AWS: Managing

What you need to know about cloud computing, **Rob Dobozy** shows you how to try it yourself using Amazon Web Services without spending a penny.

Our expert

Rob Dobozy is a SAP Technical Architect. For the last 12 months and he's been walking about with his head in the clouds.

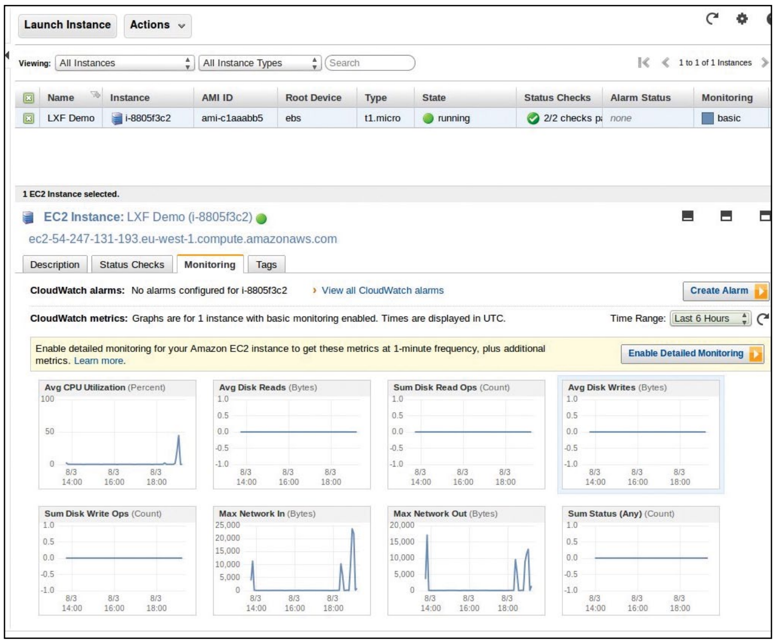
Quick tip

The instance name and IP address will change when you restart it. Make sure you use the new name when connecting to it.

In the first part of this tutorial series [**LXF171**, Get on the Cloud, Tutorials, p86] I ran through all the basic building blocks that make up the Amazon Web Service. In this second instalment you will find out how to manage and monitor the cloud infrastructure we created last time and how to deploy your own Platform as a Service (PaaS) environment in under ten minutes.

First, let's recap the key information from the first part. Your EC2 instance is running in Eu-west region (Ireland) and its type should be t1.micro as this is free for 12 months under Amazon's tier pricing system. The type designation defines how many CPUs and how much memory is available to your instance. In this case, the instance has one 8GB volume which contains the root filesystem.

The EC2 console allows you to control all the aspects of the instance including volumes and snapshots (more about snapshots later). It also allows you to do some basic monitoring. If your instance isn't running, start it now. The instance details section contains four tabs. The first one, Description, shows more detailed information about the instance. The Status Checks tab shows the status of AWS infrastructure and your instance. If you click on tab called Monitoring you should be able to see how the instance is performing. You'll notice that there are also graphs for CPU, disk and network throughput.



	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring
☒	LXF Demo	i-8805f3c2	ami-c1aaabb5	ebs	t1.micro	● running	✓ 2/2 checks pa	none	basic

1 EC2 Instance selected.

EC2 Instance: LXF Demo (i-8805f3c2)

ec2-54-247-131-193.eu-west-1.compute.amazonaws.com

CloudWatch alarms: No alarms configured for i-8805f3c2 › View all CloudWatch alarms

CloudWatch metrics: Graphs are for 1 instance with basic monitoring enabled. Times are displayed in UTC. Time Range: Last 6 Hours

Enable detailed monitoring for your Amazon EC2 instance to get these metrics at 1-minute frequency, plus additional metrics. Learn more.

› I'm not able to overload even this micro instance with my web requests.

Taking a snapshot

Before you start using the cloud for anything serious it's a good idea to make at least one backup, and more importantly, be sure that you can restore it. The Amazon EBS volume snapshot feature makes this a fairly easy process. In AWS console **(https://console.aws.amazon.com/)** select EC2 dashboard and click on the Instances link to see the list of all instances. If your instance is running stop it now. It's possible to create snapshots of EBS volumes while the instance is running, but this doesn't guarantee consistency of your data (especially if you are running databases). Making a snapshot of a running system and then trying to use it is like pulling a power cable from a server and then hoping it will start without any issues. It's, therefore, important to make sure the instance is stopped before creating a snapshot of a root disk. The step-by-step instructions on page 78 explain how to take a snapshot of an EBS volume, how to create a useable volume from it and how to replace the current volume with the new one, effectively bringing your instance to the state it was at the time the snapshot was created.

If everything went well and your instance is running happily you can remove the old unused EBS volume and the snapshot. This is to ensure you won't go over the free tier limit and won't get charged needlessly.

You can use the snapshots to do regular backups and to make backups before major changes (eg patching). A good practice is to use multiple volumes. For example you can use one for the root filesystem and one for all your data. This makes it easier to take snapshots of your precious data more frequently and with less overhead as all you need to do is stop your application (eg database) before taking a snapshot.

the cloud

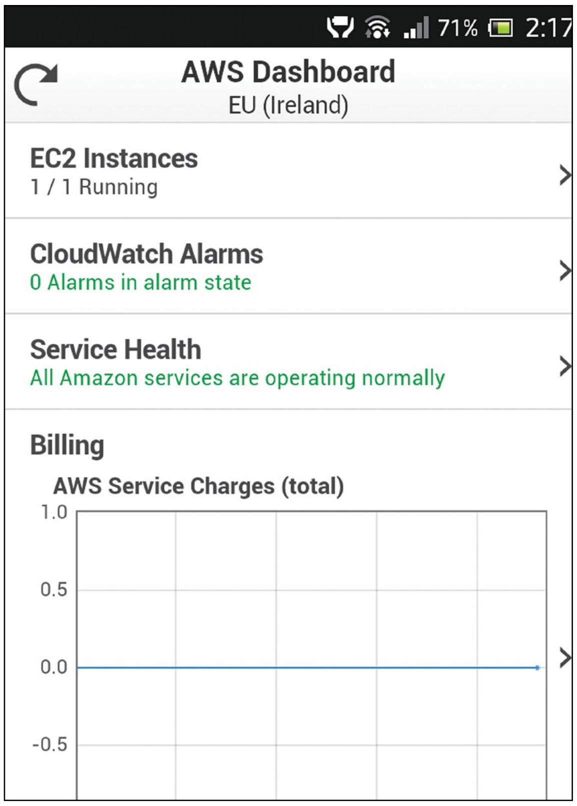
Managing multiple snapshots and doing them manually will quickly become a chore. Fortunately, Amazon is providing command line tools for Linux as well as SDKs for Android, iOS, Java, .NET, Node.js, PHP and Ruby. These tools and libraries not only let you manage snapshots, but also any aspect of your instance.

Monitoring EC2 instances

If you are not keen on doing any programming with AWS, you can use an existing application. AWS Console by AWS Mobile LLC for Android and iOS is quite simple and easy to use. It enables you to check AWS status and current charges, manage your instances and create snapshots. I'm sure it will come in handy while on holidays when you realize that you've forgot to stop your 20 EC2 instances.

Now that your cloud infrastructure is ready, you can start making and deploying your snazzy web applications. To do that, however, you need to have the three remaining components of the LAMP stack: a web server (*Apache*), database (*MySql/ MariaDB*) and execution environment (Perl/PHP/Python). Although they aren't too complicated to install and configure, there may be a more efficient way.

This brings us to the Platform as a Service (PaaS) concept, which you can use to have all the components ready within a few minutes. But what's probably even more important is the ability to quickly and automatically scale your application if there is an increased demand.


71% 2:17

AWS Dashboard
EU (Ireland)

EC2 Instances
1 / 1 Running

CloudWatch Alarms
0 Alarms in alarm state

Service Health
All Amazon services are operating normally

Billing
AWS Service Charges (total)

› Checking AWS Mobile: Everything up, green and no charges. Just the way I like it.

PaaSing to the next layer

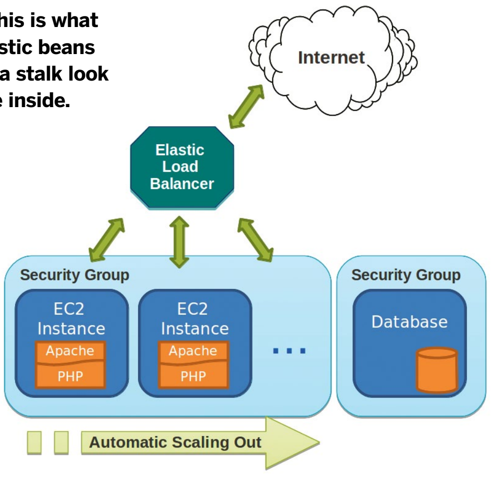
Amazon's implementation of the PaaS concept is called Elastic Beanstalk. A Beanstalk application consists of components, resources and versions. When you generate an application it will create a new EC2 instance with appropriate security settings. It will also create a load balancer component which tries to distribute load between all EC2 instances belonging to the application.

You may be wondering where the additional EC2 instances are coming from. This is an advantage of Elastic Beanstalk as you don't have to do everything yourself. You can define auto scaling rules which automatically start additional instances to accommodate increasing load. If the load on your application decreases, the additional instances are stopped.

This seems like a good time to create your first Elastic Beanstalk application. Before you do anything, it may be better to stop the EC2 instance you've used earlier. You can, obviously, have as many instances as you want, but the resources Elastic Beanstalk generates are taken from your free tier allowance, which allows one instance to run continuously for one month. If you are still in EC2 console, click on Services in the top left corner of the browser and then select Elastic Beanstalk from the menu. Follow the step-by-step instructions on page 79 (*Set up an Elastic Beanstalk demo*) to create an demonstration application.

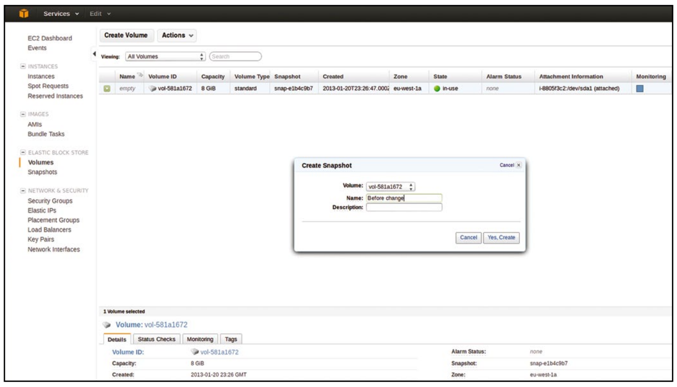
With the basic settings configured and the second version of your application running (have you noticed what's different between the versions?) you can now take your PaaS platform »

› This is what elastic beans on a stalk look like inside.



» If you missed last issue Call 0844 848 2852 or +44 1604 251045

Backing up and replacing instances

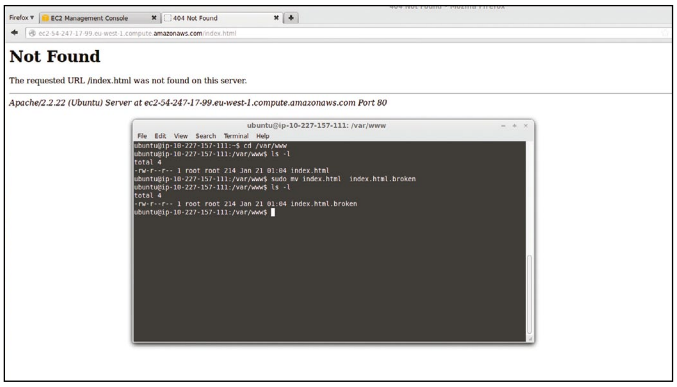

Create Snapshot

Volume: vol-581a1672

Name: Before change

Description:

1 Create snapshots
Click on Volumes in the Elastic Block Store section. There will be one volume attached to your EC2 instance. Right-click on it and select Create Snapshot. Give it a meaningful name and confirm the dialogue. You can verify that the snapshot was created in the Snapshots section (refresh may be required).

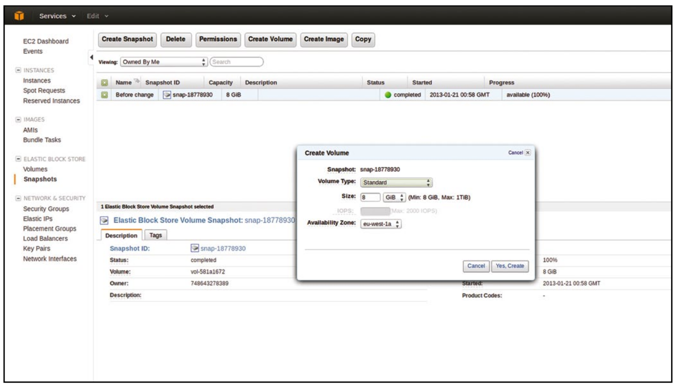

Not Found

The requested URL /index.html was not found on this server.

Apache/2.2.22 (Ubuntu) Server at ec2-54-247-17-99.eu-west-1.compute.amazonaws.com Port 80

```
ubuntu@ip-10-227-157-111:~$ cd /var/www
ubuntu@ip-10-227-157-111:/var/www$ ls -l
total 4
-rw-r--r-- 1 root root 214 Jan 21 01:04 index.html
ubuntu@ip-10-227-157-111:/var/www$ sudo mv index.html  index.html.broken
ubuntu@ip-10-227-157-111:/var/www$ ls -l
total 4
-rw-r--r-- 1 root root 214 Jan 21 01:04 index.html.broken
ubuntu@ip-10-227-157-111:/var/www$
```

2 Break it
Now you can start the instance. Note that the instance will be given a different name, so you have to modify the ssh command and the URL accordingly. Let's pretend that you've made a mistake while editing the **index.html** file in **/var/www** and rename the file as user root (don't forget **sudo**).


Create Volume

Snapshot: snap-18778930

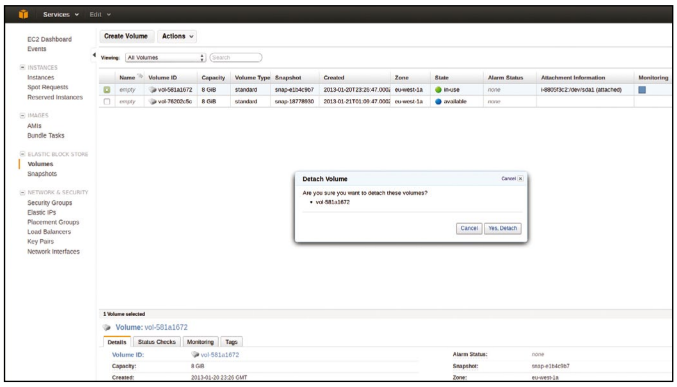
Volume Type: Standard

Size: 8 GiB (Min: 8 GiB, Max: 1TiB)

IOPS: Max: 2000 IOPS

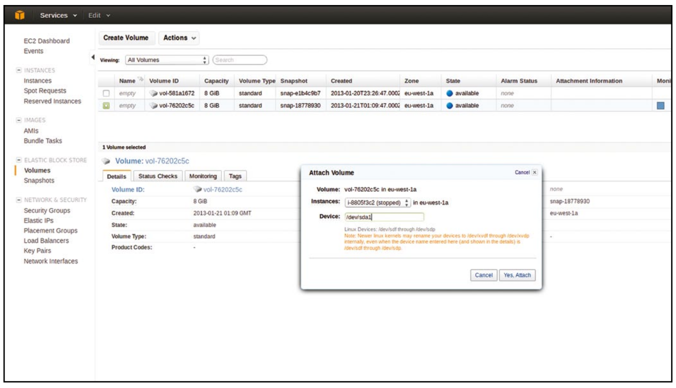
Availability Zone: eu-west-1a

3 Create volume from snapshot
Snapshots can't be used by instances directly. You need to create a volume from the snapshot first. In the snapshot section, right-click on the snapshot and select 'Create Volume from Snapshot'. Select Standard as Volume Type and make sure the availability zone is the same as the instance.


Detach Volume

Are you sure you want to detach these volumes?
- vol-581a1672

4 Detach current volume
As the root disk will be changed you will have to stop the instance volume first by right-clicking on it and selecting the Stop option. With the instance stopped, make a note of the volume attachment information (normally **/dev/sda1**) before you detach the volume from the instance.

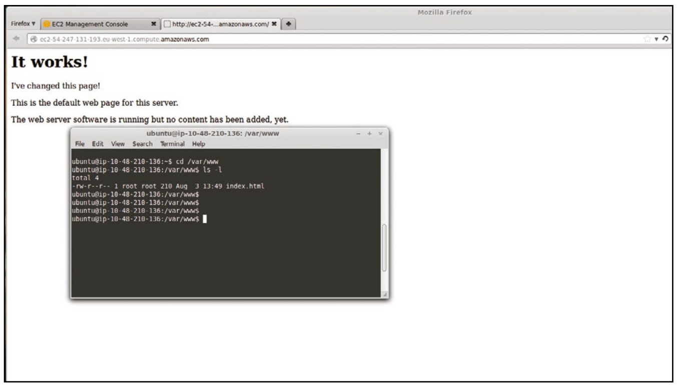

Attach Volume

Volume: vol-76202c5c in eu-west-1a

Instances: i-8805f3c2 (stopped) in eu-west-1a

Device: /dev/sda1

5 Attach a new volume
In the volumes section, attach the new volume. Select the correct instance and change the device to be the same as the old volume was (**/dev/sda1**). Now start the instance. If everything went correctly, the instance should start in a few minutes.


It works!

I've changed this page!

This is the default web page for this server.

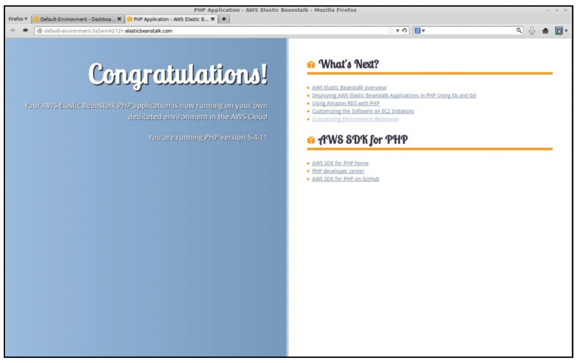
The web server software is running but no content has been added, yet.

```
ubuntu@ip-10-48-210-136:~$ cd /var/www
ubuntu@ip-10-48-210-136:/var/www$ ls -l
total 4
-rw-r--r-- 1 root root 210 Aug  3 13:49 index.html
ubuntu@ip-10-48-210-136:/var/www$
ubuntu@ip-10-48-210-136:/var/www$
ubuntu@ip-10-48-210-136:/var/www$
ubuntu@ip-10-48-210-136:/var/www$
```

6 Is it working again?
As the instance has to be restarted it will now get a new IP address and name. You will need to use the new name in a browser and **SSH** to it to check that the 'deleted' file is back and that your instance is in the original state.

» further. Hopefully, you will have noticed that the major missing LAMP component is a database. This is because the default application wizard doesn't provision any databases.

To add a relational database, go to the configuration area and scroll to the bottom. Click on the link and fill-in the required fields. Once confirmed, a new database will be created. As with all provisioning, it will take a few minutes for the database to become available.


PHP Application - AWS Elastic Beanstalk - Mozilla Firefox

default-environment-3a3wm4272h.elasticbeanstalk.com

Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 5.4.11

What's Next?

- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP

- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub

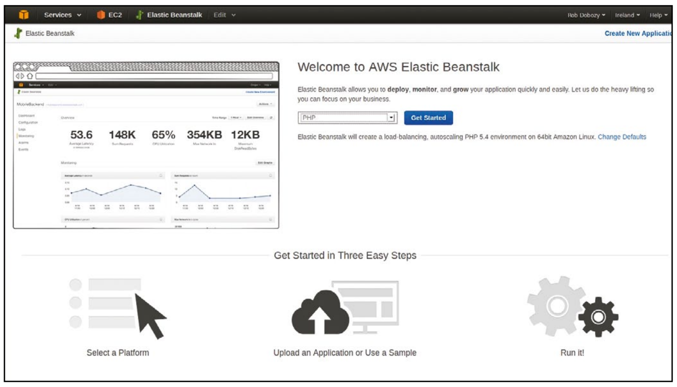
› Doesn't look much, but it's your first Elastic Beanstalk app.

You can now start deploying proper cloud based applications. We will start by downloading Wordpress from **http://wordpress.org/download**. All you need to do to install the CMS is save the ZIP file to your computer, upload it as a new version of your application in your application's Dashboard. You will need to wait for the application to go green, indicating that it's available, and then access it from a browser. You should get the Wordpress configuration wizard. Follow the wizard and provide the details of the database that you've just created. Your new Wordpress website will be up and running in no time!

All the resources that Elastic Beanstalk provisions as a part of your application will be visible from the AWS console. You can then check the load balancer, EC2 instance or RDS database with all their relevant settings and security groups. However, if you want to modify or remove your application you should do it from the separate Elastic Beanstalk console as it will modify all the resources accordingly.

You are now prepared to deploy and scale your applications in the cloud. Use this knowledge to create a cool project. Remember, the cloud's the limit! **LXF**

Set up an Elastic Beanstalk demo


Welcome to AWS Elastic Beanstalk

Elastic Beanstalk allows you to **deploy**, **monitor**, and **grow** your application quickly and easily. Let us do the heavy lifting so you can focus on your business.

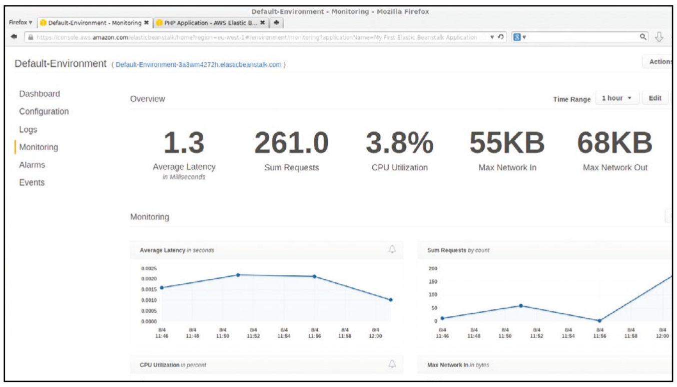
PHP | Get Started

Elastic Beanstalk will create a load-balancing, autoscaling PHP 5.4 environment on 64bit Amazon Linux. Change Defaults

Get Started in Three Easy Steps

1 Select a platform

Creating Elastic Beanstalk applications is easy. You have a choice of the following six platforms: Java, .NET, Node.js, PHP, Python and Ruby. The choice is yours, but in this tutorial we've used PHP. Press the 'Get Started' button and a sample PHP application will be created.


Default-Environment - Monitoring - Mozilla Firefox

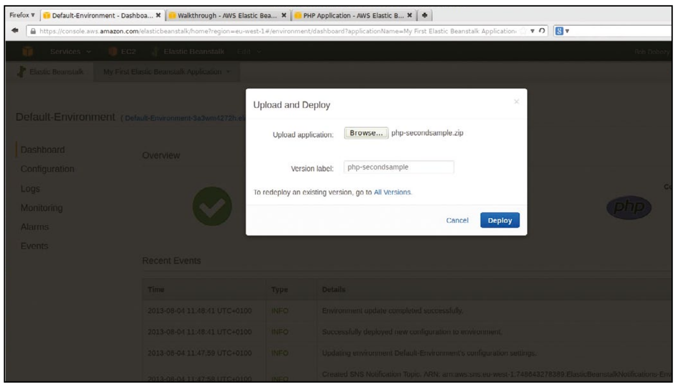
https://console.aws.amazon.com/elasticbeanstalk/home?region=eu-west-1#/environment/monitoring?applicationName=My First Elastic Beanstalk Application

Default-Environment (Default-Environment-3a3wm4272h.elasticbeanstalk.com)

1.3	261.0	3.8%	55KB	68KB
Average Latency *in Milliseconds*	Sum Requests	CPU Utilization	Max Network In	Max Network Out

2 How is it performing?

Back in the Elastic Beanstalk console, head over and click on the 'Monitoring' link. This is how AWS knows that it needs to start additional instances. You can also add more metrics to your default environment via the Edit button.


https://console.aws.amazon.com/elasticbeanstalk/home?region=eu-west-1#/environment/dashboard?applicationName=My First Elastic Beanstalk Application

Upload and Deploy

Upload application: Browse... php-secondsample.zip

Version label: php-secondsample

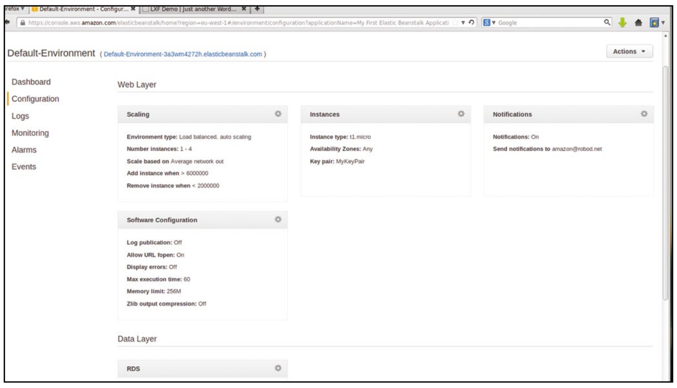
To redeploy an existing version, go to All Versions.

Cancel | Deploy

Time	Type	Details
2013-08-04 11:48:41 UTC+0100	INFO	Environment update completed successfully.
2013-08-04 11:48:41 UTC+0100	INFO	Successfully deployed new configuration to environment.
2013-08-04 11:47:59 UTC+0100	INFO	Updating environment Default-Environment's configuration settings.
2013-08-04 11:47:58 UTC+0100	INFO	Created SNS Notification Topic. ARN: arn:aws:sns:eu-west-1:748643278389:ElasticBeanstalkNotifications-Env

3 Uploading a new version

Download a second sample application from **https://elasticbeanstalk-samples-us-east-1.s3.amazonaws.com/php-secondsample.zip** and in the Dashboard section click on the 'Upload and Deploy' button.


https://console.aws.amazon.com/elasticbeanstalk/home?region=eu-west-1#/environment/configuration?applicationName=My First Elastic Beanstalk Applicati

Default-Environment (Default-Environment-3a3wm4272h.elasticbeanstalk.com)

Web Layer

Scaling
Environment type: Load balanced, auto scaling
Number instances: 1 - 4
Scale based on Average network out
Add instance when > 6000000
Remove instance when < 2000000

Instances
Instance type: t1.micro
Availability Zones: Any
Key pair: MyKeyPair

Notifications
Notifications: On
Send notifications to amazon@robod.net

Software Configuration
Log publication: Off
Allow URL fopen: On
Display errors: Off
Max execution time: 60
Memory limit: 256M
Zlib output compression: Off

Data Layer

RDS

4 The configuration

In the configuration area, add your Key Pair to the instance. This will allow you to **SSH** to it. You can also enable notifications by defining your email address. Note that some changes will result in instance being recreated with the new settings.