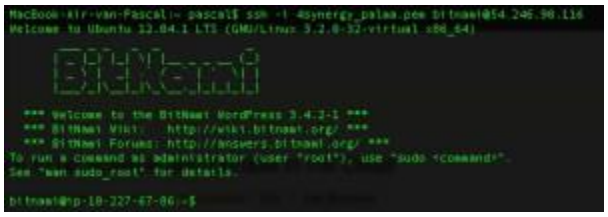
# Using Amazon RDS with your WordPress installation

It was quite easy to set up the default instance but this instance isnâ€™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 Availablity 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:



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:

The file that we are looking for is the â€˜wp-config.phpâ€™. 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 untill 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 */
  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', 'thepassword');
10
11/** MySQL hostname */
  define('DB_HOST', 'localhost:3306');
12
```

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:

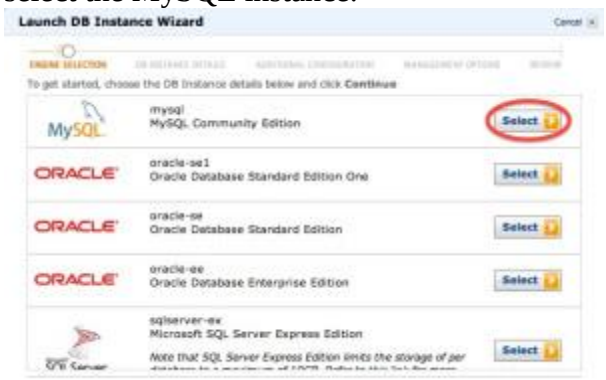
```
cd
mkdir backup
mysqldump -u bn_wordpress -pthepassword bitnami_wordpress >
backup/backup.sql
```

Now make sure the backup exists:

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

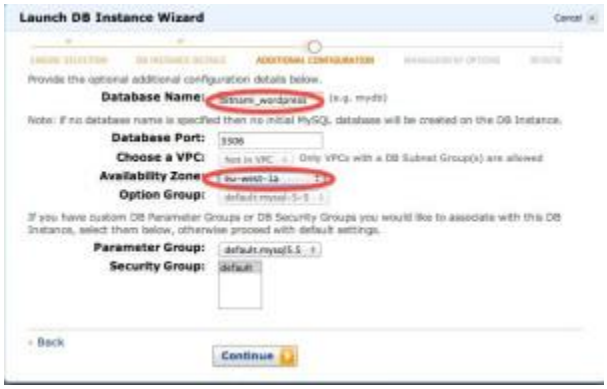
The next step is to setup a separate database on its own server in the cloud. To do this I make use of AWS RDS. In the remainder of the post I will show how to set this up. In this case I will aim for maximum performance so I will place the database server in the same AZ as the WordPress server. To find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

In the AWS Console, go to the Amazon RDS tab, click â€œLaunch DB Instanceâ€• 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 shoul dmatch the user and password we selected earlier from the â€˜wp-config.phpâ€™ file. Here is a screeshot with all settings in place:


In the next step match the database name from the â€˜wp-config.phpâ€™ file and select the correct Availability Zone:



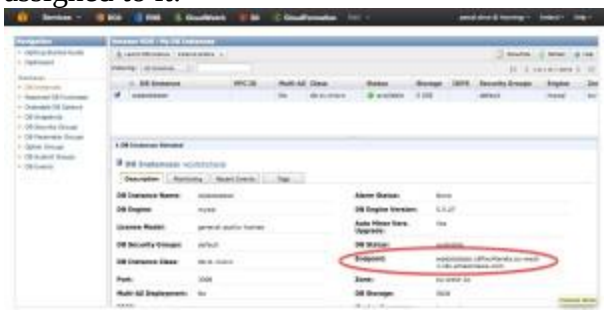

For the backups I just accept the defaults but of course in real life you would make some choices here that would match for your case:



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

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 â€˜defaultâ€™ 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 â€˜wp-config.phpâ€™ 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 */
  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.