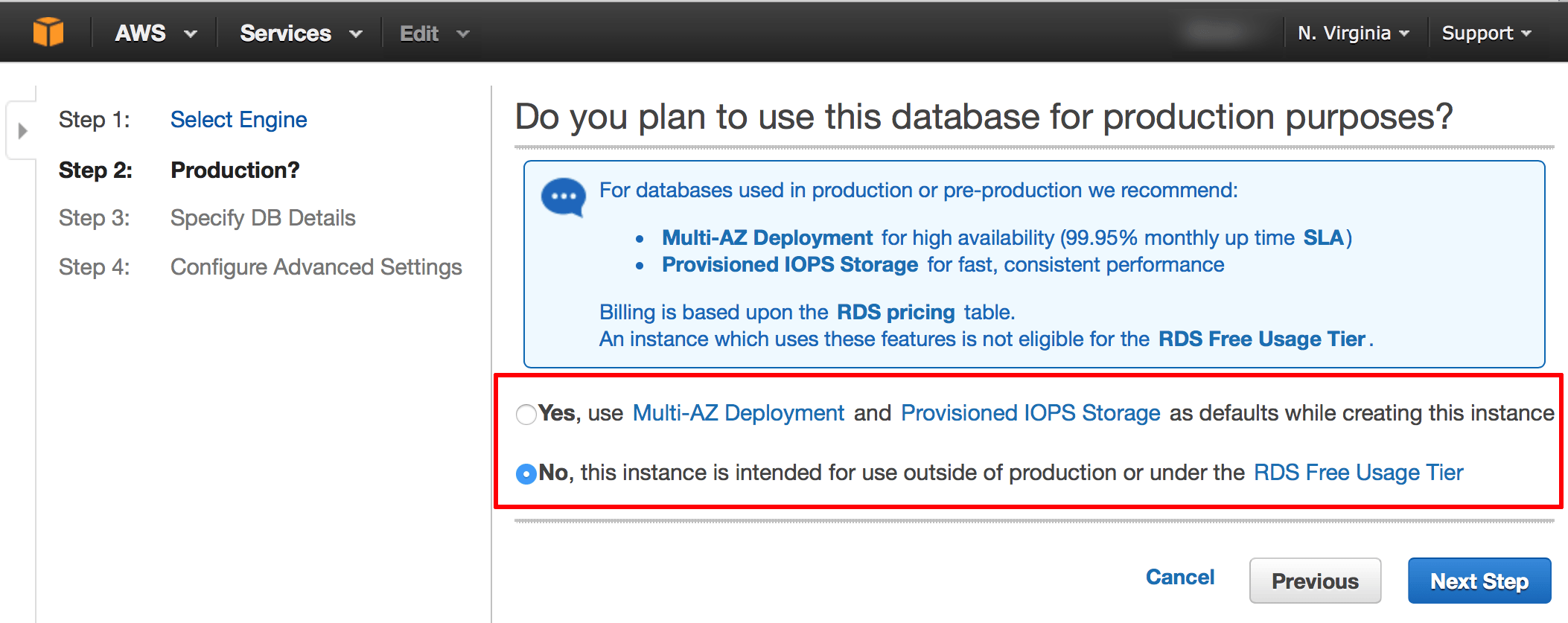
Select the "Instances" menu item and the "Launch DB Instance" command.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-1-4814ba24.png)

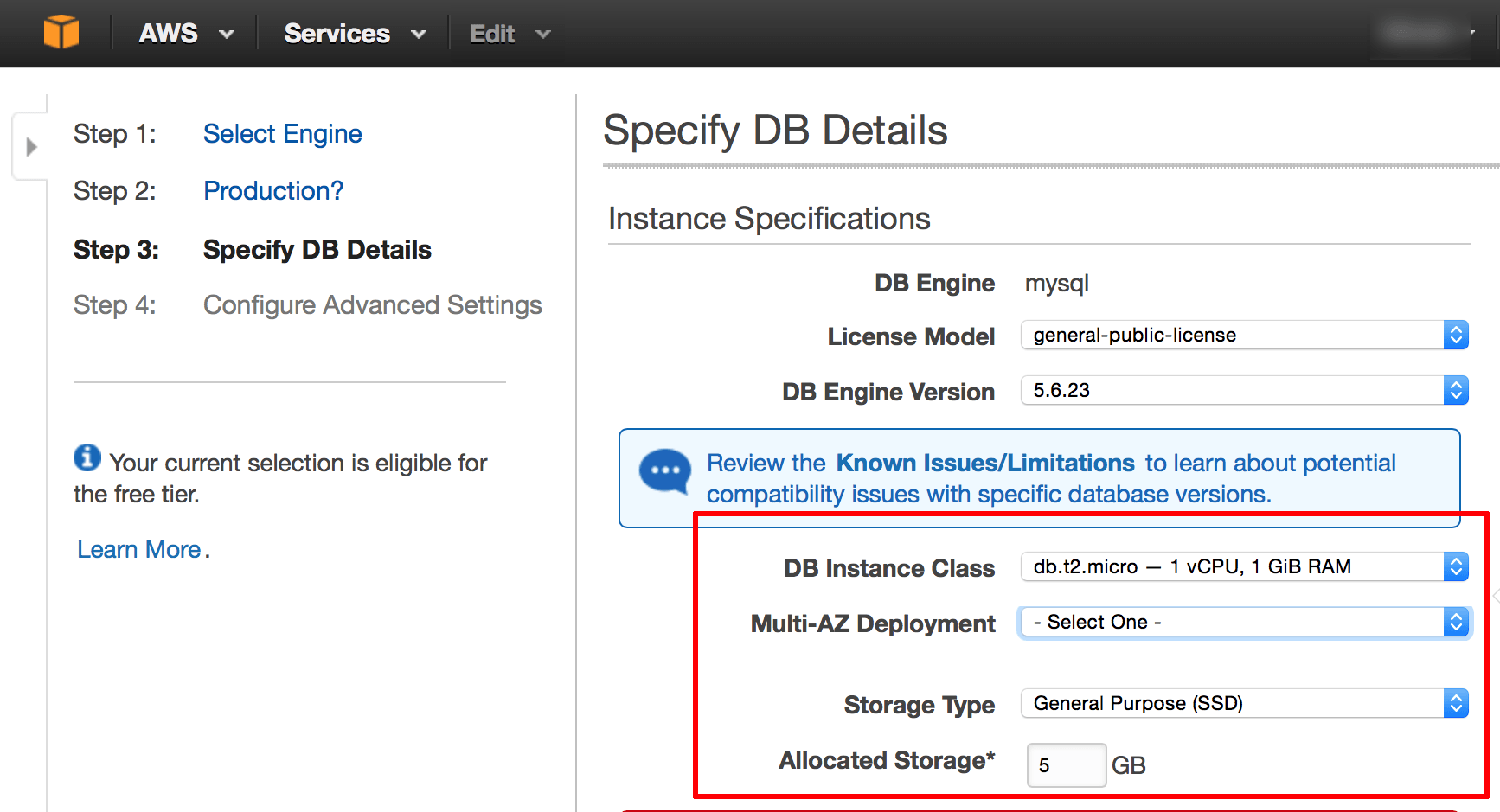
On the "Select Engine" page, select "MySQL" as the database engine. Then, click the "Select" button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-2-93cd2c6b.png)

On the resulting "Production?" page, choose between production or non-production usage scenarios. Note that the AWS Free Usage Tier only applies to non-production instances. Then, click the "Next Step" button to proceed.

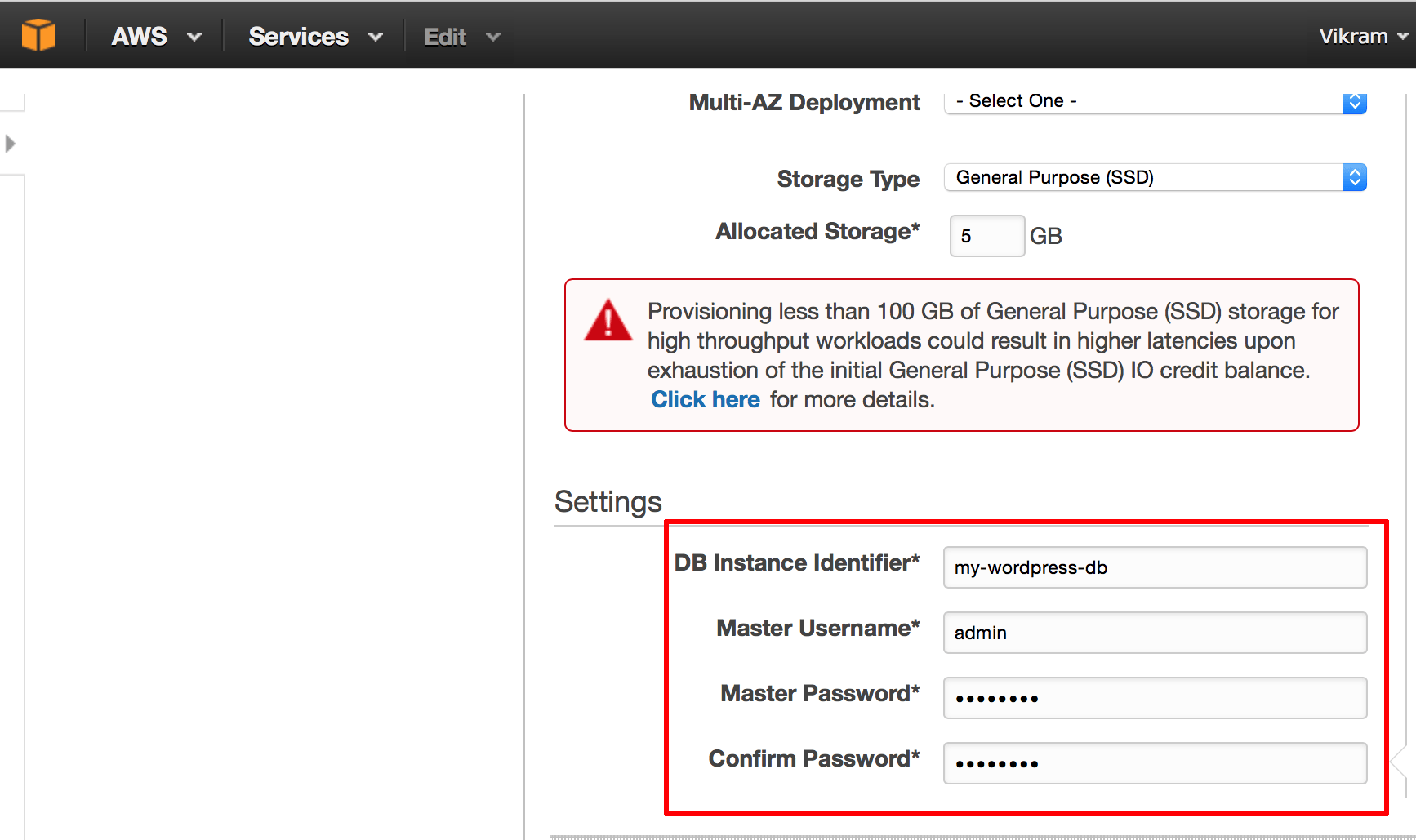
[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-3-9c2502f7.png)

On the resulting "Specify DB Details" page, enter the details for the database you wish to create, most notably the instance class, the storage type and the allocated storage size.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-4-da059896.png)

|  |
| --- |
| TIP: For small PHP applications intended for personal use, a Micro or Small instance will usually suffice. |

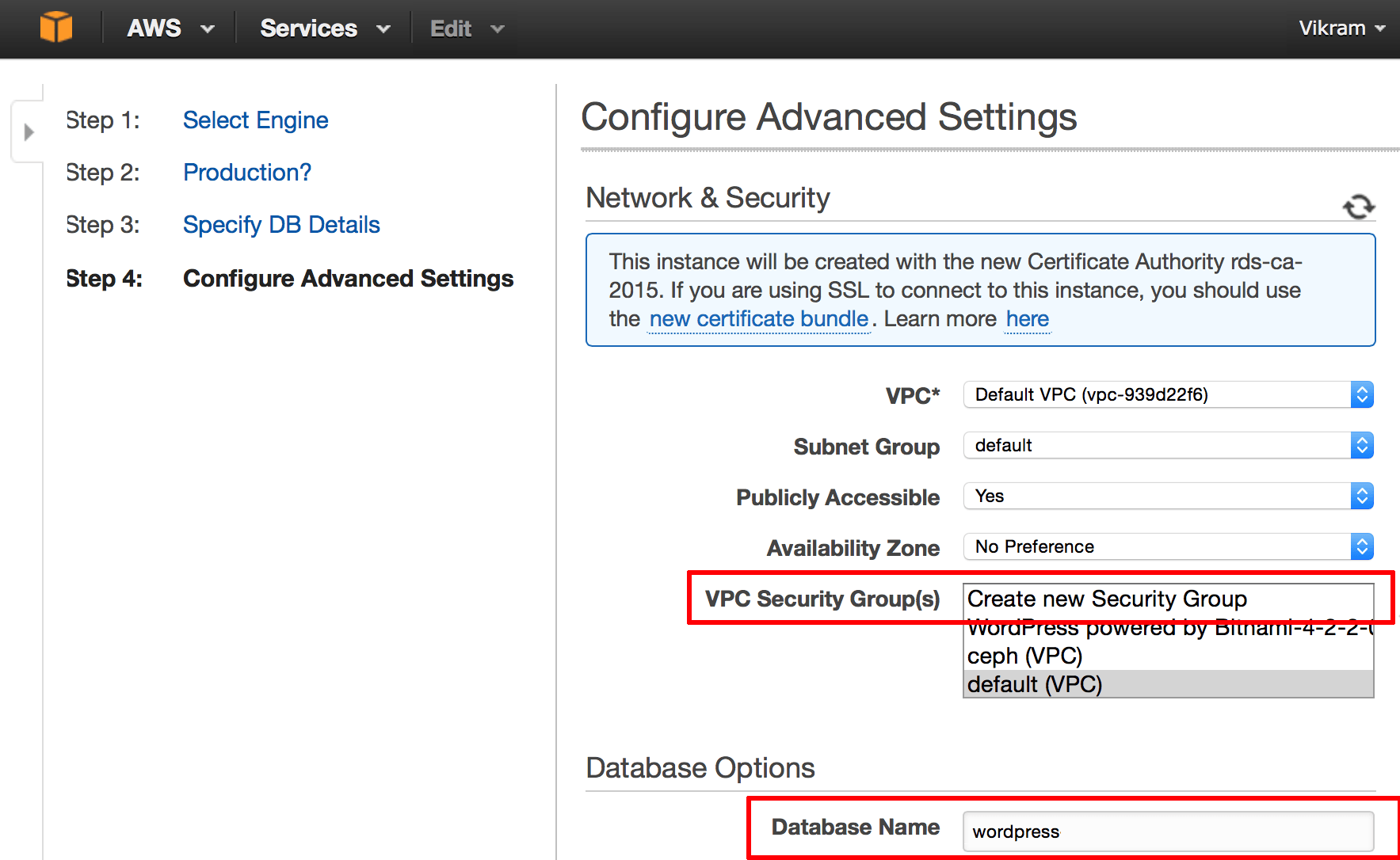
On the same page, specify a unique name for the database instance, together with a username and password. Then, click the "Next Step" button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-5-be242e86.png)

|  |
| --- |
| IMPORTANT: Make a note of these database credentials, since you will need them in a later step. |

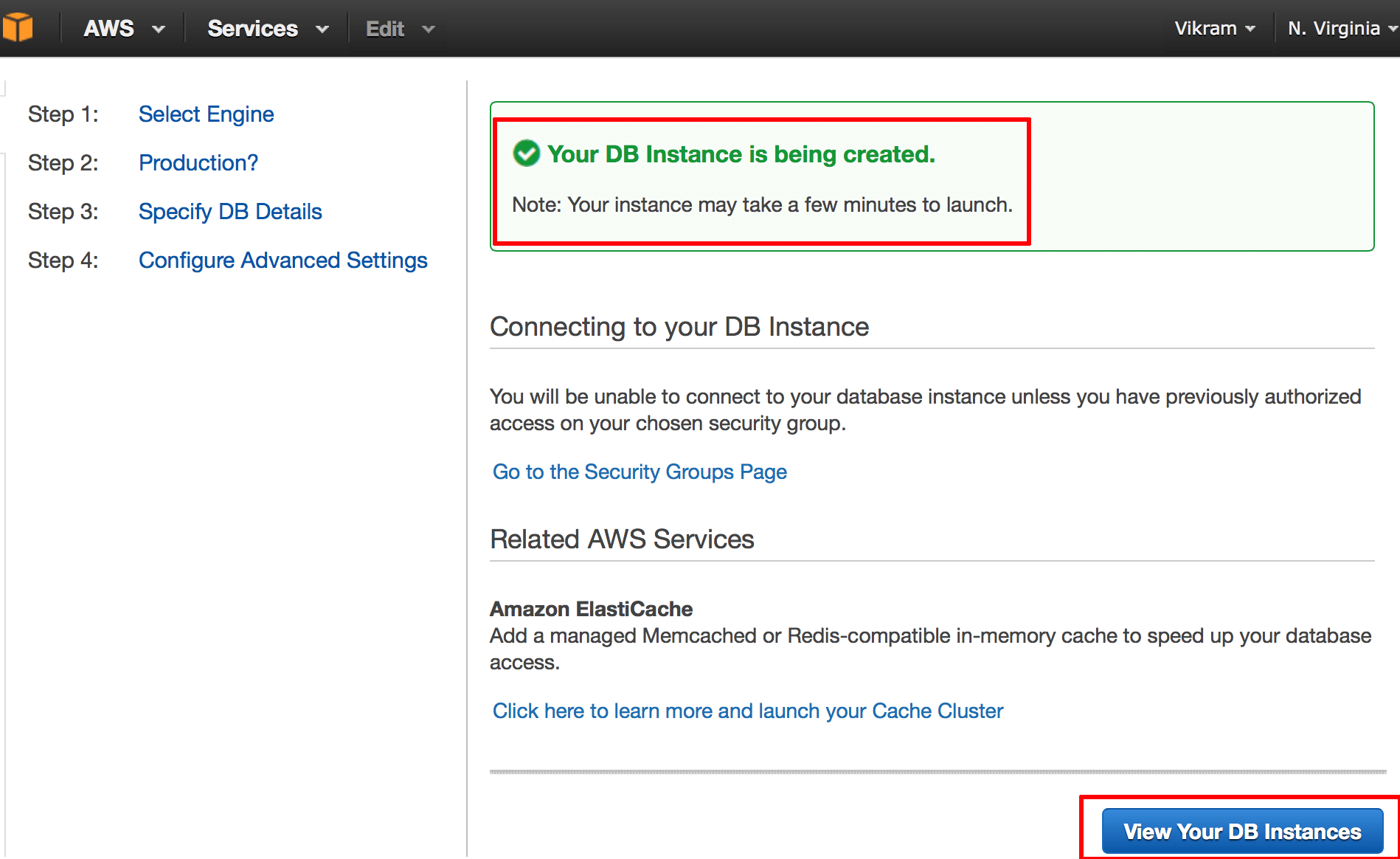
On the resulting "Configure Advanced Settings" page, leave all values at their default settings and select the option to "Create new Security Group" in the "VPC Security Groups" field. Or, if you have an existing security group that allows remote connections to the MySQL port (3306), select that security group instead.

On the same page, in the "Database Options" section, specify a name for the new database. In this example, the new database is named wordpress. This database will be automatically created once the database instance is launched.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-6-ecc39e19.png)

Click the "Launch DB Instance" button to proceed.

Your new database instance will now be launched.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-7-fadb9dc1.png)

Click the "View Your DB Instances" button to view the list of running instances.

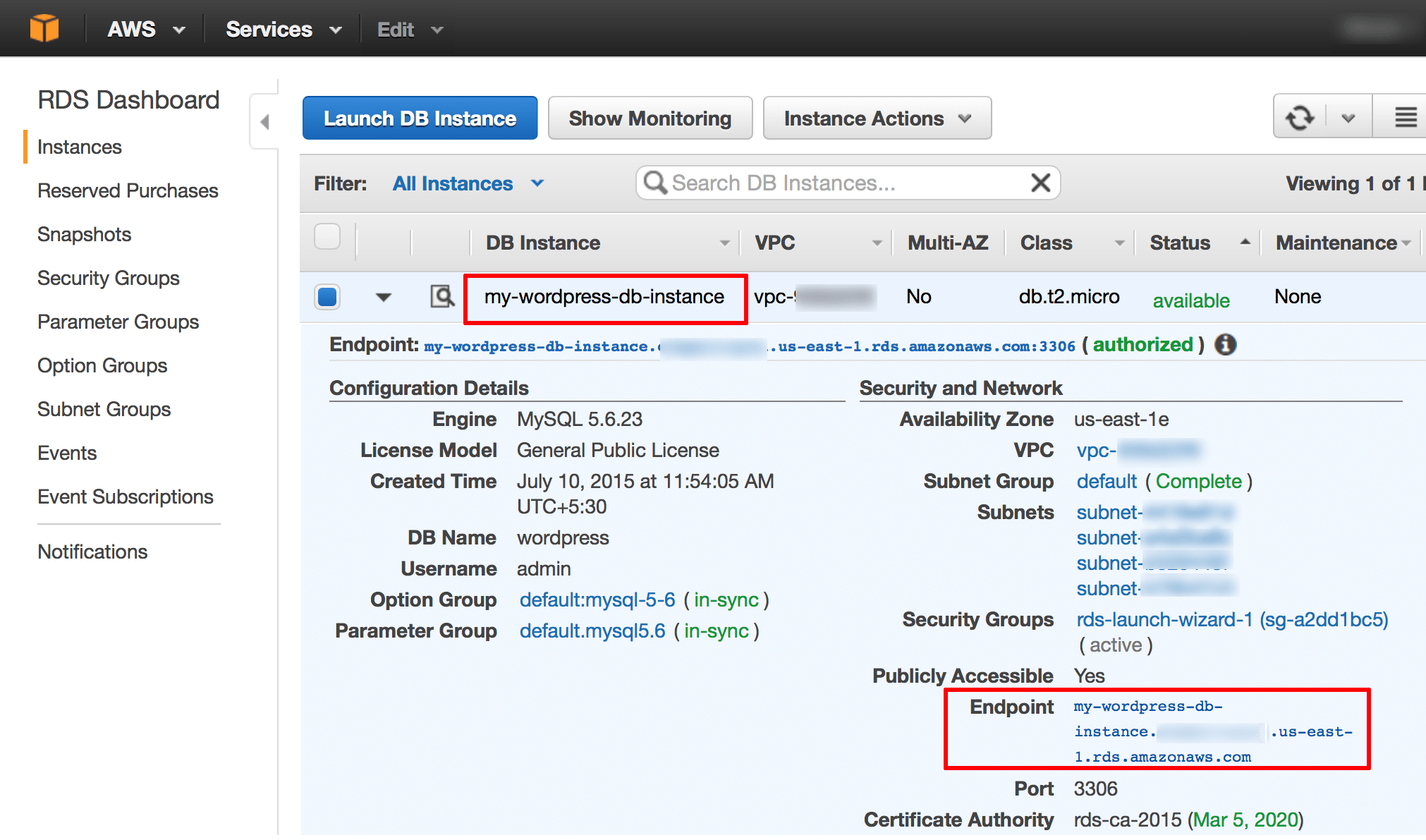
Step 2: Enable security group access

Once your database instance is running, the next step is to allow remote access to the MySQL database server. To do this:

Log in to the [Amazon RDS dashboard](http://aws.amazon.com/rds/) .

Select the "Instances" menu item.

Select and expand the record for the newly-launched MySQL database instance.

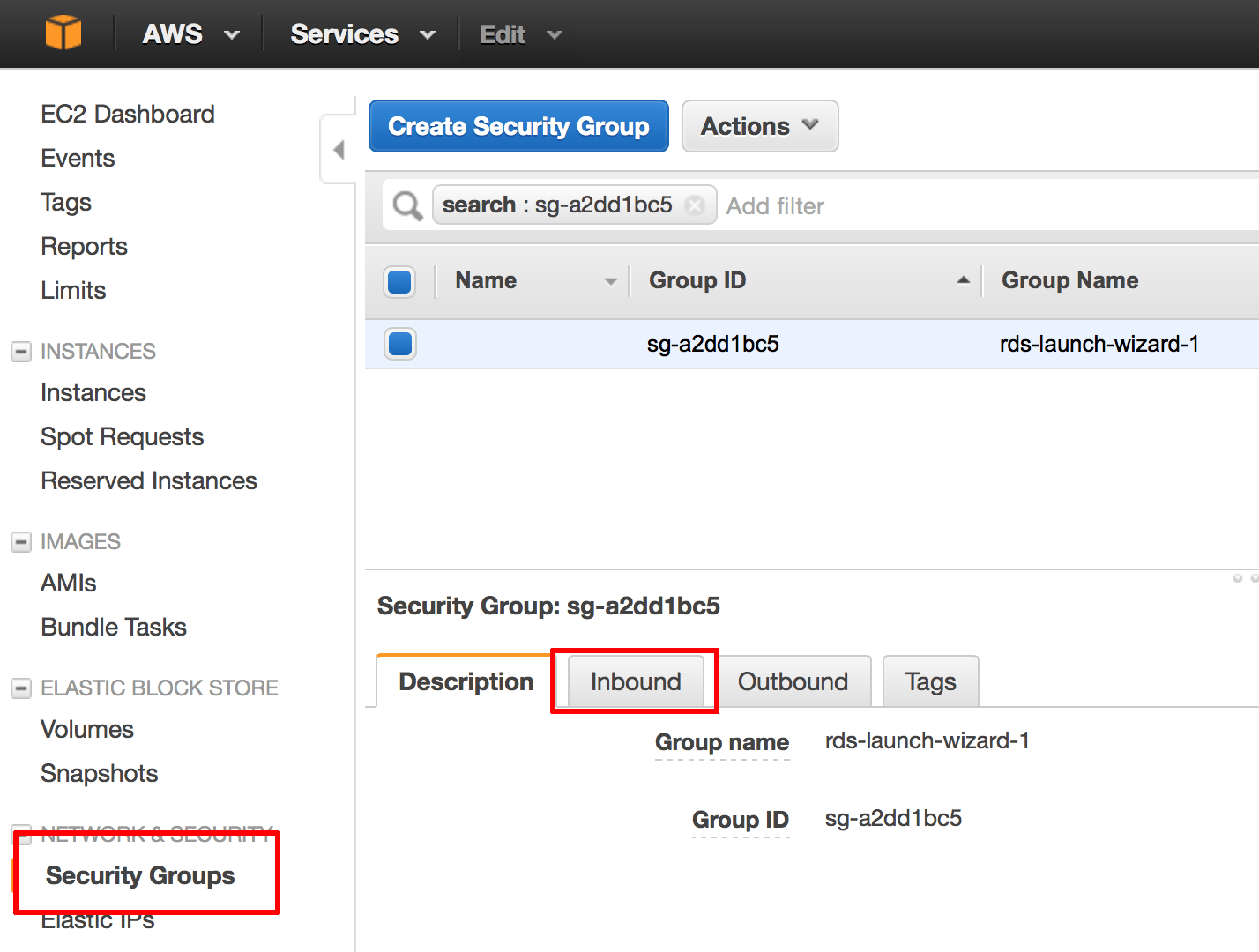
[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-8-83b5e901.png)

|  |
| --- |
| IMPORTANT: Make a note of the database endpoint, as you will need this when importing data. |

Select the current security group. This will open a new browser window containing the "Security Groups" section of the EC2 dashboard and with the current security group pre-selected.

Select the "Inbound" tab of the security group.

Click the "Edit" button.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-9-95682888.png)

In the "Edit inbound rules" pop-up window, click the "Add Rule" button and specify a new rule as follows:

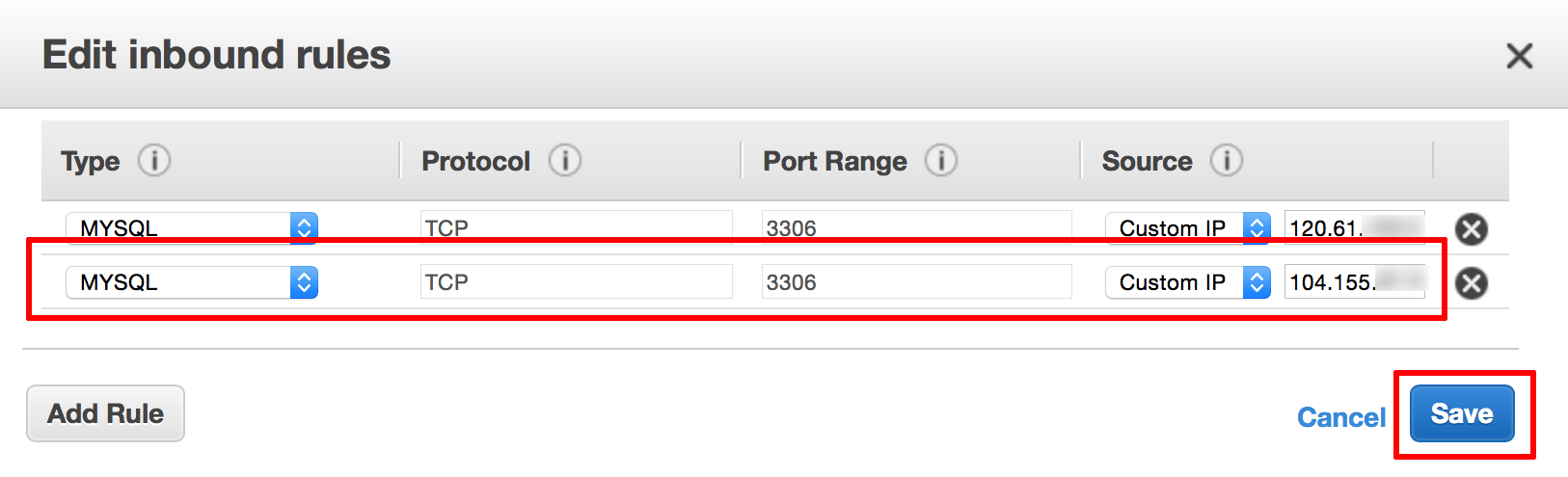
Type: MySQL

Protocol: TCP

Port range: 3306

Source: Custom IP

IP address: The public IP address of your Bitnami application server in CIDR format. For example, if the public IP address of your Bitnami application server is 101.102.103.104, specify the IP address as 101.102.103.104/32.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-10-ddda9b48.png)

Click "Save" to save your changes.

Step 3: Disable write access to the application

You can now migrate your application database to Amazon RDS. However, before you do this, you must disable write access to the application so that the original and new databases stay in sync.

The method to do this varies from application to application. In some cases, the application itself offers a "maintenance mode" which can be activated for the duration of the data migration. In others, you might need to download a maintenance mode plugin for this purpose. If neither of these options is available, you must manually disable login access by redirecting users to a static maintenance page.

In this WordPress-based example, the easiest way to disable write access is to install and activate the [WP Maintenance Mode plugin](https://wordpress.org/plugins/wp-maintenance-mode/) . This plugin will display a splash screen notification to inform users that the WordPress blog/website is down for maintenance and cannot be accessed.

Step 4: Export the application database from your Bitnami stack

The next step is to export the application database. You can either do this using the command-line *mysqldump* tool or the browser-based phpMyAdmin application.

Using the command line

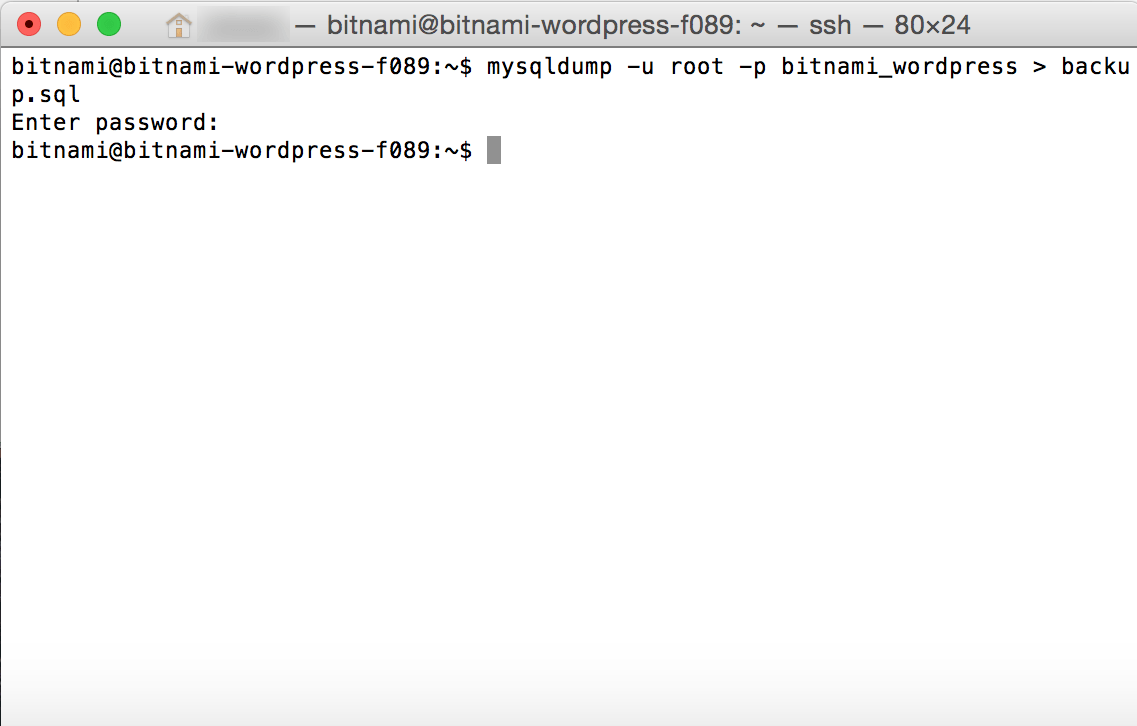
Use the *mysqldump* tool to create a backup of the current database as follows:

Obtain the password for your Bitnami application database.

Log in to the server console via SSH.

Execute the following command to create a backup of the current database. In this example, the database is named *bitnami\_wordpress* . Replace this with the name of the database you wish to export.

mysqldump -u root -p bitnami\_wordpress > backup.sql

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-11-75bf8bc9.png)

Enter the database password when prompted.

This will produce a *backup.sql* file in the current directory with the content of the selected database. Store this file carefully, as you will need it in the next step.

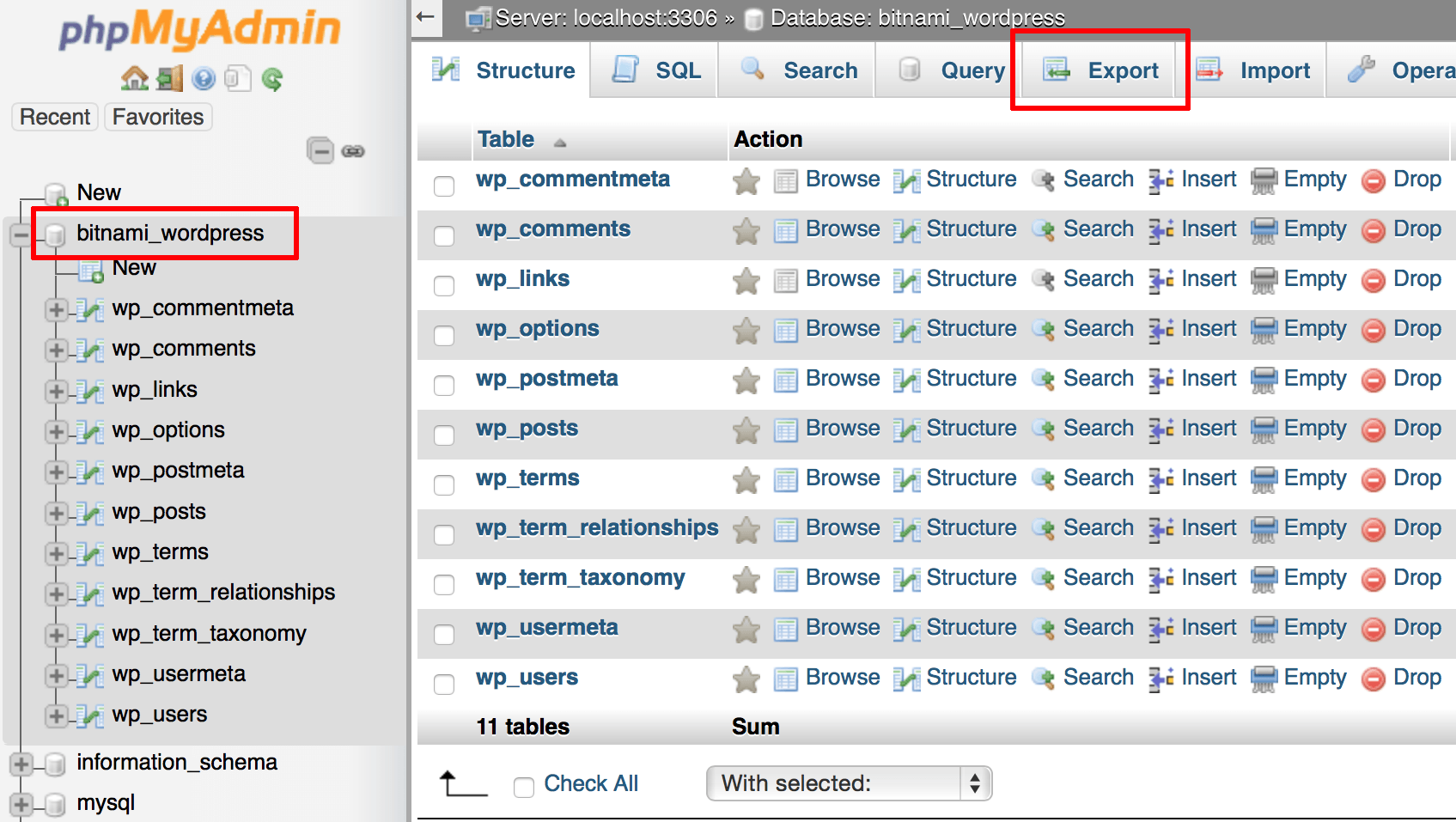
Using phpMyAdmin

You can also accomplish this with the browser-based phpMyAdmin application included by default with Bitnami stacks.

Log in to the included phpMyAdmin application [using these instructions](https://docs.bitnami.com/aws/components/phpmyadmin) .

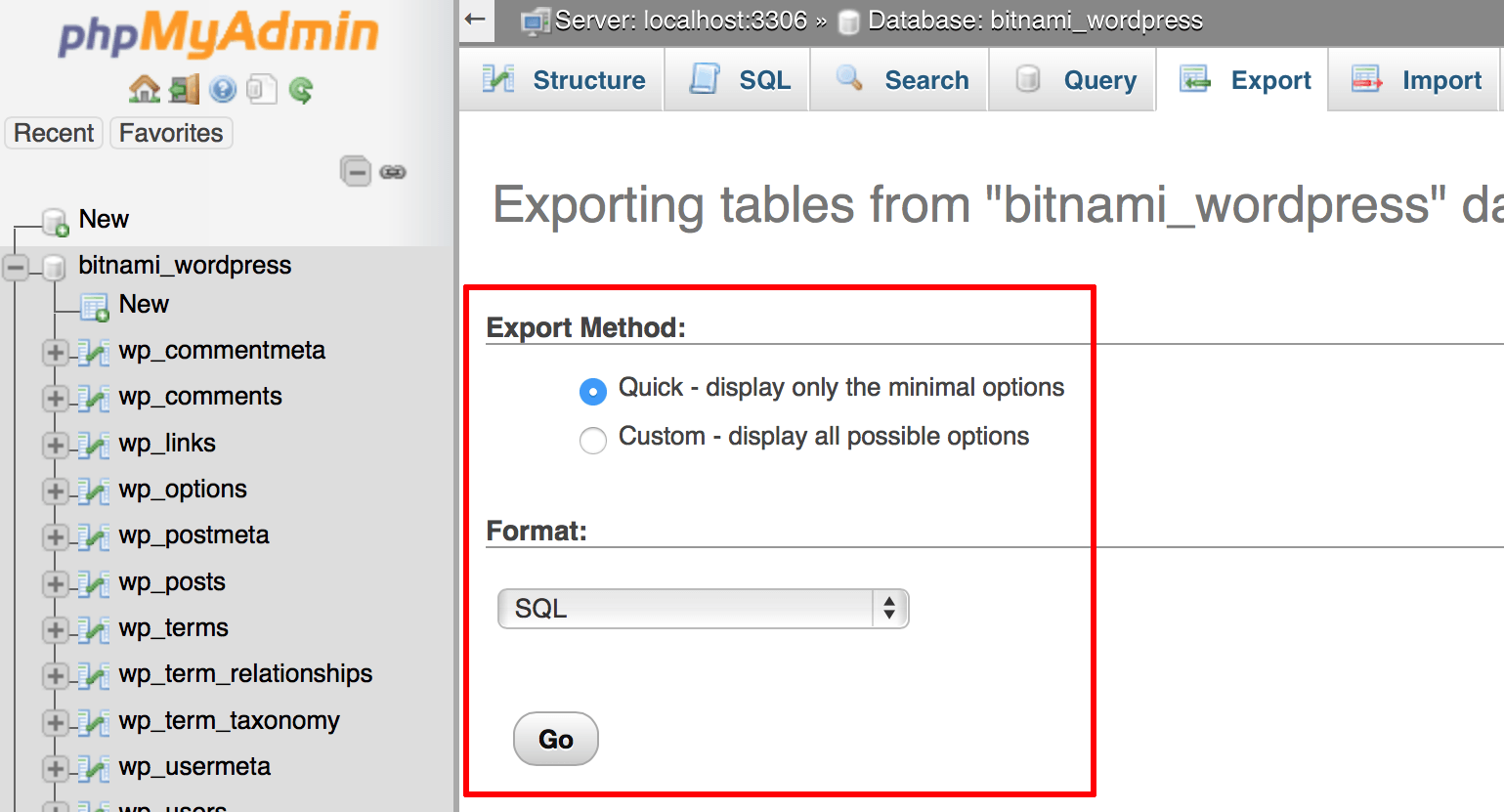
Select the application database in the left navigation menu. In this example, the database is named *bitnami\_wordpress* .

Select the "Export" menu item.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-12-21f6e2c3.png)

On the resulting page, select the "Quick" export method and the "SQL" output format.

Click "Go".

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-13-09623bd9.png)

An SQL export file will be created and downloaded to your desktop. Store this file carefully, as you will need it in the next step.

Step 5: Import the application database to Amazon RDS

The next step is to import the application database to your RDS database instance. Again, you can accomplish this either using the command-line tool or the browser-based phpMyAdmin application.

Using the command line

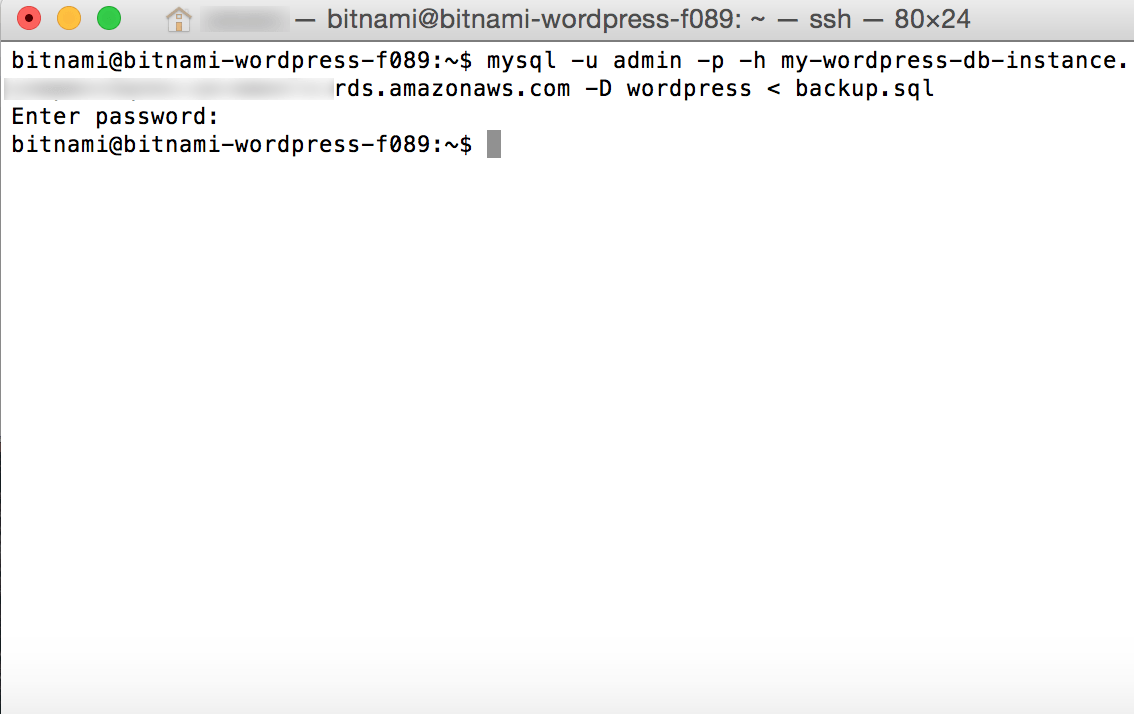
Use the *mysql* command-line client to import the backup from the previous step into your Amazon RDS database instance as follows:

Log in to the server console via SSH.

Execute the following command to import the backup file. In this example, the target database is named *wordpress* . Replace this with the name of the database you wish to import into.

mysql -u USERNAME -p -h RDS-ENDPOINT -D wordpress < backup.sql

Remember to replace the USERNAME placeholder with the master RDS username you specified in Step 1, and the RDS-ENDPOINT placeholder with the actual endpoint of your Amazon RDS database instance.

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-14-cf711257.png)

The original application database content will now be imported into the new Amazon RDS database.

Using phpMyAdmin

You can also accomplish this with the browser-based phpMyAdmin application included with the Bitnami stack. However, it is first necessary to configure phpMyAdmin to connect to your RDS database instance, as described below:

Log in to your server console via SSH.

Edit the phpMyAdmin configuration file at */opt/bitnami/apps/phpmyadmin/htdocs/config.inc.php* and add the lines below to the end of the file:

$i++; $cfg['Servers'][$i]['verbose'] = 'Amazon RDS'; $cfg['Servers'][$i]['host'] = 'RDS-ENDPOINT'; $cfg['Servers'][$i]['port'] = '3306'; $cfg['Servers'][$i]['socket'] = ''; $cfg['Servers'][$i]['connect\_type'] = 'tcp'; $cfg['Servers'][$i]['extension'] = 'mysqli'; $cfg['Servers'][$i]['auth\_type'] = 'cookie'; $cfg['Servers'][$i]['AllowNoPassword'] = false;

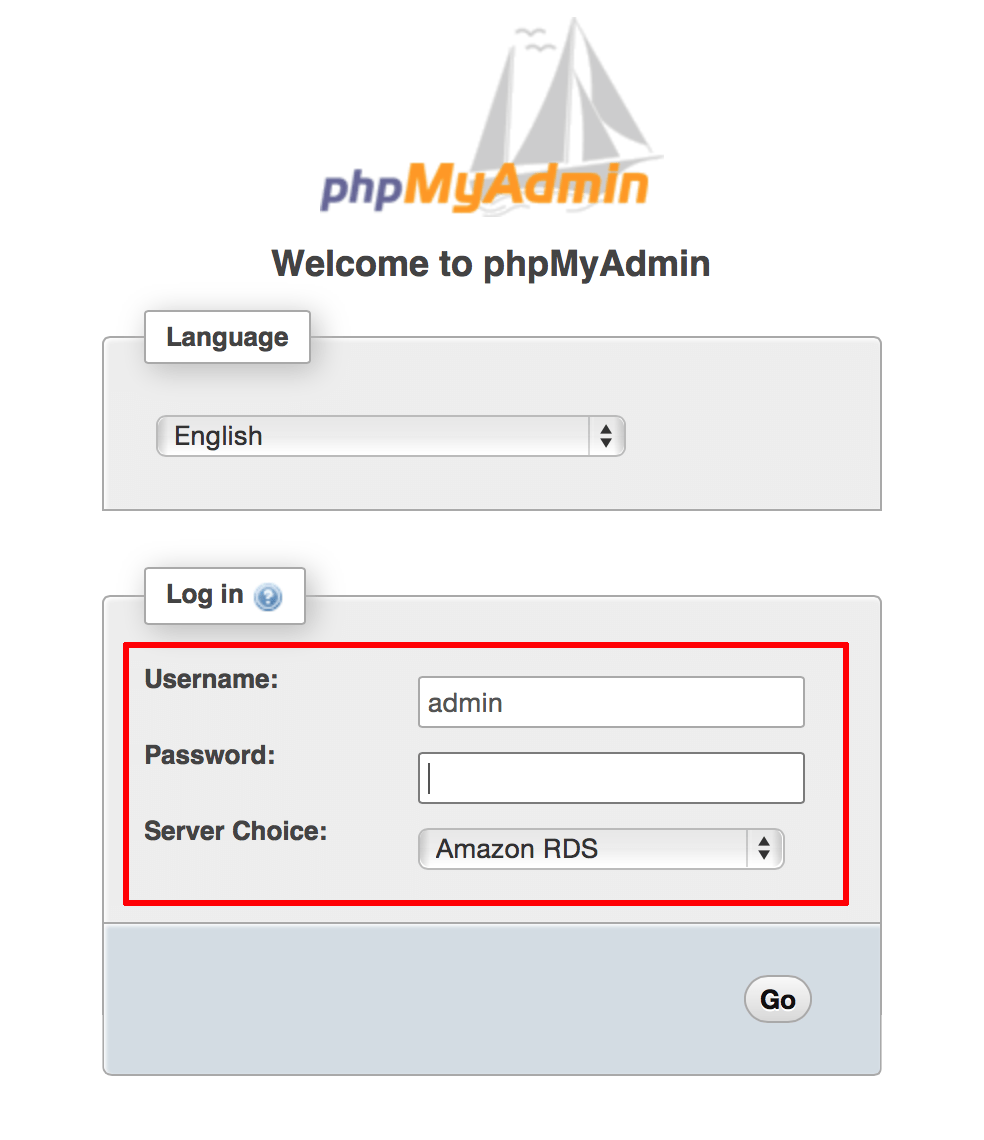
Remember to replace the RDS-ENDPOINT placeholder in the above code with the actual endpoint of your Amazon RDS database instance. These lines will make it possible to access your Amazon RDS database instance through phpMyAdmin.

Save the file.

Next:

Browse to the phpMyAdmin application [using these instructions](https://docs.bitnami.com/aws/components/phpmyadmin) and select the "Amazon RDS" server.

Log in with the database username and password you defined in Step 1.

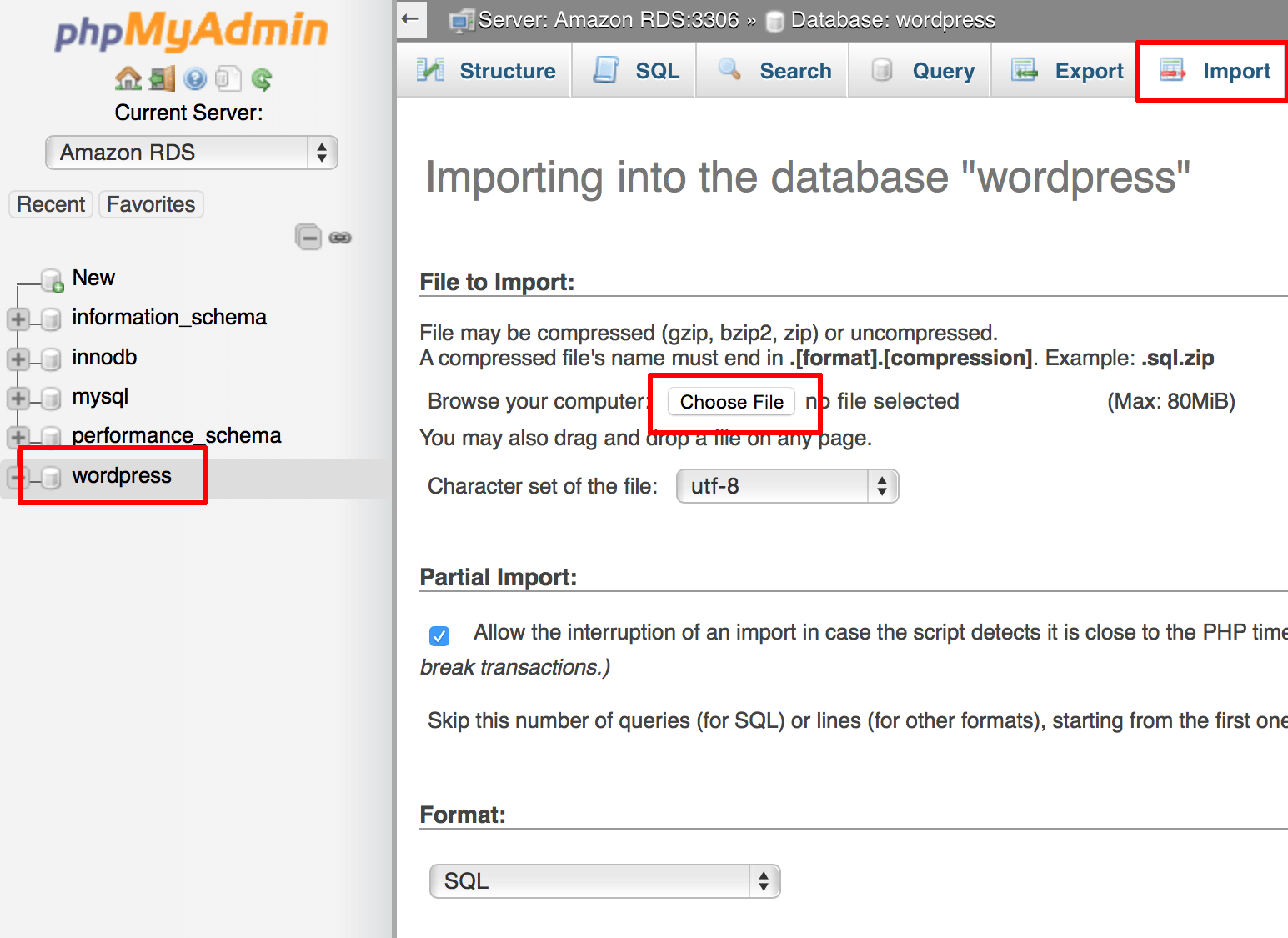
[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-15-1a8b621b.png)

Select the new database in the left navigation menu. In this example, the new database is named *wordpress* as per the input you provided in Step 1.

Select the "Import" menu item.

On the resulting page, select the SQL export file created in Step 4.

Click "Go".

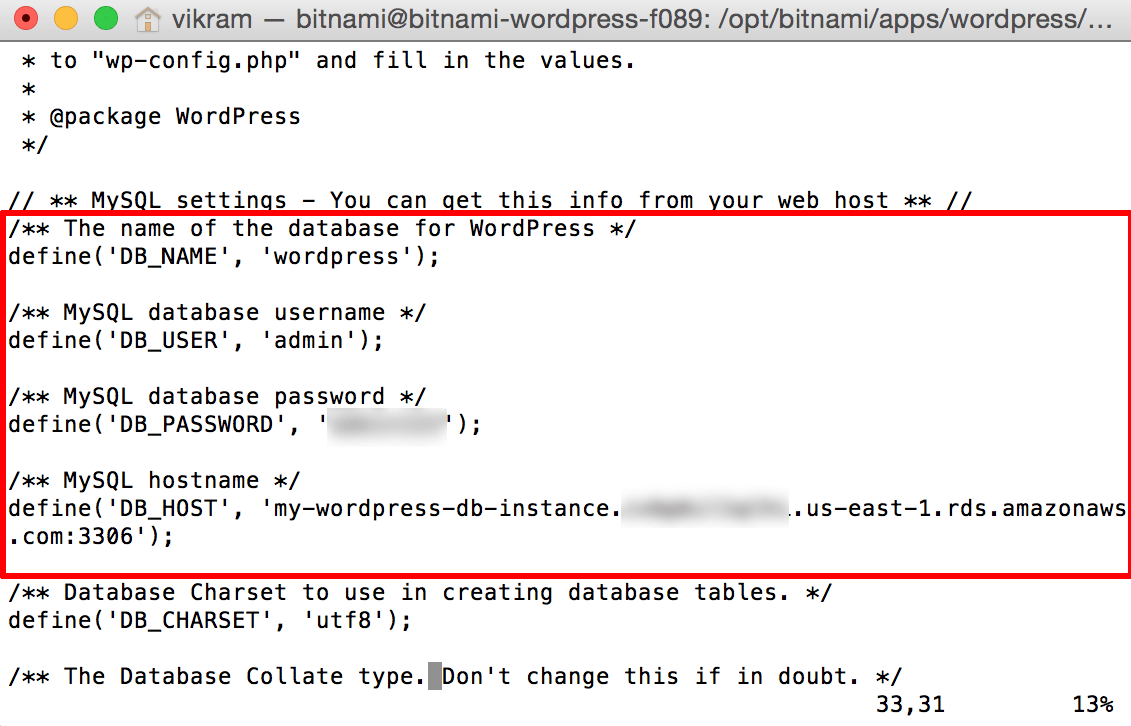
[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-16-ba4ded62.png)

The original application database content will now be imported into the new Amazon RDS database. A confirmation message will be displayed.

Step 6: Reconfigure the application to use the new database

Once your database has been transferred to Amazon RDS, the next step is to update your application configuration and point it to the new database. The procedure to do this varies from application to application, but typically involves modifying a configuration file and specifying the new database host, database username and password, and new database name.

For example, to update the WordPress configuration, edit the file at */opt/bitnami/apps/wordpress/htdocs/wp-config.php* and modify the various configuration variables as shown in the image below:

[](https://docs.bitnami.com/images/img/how_to_guides/migrate-database-rds/migrate-database-rds-17-49701cf1.png)

Step 7: Re-enable write access to the application

You can now turn off maintenance mode and re-enable full access to the application, by reversing the steps performed in Step 3.