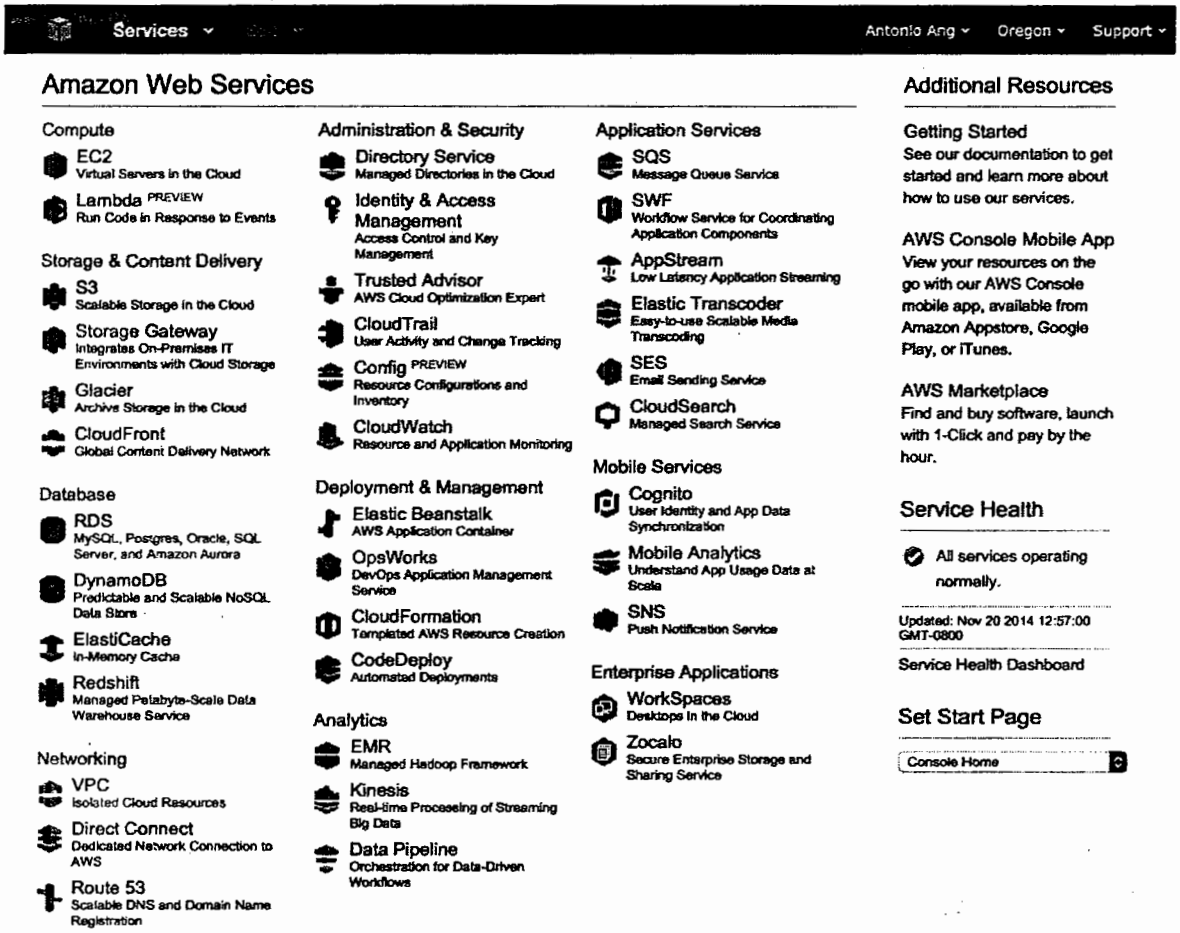
Lab 13

WORKING WITH AMAZON EC2 AUTO SCALING GROUPS

STEP 1: Log In to the Amazon Web Service Console

This laboratory experience is about Amazon Web Services and you will use the AWS Management Console in order to complete all the lab steps.



The AWS Management Console is a web control panel for managing all your AWS resources, from EC2 instances to SNS topics. The console enables cloud management for all aspects of the AWS account, including managing security credentials, or even setting up new IAM Users.

Log in to the AWS Management Console

In order to start the laboratory experience, open the Amazon Console by clicking this button:

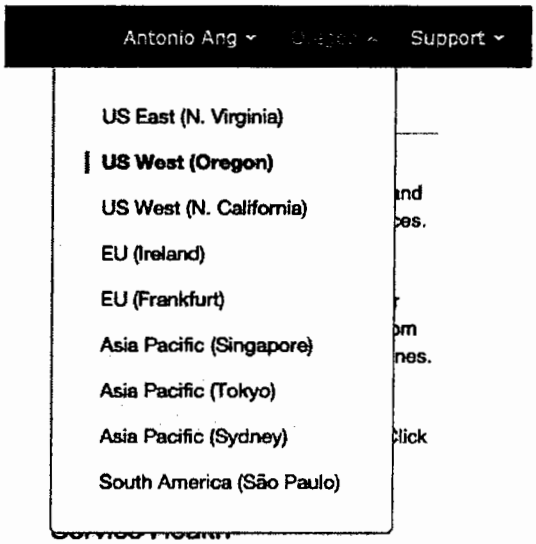
Open AWS Console

Log in with the username **xxxx** and the password **xxxx**

Select the right AWS Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but you must use the region **US West (Oregon)** for this laboratory.

You can select the **US West (Oregon)** region using the upper right dropdown menu on the AWS Console page.



STEP 2: Auto Scaling Overview

Before going to the AWS console and creating an Auto Scaling Group, let's take a quick look at the components of an Auto Scaling Group. AWS has done a great job defining them so we'll use the official definition:

Groups
Your EC2 instances are organized into groups so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances. For more information, see Auto Scaling Groups.
Launch configurations
Your group uses a launch configuration as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances. For more information, see Launch Configurations.

You can read the full documentation here
http://docs.aws.amazon.com/autoscaling/latest/userguide/WhatIsAutoScaling.html

In this lab, we will learn to create an Auto Scaling Group with these components and place it behind an Elastic Load Balancing (ELB). Don't worry if you don't fully understand all the components yet. We will talk in greater detail about each of the components as we create them.

At the end of this lab we'll have an Auto Scaling Group with some web server instances behind an ELB. Although this lab focuses on Auto Scaling, it is important to mention that to have an Auto Scaling Group behind an ELB, it is necessary to create the ELB first. In the next step, we will begin exploring elements in the AWS console by creating an ELB.

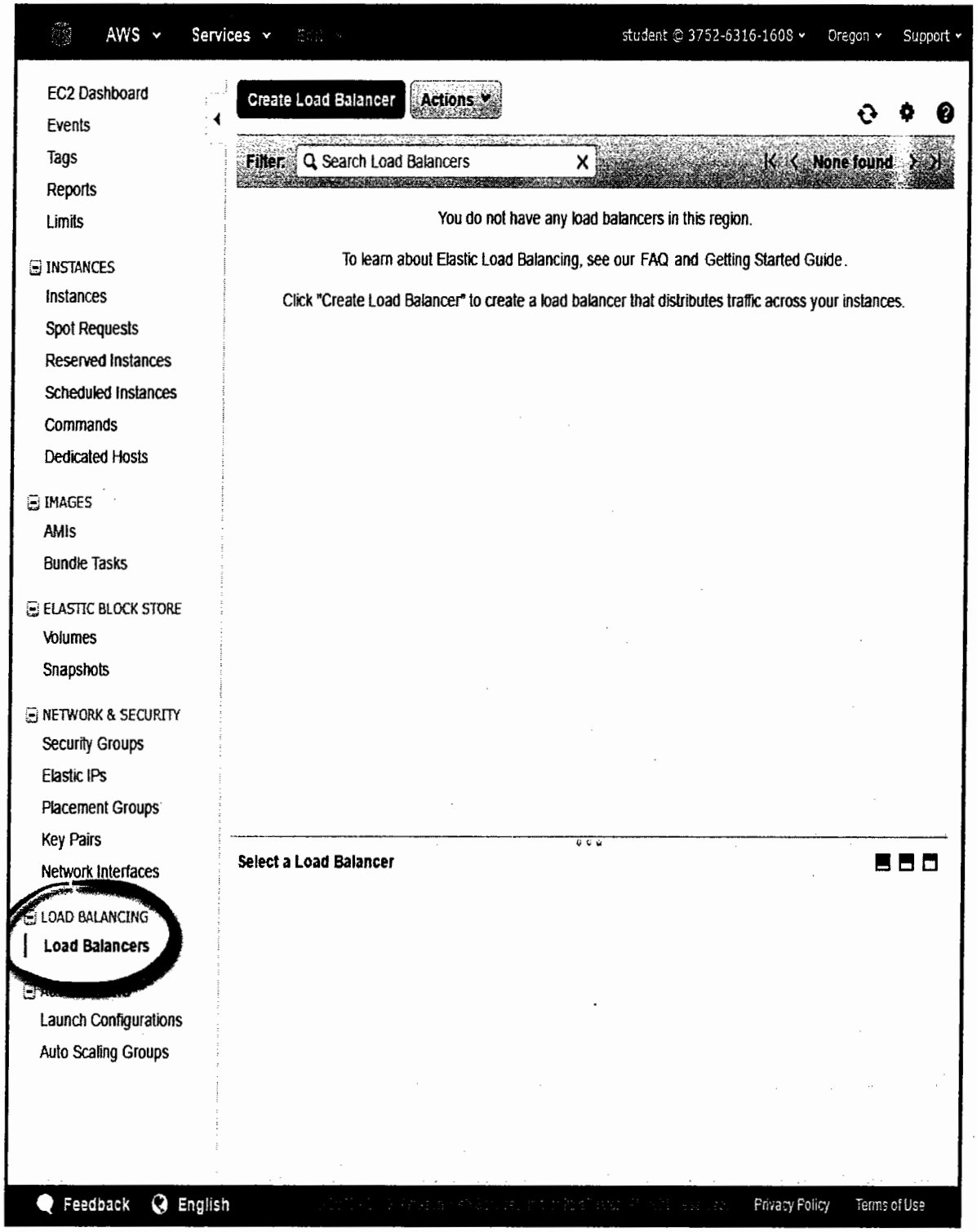
STEP 3: Create a load balancer using ELB

Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater fault tolerance in your applications and seamlessly provides the correct amount of load balancing capacity needed in response to incoming application traffic.

Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored to health. Customers can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for greater consistent application performance.

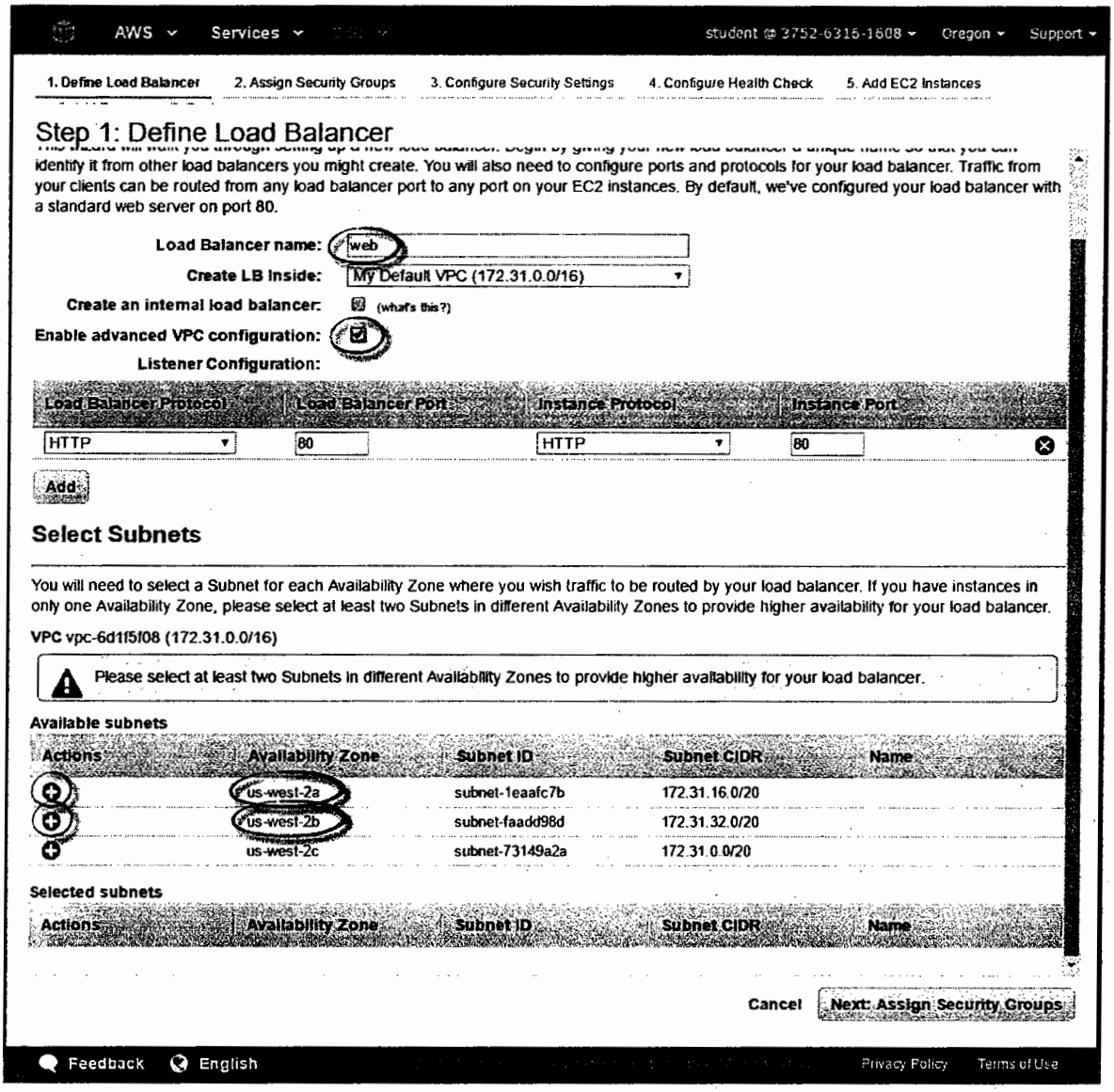
You can create your first ELB by taking the following steps:

From the EC2 dashboard, click the **Load Balancers** link in the Load Balancing group. The list of all already-created Load Balancers appears--this list will most likely be empty.



Click the blue **Create Load Balancer** button.

On the **Define Load Balancer** step, type a load balancer name (e.g., "web") and select **Enable advanced VPC configuration**



Select two subnets, one from the *us-west-2a* Availability Zone and one from the *us-west-2b*Availability Zone. Then click the **Next: Assign Security Groups** button.

AWS Services student @ 3752-6316-1608 Oregon Support

1. Define Load Balancer | 2. Assign Security Groups | 3. Configure Security Settings | 4. Configure Health Check | 5. Add EC2 Instances

Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name: web

Create LB Inside: My Default VPC (172.31.0.0/16)

Create an internal load balancer: (what's this?)

Enable advanced VPC configuration:

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-6d1f5f08 (172.31.0.0/16)

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-west-2c	subnet-73149a2a	172.31.0.0/20	

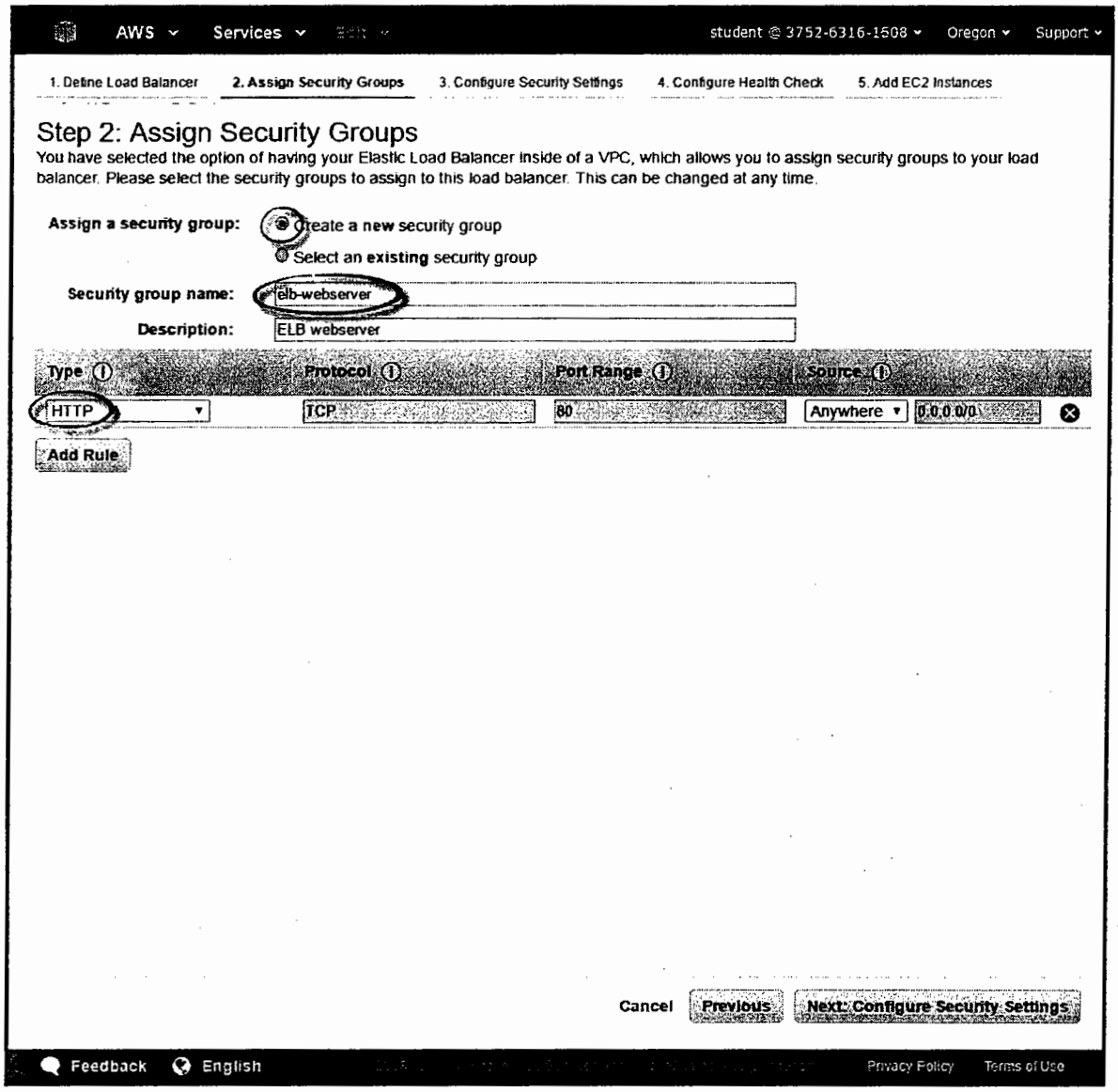
Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-west-2a	subnet-1eaafc7b	172.31.16.0/20	
	us-west-2b	subnet-faadd98d	172.31.32.0/20	

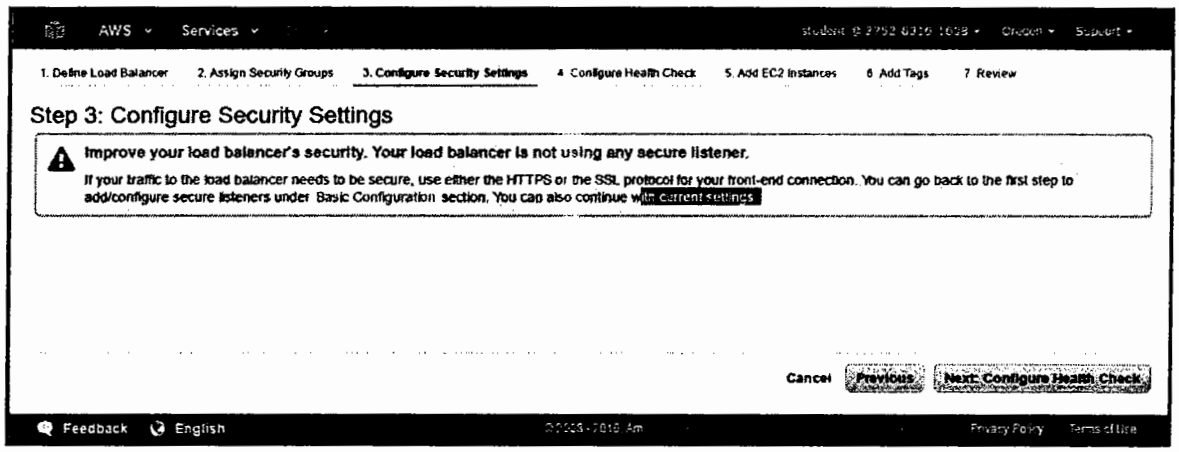
Cancel | Next: Assign Security Groups

Feedback English Privacy Policy Terms of Use

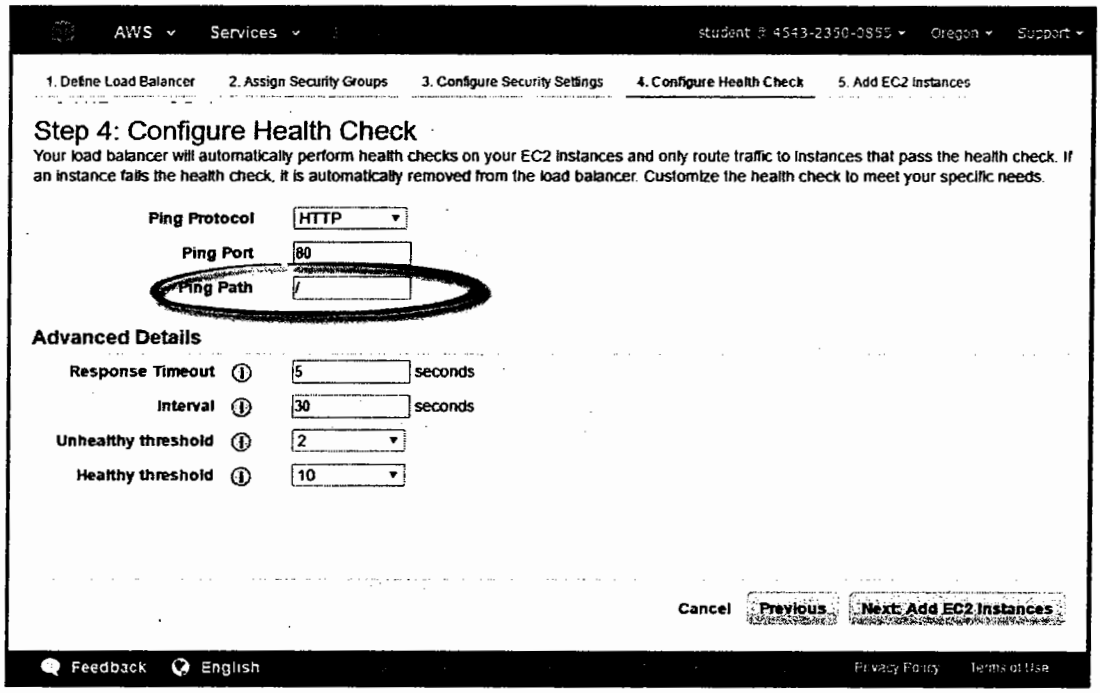
In the **Assign Security Groups** section, select ***Create a new security group***, type a Security group name (e.g., "elb-webserver") and a description. Create a single firewall rule of type *HTTP*, protocol *TCP*, port range *80*, and source *Anywhere*. Click **Next: Configure Security Settings**.



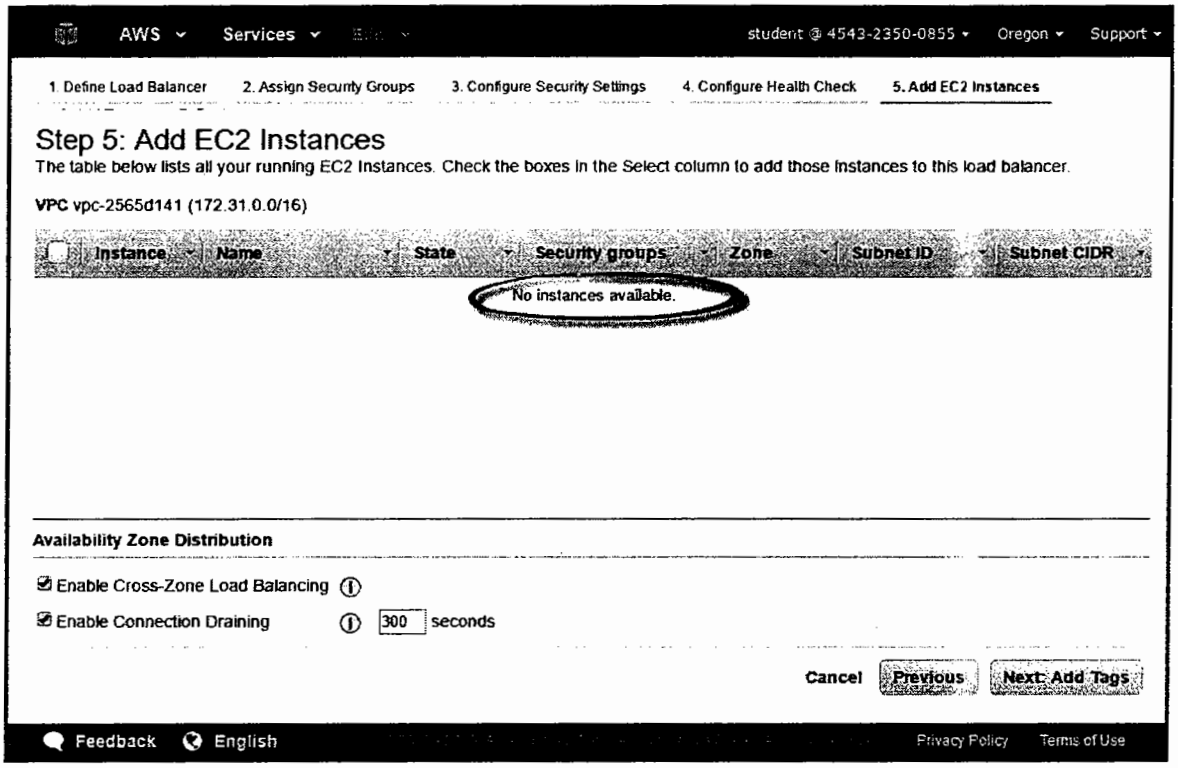
Ignore the warning in the **Configure Security Settings** section. We are only serving the HTTP protocol in this exercise, so these settings are not required. Click **Next: Configure Health Check**.



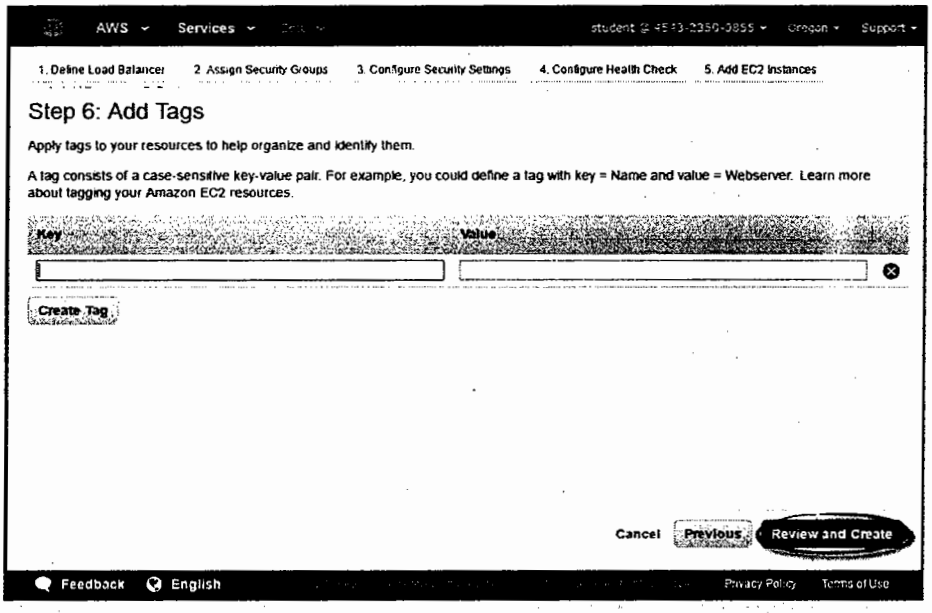
In the **Configure Health Check** section, replace the default value of **Ping Path** with a single forward slash ("/") and click **Next: Add EC2 Instances**.



In the **Add EC2 Instances** section, you should see a "No instances available" message. This is because we have yet created and launched our Auto Scaling Group. Click **Next: Add Tags to continue**.



You may leave the fields blank in the **Add Tags** section. Click the **Review and Create** button to continue.



Review your settings, then click **Create** when ready.

AWS Services student @ 4543-2350-0855 Oregon Support

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances

Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer

Edit load balancer definition

Load Balancer name: web
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)

Configure Health Check

Edit health check

Ping Target: HTTP:80/
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy threshold: 2
Healthy threshold: 10

Add EC2 Instances

Edit instances

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances:

VPC Information

Edit subnets

VPC: vpc-2565d141
Subnets: subnet-b4d603c2, subnet-e26bdb86

Security groups

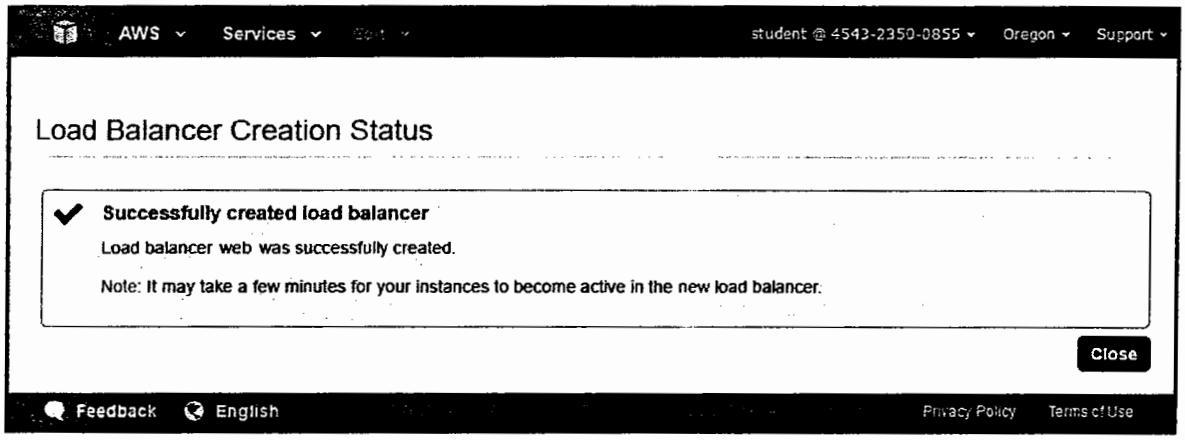
Edit security groups

Security groups: elb-webserver

Cancel Previous Create

Feedback English Privacy Policy Terms of Use

Wait for the Load Balancer Creation Status to populate with the message, "Successfully created load balancer." Click **Close.**



STEP 4: Create a Launch Configuration

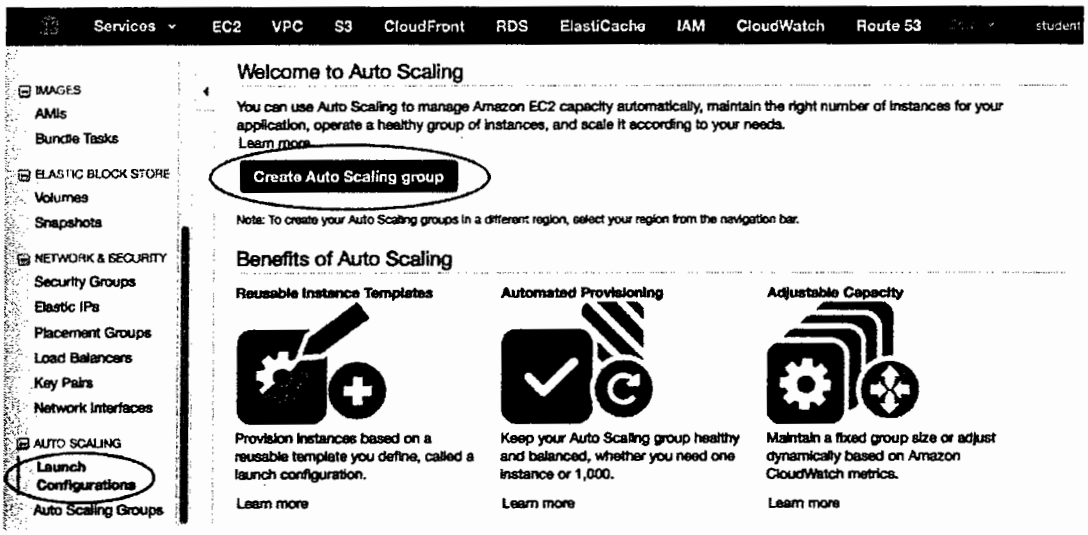
A **Launch Configuration** is a template that the Auto Scaling group uses to launch Amazon EC2 instances. If you've launched an individual EC2 instance before, you've already walked through the process of defining compute characteristics such as the instance type, security groups, and configuration scripts. A launch configuration allows you to define these same characteristics, which are then applied to any instances launched in the Auto Scaling group.

You create the launch configuration by including information such as the Amazon machine image ID to use for launching the EC2 instance, the instance type, key pairs, security groups, and block device mappings, among other configuration settings. When you create your Auto Scaling group, you must associate it with a launch configuration. You can attach only one launch configuration to an Auto Scaling group at a time and it cannot be modified.

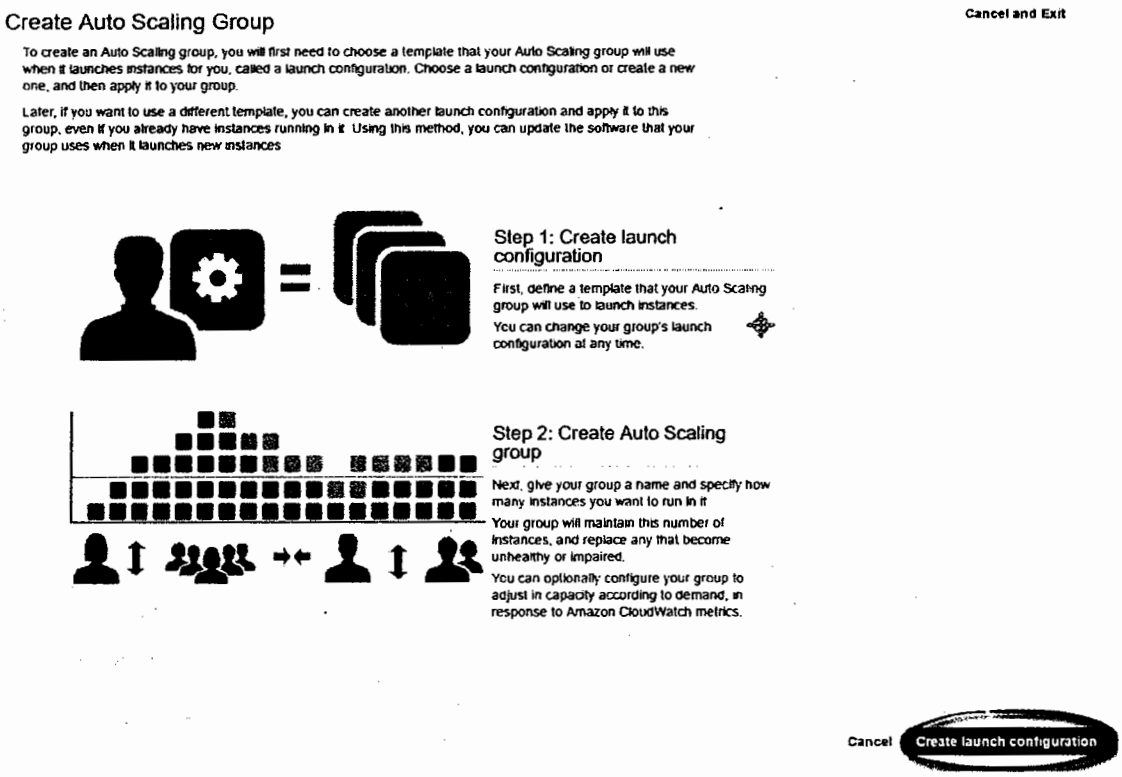
Let's start creating our Auto Scaling Group by first defining a **Launch Configuration**.

Navigate to the EC2 service from the AWS dashboard:

Open the **Launch Configurations** page and click on the **Create Auto Scaling group** button.



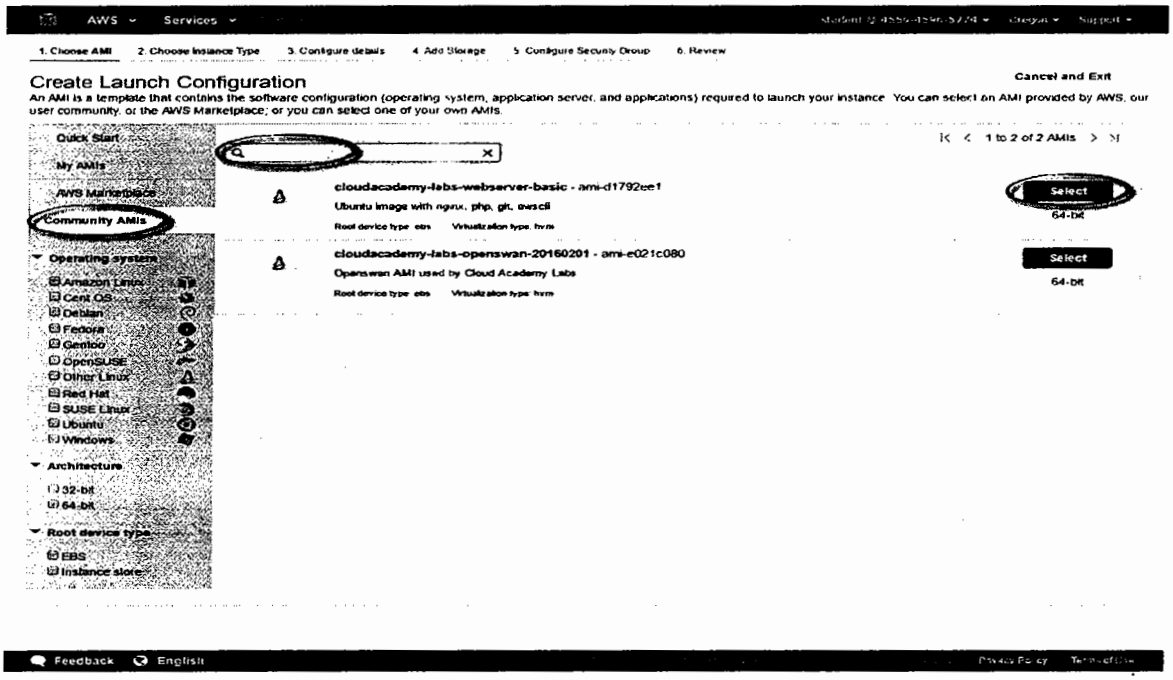
This brings you to the Create Auto Scaling group wizard. Click on the **Create Launch configuration** button.



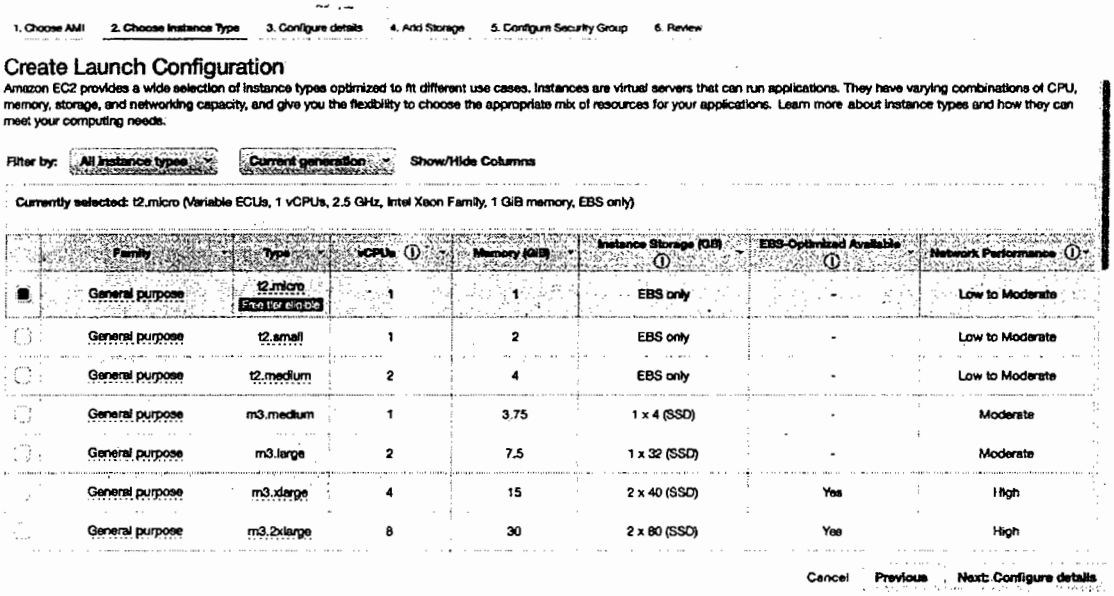
From there the AWS Management Console guides you though each required step and displays a graphical interface that is similar to the Launch Instance Wizard.

The first step is the AMI selection. You have to select the AMI that will be used by all the EC2 instances of the Auto Scaling group. The Vepsun DevOps team created a specific AMI for this laboratory. You can find it among the Community AMIs by searching for the word "vepsun" in the AMI search box.

Select the "**vepsun-labs-webserver-basic**" AMI and click Select.

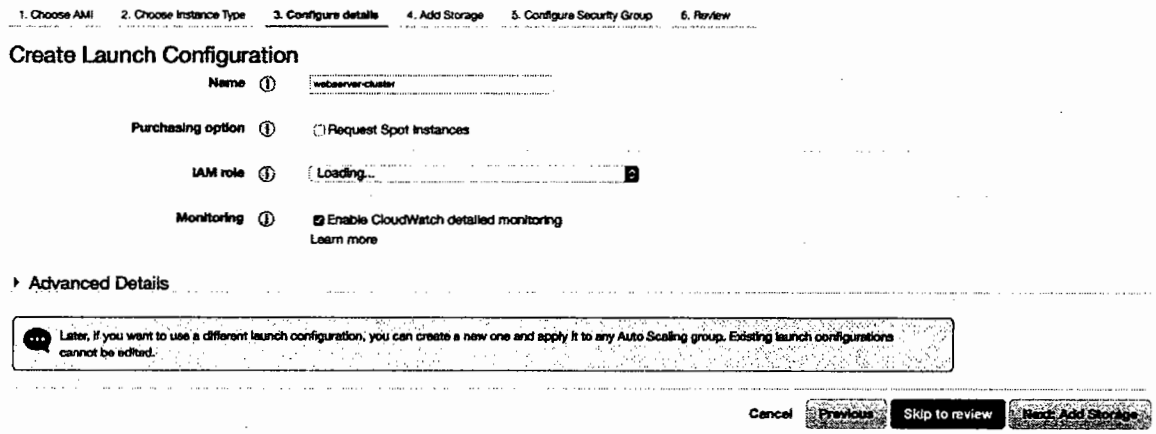


The next step is choosing the instance type. Select the t2.micro type and click on the **Next: Configure details** button.



The **Configure details** step asks you to name your launch configuration and asks if you want to enable the CloudWatch detailed monitoring for your future instances. Type a friendly name (e.g., "webserver-cluster") for the Launch Configuration and Enable detailed monitoring.

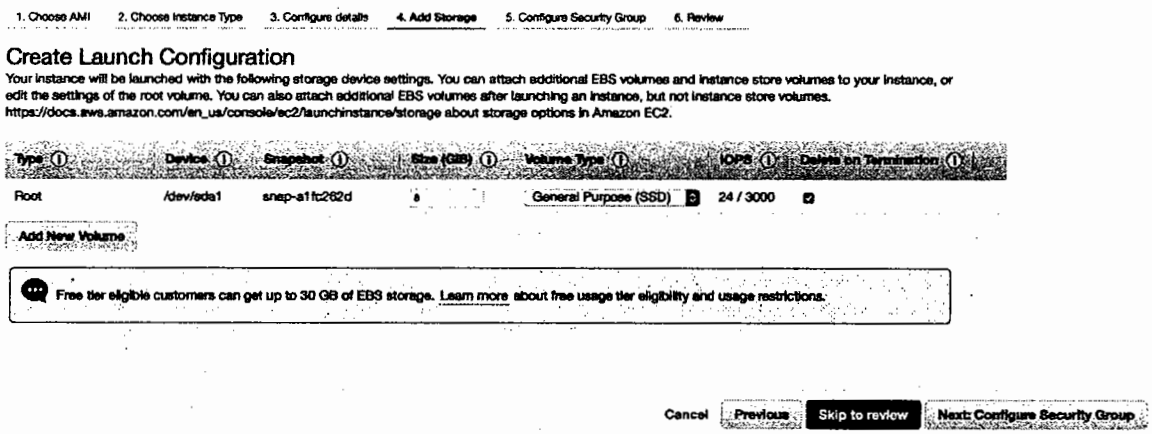
Click on **Next: Add Storage** after you have filled in the required fields.



The **Add Storage** step allows you to add or increment the size of any EBS volume linked to each EC2 instance that will be started by the Auto Scaling group.

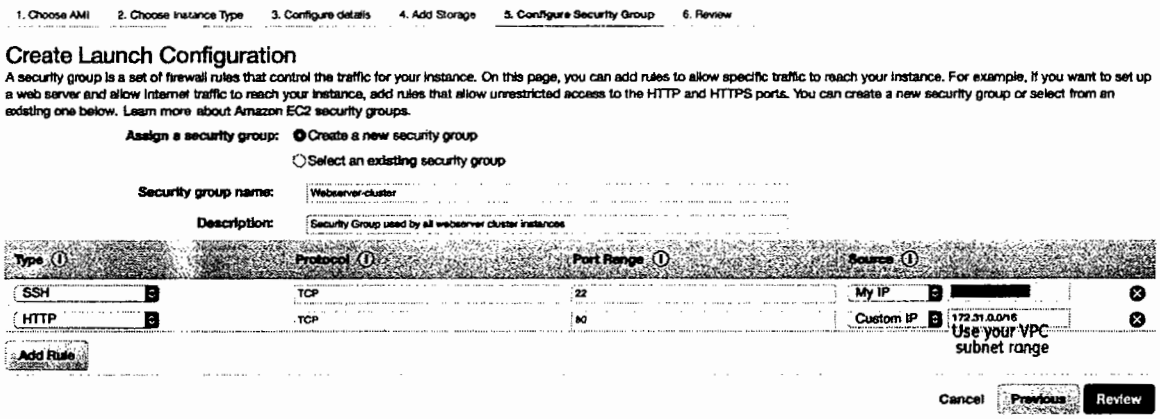
In order to complete this laboratory exercise, leave the defaults and do not add any EBS volumes. Then click on **Next: Configure Security Group**.

N.B.: You should use big EBS volumes only if your software requires storage space to process the application data. If you need to store raw or proceseed data, you should use Amazon S3, Redshift, DynamoDB or another storage/database service provided by Amazon.



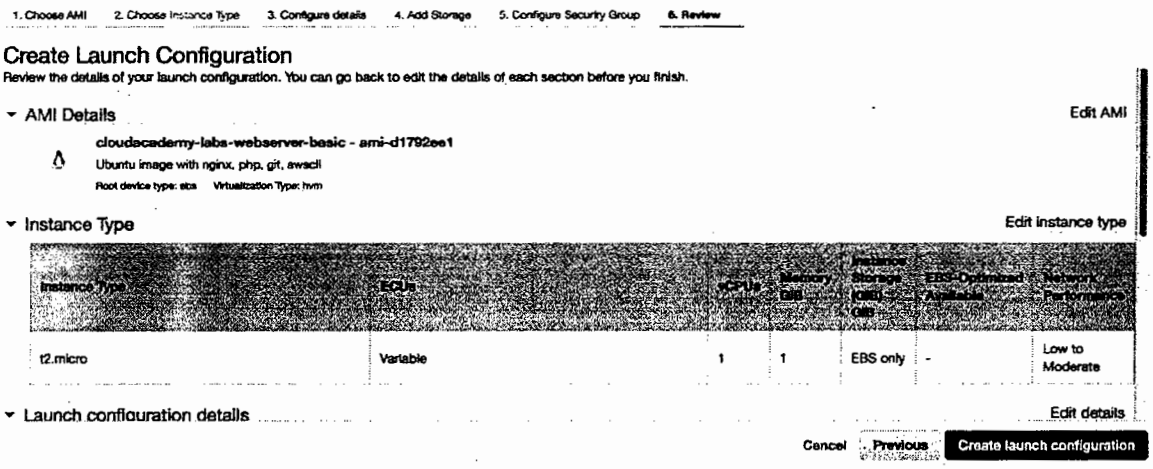
Create a new Security Group for your Auto Scaling Group. Choose a name (e.g., Webserver-cluster) and description, and add the required rules to allow inbound SSH and HTTP.

The default Amazon VPC subnet range is **172.31.0.0/16** . You can use it to allow the HTTP traffic, so the Elastic Load Balancing instance will be able route the HTTP requests to the instances of the Auto Scaling group.



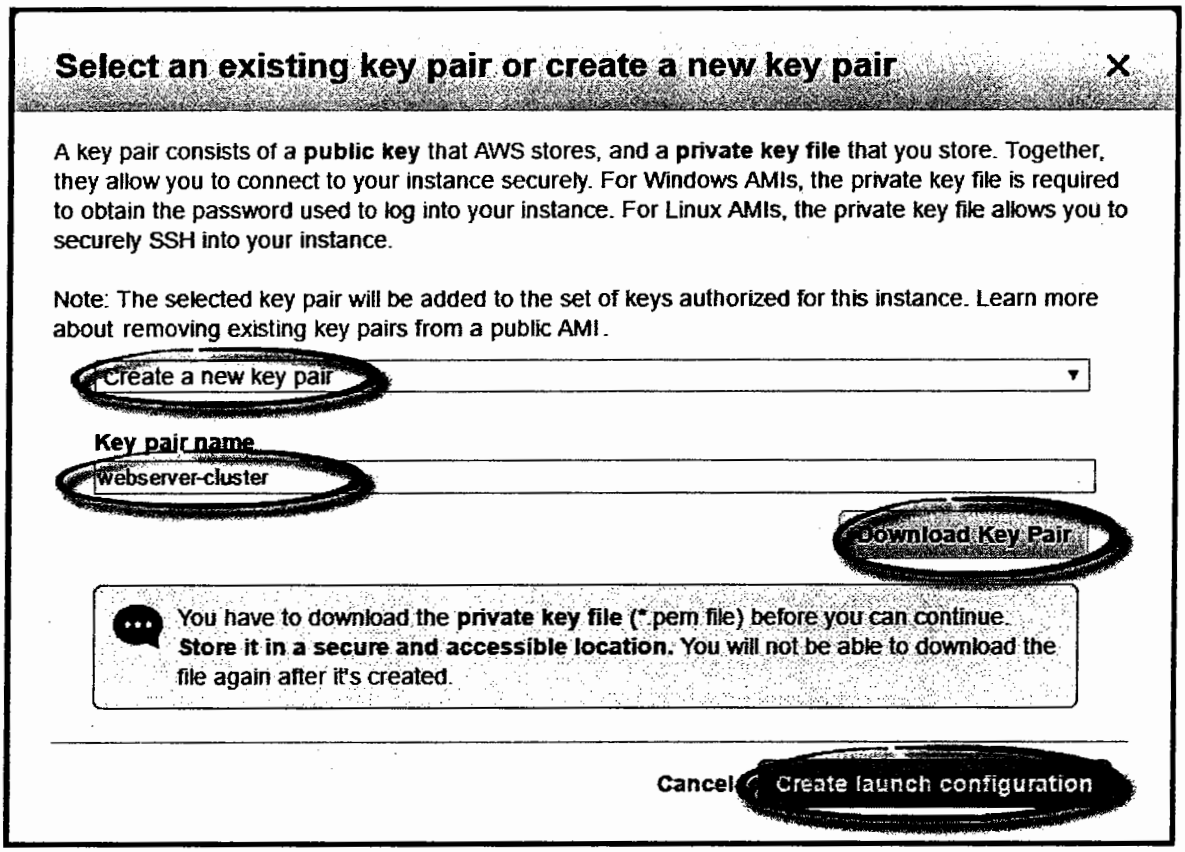
Click the blue **Review** button.

Once you have reviewed the details for accuracy, click the blue **Create launch configuration**button.



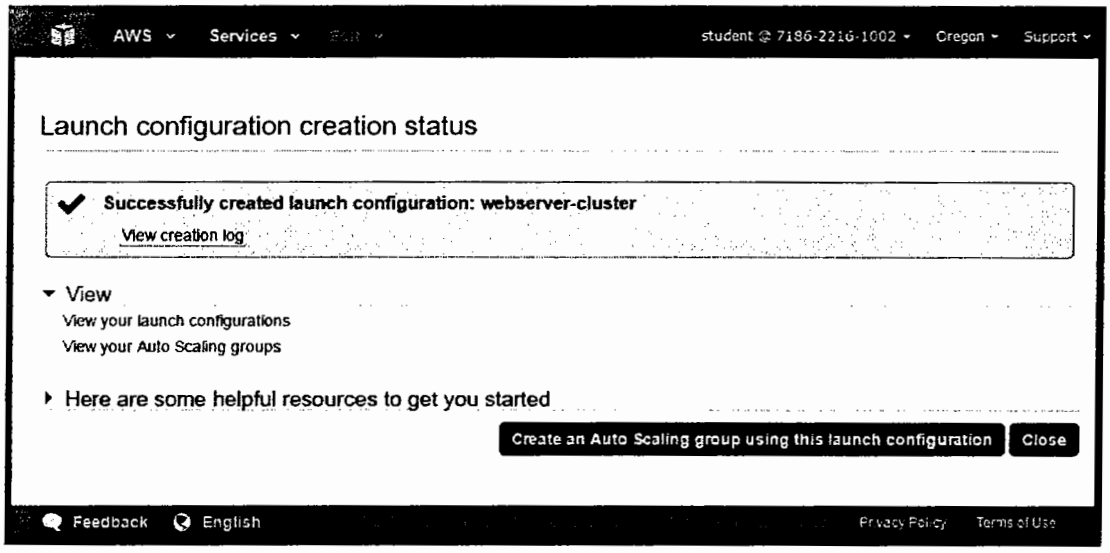
You will be presented with the S*elect an existing key pair or create a new key pair* dialogue box. Notice that you will use this Key Pair to access all the instances that are going to be launched by the Auto Scaling service with this Launch Configuration, so secure your Key Pair.

Select **Create a new key pair** from the first drop-down menu and type in a Key pair name (e.g., webserver-cluster). Click the **Download Key Pair** button. Then click the **Create Launch Configuration** button in this dialogue box.



Wait for the *Launch configuration creation status* to report, "Successfully created launch configuration." Congratulations! You created a new **Launch Configuration**.

We are going to continue creating our Auto Scaling Group in the next step. For now, just click **Close** in the AWS console and move on the next lab step.

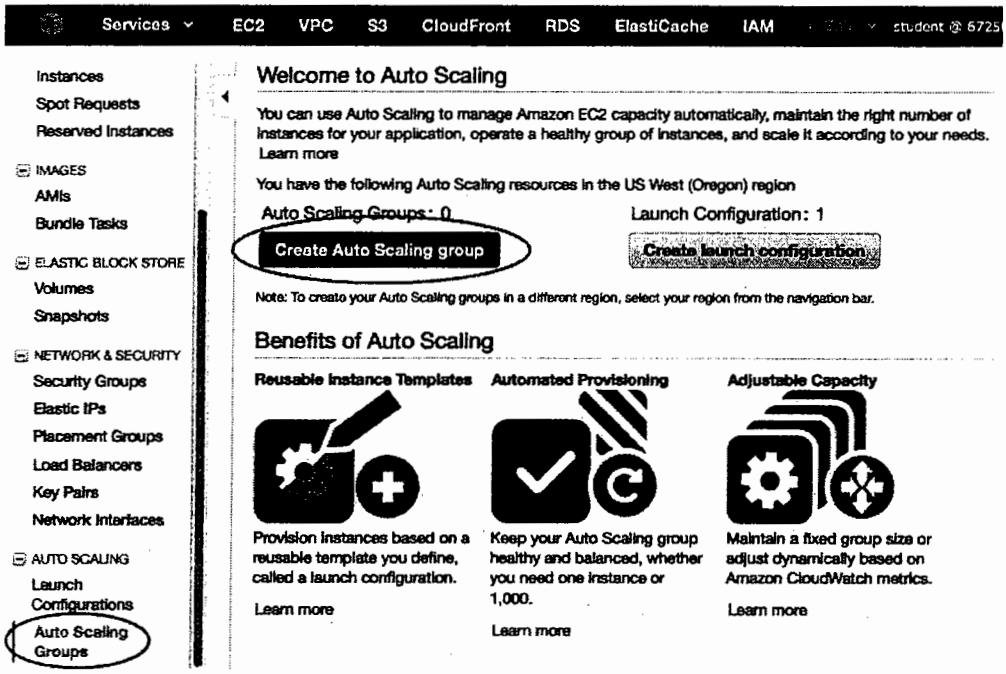


STEP 5: Create an Auto Scaling Group

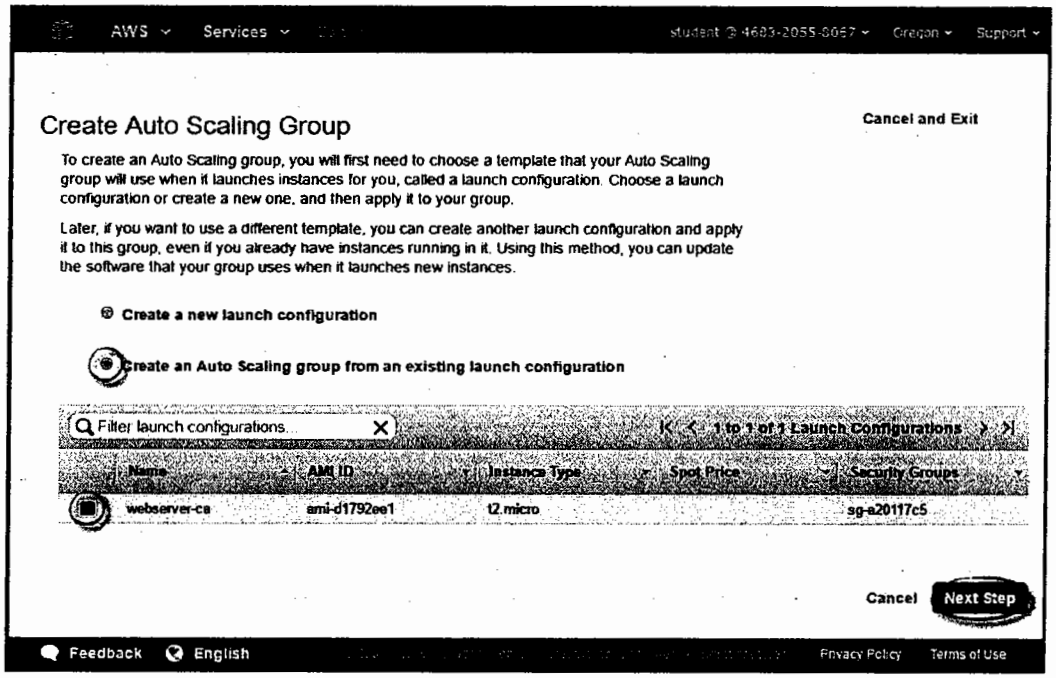
An Auto Scaling group is a representation of multiple Amazon EC2 instances that share similar characteristics and that are treated as a logical grouping for the purposes of instance scaling and management. For example, if a single application operates across multiple instances, you might want to increase or decrease the number of instances in that group to improve the performance of the application. You can use the Auto Scaling group to automatically scale the number of instances or maintain a fixed number of instances. You create Auto Scaling groups by defining the minimum, maximum, or desired number of running EC2 instances the group must have at any given point of time.

An Auto Scaling group starts by launching the minimum number (or the desired number, if specified) of EC2 instances and then increases or decreases the number of running EC2 instances automatically according to the conditions that you define. Auto Scaling also maintains the current instance levels by conducting periodic health checks on all the instances within the Auto Scaling group. If an EC2 instance within the Auto Scaling group becomes unhealthy, Auto Scaling terminates the unhealthy instance and launches a new one to replace the unhealthy instance. This automatic scaling and maintenance of the instance levels in an Auto Scaling group is the core value of the Auto Scaling service.

1. To create the Auto Scaling group, click on the **Auto Scaling Groups** link in the Auto Scaling menu group and then click the blue **Create Auto Scaling group** button.



2. Select **Create an Auto Scaling group from an existing launch configuration,** select the previously created launch configuration and click Next Step.



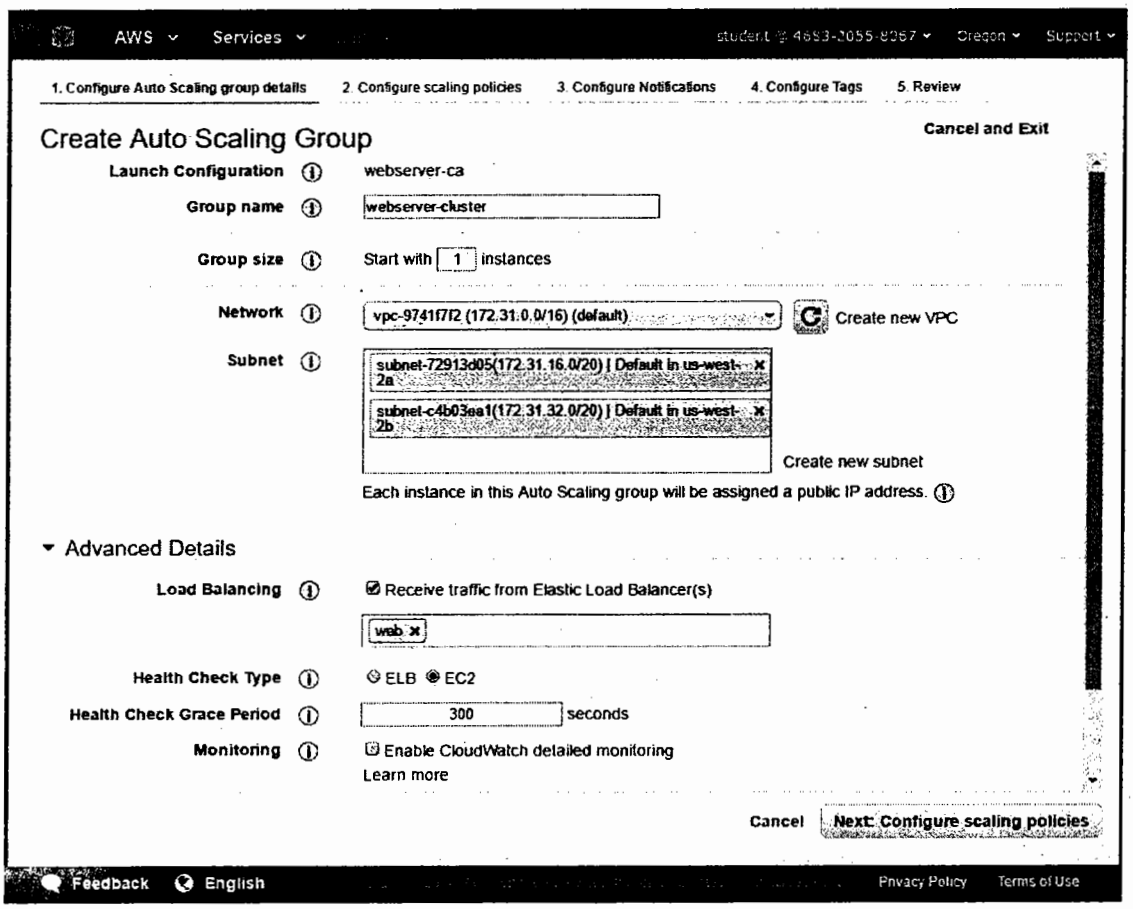
3. In the "Configure Auto Scaling group details" step, you should use the following settings:

   Group name: webserver-cluster
   Group size: 1
   Network: default
   Subnet: Select two. The default network in *us-west-2a* and the default network in *us-west-2b.*

   Open the Advanced Details, then set as follows:

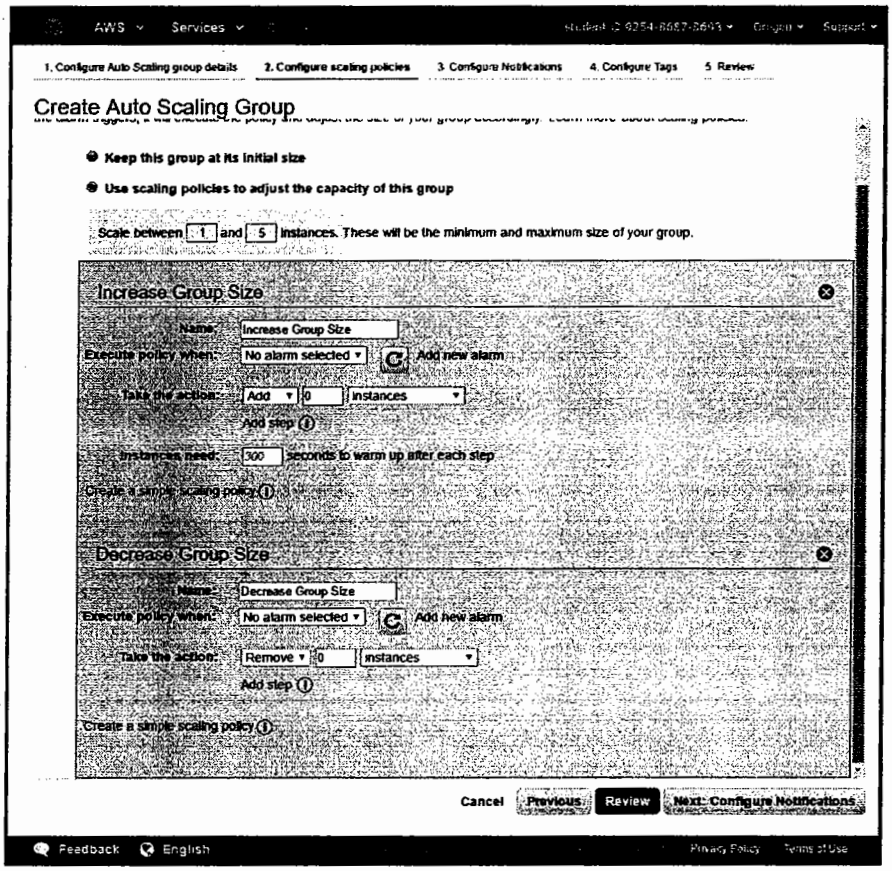
   Load Balancing: Check Receive traffic from Elastic Load Balancer(s). Select the "web" ELB you created
   Health Check Type: ELB
   Monitoring: Check *Enable CloudWatch detailed monitoring*

   Once all fields are complete, click **Next: Configure Scaling Policies**.



4. In this step, you must *Configure scaling policies*, which determine how and when your infrastructure will scale out and scale back.

   Select the *Use scaling policies to adjust the capacity of this group* button. For this lab you should set your group to scale between **1** and **5** instances.

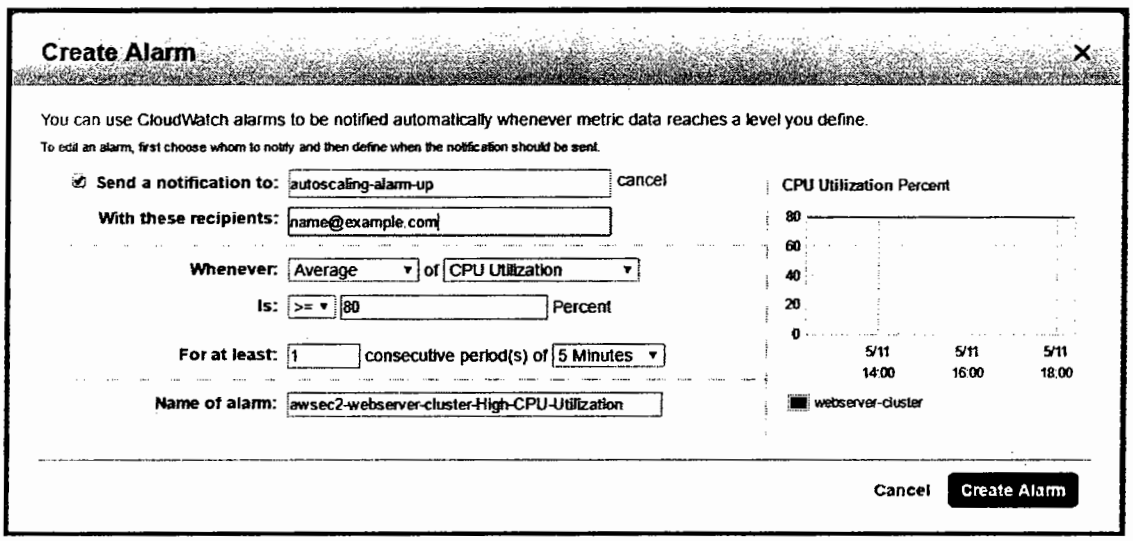


5. The Auto Scaling group policies allow you to automatically increase or decrease the group size based upon policies you define. In order to establish an Increase Group size or Decrease Group Size policy, you must create a CloudWatch Alarm and then define which action should be taken if it is triggered.

   Click *Add new alarm* under the *Increase Group Size* section. A *Create Alarm* dialogue box will pop up.

   If you want to receive a notification when the alarm is triggered, you need to set up an **SNS topic.** Check the *Send a notification to:* checkbox. Type in a name (e.g., "autoscaling-alarm-up") for the SNS topic and enter at least one email address in the recipients box.

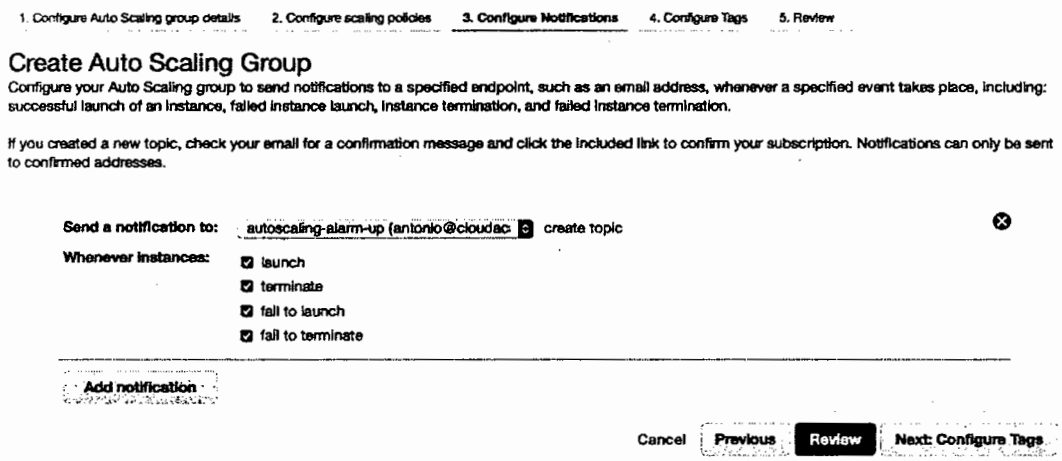
Select a metric (e.g., Average, CPU Utilization) and a constraint (e.g., >= 80 percent). Select a count and an interval (e.g., For at least **1** consecutive period of **5 minutes**). Choose a name for the alarm, and then click **Create Alarm**.



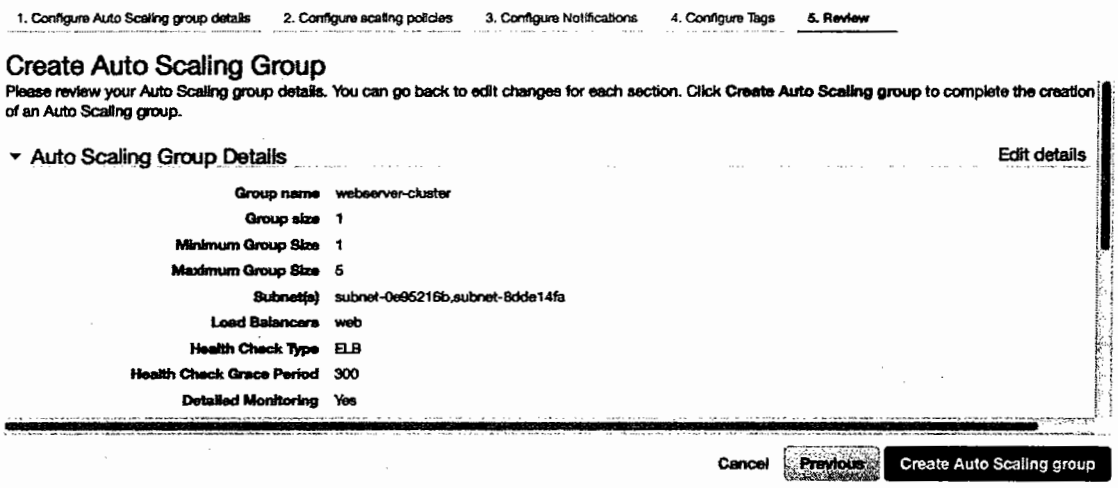
Create another alarm with whatever settings you choose for the Decrease Group Size. Click**Next: Configure Notifications**.

6. *Configure Notifications* will notify you whenever an Auto Scaling Group instance is launched or terminated -- with or without success.

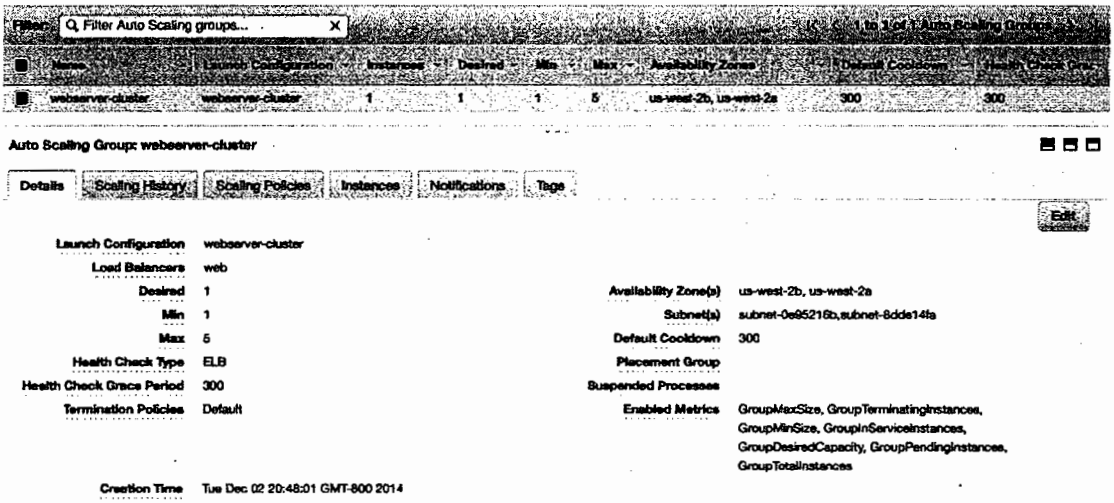
7. Click **Add notification**. You can use one of the same SNS topics previously created for the CloudWatch alarms. When you're done, click the blue **Review** button.



8. The **Review** tab allows you to review all the selected options. When you are satisfied, start the creation of your cluster by clicking on **Create Auto Scaling group**.



9. In a few minutes your cluster will be deployed and your EC2 instances will be ready to.



10. By opening the **Load Balancers** section, selecting your previously created ELB, and then opening the Instances tab, you can see the new Auto Scaling instance(s) automatically added to the ELB configuration.

Create Load Balancer | Actions

Filter: Search Load Balancers

1 to 1 of 1

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Check	C
web	web-1306826351.us-west-2....	80 (HTTP) forwarding to 80 (...	us-west-2c, us-west-2b...	1 instance	HTTP:80/index.html	D

Load balancer: web

Description | Instances | Health Check | Monitoring | Security | Listeners | Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-e0dd50ea		us-west-2b	InService	Remove from Load Balancer

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-west-2c	subnet-4bd03b12	172.31.0.0/20	0	No (Availability Zone contains no healthy instances)	Remove from Load Balancer
us-west-2b	subnet-0e95216b	172.31.32.0/20	1	Yes	Remove from Load Balancer
us-west-2a	subnet-8dde14fa	172.31.16.0/20	0	No (Availability Zone contains no healthy instances)	Remove from Load Balancer