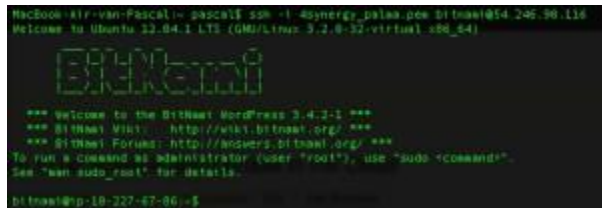


Using Amazon RDS with your WordPress installation

It was quite easy to set up the default instance but this instance isn't very durable and scalable. To improve this we are going to use a database that is installed on a separate machine. That way the current EC2 server becomes a stateless application server and ready to be installed on multiple servers in different [Availability Zones](#). To setup a database in the cloud Amazon offers the service [RDS](#). But before we create this database first make a dump of the existing database on our WordPress server.

To create a sql dump of the existing database we need to login on the server via SSH like this:



```
MacBook-Air:~$ ssh -i key.pem pascal@10-227-67-86.lia
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-32-virtual s86_64)

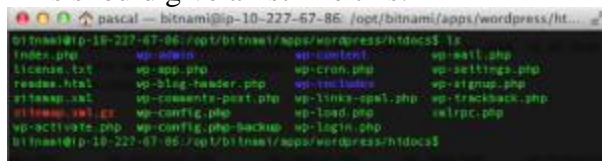
*** Welcome to the BitNami WordPress 3.4.2-1 ***
*** BitNami Wiki: http://wiki.bitnami.org ***
*** BitNami Forums: http://forums.bitnami.org ***
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-10-227-67-86:~$
```

Once we are logged in collect the username/ password and database that is used. To do that go to the htdocs directory of the WordPress installation:

```
cd /opt/bitnami/apps/wordpress/htdocs/
ls
```

This should give a list like this:



```
bitnami@ip-10-227-67-86:~$ cd /opt/bitnami/apps/wordpress/htdocs$ ls
index.php      wp-admin      wp-content     wp-mail.php
license.txt    wp-app.php    wp-cron.php    wp-settings.php
readme.html   wp-ling-header.php wp-credits     wp-signup.php
xmlrpc.php     wp-comments-post.php wp-links-opml.php wp-trackback.php
bitnami_wp.gz wp-config.php wp-load.php    xmlrpc.php
wp-activate.php wp-config.php-backup wp-login.php

bitnami@ip-10-227-67-86:~$ cd /opt/bitnami/apps/wordpress/htdocs$
```

The file that we are looking for is the `wp-config.php`. Before we continue make a backup of it:

```
sudo cp wp-config.php wp-config.php-backup
```

Now we can open the file to collect the username, etc with the command:

```
nano wp-config.php
```

Scroll down in the file until you get to the part:

```
1
2 / ** MySQL settings - You can get this info from your web host ** //
3 /** The name of the database for WordPress */
4 define('DB_NAME', 'bitnami_wordpress');
5
6 /** MySQL database username */
7 define('DB_USER', 'bn_wordpress');
8
9 /** MySQL database password */
10 define('DB_PASSWORD', 'thepassword');
11
12 /** MySQL hostname */
13 define('DB_HOST', 'localhost:3306');
```

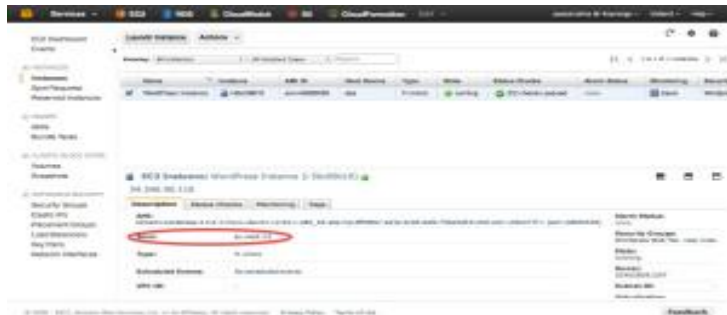
With this information available we can create the backup of the database. First create a directory in the home directory and then create the actual backup:

```
cd
mkdir backup
mysqldump -u bn_wordpress -pthepassword bitnami_wordpress >
backup/backup.sql
```

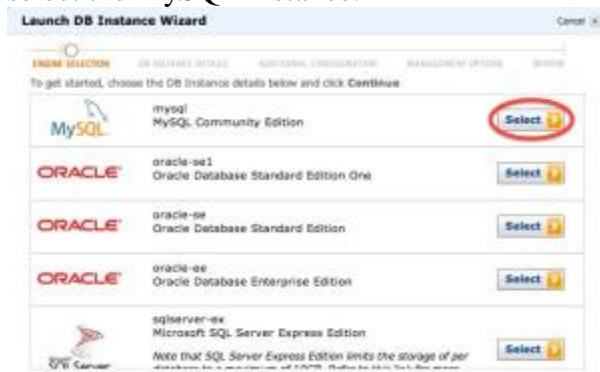
Now make sure the backup exists:

```
ls -ltr backup/  
total 36  
-rw-rw-r-- 1 bitnami bitnami 33238 Dec 11 16:23 backup.sql
```

The next step is to setup a separate database on its own server in the cloud. To do this I make use of [AWS RDS](#). In the remainder of the post I will show how to set this up. In this case I will aim for maximum performance so I will place the database server in the same AZ as the WordPress server. To find the AZ select the WordPress instance in the EC2 overview:

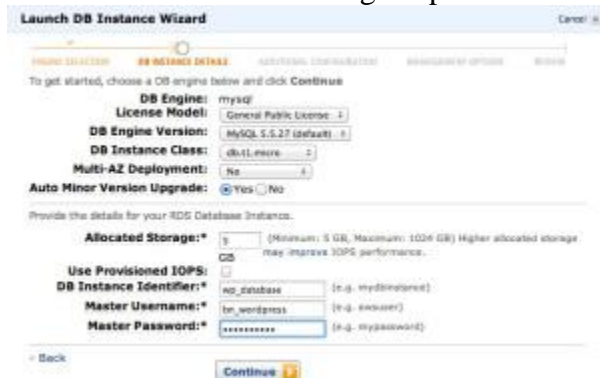


In the AWS Console, go to the Amazon RDS tab, click ‘Launch DB Instance’ • and select the MySQL instance:



In the next screen choose the Micro instance (this one is only used for demo purpose so micro is sufficient here). Set the Multi-AZ deployment to No as we are going for maximum performance, the allocated Storage is 5 GB (minimum).

The DB Instance Identifier you can put a name for this instance. The user and Password should match the user and password we selected earlier from the ‘wp-config.php’™ file. Here is a screenshot with all settings in place:



In the next step match the database name from the ‘wp-config.php’™ file and select the correct Availability Zone:

Launch DB Instance Wizard Cancel X

1. ENGINE SELECTION 2. DB INSTANCE DETAILS 3. ADDITIONAL CONFIGURATION 4. MANAGEMENT OPTIONS 5. REVIEW

Provide the optional additional configuration details below.

Database Name: bitnami_wordpress (e.g. mydb)

Note: If no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port: 3306

Choose a VPC: Not in VPC Only VPCs with a DB Subnet Group(s) are allowed

Availability Zone: eu-west-1a

Option Group: default:mysql-5-5

If you have custom DB Parameter Groups or DB Security Groups you would like to associate with this DB Instance, select them below, otherwise proceed with default settings.

Parameter Group: default:mysql5.5

Security Group: default

Back Continue

For the backups I just accept the defaults but of course in real life you would make some choices here that would match for your case:

Launch DB Instance Wizard Cancel X

1. ENGINE SELECTION 2. DB INSTANCE DETAILS 3. ADDITIONAL CONFIGURATION 4. MANAGEMENT OPTIONS 5. REVIEW

Enabled Automatic Backups: ☒ Yes ☐ No

The number of days for which automated backups are retained.

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

Backup Retention Period: 1 days

The daily time range during which automated backups are created if automated backups are enabled.

Backup Window: ☐ Select Window ☒ No Preference

The weekly time range (in UTC) during which system maintenance can occur.

Maintenance Window: ☐ Select Window ☒ No Preference

Back Continue

Launch DB Instance Wizard Cancel X

1. ENGINE SELECTION 2. DB INSTANCE DETAILS 3. ADDITIONAL CONFIGURATION 4. MANAGEMENT OPTIONS 5. REVIEW

The requested number of IOPS operations per second that the DB Instance can support. Only valid when Provisioned IOPS is selected.

Please review the information below, then click Launch DB Instance.

Engine: mysql
Engine Version: MySQL 5.5.27
License Model: general-public-license
Auto Minor Ver. Upgrade: Yes
DB Instance Class: db.t1.micro
Multi-AZ Deployment: No
Allocated Storage: 5
Provisioned IOPS: default
DB Instance Identifier: wpdatabase
Master User Name: bn_wordpress
Master User Password: *****

Database Name: bitnami_wordpress
Database Port: 3306
Availability Zone: eu-west-1a
Option Group: default:mysql-5-5
DB Parameter Group: default:mysql5.5
DB Security Group(s): default
DB Subnet Group:

Backup Retention Period: 1
Backup Window: No Preference
Maintenance Window: No Preference

Back Launch DB Instance

In the final step you get an overview of all your choices and if all is okay launch the RDS:

Launch DB Instance Wizard Cancel X

1. ENGINE SELECTION 2. DB INSTANCE DETAILS 3. ADDITIONAL CONFIGURATION 4. MANAGEMENT OPTIONS 5. REVIEW

The requested number of IOPS operations per second that the DB Instance can support. Only valid when Provisioned IOPS is selected.

Please review the information below, then click Launch DB Instance.

Engine: mysql
Engine Version: MySQL 5.5.27
License Model: general-public-license
Auto Minor Ver. Upgrade: Yes
DB Instance Class: db.t1.micro
Multi-AZ Deployment: No
Allocated Storage: 5
Provisioned IOPS: default
DB Instance Identifier: wpdatabase
Master User Name: bn_wordpress
Master User Password: *****

Database Name: bitnami_wordpress
Database Port: 3306
Availability Zone: eu-west-1a
Option Group: default:mysql-5-5
DB Parameter Group: default:mysql5.5
DB Security Group(s): default
DB Subnet Group:

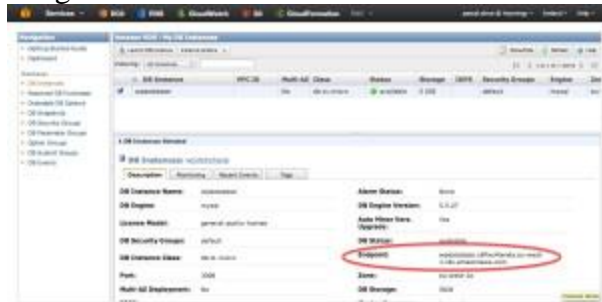
Backup Retention Period: 1
Backup Window: No Preference
Maintenance Window: No Preference

Back Launch DB Instance

You can see your instance in the overview while being created (will take a few minutes since it also creates a backup):



When the database is created you can select it to see the details and copy the public address assigned to it:



One thing we have to do before we can put the backup of the original WP database on the new one is to open up the access to the machine. By default it comes with the standard EC2 security: trust no one. To make the server available open up the security group screen and add the Security Group that is also associated with the WordPress EC2 instance (please note that the name of the DB security group is 'default'™ and has nothing to do with the EC2 security group created earlier):



Now go back to the SSH session on the WordPress server and put the backup into the new DB:

```
mysql -u bn_wordpress -pmypassword --database=bitnami_wordpress --
host=wpdatabase.c6flxs4tenda.eu-west-1.rds.amazonaws.com <
backup/backup.sql
```

where you have to make sure that the host matches your database endpoint.

Now configure the WordPress instance to use this database instead of the local one. Modify the 'wp-config.php'™ so the host matches the remote host:

```
1 // ** MySQL settings - You can get this info from your web host ** //
2 /** The name of the database for WordPress */
3 define('DB_NAME', 'bitnami_wordpress');
4
5 /** MySQL database username */
6 define('DB_USER', 'bn_wordpress');
7
8 /** MySQL database password */
9 define('DB_PASSWORD', 'd314c809ea');
```

```

9
10/** MySQL hostname */
11define('DB_HOST', 'wpdatabase.c6flxs4tenda.eu-west-
121.rds.amazonaws.com:3306');
13
14/** Database Charset to use in creating database tables. */
15define('DB_CHARSET', 'utf8');
16
17/** The Database Collate type. Don't change this if in doubt. */
18define('DB_COLLATE', '');

```

Thatâ€™s it. Now test your installation by opening a web browser with your fixed IP and you should see your blog again:



If it is working then this is a good point to create a snapshot of your database so you can easily reinstall it later.