CS 4001/7001 Cloud Computing
Spring 2015
Lab # 3 - Platform/Application Provisioning and Auto Scaling Adaptation

Dr. Prasad Calyam & Ronny Bazan Antequera (Contact: calyamp@missouri.edu)

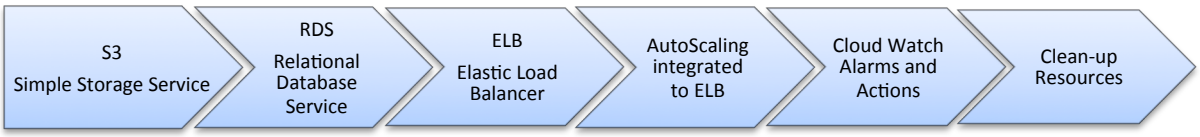
1. Purpose of the Lab

Launch new instances through Images taken from your current instance state, include them to a load balancer configuration, and apply CloudWatch alarms for automatically scaling up and scaling down using AutoScaling service adaptations based on usage load demands.

2. References to guide Lab work

- *Elastic Load Balancing* – http://aws.amazon.com/documentation/elasticloadbalancing
- *Auto Scaling* – http://aws.amazon.com/documentation/autoscaling
- *Cloud Watch* – http://aws.amazon.com/documentation/cloudwatch
- *Chapter 2,3,5,6, Programming Amazon EC2 – Jurg van Vliet & Flavia Paganelli*
- Address book with PHP and MySQL: http://php.about.com/od/finishedphp1/ss/address_book.htm
- MySQL Manual – Posted in Blackboard

3. Lab Steps and output collection guidelines

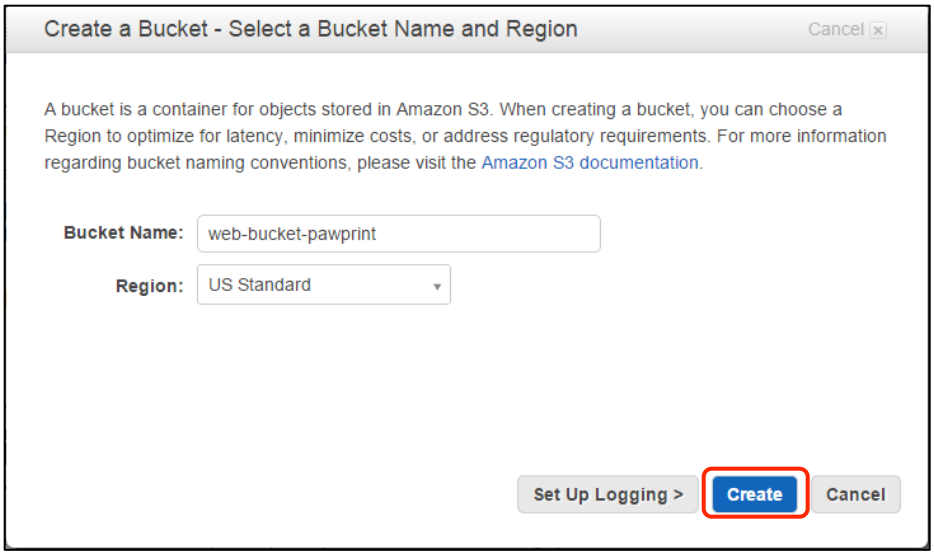


In this Lab, you will learn about AWS services that allow you to store information in a public cloud (S3), launch Relational Databases (RDS) and access them from your instances. You will apply concepts of load balancing by working with multiple instances (ELB), install AutoScaling tools for scaling up and scaling down your infrastructure and integrate them to the ELB for an efficient use of your resources. Integration will involve creation of policies that will be triggered according to the monitoring of usage load of the resources (CloudWatch) i.e., demand of users. Finally, you will clean-up the configuration and instances created in this Lab.

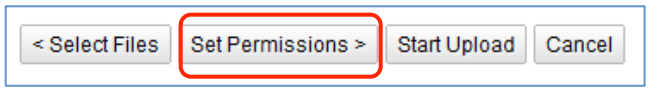
3.1 Using AWS Simple Storage Service – S3

- In EC2 service, start the instance created in Lab 2. (Right click over the instance and select 'Start')

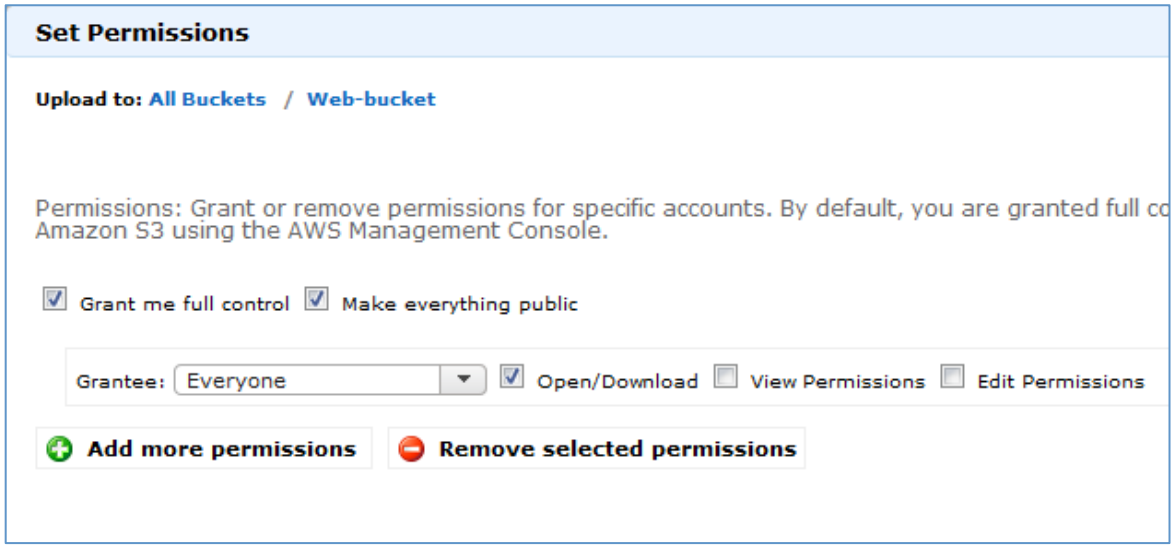
- 3.1.1 In services (top left part of your screen) select S3 service and click on 'Create Bucket' button. Add 'web-bucket-pawprint'. Select 'US Standard' in Region name.



- Open your bucket content and click on 'Upload' option. Then on 'Add Files', select 'VIMAN Lab' logo that is posted in Blackboard. Click on 'Set Details' and then 'Set Permissions'

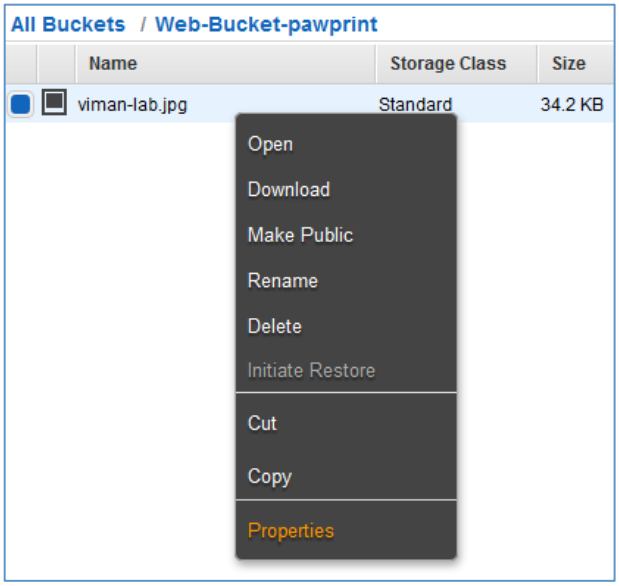


- Set up the following configuration and click on 'Start upload'



- Copy the picture link that is located in properties similar to:

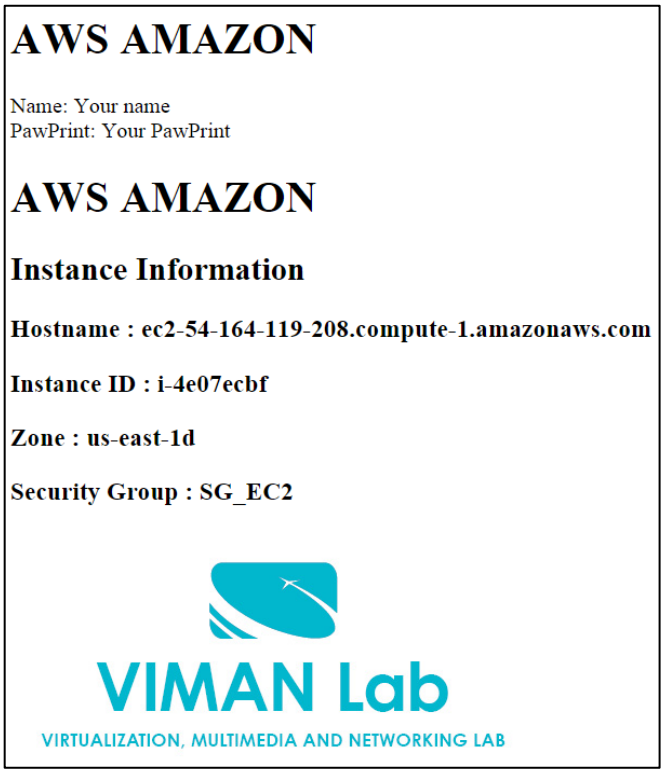
  Link: https://s3.amazonaws.com/Web-Bucket-pawprint/viman-lab.jpg



- Edit your index.php that you created in Lab2 and add the following line; use your link as a source

```
<img src="https://s3.amazonaws.com/web-bucket-pawprint/viman-lab.jpg">
```

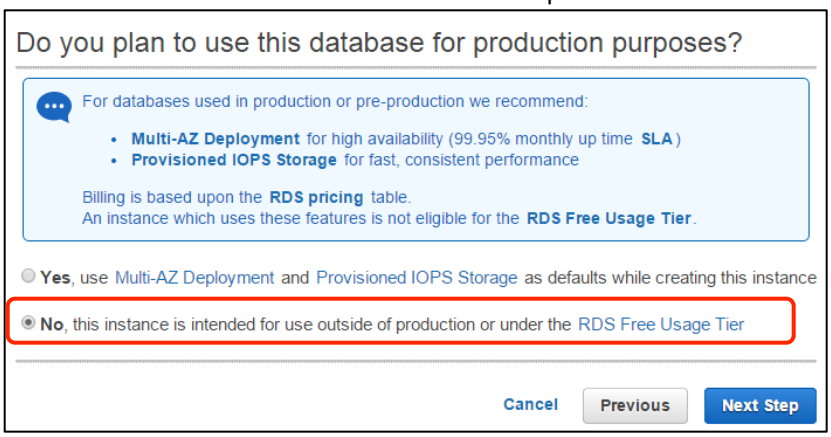
- Now using your running instance [public dns] in a web browser you will be able to see your uploaded 'VIMAN Lab' logo that is stored in Amazon S3.



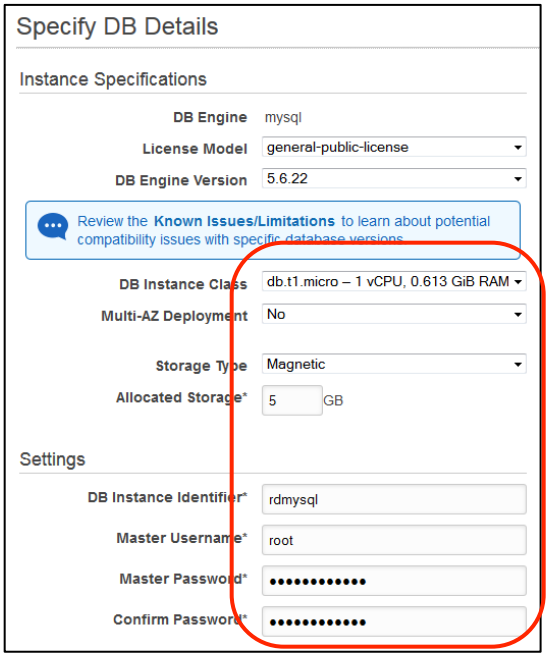
3.2 Using AWS Relational Database Service - RDS

3.2.1 Select RDS service and click on 'Get Started Now'. In 'Engine selection' tab choose MySQL database

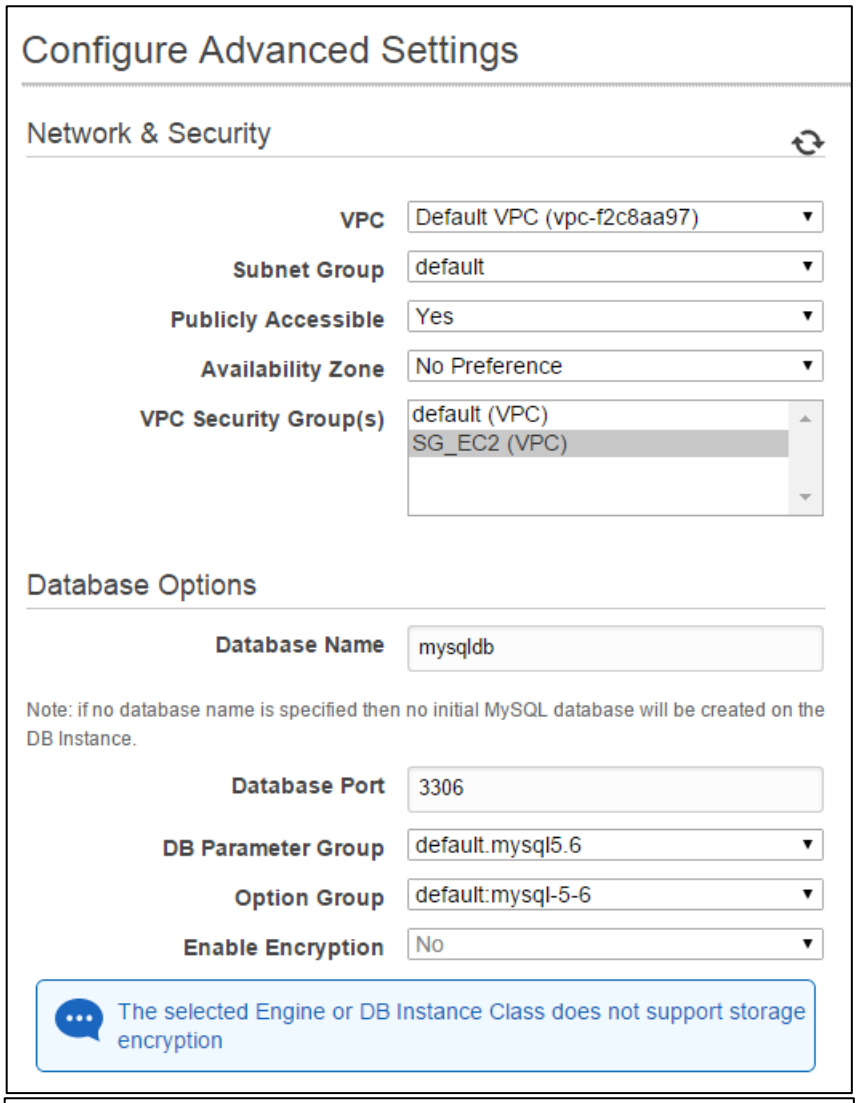
- Select 'No' in the next screen and continue with the 'Next Step'.

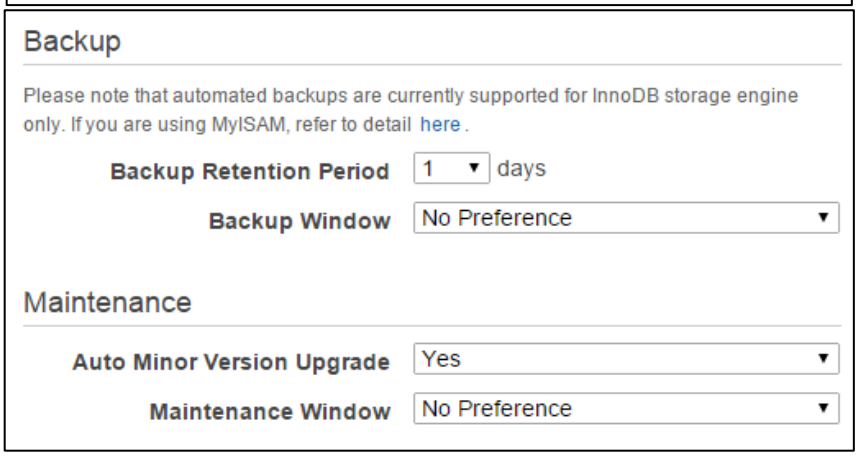


- In 'DB Instance Details' make sure to set up the following configuration, use 'rootpassword' as a password and click next.

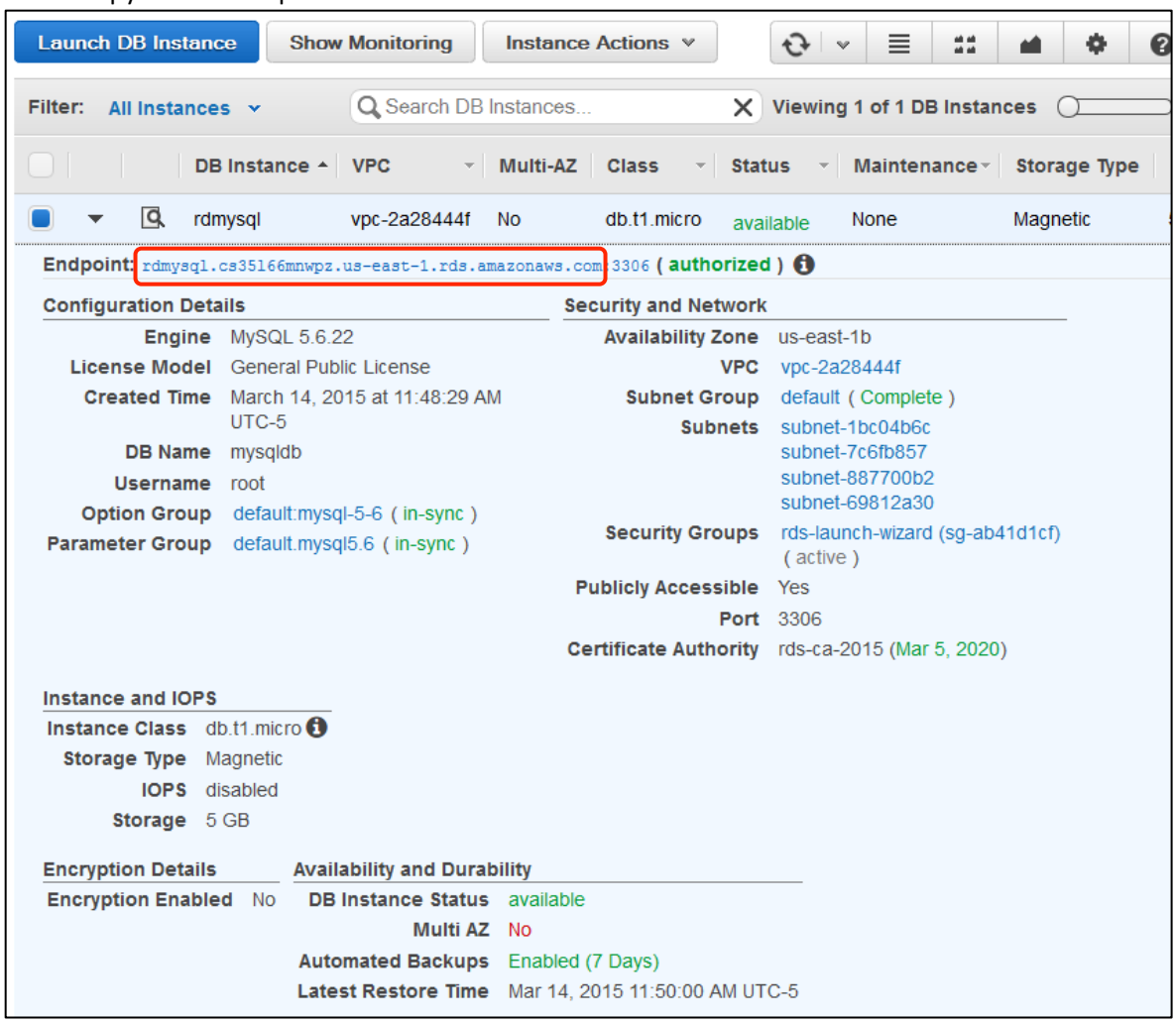


- Add ‘mysqldb’ to ‘Database Name’ ensure rest of the settings are as in the following image and click ‘Launch DB Instance’.

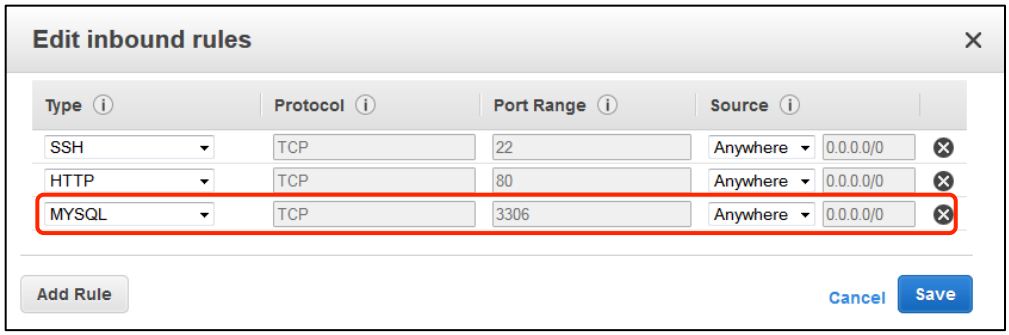




- Note that it will take a few minutes to change status from 'creating' to 'backing-up' and finally 'available' status. Copy the 'EndPoint' of the database by displaying the details of your database. Omit copy the ':3306' port number. You will use this information later.



- To access this DB instance, you will need to add a new 'MYSQL' inbound rule in the security group SG_EC2.



3.2.2 Connecting to the database from your EC2 instance. (Note: Make sure you are familiar with the MySQL tutorial referenced in Section 2 of this Lab)

mysql -u root -p --database=mysqldb --host=[EndPoint]

```
[ec2-user@ip-172-31-54-49 html]$ mysql -u root -p --database=mysqldb --host=rdmysql.cs35l66mnwpz.us-east-1.rds.
amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 5.6.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3.2.3 Create a table and insert information into the database. There are many options to do that (creating a file, input line by line, input all in one line…)

```
CREATE TABLE address (
        id INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(35),
        phone VARCHAR(20),
        email VARCHAR(30));
INSERT INTO address (name, phone, email)
VALUES
        ("George", "1112223333", "george@abc.com"),
        ("Jhon", "2223334444", "jhon@abc.com"),
        ("Charles", "3334445555", "charles@abc.com");
```

3.2.4 Copy the provided file 'addressbook.php' into your instance under '/var/www/html'. Note that the 'html' folder is protected, you might want to copy the file to /home/ec2-user directory first.

e.g. in Linux/Unix #scp –i key-ec2.pem addressbook.php ec2-user@[public dns]:/home/ec2-user
e.g. in Windows c:\>pscp –i key-ec2-putty.ppk addressbook.php ec2-user@[public dns]: /home/ec2-user

```
root@ubuntu:/opt/aws# scp -i key-ec2.pem addressbook.php ec2-user@ec2-54-164-119
-208.compute-1.amazonaws.com:/home/ec2-user/
addressbook.php                               100% 3308     3.2KB/s   00:00
```

```
[ec2-user@ip-172-31-48-192 ~]$ sudo cp /home/ec2-user/addressbook.php /var/www/html/
[ec2-user@ip-172-31-48-192 ~]$
```

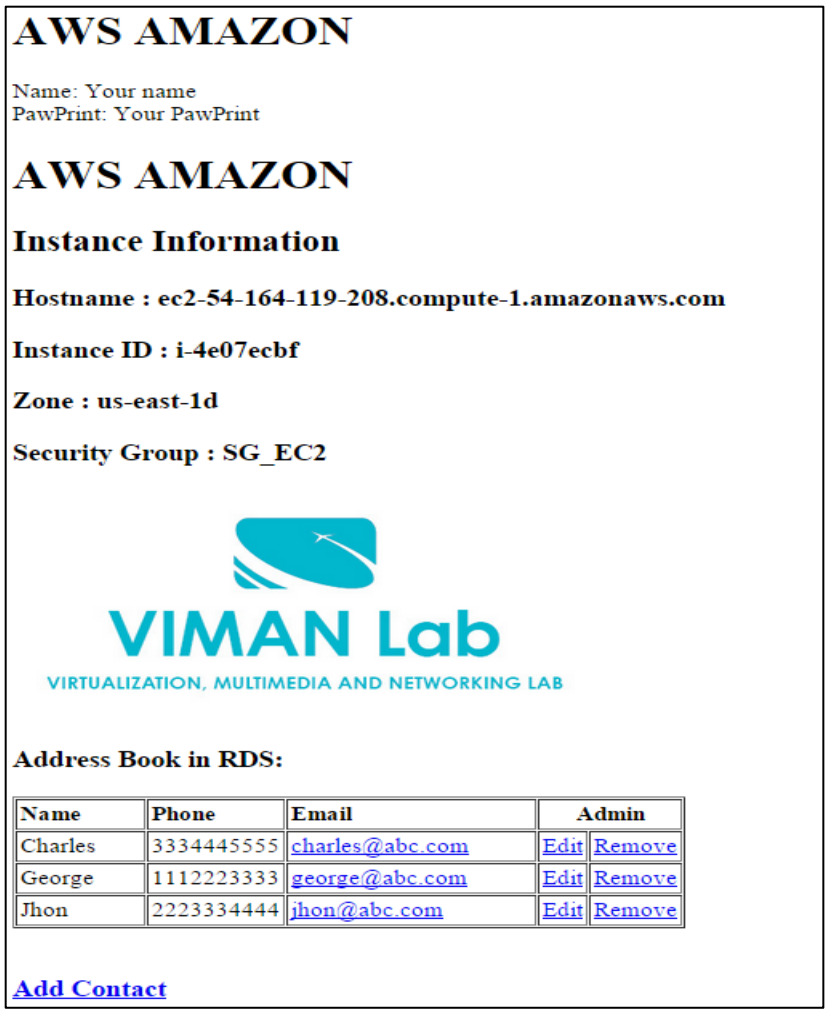
3.2.5 Create a new file called /var/www/html/mysqlinfo.php, populate it with the provided information. (Use the complete end-point including the port number).

```
<?php
 $RDS_URL="[EndPoint]";
 $RDS_DB="mysqldb";
 $RDS_user="root";
 $RDS_pwd="rootpassword";
?>
```

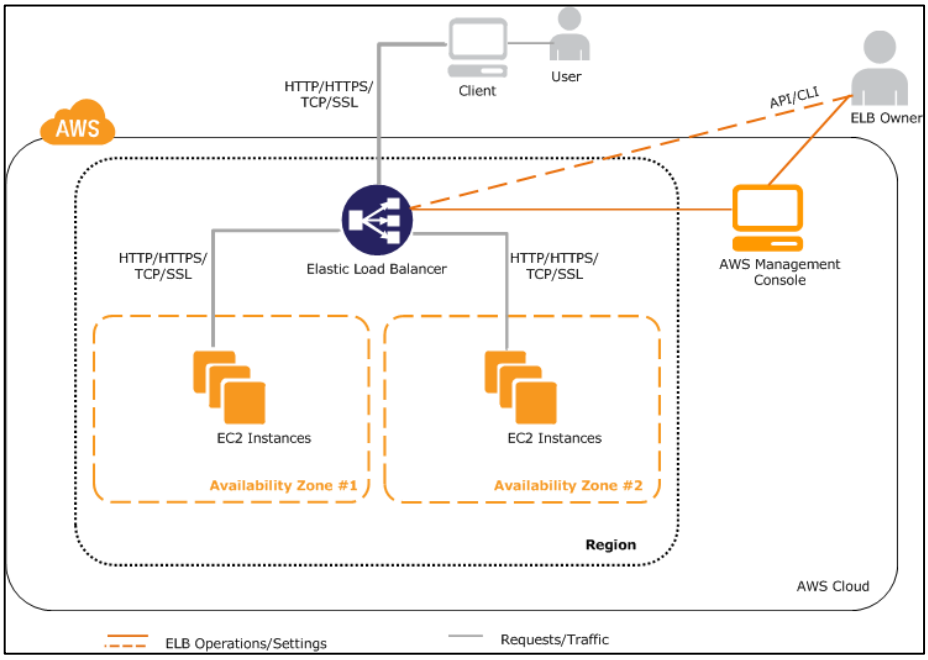
3.2.6 Add the next code to your index.php

```
<?php include 'mysqlinfo.php';
    if($RDS_URL != "") {
        include 'addressbook.php';
    }
?>
```

3.2.7 Now using your running instance [public dns] in a web browser you will be able to access your database with your RDS configuration as shown in the following figure. You have successfully created a simple web application in a public cloud !



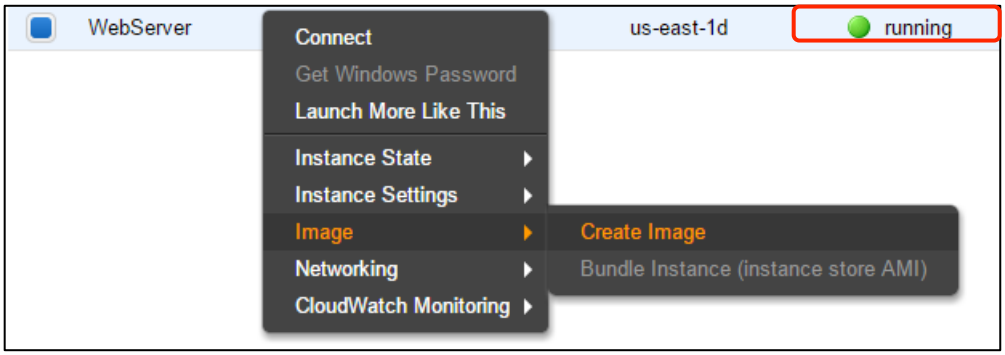
3.3 Using AWS Elastic Load Balancer (ELB)



Elastic Load Balancer components and architecture

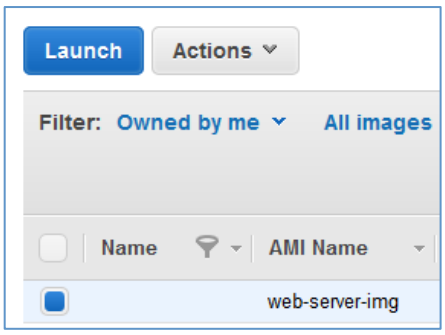
Previous figure shows the different components that ELB uses to manage several instances with the same application that can be accessed through a common DNS entry point.

3.3.1. In EC2, Create an Image (EBS AMI) from your running instance; Use Image Name: web-server-img. In the case that you accidentally terminate an instance, you can use this image to deploy a new one with all the configuration done before to take the image.

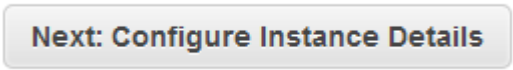


- Verify that the Image status is available in 'AMIs' under 'IMAGES' option in the left panel.

- Select the image and click on the Launch Button.

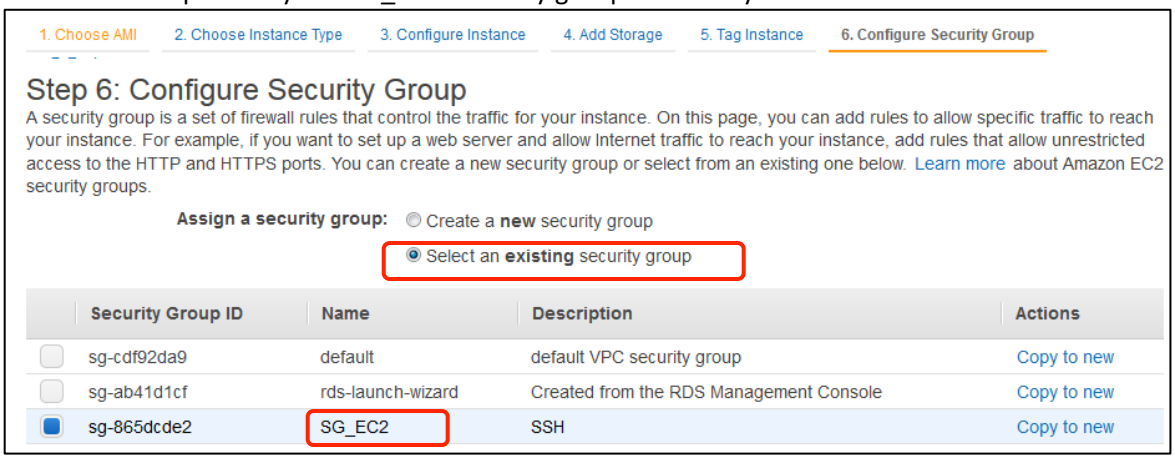


- Click 'Next: Configure Instance Details'

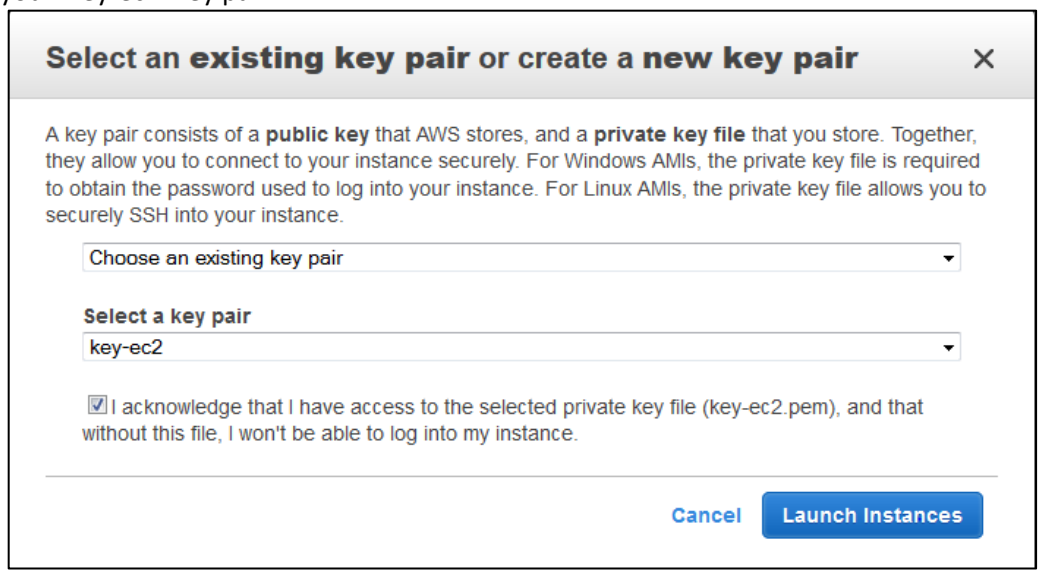


- Click on 'Next: Add Storage' and on 'Next: Tag Instance', add the value name 'WebServerFromImage' and click on "Next: Configure Security Group'

- In the next step select your 'SG_EC2' security group and finally click on 'Launch'.

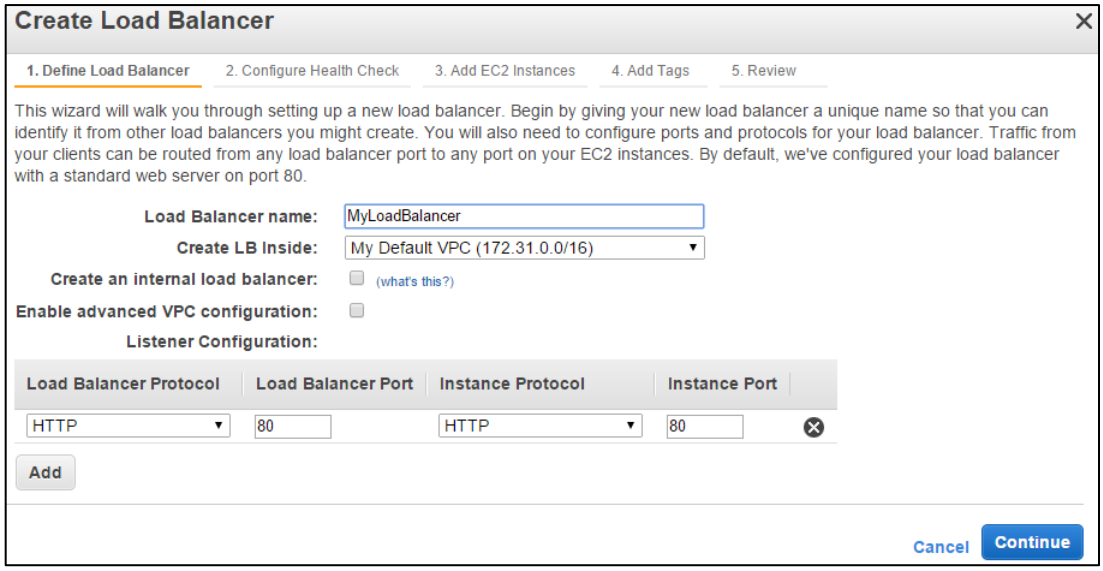


- Select your 'key-ec2' key pair.

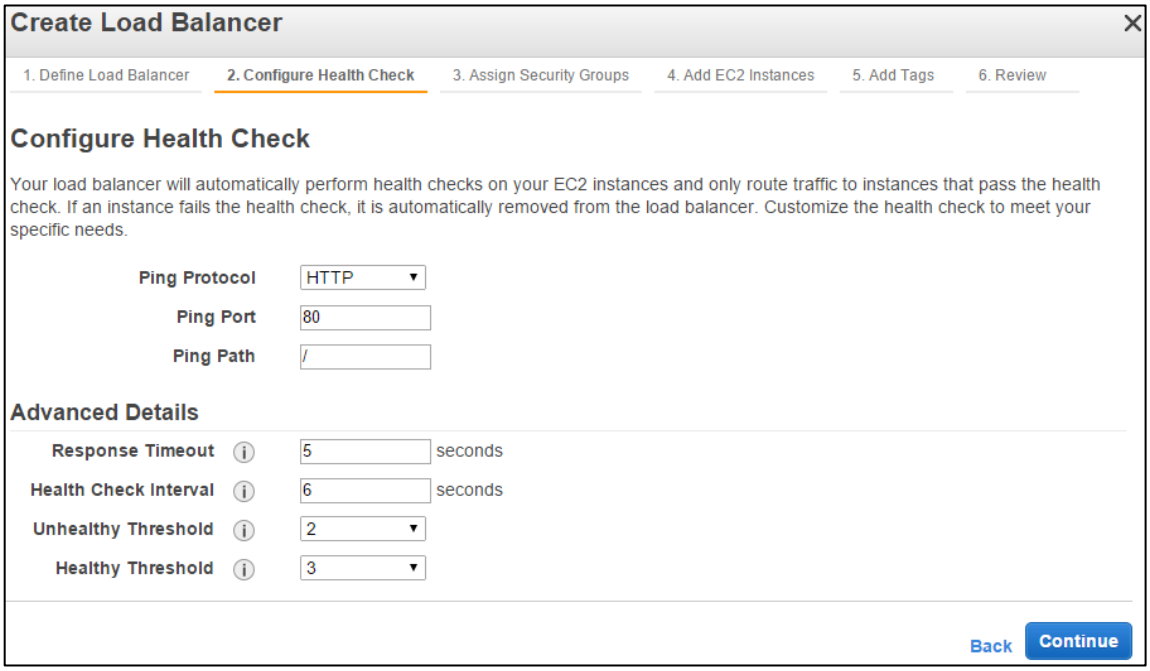


- Once the new instance is running, copy the new [public dns] to a browser and verify that the new instance has different Metadata information than the original (Instance ID and Hostname). Notice that both instances are accessing the same RDS database.

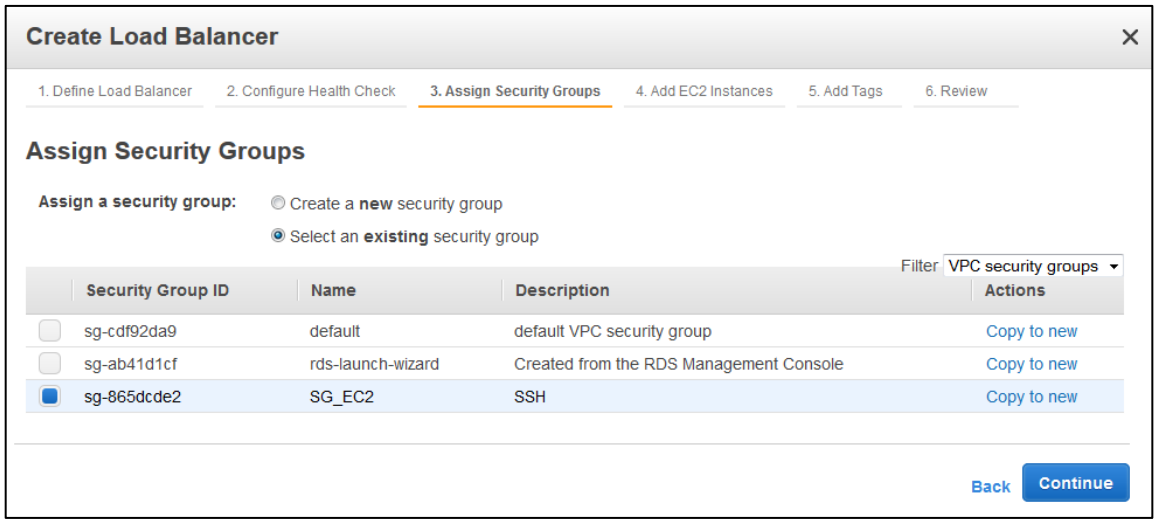
3.3.2 In the Left panel of the EC2 service, create a ELB in 'Load Balancers' and add 'MyLoadBalancer' to the name field



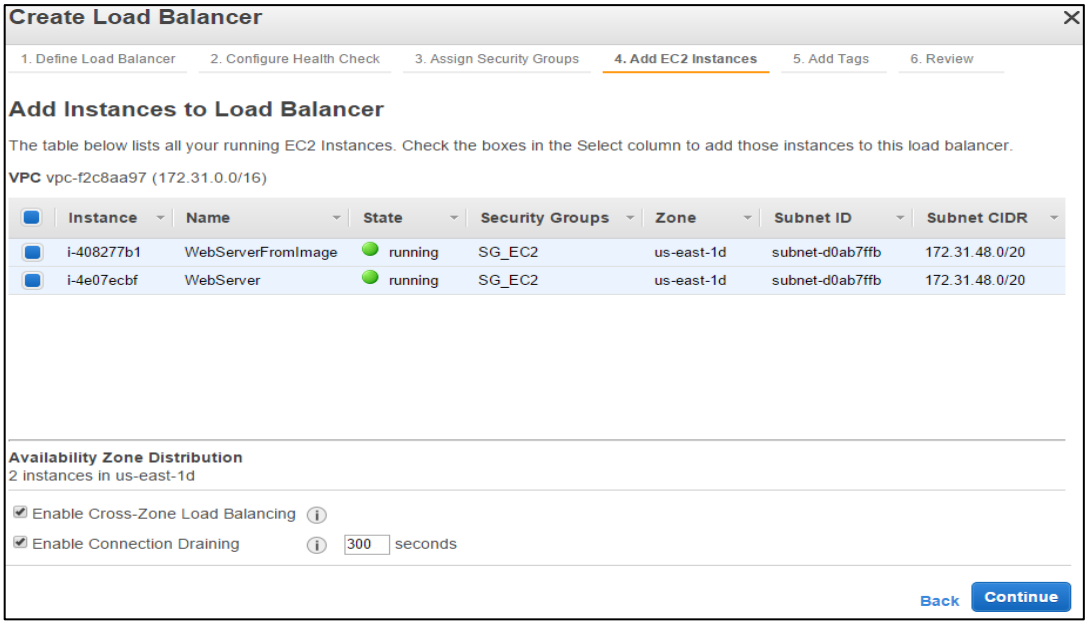
3.3.3 Setup the following values for your 'New Load Balancer'



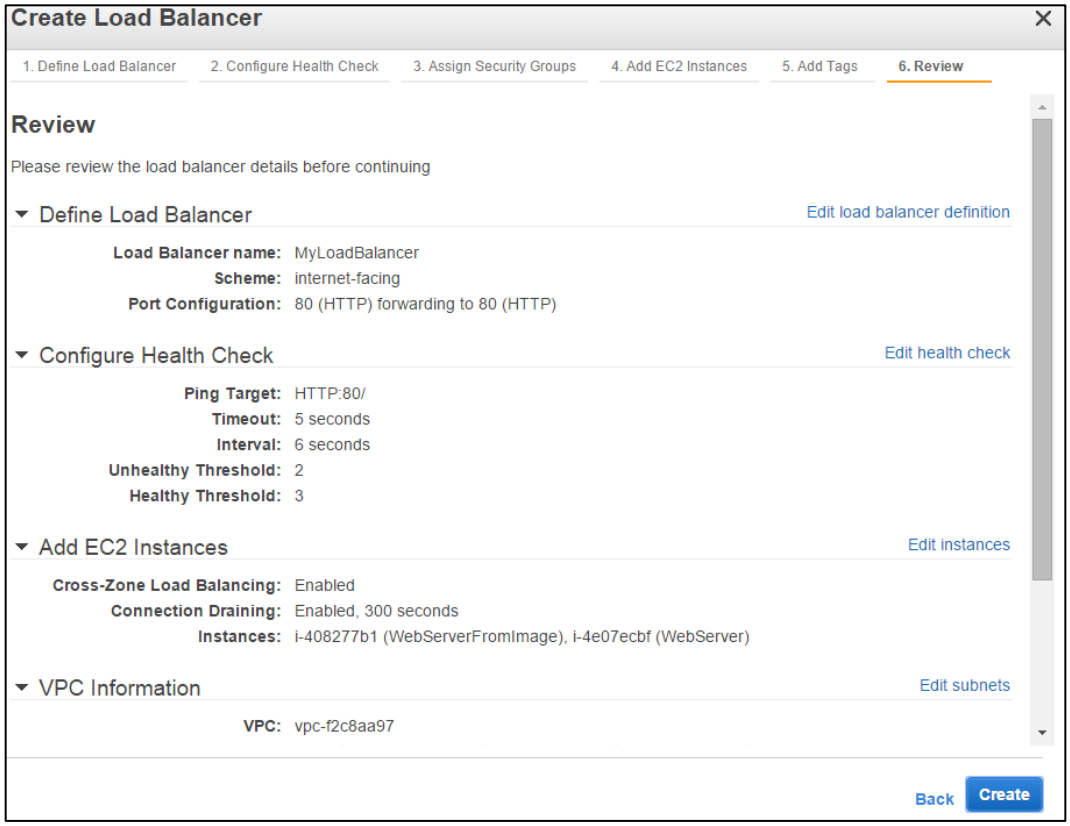
3.3.4 Select the 'SG_EC2' security group



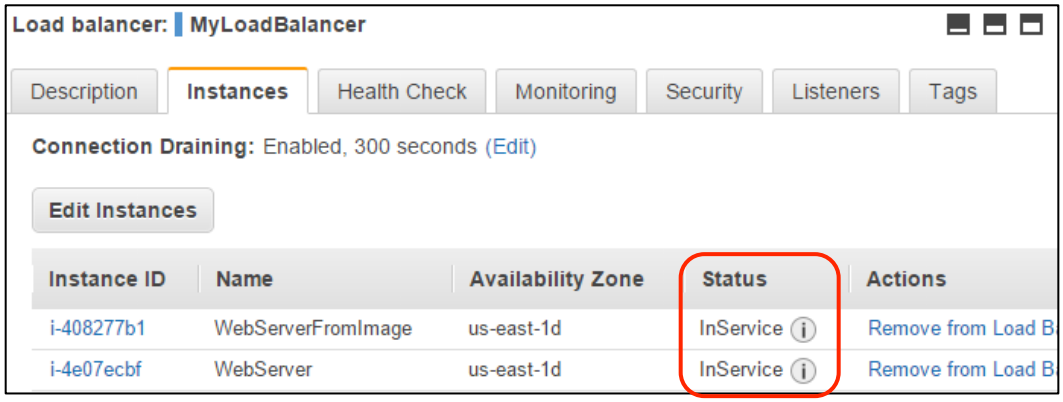
3.3.5 Select the two running instances and click on ‘Continue’ and then ‘Continue’.



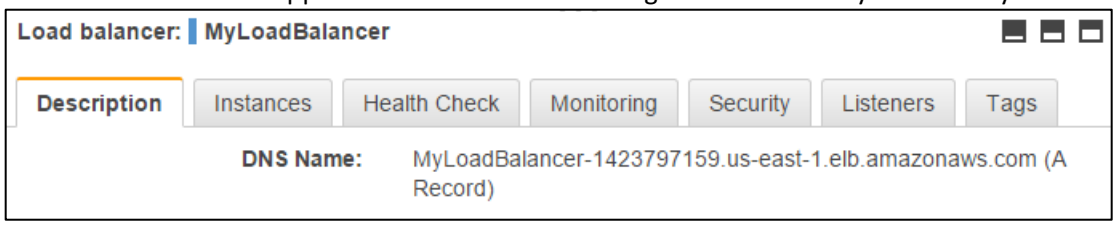
3.3.6 Add Key as ‘Name’ and Value as ‘WebServerLoadBalancer’ and click ‘Continue’. Review the information and click ‘Create’



3.3.7 Once created it will take time for the Instances to register with the load balancer. Proceed when both the instances are 'InService' in the Instances tab.



3.3.8 After that, copy the (A Record) link from 'Description' tab to a web browser and verify that the content shown is swapped between the two running instances when you refresh your browser.



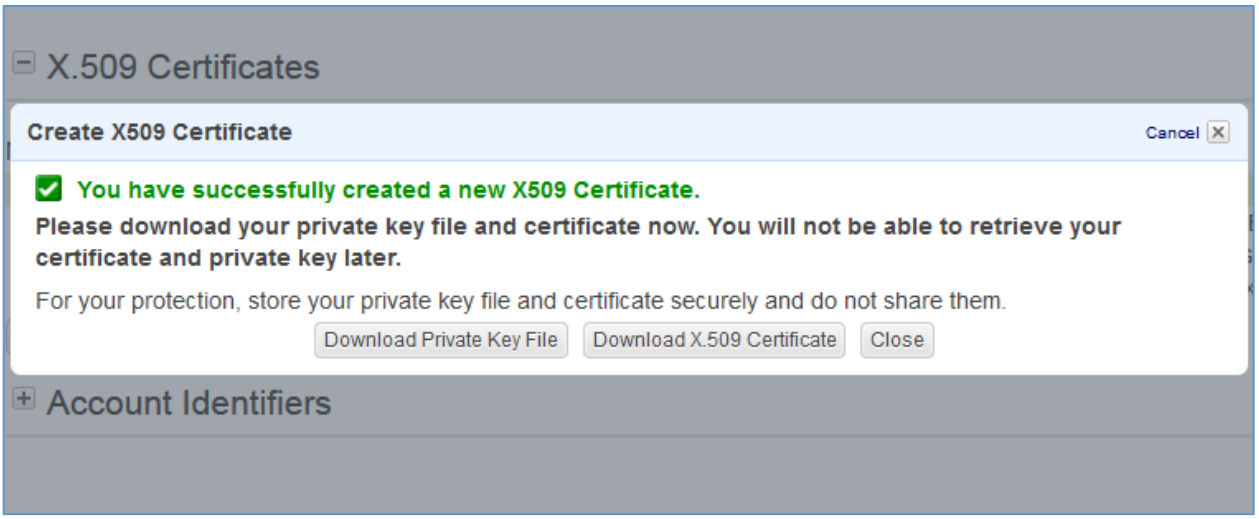
- You will end-up with two running instances that are accessed from a common 'DNS Name' using a Load Balancer.

3.4 Using AWS AutoScaling Group

3.4.1 In your **local** linux machine, inside /opt/tools/ directory download and unzip the AutoScaling tools command line. (*Create the directory if it doesn't exist*)

```
#sudo su
#wget ec2-downloads.s3.amazonaws.com/AutoScaling-2011-01-01.zip
#unzip AutoScaling-2011-01-01.zip
```

From your AWS account, click in 'security credentials' and create a X.509 Certificate, you will able to download a 'Private Key File' and 'X.509 Certificate'; save those files in /opt/tools directory.



- Using the 'credential-file.path.template' inside /opt/tools/AutoScaling-1.0.61.6, create a 'credential-file-path' file in the same directory and add the 'Access Key Id' and 'Secret Access Key' information obtained in Lab 2 Step 3.3.1

  Change permissions
  #chmod 400 credential-file-path

  PATH and environment variables

  For Ubuntu, Xubuntu, open the bash file:
  #nano /etc/bash.bashrc

  At the end of the file add:

  ```
  export JAVA_HOME=/usr/
  export AWS_AUTO_SCALING_HOME=/opt/tools/AutoScaling-1.0.61.6/
  export PATH=${AWS_AUTO_SCALING_HOME}/bin:$PATH
  export AWS_CREDENTIAL_FILE=/opt/tools/AutoScaling-1.0.61.6/credential-file-path
  ```

For the changes to take effect, **open a new terminal** on your current terminal run:
#source /etc/bash.bashrc

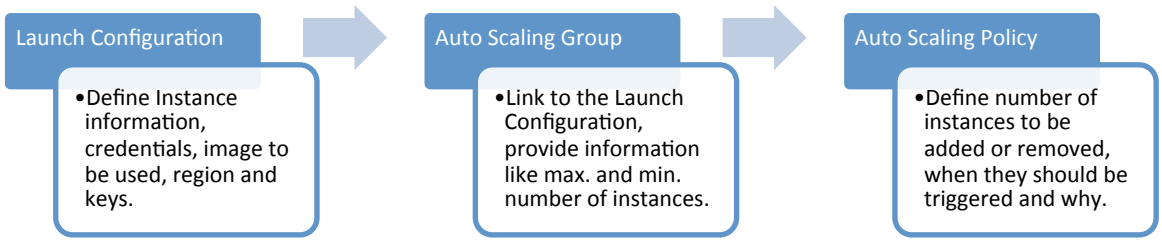
Verify that the environment is properly setup!

#as-cmd

A list of Auto-Scaling commands will be displayed in your screen.

```
[ec2-user@ip-10-164-46-180 AutoScaling-1.0.61.2]$ as-cmd
Command Name                                Description

------------                                -----------

as-create-auto-scaling-group                Create a new Auto Scaling group.

as-create-launch-config                     Creates a new launch configuration.

as-create-or-update-tags                    Create or update tags.

as-delete-auto-scaling-group                Deletes the specified Auto Scaling g
roup.
as-delete-launch-config                     Deletes the specified launch configu
ration.
as-delete-notification-configuration        Deletes the specified notification c
onfiguration.
as-delete-policy                            Deletes the specified policy.

as-delete-scheduled-action                  Deletes the specified scheduled acti
on.
as-delete-tags                              Delete the specified tags

as-describe-adjustment-types                Describes all policy adjustment type
s.
as-describe-auto-scaling-groups             Describes the specified Auto Scaling
 groups.
as-describe-auto-scaling-instances          Describes the specified Auto Scaling
 instances.
as-describe-auto-scaling-notification-types Describes all Auto Scaling notificat
ion types.
as-describe-launch-configs                  Describes the specified launch confi
gurations.
as-describe-metric-collection-types         Describes all metric colle... metric
 granularity types.
as-describe-notification-configurations     Describes all notification...given A
uto Scaling groups.
as-describe-policies                        Describes the specified policies.
```

Launch Configuration and Auto Scaling Group



3.4.3 Now that your local machine is configured, run the following commands in order to create a Launch Configuration. Use your AMI information below. (**Note**: You can copy all this commands in only one line without "\" if you feel more comfortable or just type line by line, verify the spaces between commands).

```
as-create-launch-config                          \
    --instance-type t1.micro                     \
    --aws-credential-file credential-file-path   \
    --region us-east-1                           \
    --image-id ami-xxxxxxxx                      \
    --key key-ec2                                \
    --group 'SG_EC2'                             \
    --launch-config ec2-launch-configuration
```

An 'OK-Created launch config' message will be received.

```
root@ubuntu:/opt/tools/AutoScaling-1.0.61.6# as-create-launch-config --instance-type t1.micro
 --aws-credential-file credential-file-path --region us-east-1 --image-id ami-28c5e540 --key
key-ec2 --group 'SG_EC2' --launch-config ec2-launch-configuration
OK-Created launch config
```

Run the 'as-describe-launch-configs' to see details of your configuration.

```
root@ubuntu:/opt/tools/AutoScaling-1.0.61.3# as-describe-launch-configs
LAUNCH-CONFIG  ec2-launch-configuration  ami-274a164e  t1.micro
```

Auto Scaling Group creation. **(Use the same zone as your instances)**

```
as-create-auto-scaling-group ec2-autoscaling-group \
    --aws-credential-file credential-file-path     \
    --region us-east-1                             \
    --availability-zones us-east-1x                \
    --launch-configuration ec2-launch-configuration \
    --load-balancers MyLoadBalancer                \
    --max-size 4                                   \
    --min-size 1
```

You will receive an 'OK-Created AutoScalingGroup' message

```
root@ubuntu:/opt/tools/AutoScaling-1.0.61.3# as-create-auto-scaling-group ec2-autoscaling-group --aws-credential-file cred
ential-file-path --region us-east-1 --availability-zones us-east-1c --launch-configuration ec2-launch-configuration --load
-balancers MyLoadBalancer --max-size 4 --min-size 1
OK-Created AutoScalingGroup
```

Verify the Auto Scaling Group is created with 'as-describe-auto-scaling-groups'.

```
root@ubuntu:/opt/tools/AutoScaling-1.0.61.3# as-describe-auto-scaling-groups
AUTO-SCALING-GROUP  ec2-autoscaling-group  ec2-launch-configuration  us-east-1c  MyLoadBalancer  1  4  1  Default
INSTANCE  i-f9df3380  us-east-1c  InService  Healthy  ec2-launch-configuration
```

Check the Scaling activities: with the 'as-describe-scaling-activities' command

```
root@ubuntu:/opt/tools/AutoScaling-1.0.61.3# as-describe-scaling-activities
ACTIVITY  a391ef98-5634-4900-80c8-3cef2be1117f  2013-10-22T22:37:12Z  ec2-autoscaling-group  Successful
```

3.4.4 Confirm that a new instance is created and automatically joined to the 'Load Balancers'. Using your Load Balancer [DNS name], refresh your browser to swap among the 3 instances.

		i-f9df3380	t1.micro	us-east-1c	running	Initializing	None
	Web Server	i-58aa9a32	t1.micro	us-east-1c	running	2/2 check...	None
	WebServerFr...	i-0c372f77	t1.micro	us-east-1c	running	2/2 check...	None

3.4.5 Auto Scaling integrated into the ELB

Remove the first two instances from 'Load Balancers' and terminate them in order to launch new instances automatically.

		i-f9df3380	t1.micro	us-east-1c	running
	Web Server	i-58aa9a32	t1.micro	us-east-1c	shutting-do...
	WebServerFr...	i-0c372f77	t1.micro	us-east-1c	shutting-do...

3.4.6 Set up Auto Scaling policy by entering the following commands:

Create Scale Up Policy

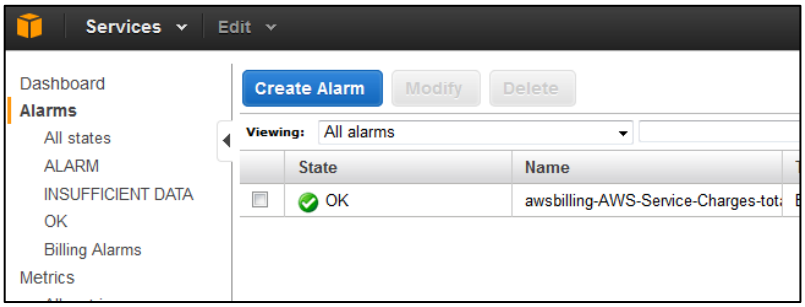
```
as-put-scaling-policy ec2-ScaleUpPolicy                \
    --aws-credential-file credential-file-path         \
    --region us-east-1                                 \
    --auto-scaling-group ec2-autoscaling-group         \
    --adjustment=1                                     \
    --type ChangeInCapacity                            \
    --cooldown 300
```

Create Scale Down Policy

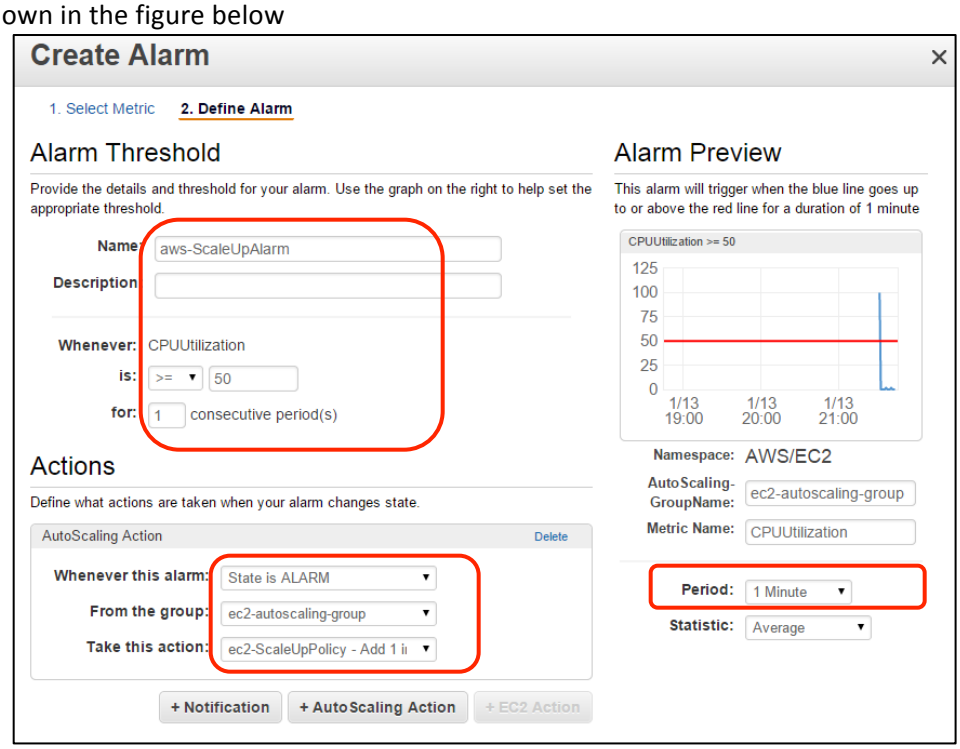
```
as-put-scaling-policy ec2-ScaleDownPolicy          \
    --aws-credential-file credential-file-path     \
    --region us-east-1                             \
    --auto-scaling-group ec2-autoscaling-group     \
    "--adjustment=-1"                              \
    --type ChangeInCapacity                        \
    --cooldown 300
```

3.5 Using AWS CloudWatch

3.5.1 Go to 'CloudWatch' service and select 'Alarms' in the left panel.

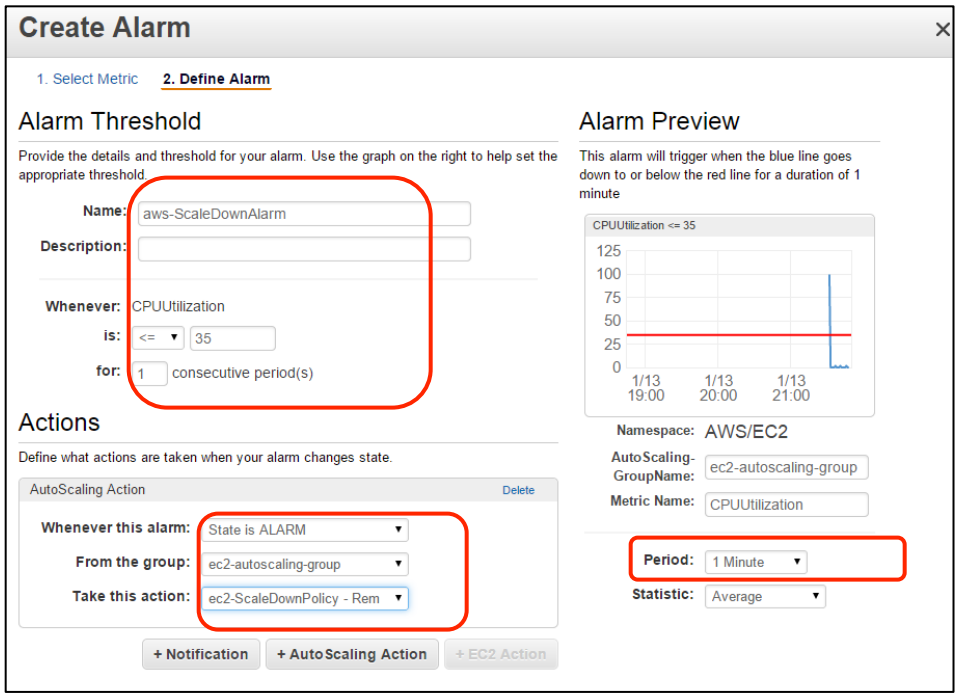


Search for 'ec2-autoscaling-group', select 'CPU Utilization' , click 'Next', then add an auto scaling action by clicking on '+ Auto Scaling Action' and remove the Notification by clicking on 'Delete'. Finally enter values as shown in the figure below

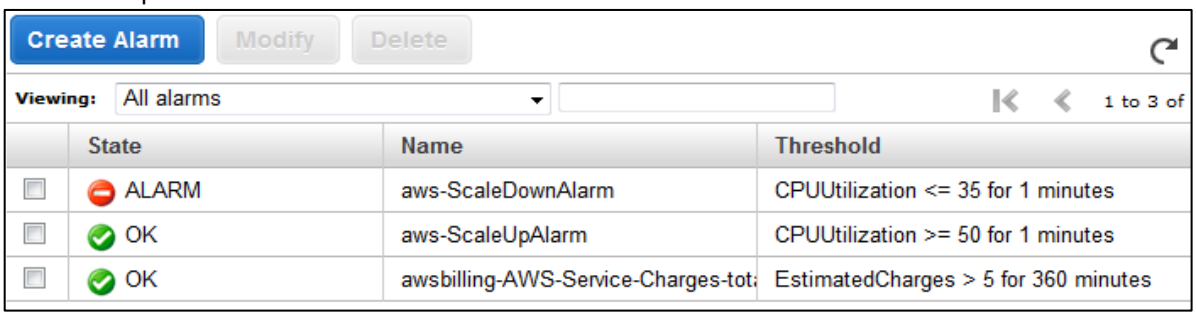


Confirm your configuration and ensure all details are accurate and then click on 'Create Alarm'.

Similarly to the 'Scale Up' alarm, create a new alarm for 'Scale Down' policy. Follow steps as before and ensure that the values entered are as displayed in the following figure and click on 'Create Alarm'.



You will end up with two new alarms



3.5.2 Trigger the alarms

Ensure that only one EC2 instance is running before executing the following commands.
Login to your EC2 instance and run the 'vmstat 1' command to check CPU usage.

In another terminal, login again to the same instance and run the command '**Apache server benchmarking tool (ab)**' in order to stress the CPU and activate the alarm.

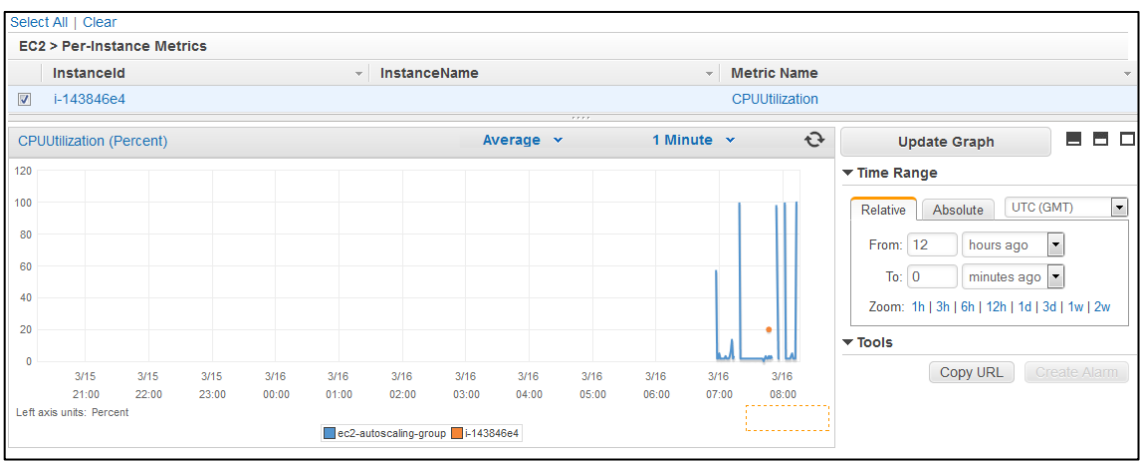
#ab -n 1000000 -c 1 http://localhost/drupal7/

Above command will generate a large number of user requests through the ab benchmarking tool. Note the output of the ab tool.

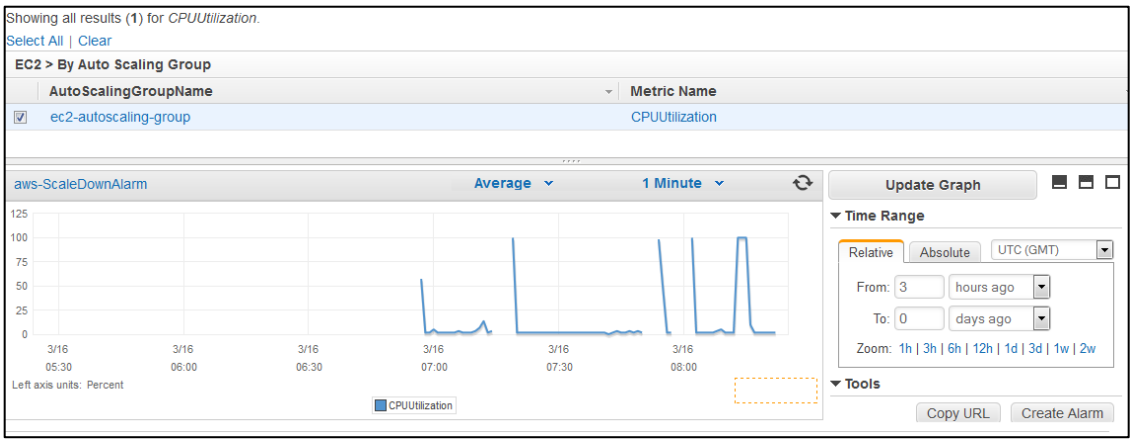
```
Terminal - ec2-user@ip-10-245-190-6:~
File  Edit  View  Terminal  Tabs  Help
[ec2-user@ip-10-245-190-6 ~]$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 380860   9000 161300    0    0    75    90   33   42  1  0 95  2  2
 1  0      0 378248   9000 162324    0    0     0     0  255 19488 28 71  0  0  1
 1  0      0 375892   9008 163308    0    0     0    24  259 19480 29 69  0  1  1
 1  0      0 374156   9008 164092    0    0     0     0  304 15563 26 66  0  0  8
 1  0      0 372296   9008 164800    0    0     0     0  252 14324 26 66  0  0  8
 1  0      0 370800   9008 165432    0    0     0     0  252 12238 33 53  0  0 14
 1  0      0 368940   9008 166216    0    0     0     0  256 15977 29 66  0  0  5
 1  0      0 357400   9016 167044    0    0     0     0  [illegible]  0  0  5
Terminal - ec2-user@ip-10-245-190-6:~
File  Edit  View  Terminal  Tabs  Help
[ec2-user@ip-10-245-190-6 ~]$ ab -n 1000000 -c 1 http://localhost/drupal7/'
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
```

Interrupt the process when a new instance is created.

3.5.3 In CloudWatch metrics option, search for EC2 Auto Scaling Group and see how the CPU utilization has increased. (***Capture this screen to submit as part of your report submission for grading***)



3.5.4 In Alarms, verify that the Scale Up alarm has triggered, and after a few minutes verify that the Scale Down has triggered as well, the process for scaling down should be automatic. (***Capture this screen as part of your report submission for grading***).



- Using your LoadBalancer 'DNS Name' as you did in step 3.3.8, you will be able to access the instances automatically.

4. What to turn in for Grading?

1. Create a 'new-lc' launch-configuration with 'new-cf' as a source file for your credential, 'ami-xxx' image available in the EC2, 'new-key' key, 'new-group' group and configure to launch a free instance type. Describe the commands used.

2. Define an Auto Scaling Group called 'new-asg' with a max scalability of 5 instances, using 'new-cf' as name source file for your credential, 'new-lc' as launch configuration name and 'ELB' load balancer. Describe the commands used.

3. Create a 'scale-up-policy' for scaling up with 2 new instances every time that a change in the capacity of your running instances is detected; use 'new-asg' as name for your auto-scaling group and specify 4 minutes interval for evaluating conditions before taking cool down action. Describe the commands used.

4. Which AWS service and metric do you use for joining and triggering your policies?

5. Add the screenshots taken in Steps 3.5.3 and 3.5.4 to your report with your name on it.

6. Send an email to T.A Amit (ar442@mail.missouri.edu) with subject as "AWS" and a link to your load balancer in the body of the email. Once your grade will be posted on Blackboard you can execute the following commands.

Clean-up resources.
Cleaning up resources by removing the two alarms in the CloudWatch Dashboard and the Policies in the instance.

Perform the following command,

```
as-delete-policy                          \
--aws-credential-file credential-file-path \
--name ec2-ScaleDownPolicy                \
--auto-scaling-group ec2-autoscaling-group
```

Do the same for your Scale Up Policy

Remove your AutoScalingGroup and launch configuration use [--force-delete] option if necessary

```
as-delete-auto-scaling-group ec2-autoscaling-group
```

Finally, remove your launch configuration

```
as-delete-launch-config ec2-launch-configuration
```