Tutorial: Connecting your WordPress website to a MySQL managed database in Amazon Lightsail

Last updated: June 27, 2019

Crucial WordPress website data for posts, pages, and users, is stored on the MySQL database that is running on the instance in Amazon Lightsail. If the WordPress instance fails, your data may become unrecoverable. To avoid this scenario, you should transfer your website data to a MySQL managed database in Amazon Lightsail.

This tutorial shows you how to transfer your WordPress website data for posts, pages, and users, to a MySQL managed database in Lightsail. It also shows you how to edit the WordPress configuration file so that your WordPress website connects to the new managed database, and stops using the database running on the instance.

Contents

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already:

- Create a WordPress instance in Lightsail, and make sure that it's in a running state. For more information, see Tutorial: Launch and configure a WordPress instance in Amazon Lightsail.
- Create a MySQL managed database in Lightsail in the same AWS Region as your WordPress instance, and make sure it's in a running state. WordPress works with all of the MySQL database options available in Lightsail. For more information, see Creating a database in Amazon Lightsail.
- Enable public and data import modes for your MySQL managed database. You can disable these modes after completing the steps in this tutorial. For more information, see Configuring the public mode for your database in Amazon Lightsail and Configuring the data import mode for your database in Amazon Lightsail.

Step 2: Transfer the WordPress database to a MySQL managed database in Lightsail

Complete these steps to transfer your WordPress website data for posts, pages, and users, to your MySQL managed database in Lightsail.

1. Sign in to the Lightsail console.

2. In the **Instances** tab, choose the browser-based SSH client icon for your WordPress instance.
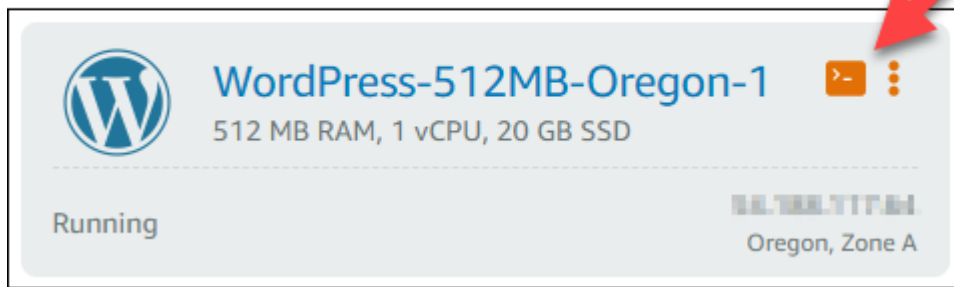
3. After the browser-based SSH client is connected to your WordPress instance, enter the following command to transfer the `bitnami_wordpress` database on the instance to your MySQL managed database. Be sure to replace *DbUserName* with the user name for your managed database, and replace *DbEndpoint* with the endpoint address for your managed database.

```
mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary  -p$(cat
/home/bitnami/bitnami_application_password) | mysql -u DbUserName --host
DbEndpoint --password
```

Example:

```
mysqldump -u root --databases bitnami_wordpress --single-transaction
        --compress --order-by-primary -p$(cat
/home/bitnami/bitnami_application_password) | mysql
        -u dbmasteruser --host
          http://ls-a3420cc0b7a6c82a21d611e6e58198cf01c.czowadgeezqi.us-
west-2.rds.amazonaws.com
        --password
```

4. At the prompt, enter the password for your MySQL managed database, and press **Enter**.

Note

You will not be able to see the password as it is being typed. You should copy and paste the password into the SSH client window so that you don't mistype it.

5. A result similar to the following displays if the data was successfully transferred.

If you get an error, confirm that you're using the correct database user name, password, or endpoint, and try again.

Step 3: Configure WordPress to connect to your MySQL managed database

Complete these steps to edit the WordPress configuration file so that your website connects to your MySQL managed database.

1. In the browser-based SSH client that is connected to your WordPress instance, enter the following command to create a backup of the WordPress configuration file in case something goes wrong:

```
cp /opt/bitnami/apps/wordpress/htdocs/wp-config.php
/opt/bitnami/apps/wordpress/htdocs/wp-config.php-backup
```

2. Enter the following command to open the WordPress configuration file using the Nano text editor:

```
nano /opt/bitnami/apps/wordpress/htdocs/wp-config.php
```

3. Press the down arrow on your keyboard to scroll down until you find the values for **DB_USER**, **DB_PASSWORD**, and **DB_HOST** as shown in the following example.

4. Modify the following values:

   o **DB_USER** — Edit this to match the master user name for the MySQL managed database. The default master user name for Lightsail managed databases is dbmasteruser.

   o **DB_PASSWORD** — Edit this to match the strong password for the MySQL managed database. For more information, see Managing your database password in Amazon Lightsail.

   o **DB_HOST** — Edit this to match the endpoint for the MySQL managed database. Be sure to add the `:3306` port number at the end of the host address. For example: `ls-c6d76d20d6s6d7a695e26.csodadgdeaqi.us-west-2.rds.amazonaws.com:3306`.

The result should look like the following:

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'bitnami_wordpress');

/** MySQL database username */
define('DB_USER', 'dbmasteruser');

/** MySQL database password */
define('DB_PASSWORD', '*'Q+s)         _?1|jY');

/** MySQL hostname */
define('DB_HOST', 'ls-c6d76d20f14d2d         ca7a695e26.czo      zqi.us-west-2.rds.amazonaws.com:3306');
```

5. Press **Ctrl+X** to exit Nano, then press **Y** and **Enter** to save your edits to the WordPress configuration file. **Note**
   If for any reason you need to restore the original `wp-config.php` file, enter the following command to restore it using the backup you created earlier in this tutorial:

   ```
   cp /opt/bitnami/apps/wordpress/htdocs/wp-config.php-backup
   /opt/bitnami/apps/wordpress/htdocs/wp-config.php
   ```

6. Enter the following command to restart the web services:

   ```
   sudo /opt/bitnami/ctlscript.sh restart
   ```

   A result similar to the following displays when the services have restarted:

   ```
   bitnami@ip-172-26-13-236:~$ sudo /opt/bitnami/ctlscript.sh restart
   Syntax OK
   /opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
   /opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
   /opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
   /opt/bitnami/mysql/scripts/ctl.sh : mysql  started at port 3306
   /opt/bitnami/php/scripts/ctl.sh : php-fpm started
   Syntax OK
   /opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
   bitnami@ip-172-26-13-236:~$ █
   ```

   Congratulations! Your WordPress site is now configured to use the MySQL managed database.

Step 4: Complete the next steps

Complete these steps after you're done connecting your WordPress website to a MySQL managed database.

- Create a snapshot of your WordPress instance. For more information, see Creating a snapshot of your Linux or Unix instance in Amazon Lightsail.
- You should also create a snapshot of the MySQL managed database. For more information, see Creating a snapshot of your database in Amazon Lightsail.
- Disable public and data import modes for the MySQL managed database. For more information, see Configuring the public mode for your database in Amazon Lightsail and Configuring the data import mode for your database in Amazon Lightsail.