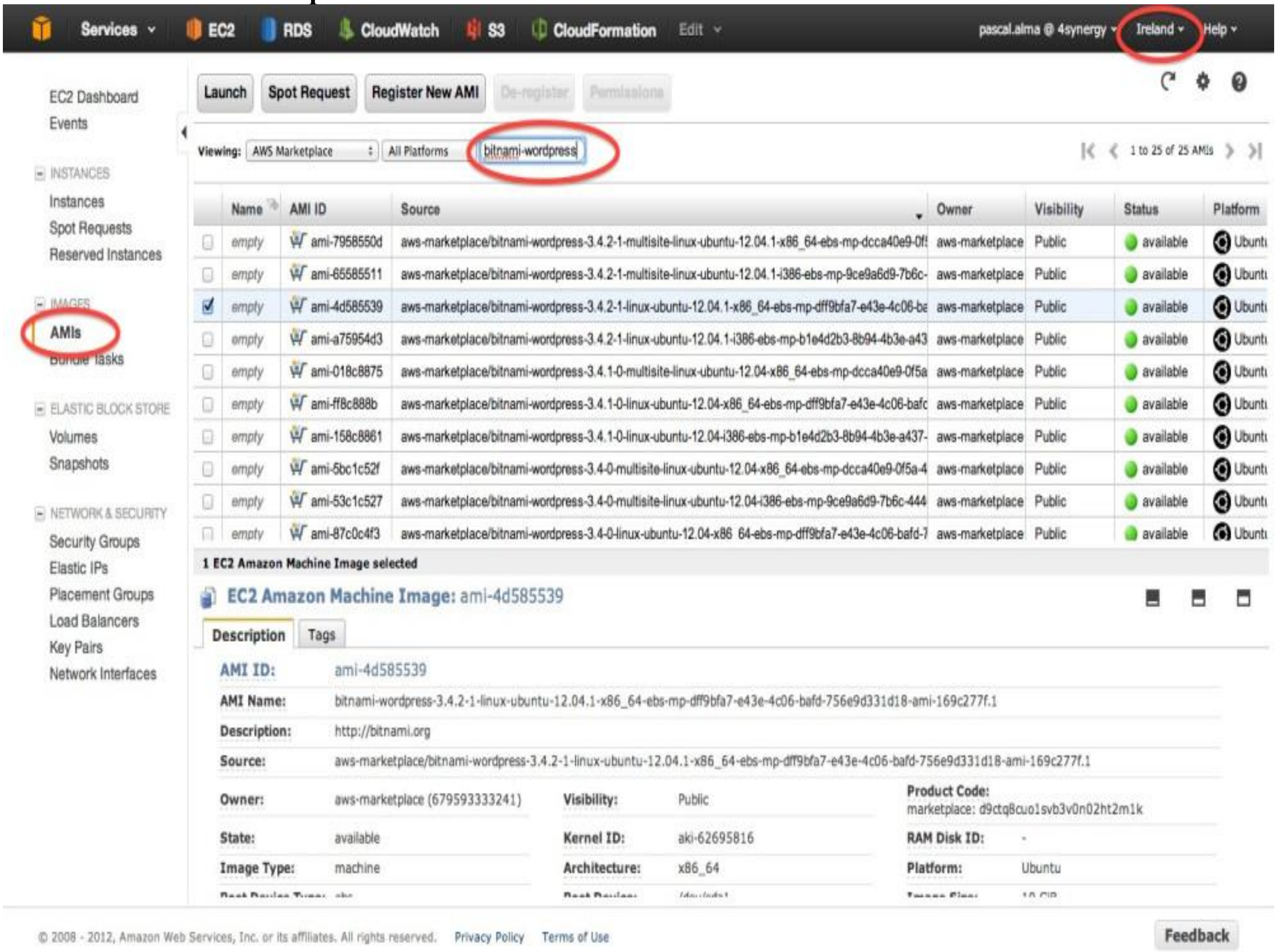
Working with Amazon Web Services (EC2)

In this demo I will install a "pre-baked" AMI with a complete WordPress setup installed. There are thousands of these AMI's available in the AWS community. For this one launch the AWS Management Console at http://aws.amazon.com and choose the EC2 tab. In the AWS console make certain you selected the correct region and select 'Images-AMIs' in the left menu and search for "bitnami-wordpress":



Search: bitnami-wordpress

	Name	AMI ID	Source	Owner	Visibility	Status	Platform
□	empty	ami-7958550d	aws-marketplace/bitnami-wordpress-3.4.2-1-multisite-linux-ubuntu-12.04.1-x86_64-ebs-mp-dcca40e9-0f	aws-marketplace	Public	available	Ubunt
□	empty	ami-65585511	aws-marketplace/bitnami-wordpress-3.4.2-1-multisite-linux-ubuntu-12.04.1-i386-ebs-mp-9ce9a6d9-7b6c-	aws-marketplace	Public	available	Ubunt
☑	empty	ami-4d585539	aws-marketplace/bitnami-wordpress-3.4.2-1-linux-ubuntu-12.04.1-x86_64-ebs-mp-dff9bfa7-e43e-4c06-ba	aws-marketplace	Public	available	Ubunt
□	empty	ami-a75954d3	aws-marketplace/bitnami-wordpress-3.4.2-1-linux-ubuntu-12.04.1-i386-ebs-mp-b1e4d2b3-8b94-4b3e-a43	aws-marketplace	Public	available	Ubunt
□	empty	ami-018c8875	aws-marketplace/bitnami-wordpress-3.4.1-0-multisite-linux-ubuntu-12.04-x86_64-ebs-mp-dcca40e9-0f5a	aws-marketplace	Public	available	Ubunt
□	empty	ami-ff8c888b	aws-marketplace/bitnami-wordpress-3.4.1-0-linux-ubuntu-12.04-x86_64-ebs-mp-dff9bfa7-e43e-4c06-bafc	aws-marketplace	Public	available	Ubunt
□	empty	ami-158c8861	aws-marketplace/bitnami-wordpress-3.4.1-0-linux-ubuntu-12.04-i386-ebs-mp-b1e4d2b3-8b94-4b3e-a437-	aws-marketplace	Public	available	Ubunt
□	empty	ami-5bc1c52f	aws-marketplace/bitnami-wordpress-3.4-0-multisite-linux-ubuntu-12.04-x86_64-ebs-mp-dcca40e9-0f5a-4	aws-marketplace	Public	available	Ubunt
□	empty	ami-53c1c527	aws-marketplace/bitnami-wordpress-3.4-0-multisite-linux-ubuntu-12.04-i386-ebs-mp-9ce9a6d9-7b6c-444	aws-marketplace	Public	available	Ubunt
□	empty	ami-87c0c4f3	aws-marketplace/bitnami-wordpress-3.4-0-linux-ubuntu-12.04-x86_64-ebs-mp-dff9bfa7-e43e-4c06-bafd-7	aws-marketplace	Public	available	Ubunt

1 EC2 Amazon Machine Image selected

EC2 Amazon Machine Image: ami-4d585539

AMI ID: ami-4d585539

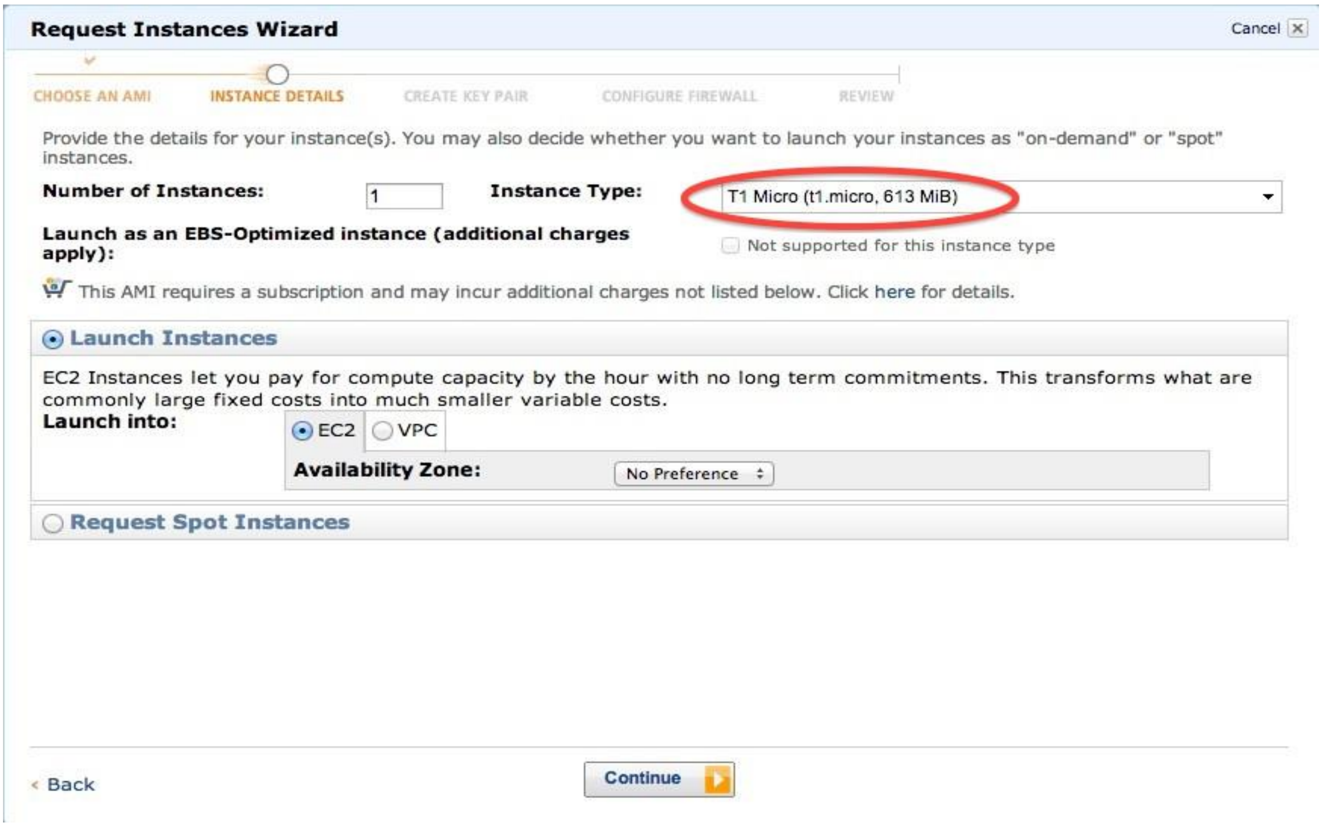
AMI Name: bitnami-wordpress-3.4.2-1-linux-ubuntu-12.04.1-x86_64-ebs-mp-dff9bfa7-e43e-4c06-bafd-756e9d331d18-ami-169c277f.1

Description: http://bitnami.org

Source: aws-marketplace/bitnami-wordpress-3.4.2-1-linux-ubuntu-12.04.1-x86_64-ebs-mp-dff9bfa7-e43e-4c06-bafd-756e9d331d18-ami-169c277f.1

Owner:	aws-marketplace (679593333241)	Visibility:	Public	Product Code:	marketplace: d9ctq8cuo1svb3v0n02ht2m1k
State:	available	Kernel ID:	aki-62695816	RAM Disk ID:	-
Image Type:	machine	Architecture:	x86_64	Platform:	Ubuntu
Root Device Type:	ebs	Root Device:	/dev/sda1	Image Size:	10 GiB

Select the AMI of your choice (I have 64-bit Ubuntu running WordPress 3.4.2.1) and click 'Launch'. Now simply follow the steps of the wizard (although there is a lot to tell about almost every option in the wizard I simply go for the most default installation). At the *Instance Details* step make sure you select a Micro instance (to cut down the cost of having this demo running):



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** T1 Micro (t1.micro, 613 MiB)

Launch as an EBS-Optimized instance (additional charges apply): □ Not supported for this instance type

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● **Launch Instances**

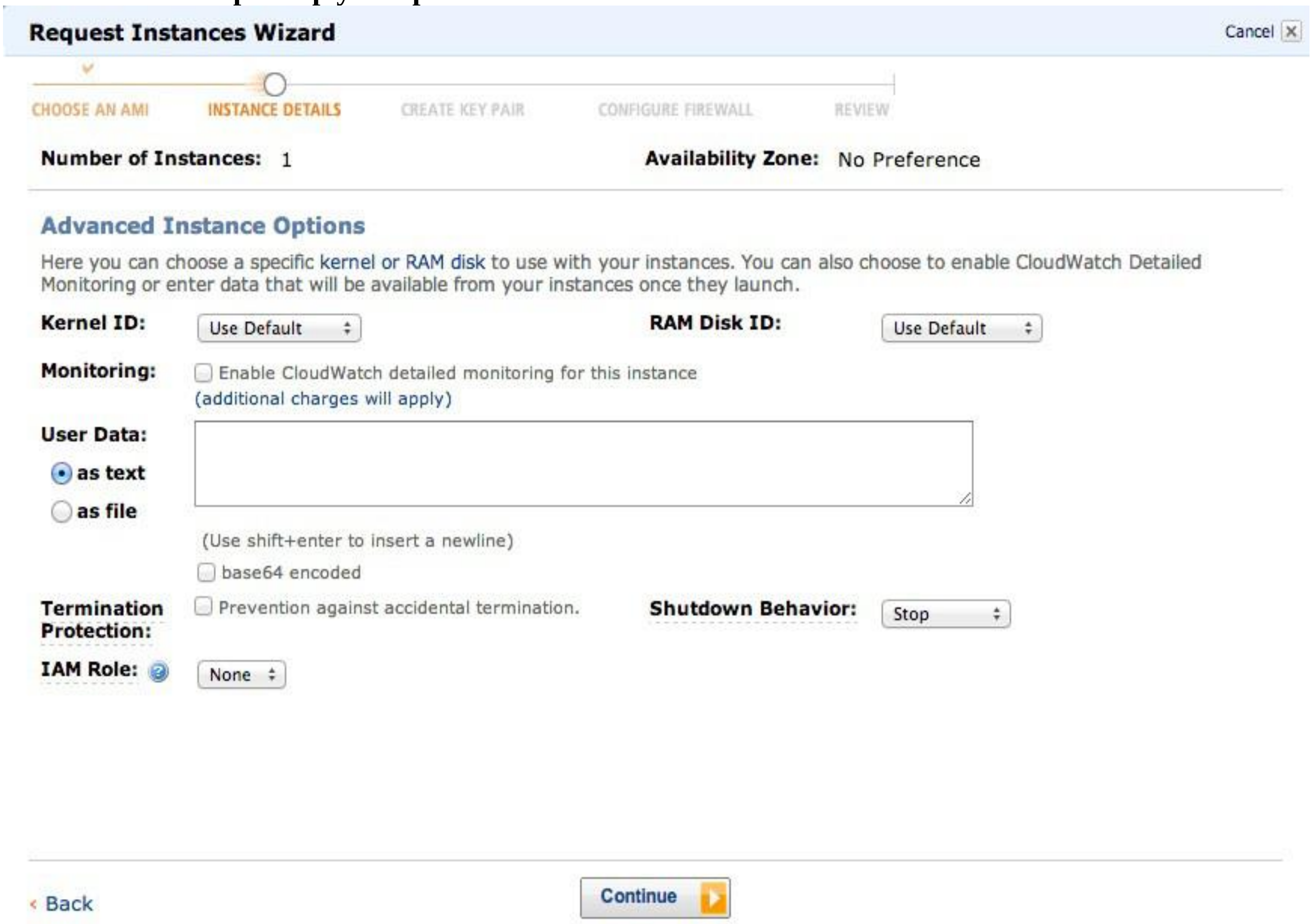
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ● EC2 ○ VPC

Availability Zone: No Preference

○ **Request Spot Instances**

In the next two steps simply accept all defaults:



Number of Instances: 1 **Availability Zone:** No Preference

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default **RAM Disk ID:** Use Default

Monitoring: □ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:

● **as text**

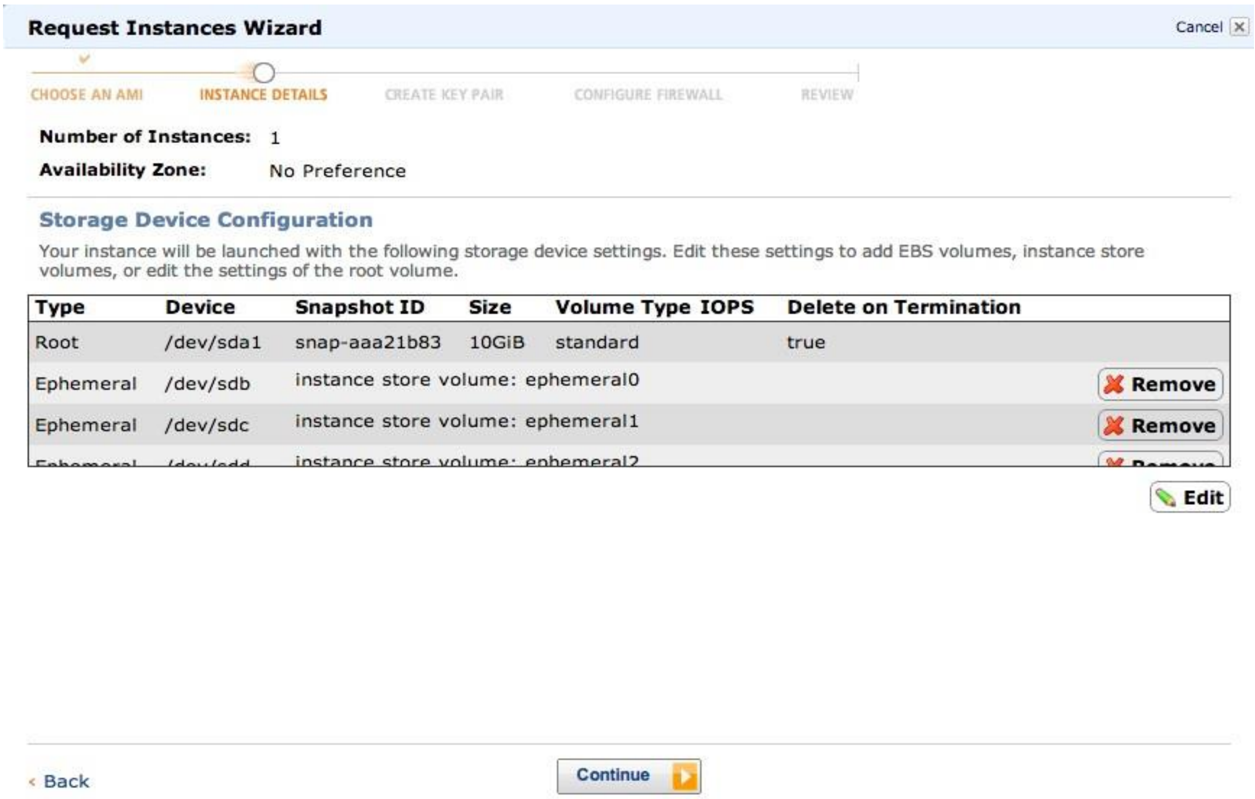
○ **as file**

(Use shift+enter to insert a newline)

□ base64 encoded

Termination Protection: □ Prevention against accidental termination. **Shutdown Behavior:** Stop

IAM Role: None



Request Instances Wizard

Number of Instances: 1

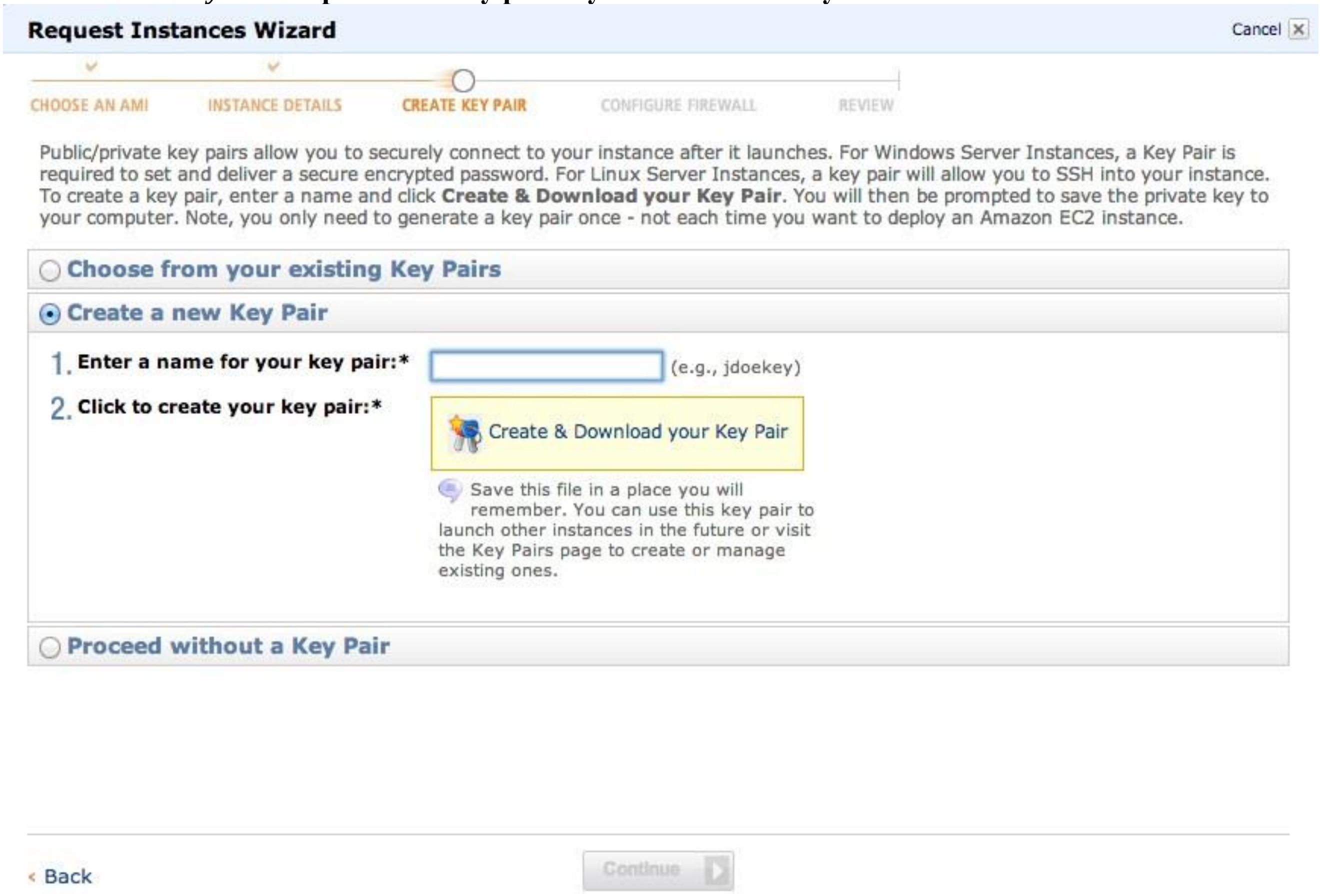
Availability Zone: No Preference

Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination	
Root	/dev/sda1	snap-aaa21b83	10GiB	standard		true	
Ephemeral	/dev/sdb	instance store volume: ephemeral0					Remove
Ephemeral	/dev/sdc	instance store volume: ephemeral1					Remove
Ephemeral	/dev/sdd	instance store volume: ephemeral2					Remove

At the *Create Key Pair* step create a key pair if you haven’t already done so:



Request Instances Wizard

Public/private key pairs allow you to securely connect to your instance after it launches. For Windows Server Instances, a Key Pair is required to set and deliver a secure encrypted password. For Linux Server Instances, a key pair will allow you to SSH into your instance. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

○ Choose from your existing Key Pairs

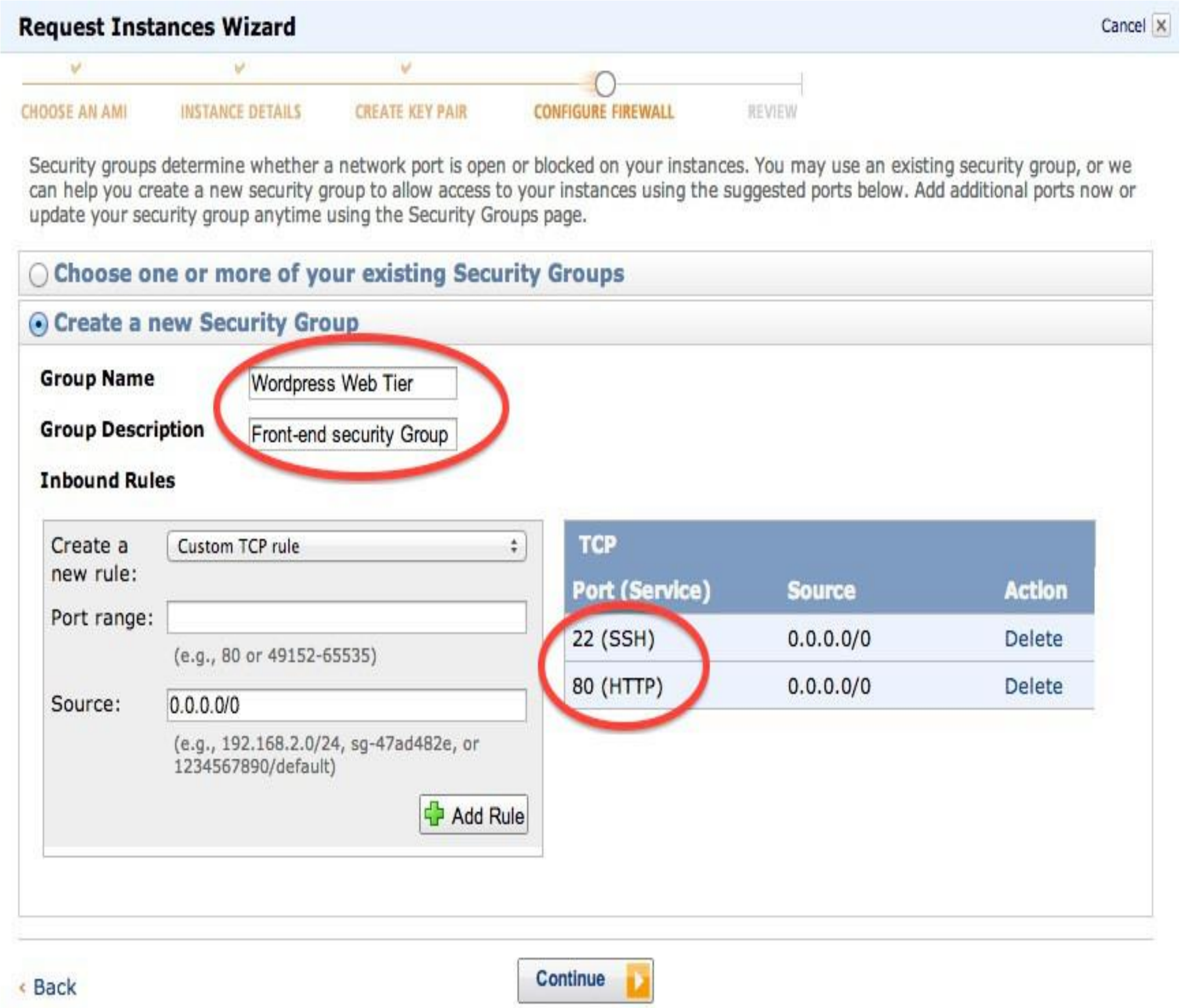
● Create a new Key Pair

1. **Enter a name for your key pair:*** (e.g., jdoekey)
2. **Click to create your key pair:***

Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

○ Proceed without a Key Pair

The next step *Configure Firewall* create a Security Group so we are able to access our Linux instance from the outside world by HTTP and SSH:



Request Instances Wizard

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

○ Choose one or more of your existing Security Groups

● Create a new Security Group

Group Name: Wordpress Web Tier

Group Description: Front-end security Group

Inbound Rules

Create a new rule: Custom TCP rule

Port range:

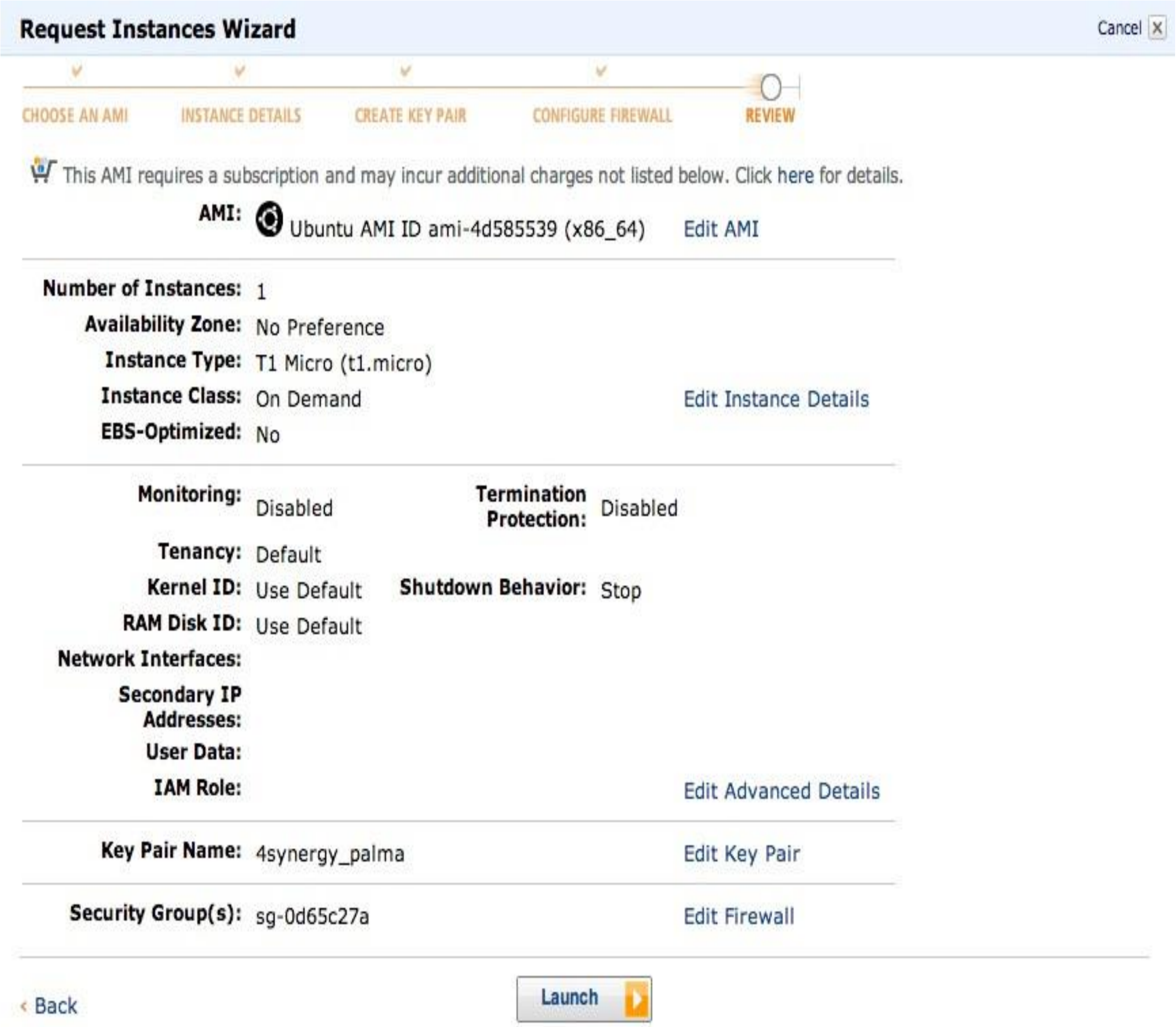
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete

Finally review all your settings and if it is okay then launch the instance:



This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

AMI: Ubuntu AMI ID ami-4d585539 (x86_64) Edit AMI

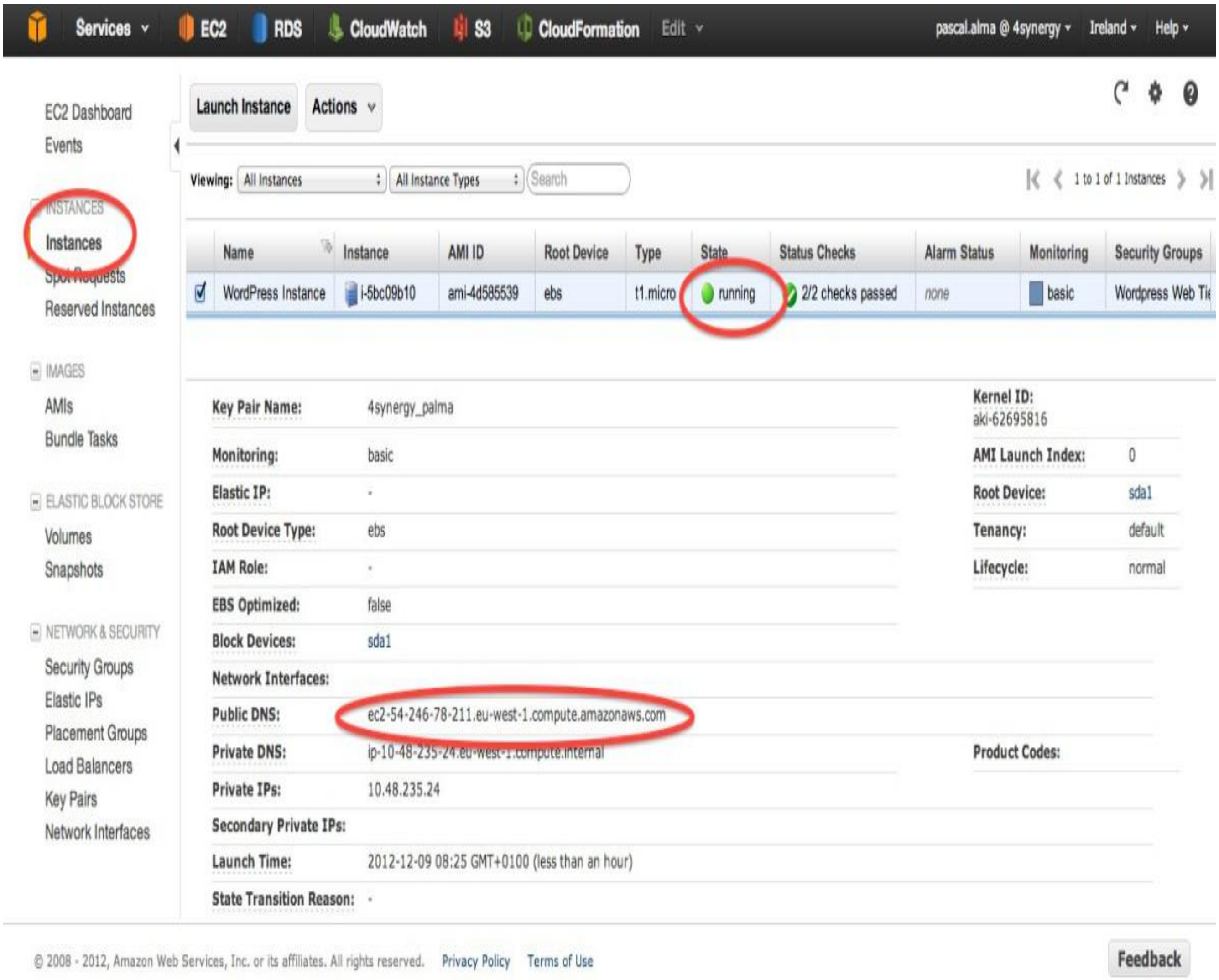
Number of Instances: 1
Availability Zone: No Preference
Instance Type: T1 Micro (t1.micro)
Instance Class: On Demand Edit Instance Details
EBS-Optimized: No

Monitoring: Disabled Termination Protection: Disabled
Tenancy: Default
Kernel ID: Use Default Shutdown Behavior: Stop
RAM Disk ID: Use Default
Network Interfaces:
Secondary IP Addresses:
User Data:
IAM Role: Edit Advanced Details

Key Pair Name: 4synergy_palma Edit Key Pair

Security Group(s): sg-0d65c27a Edit Firewall

After a few seconds you should be able to see your new instance in the Instances overview in the AWS Management Console:



Viewing: All Instances | All Instance Types

	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Security Groups
☑	WordPress Instance	i-5bc09b10	ami-4d585539	ebs	t1.micro	● running	✓ 2/2 checks passed	none	basic	Wordpress Web Ti

Key Pair Name:	4synergy_palma	Kernel ID:	aki-62695816
Monitoring:	basic	AMI Launch Index:	0
Elastic IP:	-	Root Device:	sda1
Root Device Type:	ebs	Tenancy:	default
IAM Role:	-	Lifecycle:	normal
EBS Optimized:	false		
Block Devices:	sda1		
Network Interfaces:			
Public DNS:	ec2-54-246-78-211.eu-west-1.compute.amazonaws.com		
Private DNS:	ip-10-48-235-24.eu-west-1.compute.internal	Product Codes:	
Private IPs:	10.48.235.24		
Secondary Private IPs:			
Launch Time:	2012-12-09 08:25 GMT+0100 (less than an hour)		
State Transition Reason:	-		

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

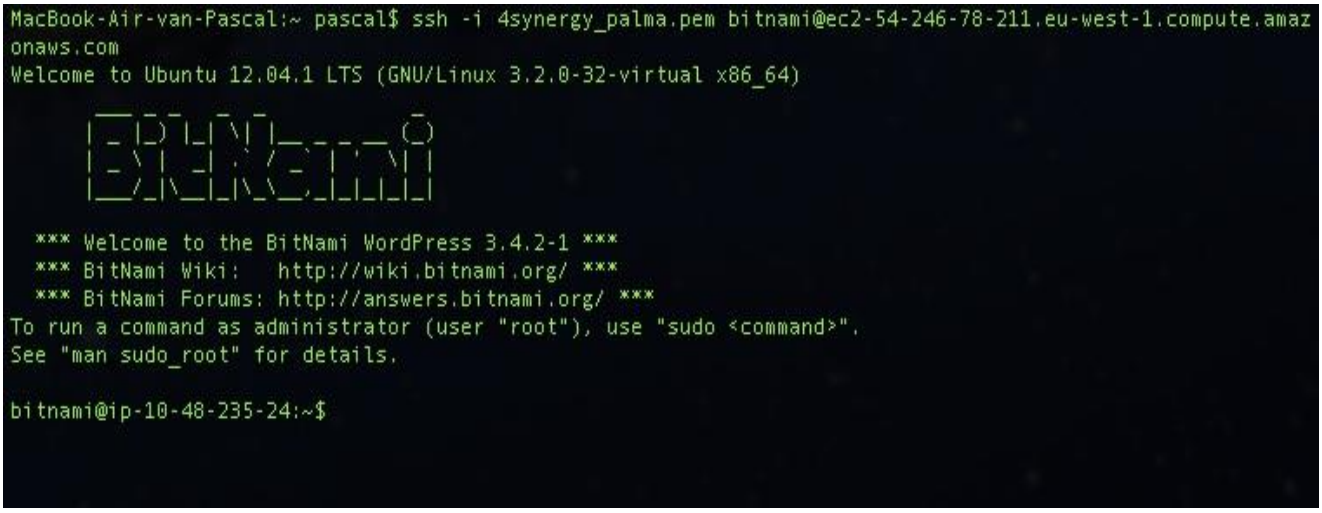
Please pay specific attention to the public DNS that is assigned to the instance. As you might be aware this DNS will change after you stopped and started this instance. So I use it now to show how to connect to the instance via SSH but you should not use this DNS as if it was a fixed IP/DNS. To work around this issue Amazon supplies Elastic IP which I show in another post. To connect with SSH to this instance open up a Terminal on your Mac and type the following commands.

First I had to narrow the permissions on the downloaded pem file for which I used:

```
chmod u=r,go= 4synergy_palma.pem
```

Then I was able to connect to the Bitnami instance via SSH with:

`ssh -i 4synergy_palma.pem bitnami@ec2-54-246-78-211.eu-west-1.compute.amazonaws.com` as you can see here:

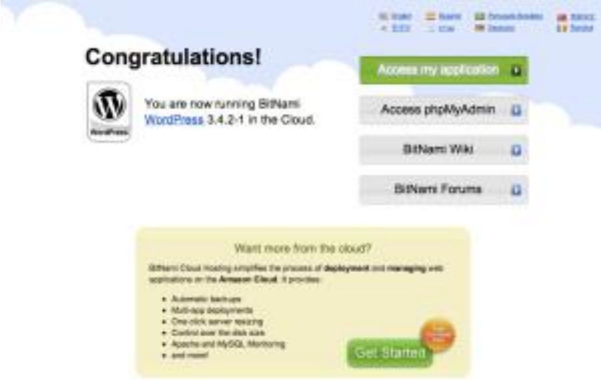


```
MacBook-Air-van-Pascal:~ pascal$ ssh -i 4synergy_palma.pem bitnami@ec2-54-246-78-211.eu-west-1.compute.amaz
onaws.com
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-32-virtual x86_64)
       ___ _ _   _  _
      | _ |_) |_| \| |__ _ _ __  (_)
      | _ \ |  _| .` / _` | '  \| |
      |___/_|\__|_|\_\__,_|_|_|_|_|

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

bitnami@ip-10-48-235-24:~$
```

And of course we can enter our WordPress by opening the home URL in a web browser:



Congratulations!

You are now running BitNami WordPress 3.4.2-1 in the Cloud.

Want more from the cloud?

BitNami Cloud Hosting simplifies the process of **deployment** and **managing** web applications on the **Amazon Cloud**. It provides:

- Automatic back-ups
- Multi-app deployments
- One-click server resizing
- Control over the disk size
- Apache and MySQL Monitoring
- and more!

In the next posts I will extend this example to make it a more robust and scalable solution.

🙂