AUTOSCALING

- Autoscaling is a service provided by AWS that automatically increasing the number of EC2 instances during demand spikes to maintain performance and decrease the number of EC2 instances during lulls to reduce costs.

- This autoscaling will increase or decrease the number of instances based on chosen **Cloudwatch** metrics.

- Autoscaling helps you to maintain application availability as per the conditions you define.

For example: If your application's demand increases unexpectedly, autoscaling can automatically scale up (add instance) to meet the demand and terminate instances when the demand decreases. This is known as "elasticity" in the AWS environment.

- Autoscaling is well suited both to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

-Autoscaling is also used to help ensure that you are running your desired number of amazon EC2 instances.

<u>Components of Autoscaling</u>

Autoscaling has 2 main components

1) Launch configuration
2) Auto scaling Group

Launch Configuration : The "EC2" template used when the auto scaling group needs to provision an additional instance (i.e. AMI, instance type, user-data, storage, tags, security groups etc)

Auto scaling Group :

All the rules and settings that govern if/when an EC2 instance is automatically provisioned or terminated.

For example

1) number of MIN & MAX allows instances
2) VPC & AZs to launch instances into
3) if provisioned instances should receive traffic from a ELB
4) Scaling policies (cloudwatch metrics thresholds that trigger scaling)
5) SNS notifications (to keep you informed when scaling occurs)

NOTE: To make Highly Available & Fault tolerant architecture, it MUST have and ELB serving traffic to and AutoScaling Group with a MIN of two instances located in separate availability zones.

Learn to automatically scale up or down your EC2 infrastructure using Auto Scaling Groups

This lab introduces the basics of Auto Scaling in Amazon Web Services. The Amazon Web Services (AWS) Auto Scaling service automatically adds or removes compute resources allocated for your cloud application, in response to changes in demand. For applications configured to run on a cloud infrastructure, scaling is an important part of cost control and resource management.

Scaling is the ability to increase or decrease the compute capacity of your application either by changing the number of servers (horizontal scaling) or by changing the size of the servers (vertical scaling).

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to the defined conditions. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. AS is well suited to applications that have stable demand patterns, or that experience hourly, daily, or weekly variability in usage.

By completing this lab you will learn about:

- Configuring Auto Scaling to automatically launch web server instances
- Building an elastic cluster by integrating Auto Scaling with an Elastic Load Balancer
- Setting CloudWatch alarms to automatically adjust the size of the web farm based on CPU utilization
- Utilizing Auto Scaling to ensure the availability of steady state resources

AGENDA:

You'll build and learn following these steps:

Log In to the Amazon Web Service Console
Your first step to start the laboratory experience

Auto Scaling Overview
Give an overview about the creation process of an Auto Scaling Group

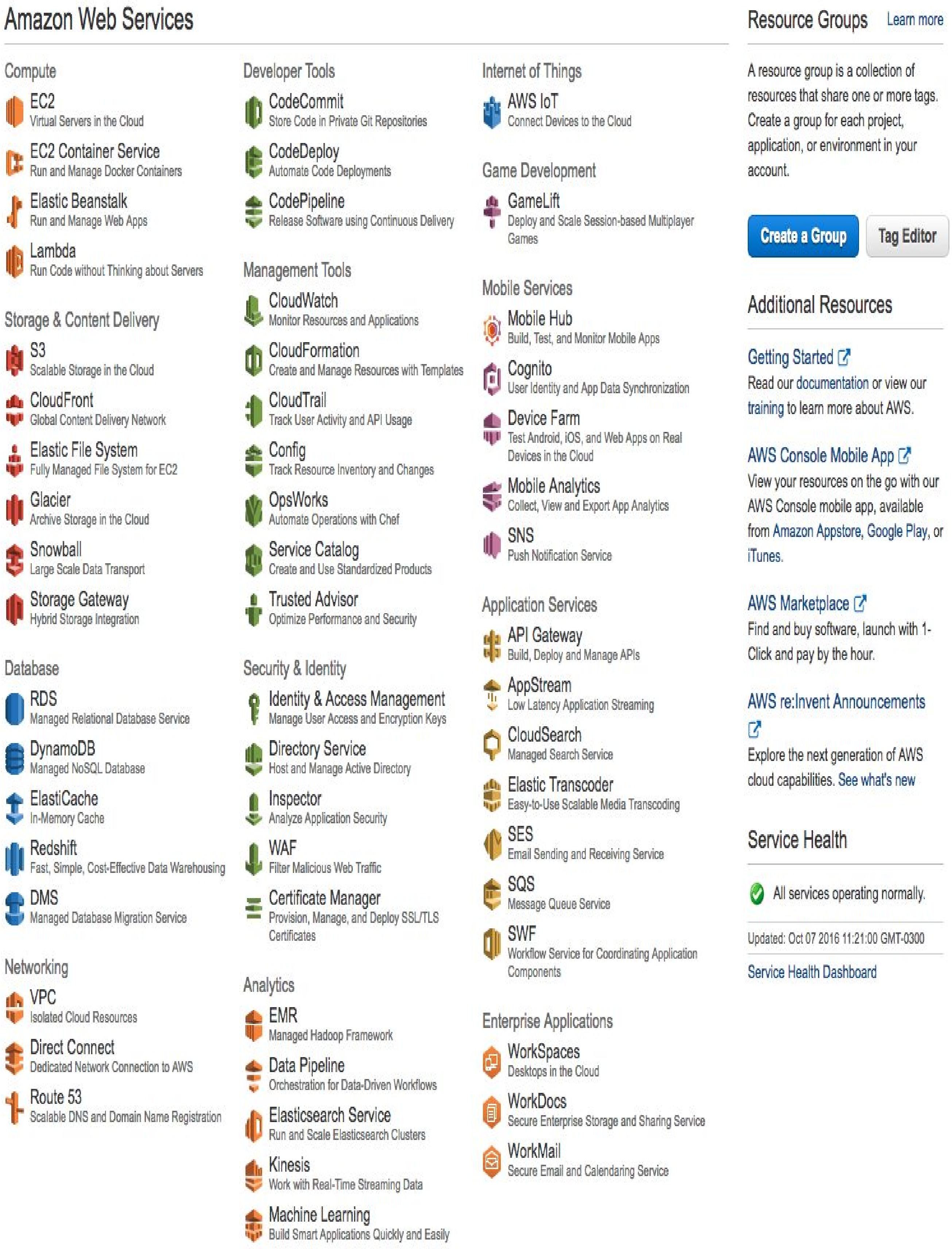
Create a load balancer using ELB
How to create a load balancer using Elastic Load Balancing service.

Create a Launch Configuration
How to create an Auto Scaling Launch Configuration

Create an Auto Scaling Group
How to create an Auto Scaling Group by using a specific Launch Configuration

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.



Amazon Web Services

Compute

EC2
Virtual Servers in the Cloud

EC2 Container Service
Run and Manage Docker Containers

Elastic Beanstalk
Run and Manage Web Apps

Lambda
Run Code without Thinking about Servers

Storage & Content Delivery

S3
Scalable Storage in the Cloud

CloudFront
Global Content Delivery Network

Elastic File System
Fully Managed File System for EC2

Glacier
Archive Storage in the Cloud

Snowball
Large Scale Data Transport

Storage Gateway
Hybrid Storage Integration

Database

RDS
Managed Relational Database Service

DynamoDB
Managed NoSQL Database

ElastiCache
In-Memory Cache

Redshift
Fast, Simple, Cost-Effective Data Warehousing

DMS
Managed Database Migration Service

Networking

VPC
Isolated Cloud Resources

Direct Connect
Dedicated Network Connection to AWS

Route 53
Scalable DNS and Domain Name Registration

Developer Tools

CodeCommit
Store Code in Private Git Repositories

CodeDeploy
Automate Code Deployments

CodePipeline
Release Software using Continuous Delivery

Management Tools

CloudWatch
Monitor Resources and Applications

CloudFormation
Create and Manage Resources with Templates

CloudTrail
Track User Activity and API Usage

Config
Track Resource Inventory and Changes

OpsWorks
Automate Operations with Chef

Service Catalog
Create and Use Standardized Products

Trusted Advisor
Optimize Performance and Security

Security & Identity

Identity & Access Management
Manage User Access and Encryption Keys

Directory Service
Host and Manage Active Directory

Inspector
Analyze Application Security

WAF
Filter Malicious Web Traffic

Certificate Manager
Provision, Manage, and Deploy SSL/TLS Certificates

Analytics

EMR
Managed Hadoop Framework

Data Pipeline
Orchestration for Data-Driven Workflows

Elasticsearch Service
Run and Scale Elasticsearch Clusters

Kinesis
Work with Real-Time Streaming Data

Machine Learning
Build Smart Applications Quickly and Easily

Internet of Things

AWS IoT
Connect Devices to the Cloud

Game Development

GameLift
Deploy and Scale Session-based Multiplayer Games

Mobile Services

Mobile Hub
Build, Test, and Monitor Mobile Apps

Cognito
User Identity and App Data Synchronization

Device Farm
Test Android, iOS, and Web Apps on Real Devices in the Cloud

Mobile Analytics
Collect, View and Export App Analytics

SNS
Push Notification Service

Application Services

API Gateway
Build, Deploy and Manage APIs

AppStream
Low Latency Application Streaming

CloudSearch
Managed Search Service

Elastic Transcoder
Easy-to-Use Scalable Media Transcoding

SES
Email Sending and Receiving Service

SQS
Message Queue Service

SWF
Workflow Service for Coordinating Application Components

Enterprise Applications

WorkSpaces
Desktops in the Cloud

WorkDocs
Secure Enterprise Storage and Sharing Service

WorkMail
Secure Email and Calendaring Service

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Additional Resources

Getting Started

Read our documentation or view our training to learn more about AWS.

AWS Console Mobile App

View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace

Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements

Explore the next generation of AWS cloud capabilities. See what's new

Service Health

✓ All services operating normally.

Updated: Oct 07 2016 11:21:00 GMT-0300

Service Health Dashboard

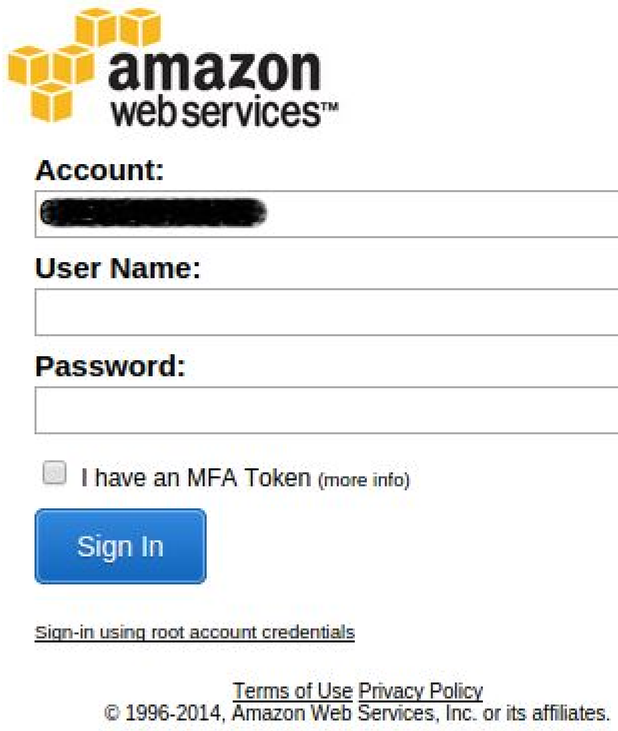
The AWS Management Console is a web control panel for managing all your AWS resources, from EC2 instances to SNS topics. The console enables cloud management for all aspects of the AWS account, including managing security credentials, or even setting up new IAM Users.

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

We created a Console User just for you. Log in with the username `student` and the password `Ca1_K0Q2g0ug` .



amazon web services™

Account:

[redacted]

User Name:

Password:

□ I have an MFA Token (more info)

Sign In

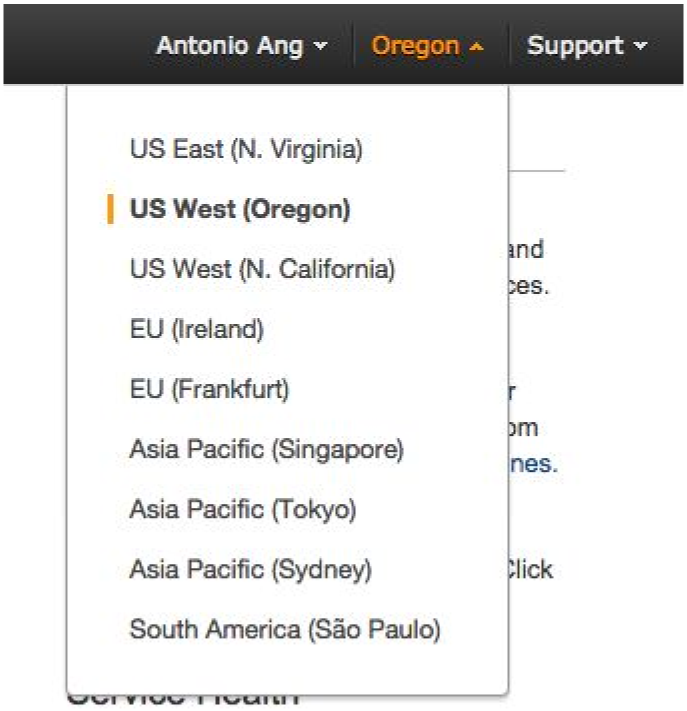
Sign-in using root account credentials

Terms of Use Privacy Policy
© 1996-2014, Amazon Web Services, Inc. or its affiliates.

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

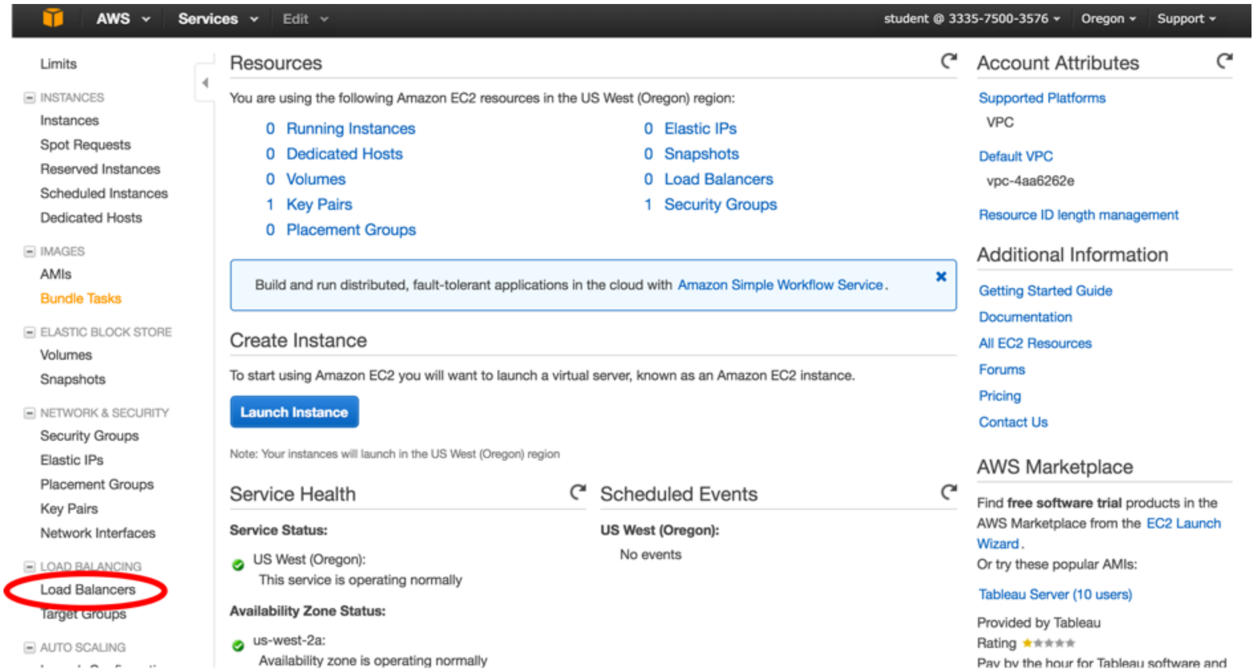
Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater fault tolerance in your applications and seamlessly provides the correct amount of load balancing capacity needed in response to incoming application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

1. Select EC2 from the AWS Service List

2. 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.



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
1 Key Pairs	1 Security Groups
0 Placement Groups	

Build and run distributed, fault-tolerant applications in the cloud with Amazon Simple Workflow Service.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (Oregon) region

Service Health

Service Status:

US West (Oregon):
This service is operating normally

Availability Zone Status:

us-west-2a:
Availability zone is operating normally

Scheduled Events

US West (Oregon):
No events

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-4aa6262e

Resource ID length management

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

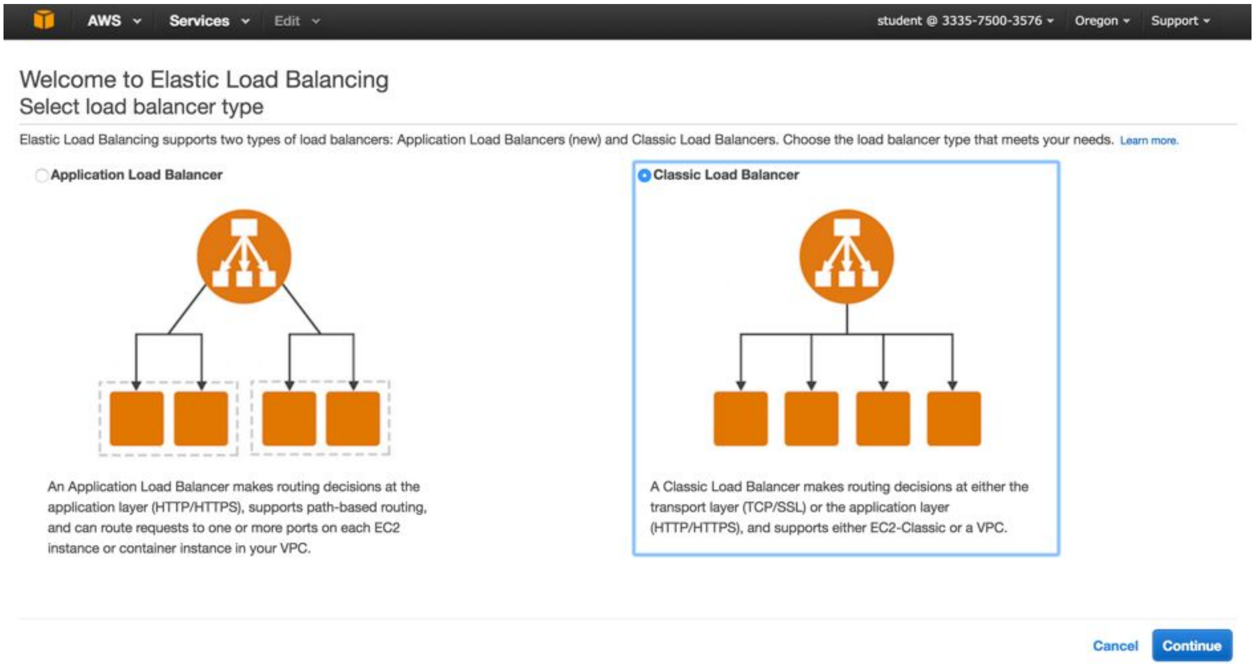
Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

Tableau Server (10 users)

Provided by Tableau
Rating ★★★★★
Pay by the hour for Tableau software and

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

4. Select the **Classic Load Balancer** option and click Continue



Welcome to Elastic Load Balancing
Select load balancer type

Elastic Load Balancing supports two types of load balancers: Application Load Balancers (new) and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more.

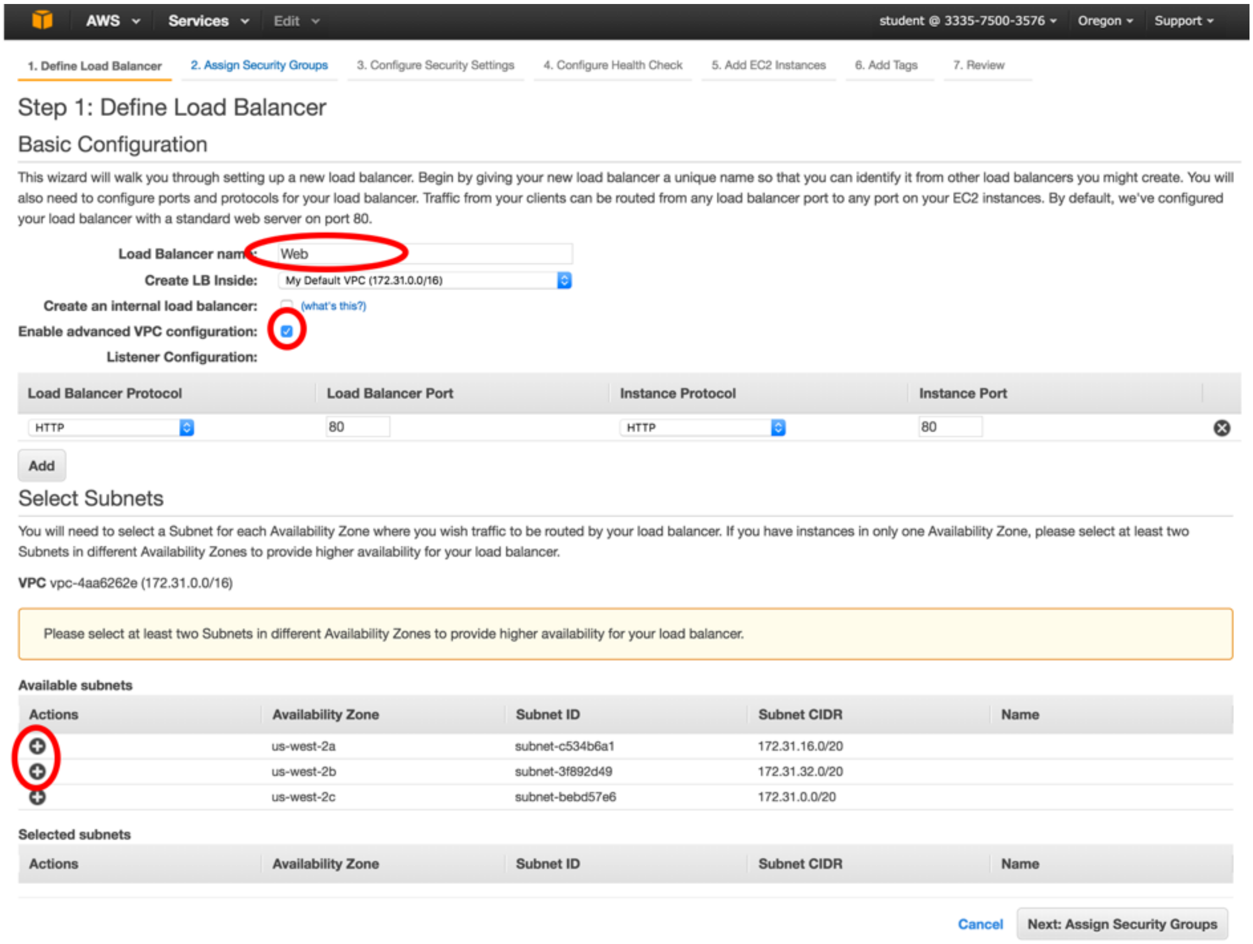
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each EC2 instance or container instance in your VPC.

A Classic Load Balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS), and supports either EC2-Classic or a VPC.

5. On the **Define Load Balancer** step, type a load balancer name (e.g., "web")

6. Select **Enable advanced VPC configuration**

7. Select two subnets, one from the **us-west-2a Availability Zone** and one from the **us-west-2b Availability Zone**



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

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-4aa6262e (172.31.0.0/16)

Please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊕	us-west-2a	subnet-c534b6a1	172.31.16.0/20	
⊕	us-west-2b	subnet-3f892d49	172.31.32.0/20	
⊕	us-west-2c	subnet-bebd57e6	172.31.0.0/20	

Selected subnets

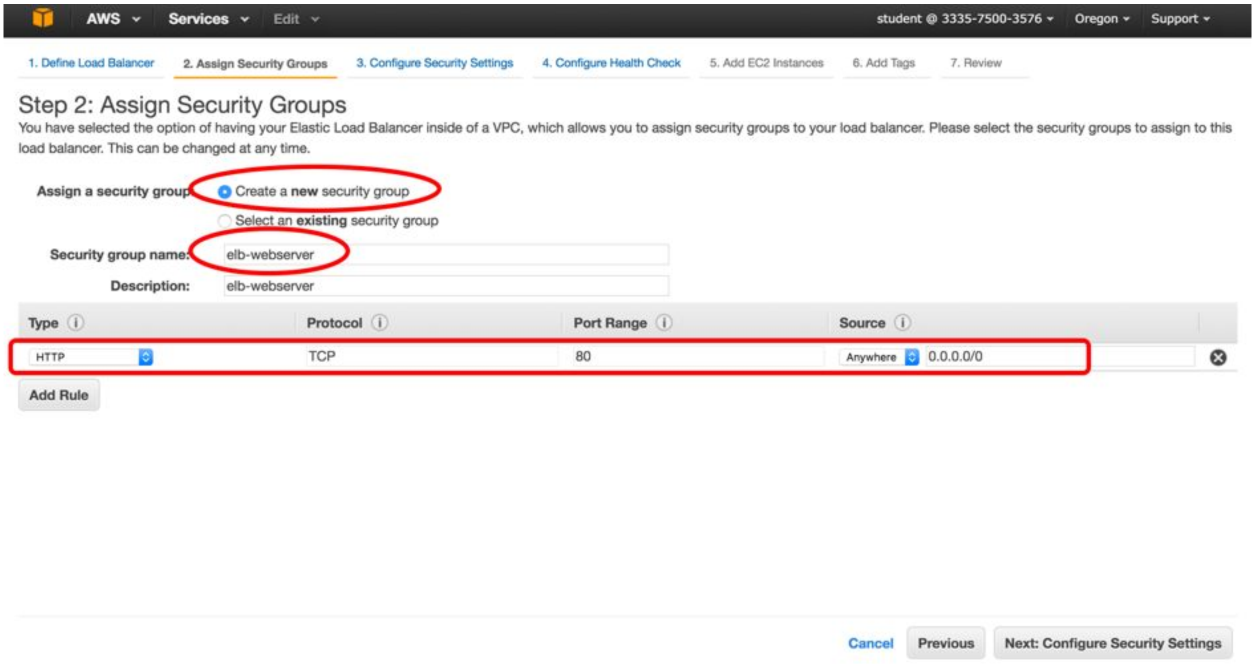
Actions	Availability Zone	Subnet ID	Subnet CIDR	Name

8. Then click the **Next: Assign Security Groups** button

9. In the **Assign Security Groups** section, select ***Create a new security group***

10. Type a Security group name (e.g., "elb-webserver") and a description

11. Create a single firewall rule of *type* **HTTP**, *protocol* **TCP**, *port range* **80**, and *source* **Anywhere**



Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

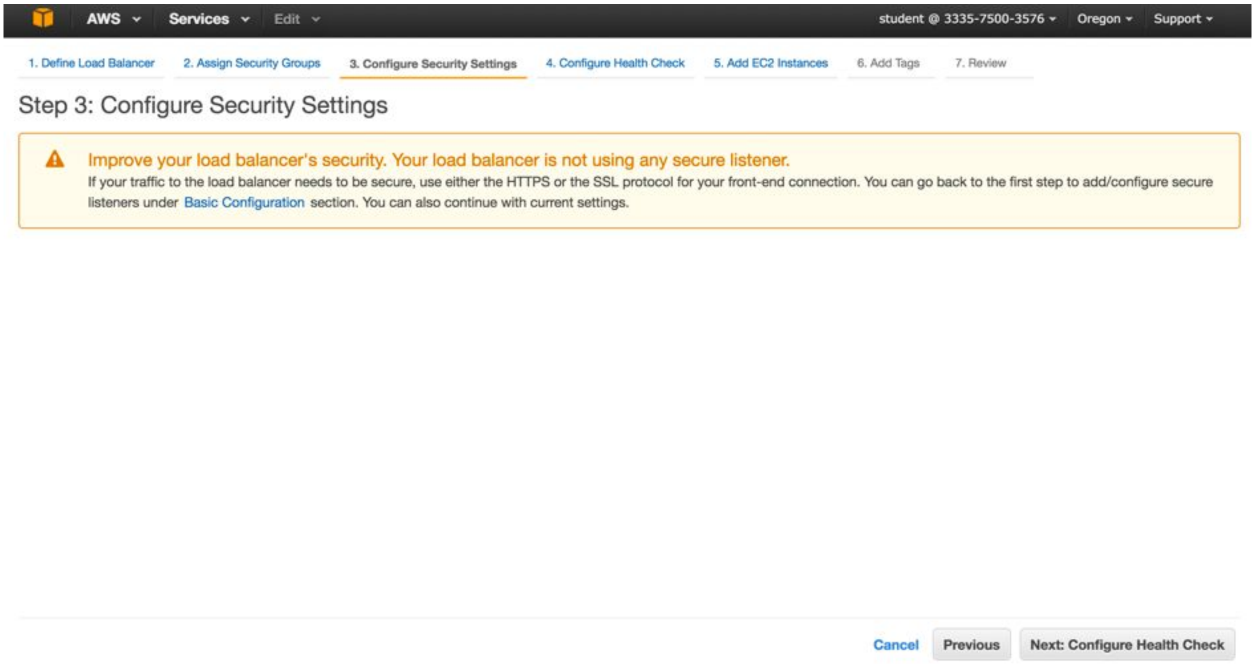
Security group name: elb-webserver

Description: elb-webserver

Type	Protocol	Port Range	Source
HTTP	TCP	80	Anywhere 0.0.0.0/0

12. Click **Next: Configure Security Settings**.

13. 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

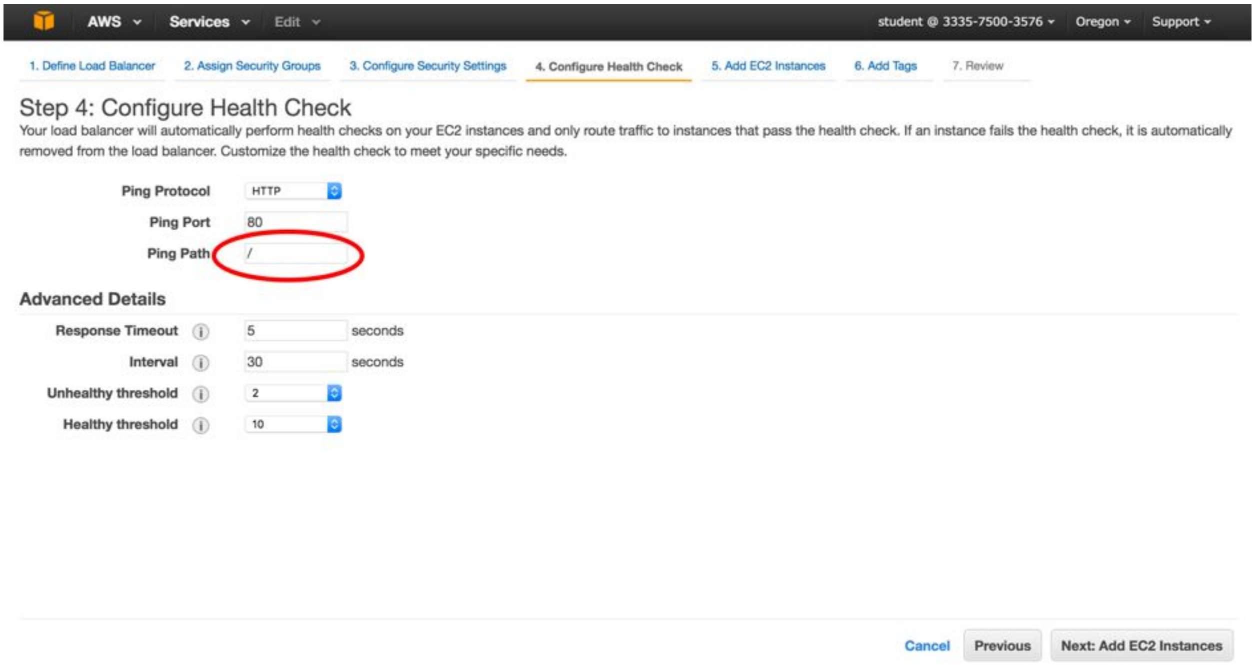


Step 3: Configure Security Settings

Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use either the HTTPS or the SSL protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings.

14. Click **Next: Configure Health Check**.

15. In the **Configure Health Check** section, replace the default value of **Ping Path** with a single forward slash ("/")



Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

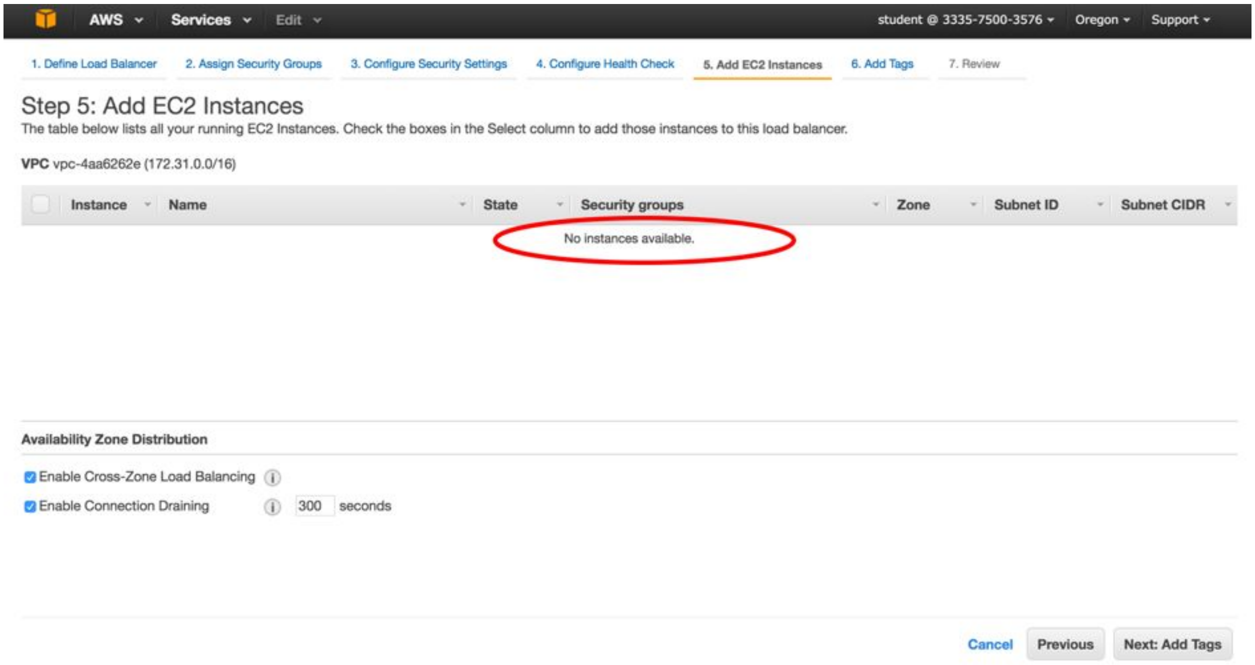
Ping Protocol: HTTP
Ping Port: 80
Ping Path: /

Advanced Details

Response Timeout: 5 seconds
Interval: 30 seconds
Unhealthy threshold: 2
Healthy threshold: 10

16. Click **Next: Add EC2 Instances**

17. 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



Step 5: Add EC2 Instances

The table below lists all your running EC2 Instances. Check the boxes in the Select column to add those instances to this load balancer.

VPC vpc-4aa6262e (172.31.0.0/16)

□	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
			No instances available.				

Availability Zone Distribution

☑ Enable Cross-Zone Load Balancing

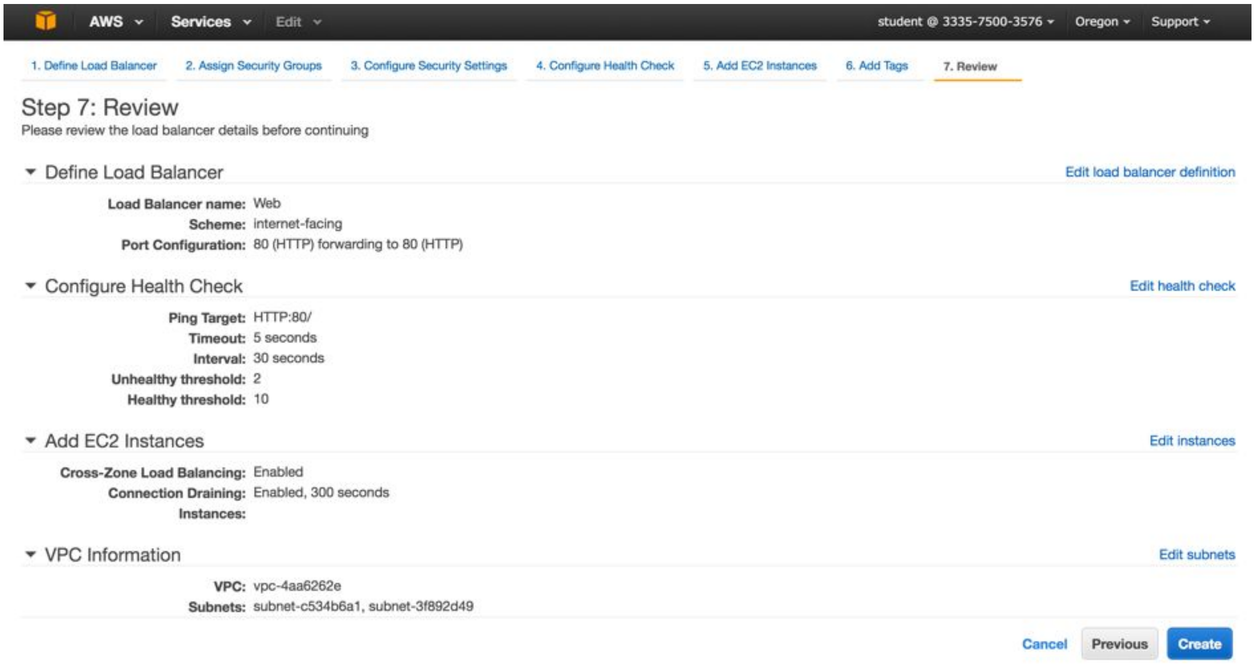
☑ Enable Connection Draining 300 seconds

18. Click **Next: Add Tags to continue**

19. You may leave the fields blank in the **Add Tags** section

20. Click the **Review and Create** button to continue

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



Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer

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

Configure Health Check

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

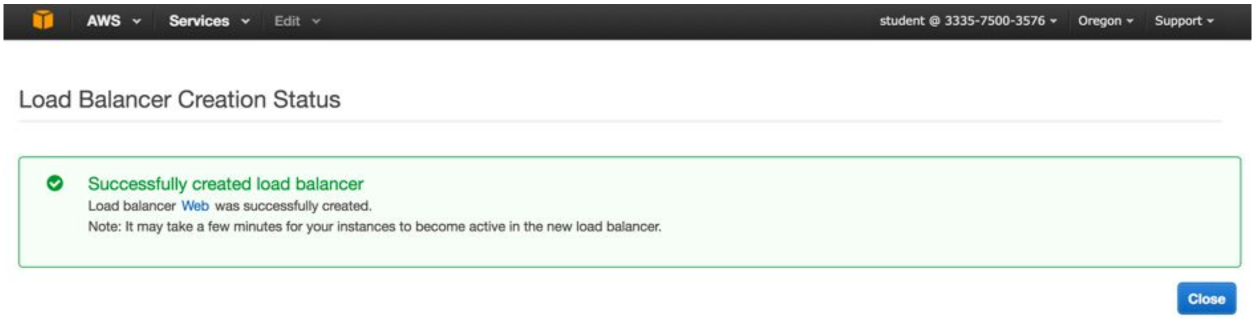
Add EC2 Instances

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

VPC Information

VPC: vpc-4aa6262e
Subnets: subnet-c534b6a1, subnet-3f892d49

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



Load Balancer Creation Status

Successfully created load balancer

Load balancer Web was successfully created.

Note: It may take a few minutes for your instances to become active in the new load balancer.

Step 4 Create a Launch Configuration

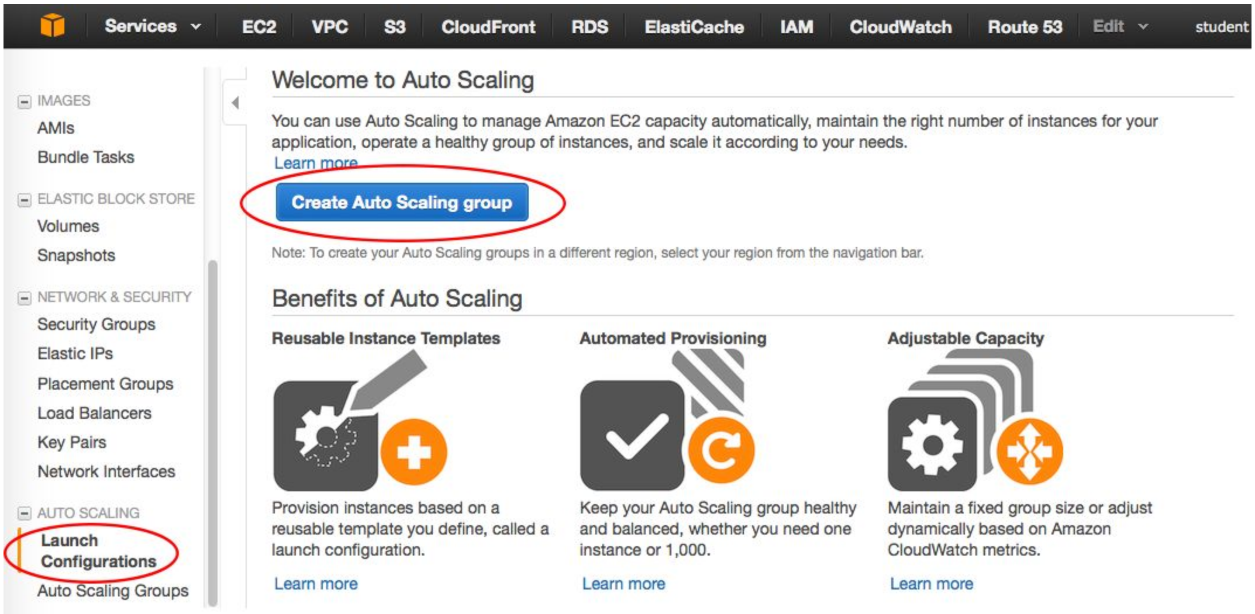
A **Launch Configuration** is a template that the Auto Scaling group uses to launch Amazon EC2 instances. If you've launched an individual EC2 instance before, you've already walked through the process of defining compute characteristics such as the instance type, security groups, and configuration scripts.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

1.Navigate to the EC2 service from the AWS dashboard:

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



Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs.

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

Benefits of Auto Scaling

Reusable Instance Templates

Provision instances based on a reusable template you define, called a launch configuration.

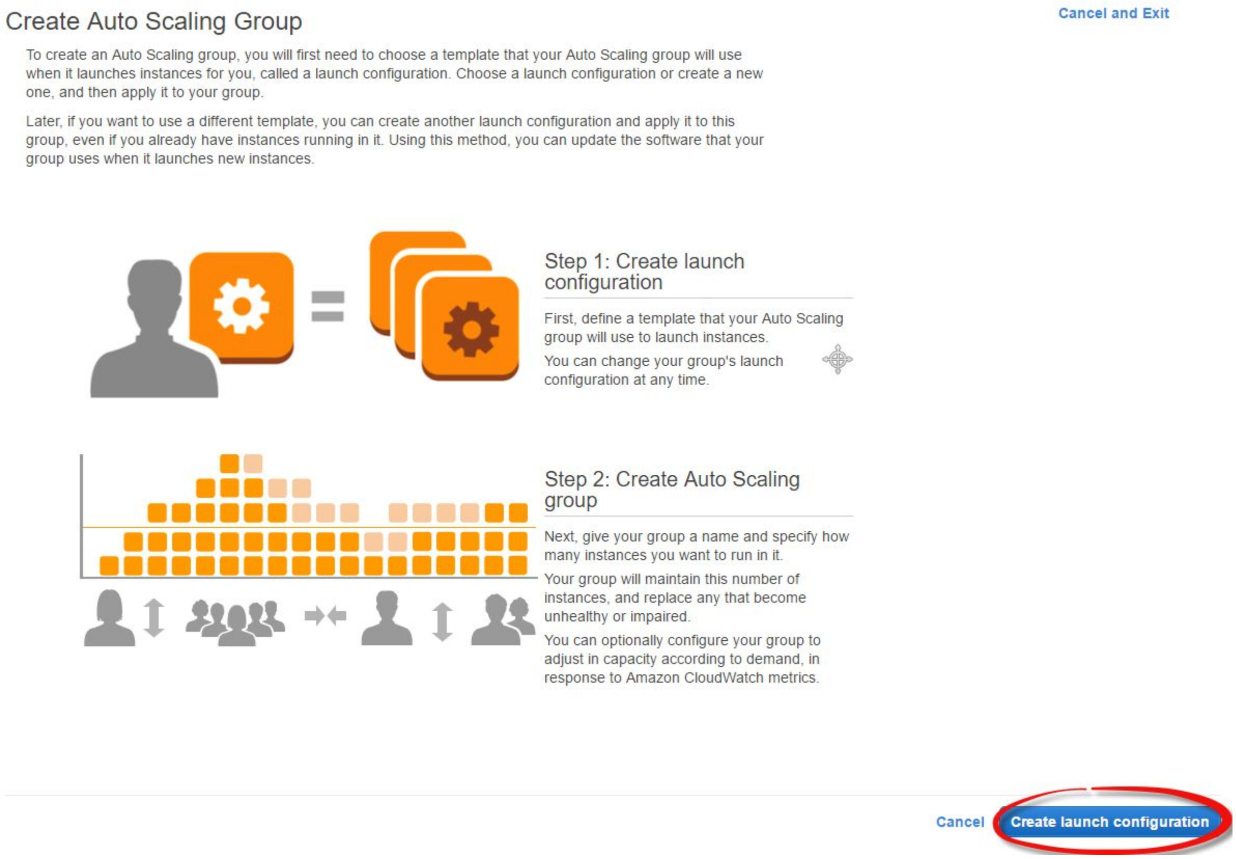
Automated Provisioning

Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.

Adjustable Capacity

Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.

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



Create Auto Scaling Group

Cancel and Exit

To create an Auto Scaling group, you will first need to choose a template that your Auto Scaling group will use when it launches instances for you, called a launch configuration. Choose a launch configuration or create a new one, and then apply it to your group.

Later, if you want to use a different template, you can create another launch configuration and apply it to this group, even if you already have instances running in it. Using this method, you can update the software that your group uses when it launches new instances.

Step 1: Create launch configuration

First, define a template that your Auto Scaling group will use to launch instances.

You can change your group's launch configuration at any time.

Step 2: Create Auto Scaling group

Next, give your group a name and specify how many instances you want to run in it.

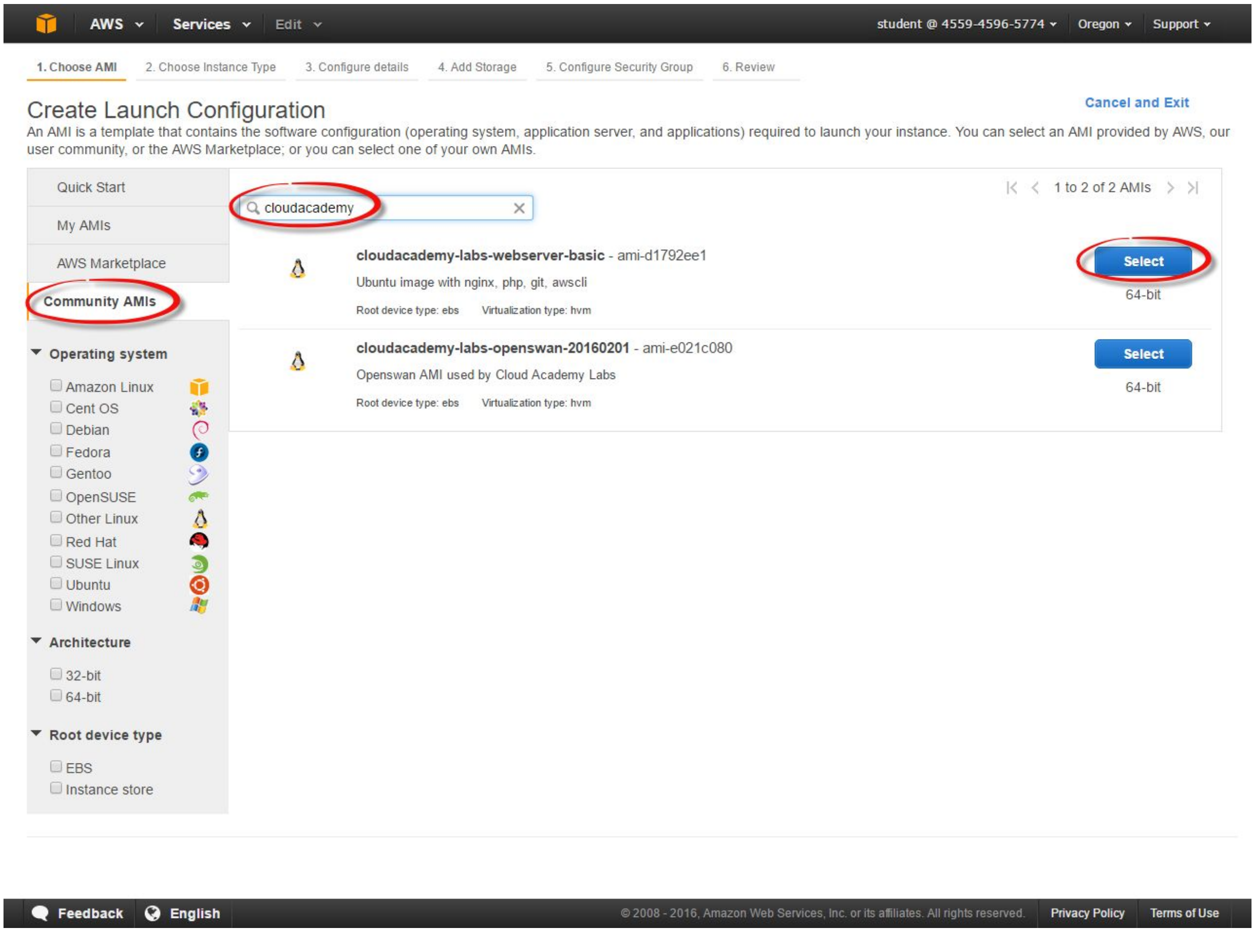
Your group will maintain this number of instances, and replace any that become unhealthy or impaired.

You can optionally configure your group to adjust in capacity according to demand, in response to Amazon CloudWatch metrics.

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 Cloud Academy DevOps team created a specific AMI for this laboratory. You can find it among the Community AMIs by searching for the word "cloudacademy" in the AMI search box.

3. Select the "**cloudacademy-labs-webserver-basic" -** (ami-d1792dee1) AMI and click Select.



Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

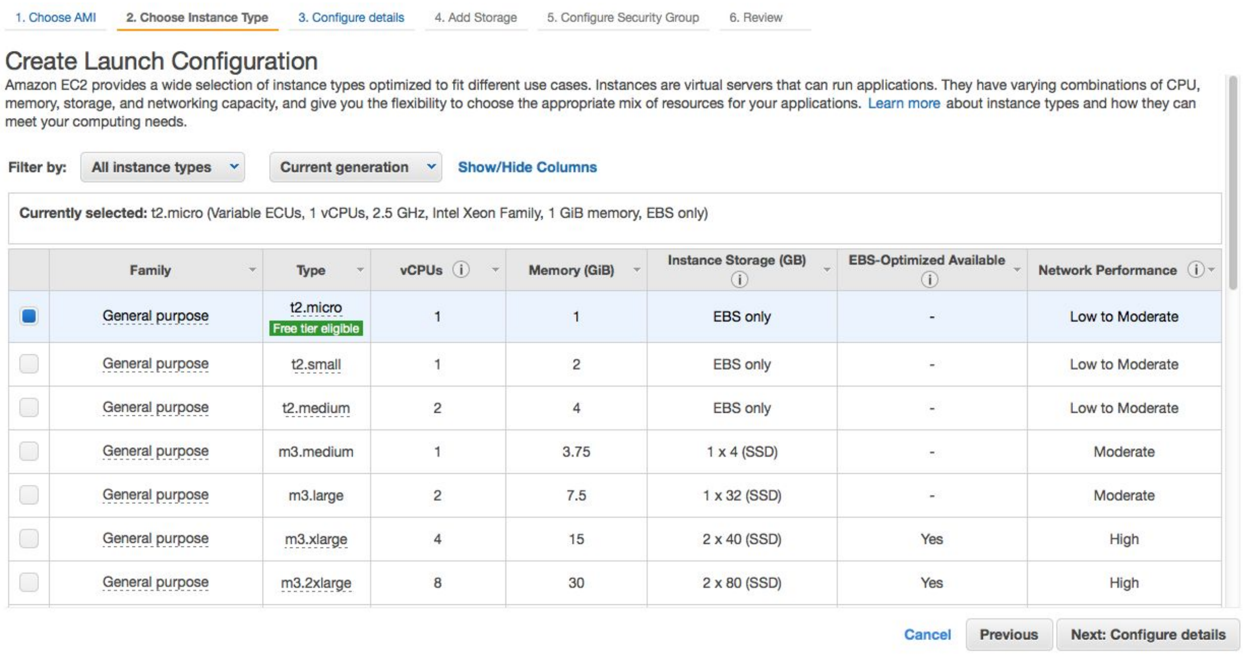
cloudacademy

cloudacademy-labs-webserver-basic - ami-d1792ee1
Ubuntu image with nginx, php, git, awscli
Root device type: ebs Virtualization type: hvm
64-bit

cloudacademy-labs-openswan-20160201 - ami-e021c080
Openswan AMI used by Cloud Academy Labs
Root device type: ebs Virtualization type: hvm
64-bit

The next step is choosing the instance type

4. Select the t2.micro type and click on the **Next: Configure details** button.



Create Launch Configuration

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

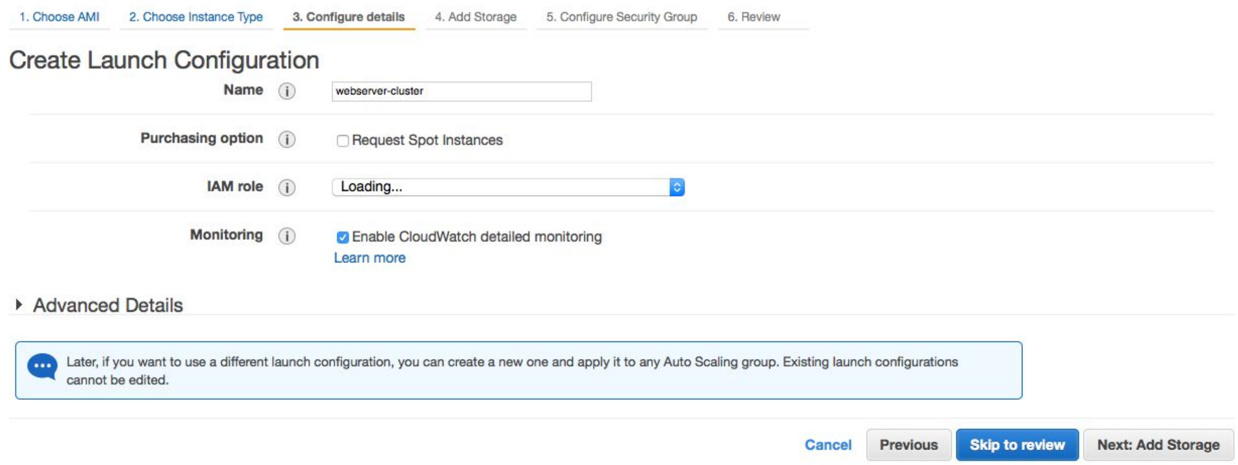
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
□	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
□	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
□	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

5. The **Configure details** step asks you to name your launch configuration, enter a friendly name (e.g., webserver-cluster')

6. Enable detailed monitoring

7. Click on **Next: Add Storage**



Create Launch Configuration

Name: webserver-cluster

Purchasing option: □ Request Spot Instances

IAM role: Loading...

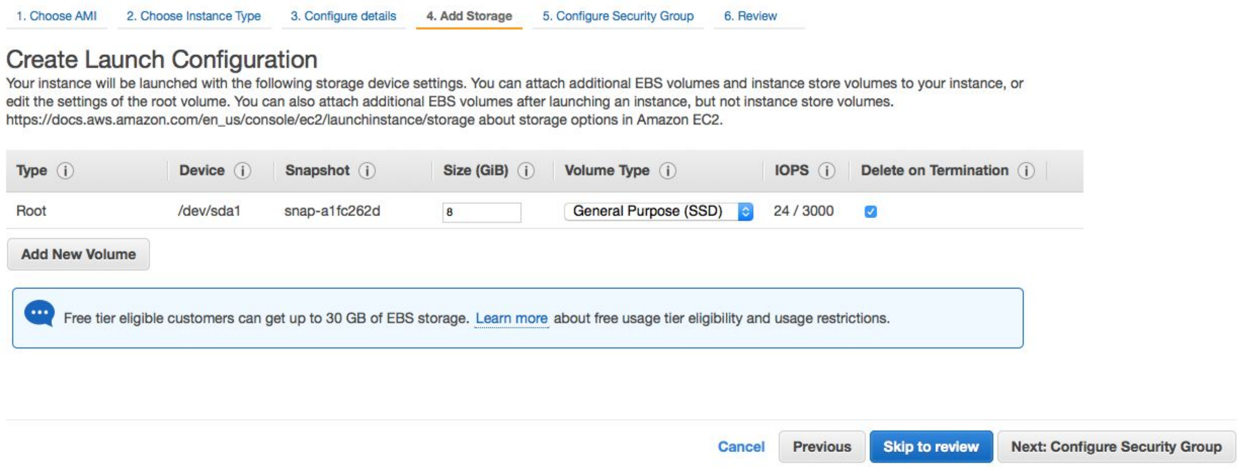
Monitoring: ☑ Enable CloudWatch detailed monitoring
Learn more

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

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.

8. 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 Launch Configuration

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. https://docs.aws.amazon.com/en_us/console/ec2/launchinstance/storage about storage options in Amazon EC2.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-a1fc262d	8	General Purpose (SSD)	24 / 3000	☑

Free tier eligible customers can get up to 30 GB of EBS storage. Learn more about free usage tier eligibility and usage restrictions.

9. **Create a new Security Group** for your Auto Scaling Group.

10. Choose a name (e.g., Webserver-cluster) and description

11. Add 2 rules:

1st Rule

Type=SSH

Protocol=TCP

Port Range=22

Source=My IP

2nd Rule

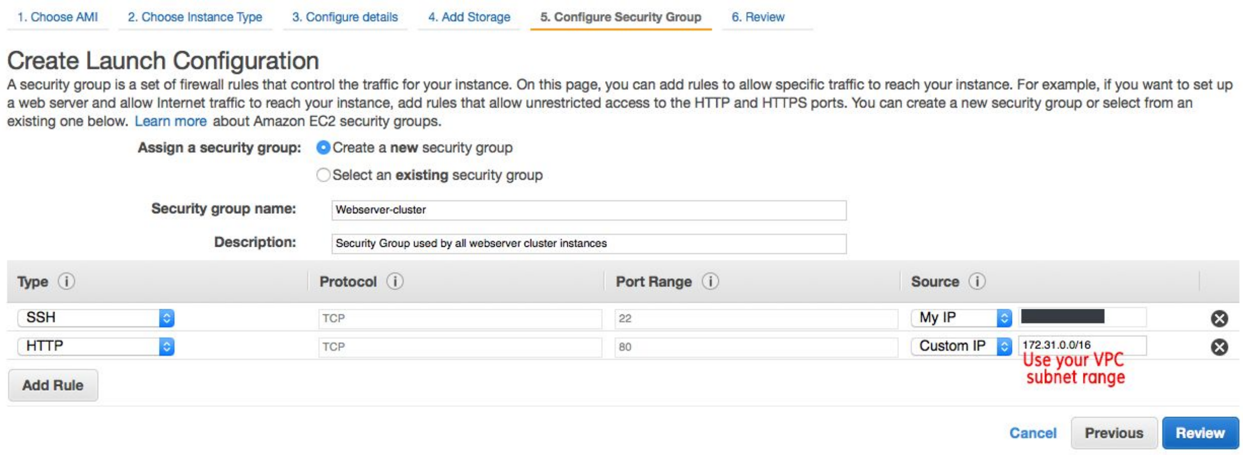
Type=HTTP

Protocol=TCP

Port Range=80

Source=Custom IP (enter 172.31.0.0/16 for the IP)

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.



Create Launch Configuration

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: Webserver-cluster

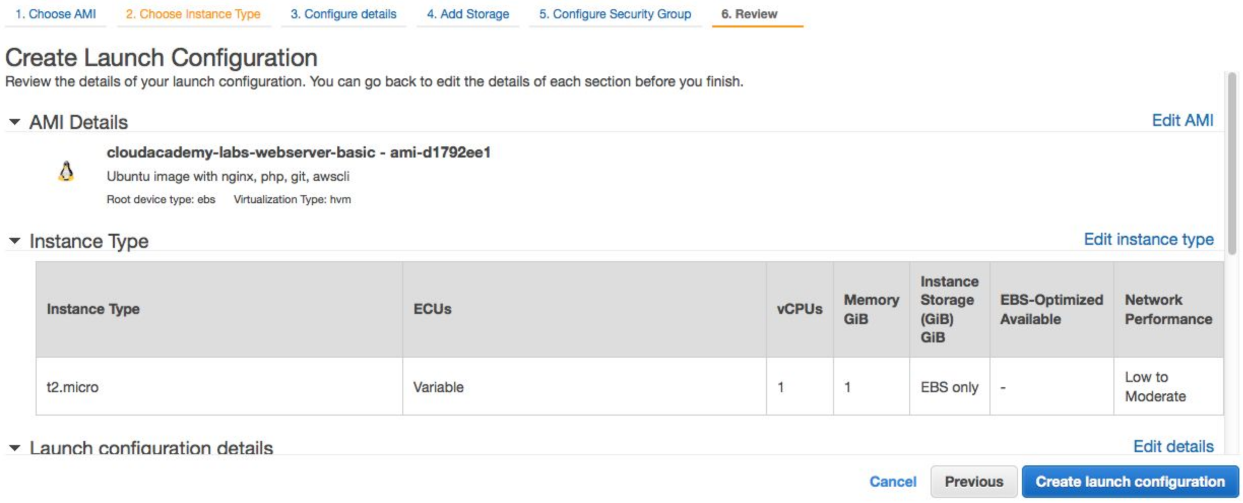
Description: Security Group used by all webserver cluster instances

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP [redacted]
HTTP	TCP	80	Custom IP 172.31.0.0/16

Use your VPC subnet range

12. Click the blue **Review** button.

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



Create Launch Configuration

Review the details of your launch configuration. You can go back to edit the details of each section before you finish.

AMI Details

cloudacademy-labs-webserver-basic - ami-d1792ee1

Ubuntu image with nginx, php, git, awscli

Root device type: ebs Virtualization Type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB) GiB	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

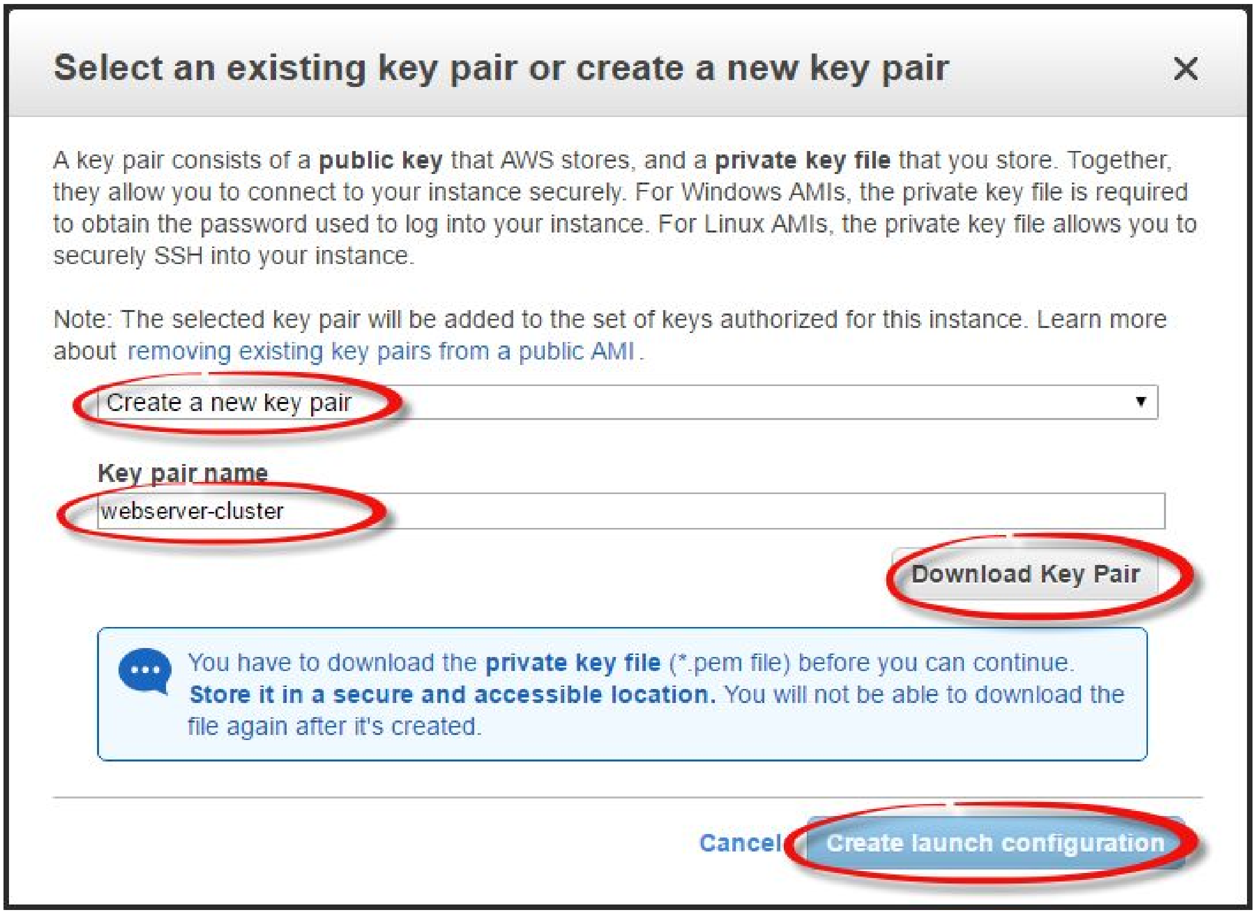
Launch configuration details

You will be presented with the *Select 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.

14. Select **Create a new key pair** from the first drop-down menu and type in a Key pair name (e.g., webserver-cluster).

15. Click the **Download Key Pair** button

16. Then click the **Create Launch Configuration** button in this dialogue box



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name: webserver-cluster

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

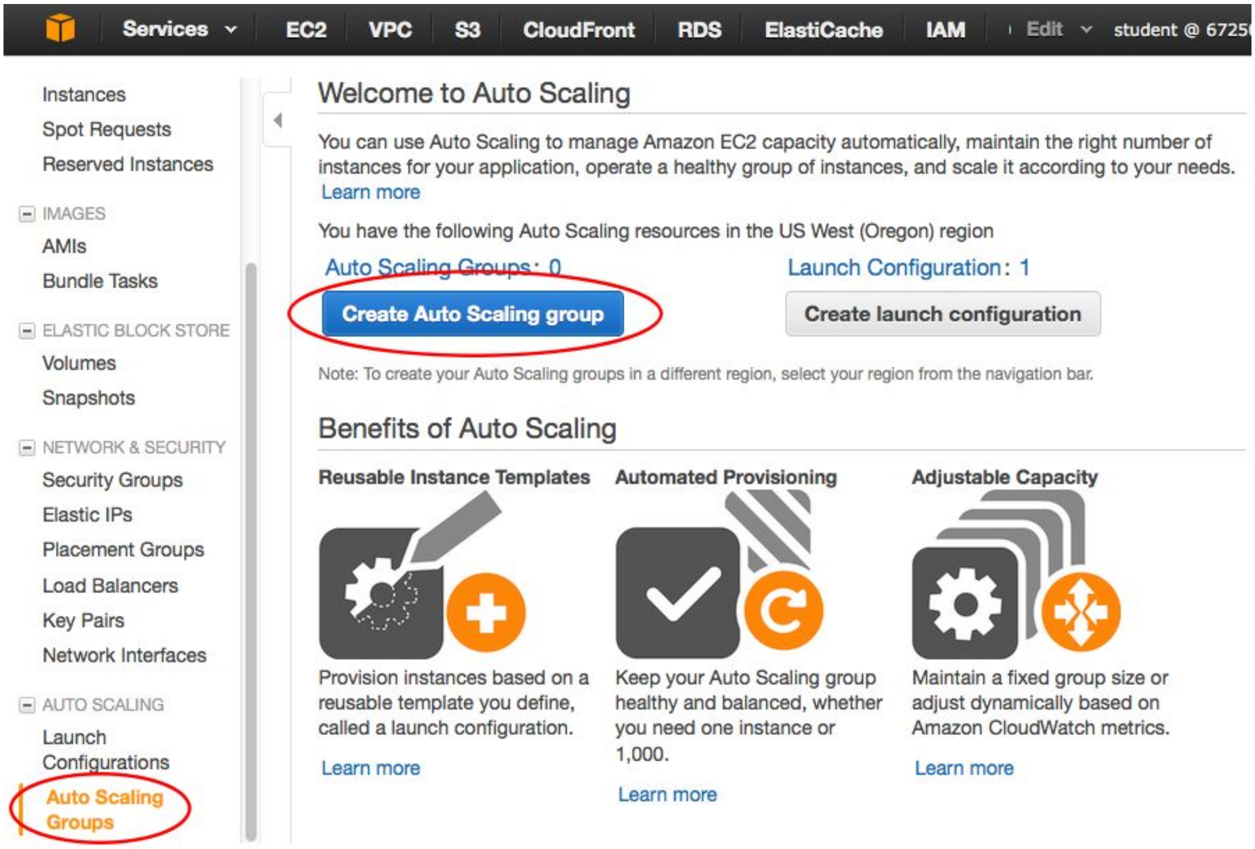
17. Next a screen will appear that tries to configure an Auto Scaling Group, however at this point click on Cancel as we will create this from scratch in the next step

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.



Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs. Learn more

You have the following Auto Scaling resources in the US West (Oregon) region

Auto Scaling Groups: 0

Launch Configuration: 1

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

Benefits of Auto Scaling

Reusable Instance Templates

Provision instances based on a reusable template you define, called a launch configuration.

Learn more

Automated Provisioning

Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.

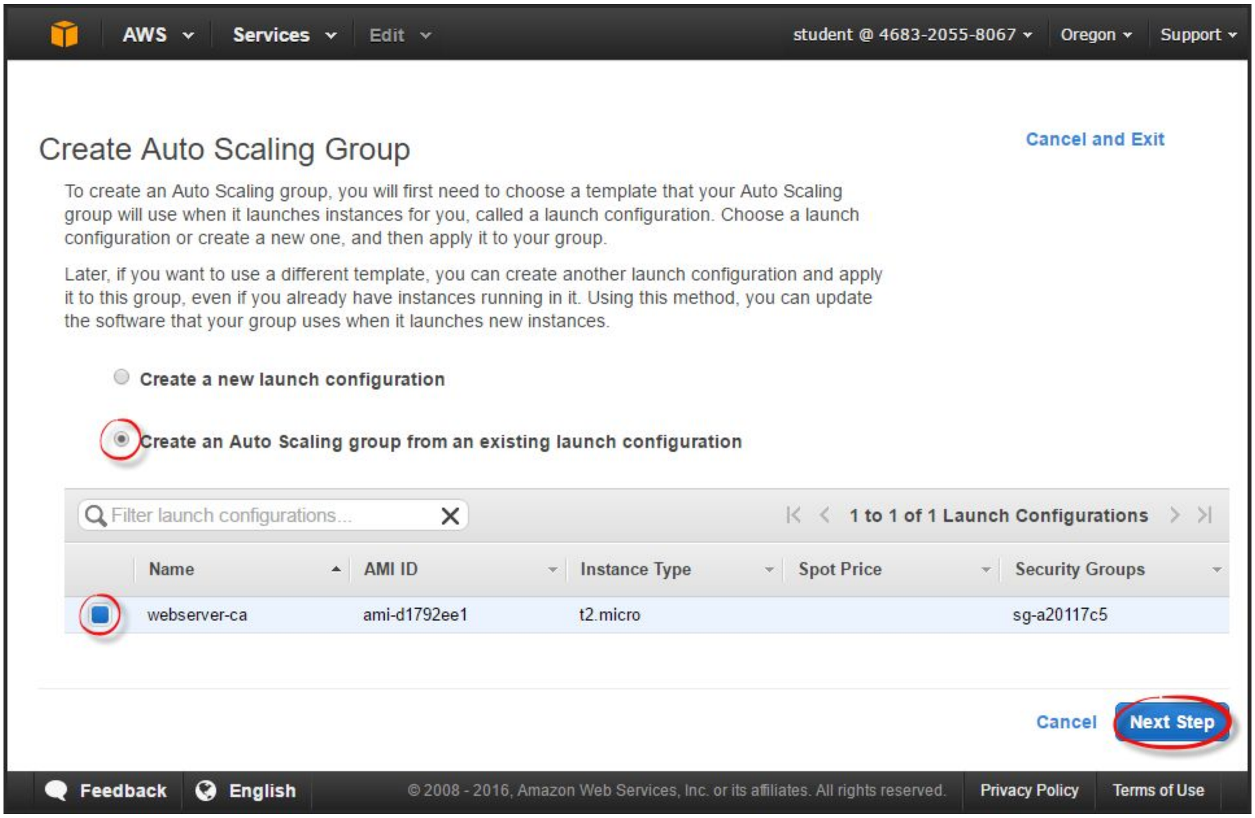
Learn more

Adjustable Capacity

Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.

Learn more

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


AWS
Services
Edit
student @ 4683-2055-8067
Oregon
Support
Cancel and Exit
Filter launch configurations...
1 to 1 of 1 Launch Configurations
Cancel
Next Step
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Create Auto Scaling Group

To create an Auto Scaling group, you will first need to choose a template that your Auto Scaling group will use when it launches instances for you, called a launch configuration. Choose a launch configuration or create a new one, and then apply it to your group.

Later, if you want to use a different template, you can create another launch configuration and apply it to this group, even if you already have instances running in it. Using this method, you can update the software that your group uses when it launches new instances.

○ Create a new launch configuration

● Create an Auto Scaling group from an existing launch configuration

	Name	AMI ID	Instance Type	Spot Price	Security Groups
■	webserver-ca	ami-d1792ee1	t2.micro		sg-a20117c5

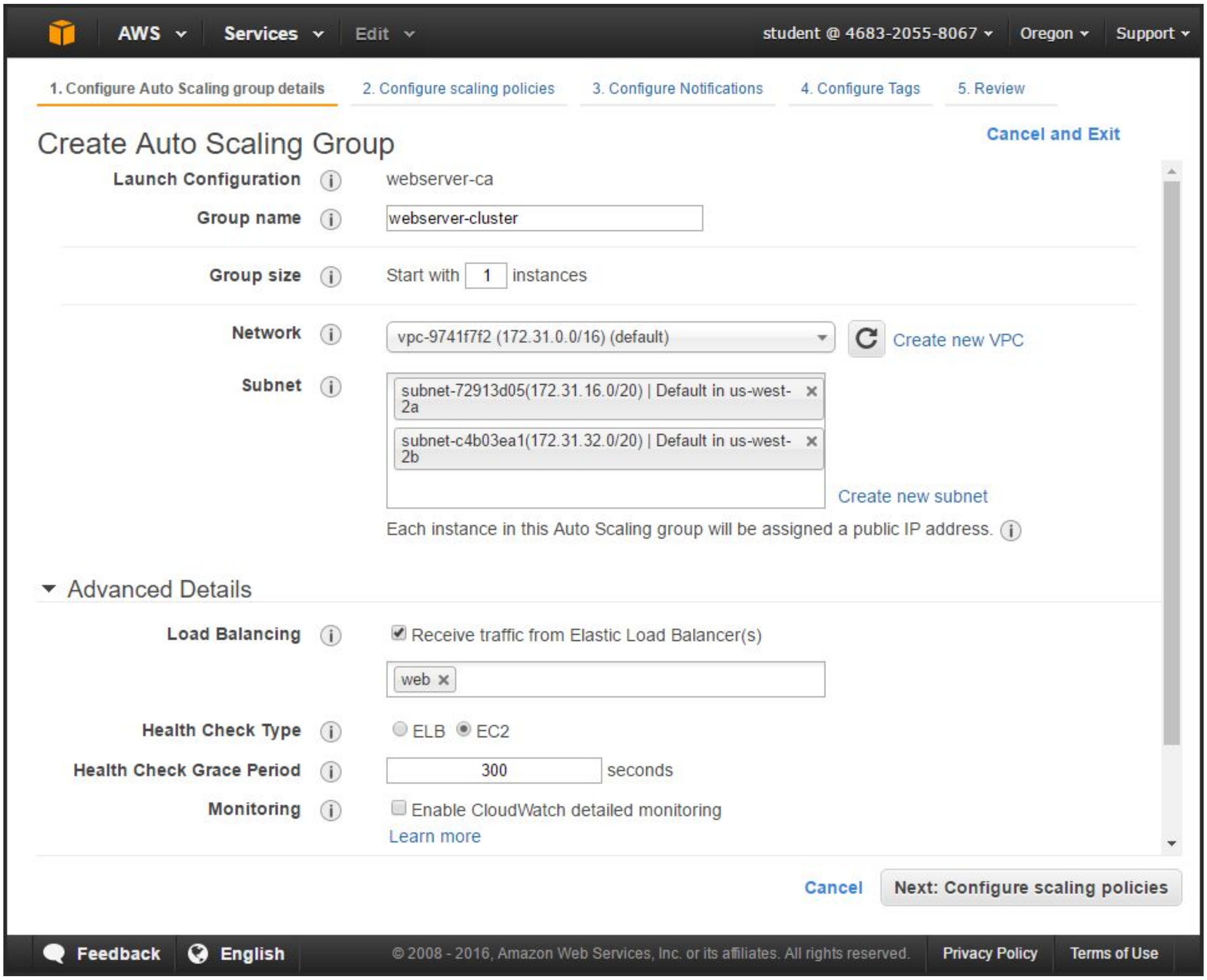
3. In the "**Configure Auto Scaling group details**" step, you should use the following settings:

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.*

4. 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*

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


AWS
Services
Edit
student @ 4683-2055-8067
Oregon
Support
1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review
Cancel and Exit
Create new VPC
Create new subnet
Learn more
Cancel
Next: Configure scaling policies
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Create Auto Scaling Group

Launch Configuration: webserver-ca

Group name: webserver-cluster

Group size: Start with 1 instances

Network: vpc-9741f7f2 (172.31.0.0/16) (default)

Subnet: subnet-72913d05(172.31.16.0/20) | Default in us-west-2a
subnet-c4b03ea1(172.31.32.0/20) | Default in us-west-2b

Each instance in this Auto Scaling group will be assigned a public IP address.

Advanced Details

Load Balancing: ☑ Receive traffic from Elastic Load Balancer(s)
web

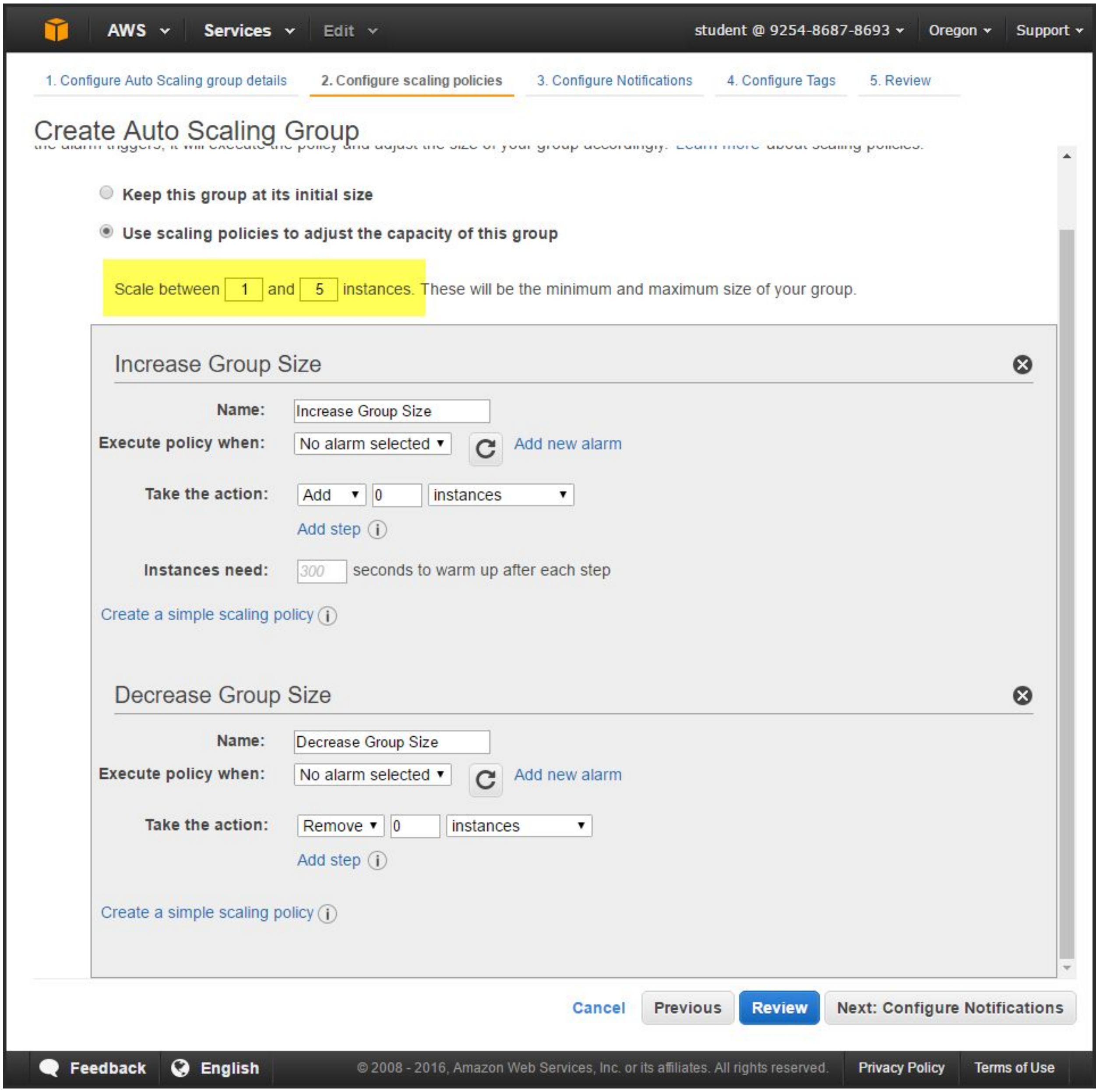
Health Check Type: ○ ELB ● EC2

Health Check Grace Period: 300 seconds

Monitoring: □ Enable CloudWatch detailed monitoring

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

6. 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.


AWS
Services
Edit
student @ 9254-8687-8693
Oregon
Support
1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review
Cancel
Previous
Review
Next: Configure Notifications
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Create Auto Scaling Group

the alarm triggers, it will execute the policy and adjust the size of your group accordingly. Learn more about scaling policies.

○ Keep this group at its initial size

● Use scaling policies to adjust the capacity of this group

Scale between 1 and 5 instances. These will be the minimum and maximum size of your group.

Increase Group Size

Name: Increase Group Size

Execute policy when: No alarm selected ▾ Add new alarm

Take the action: Add ▾ 0 instances ▾

Add step

Instances need: 300 seconds to warm up after each step

Create a simple scaling policy

Decrease Group Size

Name: Decrease Group Size

Execute policy when: No alarm selected ▾ Add new alarm

Take the action: Remove ▾ 0 instances ▾

Add step

Create a simple scaling policy

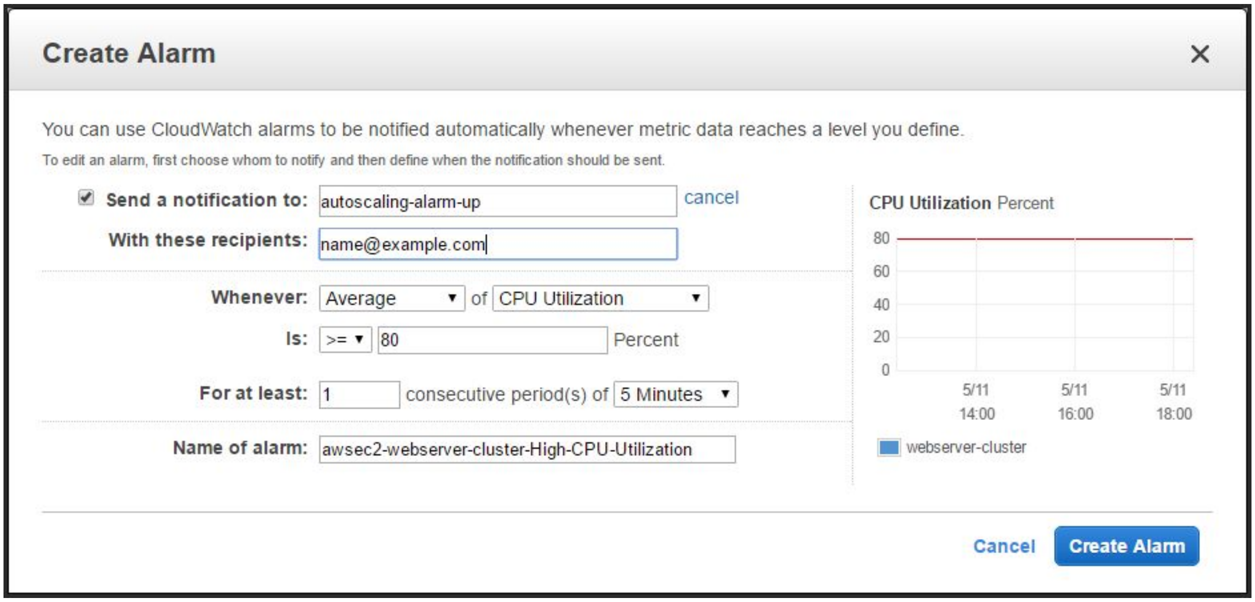
The Auto Scaling group policies allow you to automatically increase or decrease the group size based upon policies you define. In order to establish an Increase Group size or Decrease Group Size policy, you must create a CloudWatch Alarm and then define which action should be taken if it is triggered.

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

8. 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.

9. 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 Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

☑ Send a notification to: autoscaling-alarm-up

With these recipients: name@example.com

Whenever: Average of CPU Utilization

Is: >= 80 Percent

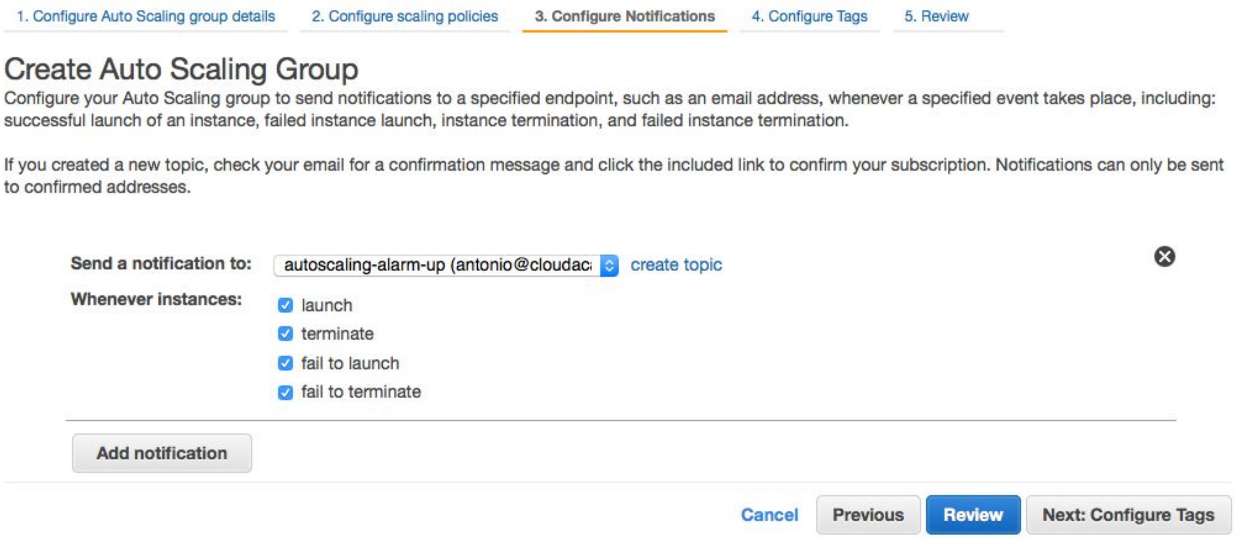
For at least: 1 consecutive period(s) of 5 Minutes

Name of alarm: awsec2-webserver-cluster-High-CPU-Utilization

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

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

12. 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.



1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review

Create Auto Scaling Group

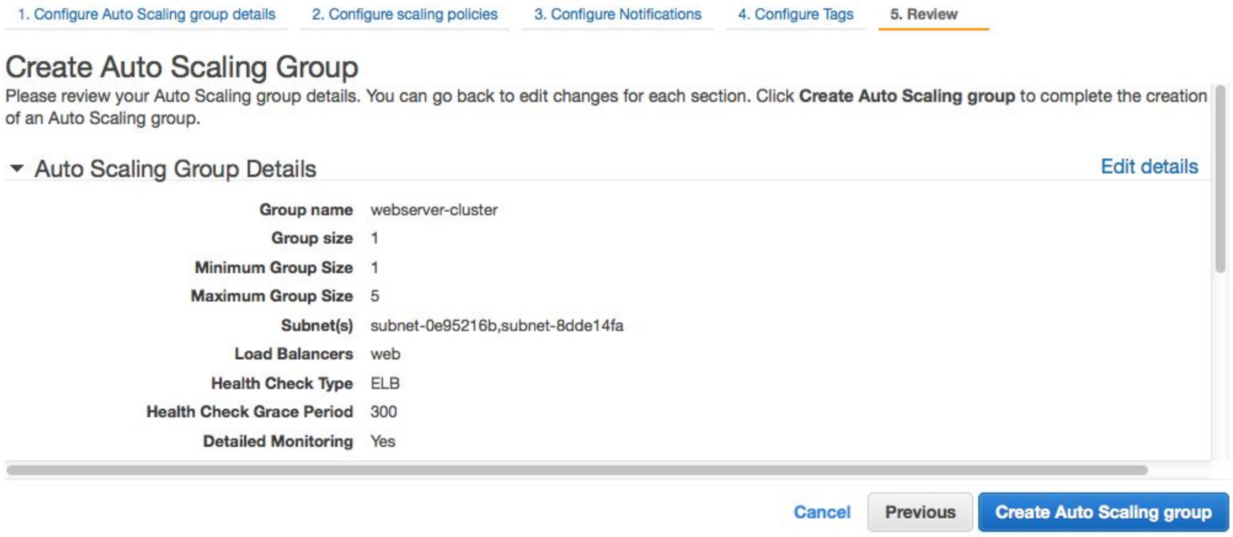
Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes place, including: successful launch of an instance, failed instance launch, instance termination, and failed instance termination.

If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications can only be sent to confirmed addresses.

Send a notification to: autoscaling-alarm-up (antonio@cloudac

Whenever instances:
☑ launch
☑ terminate
☑ fail to launch
☑ fail to terminate

13. 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**.



1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review

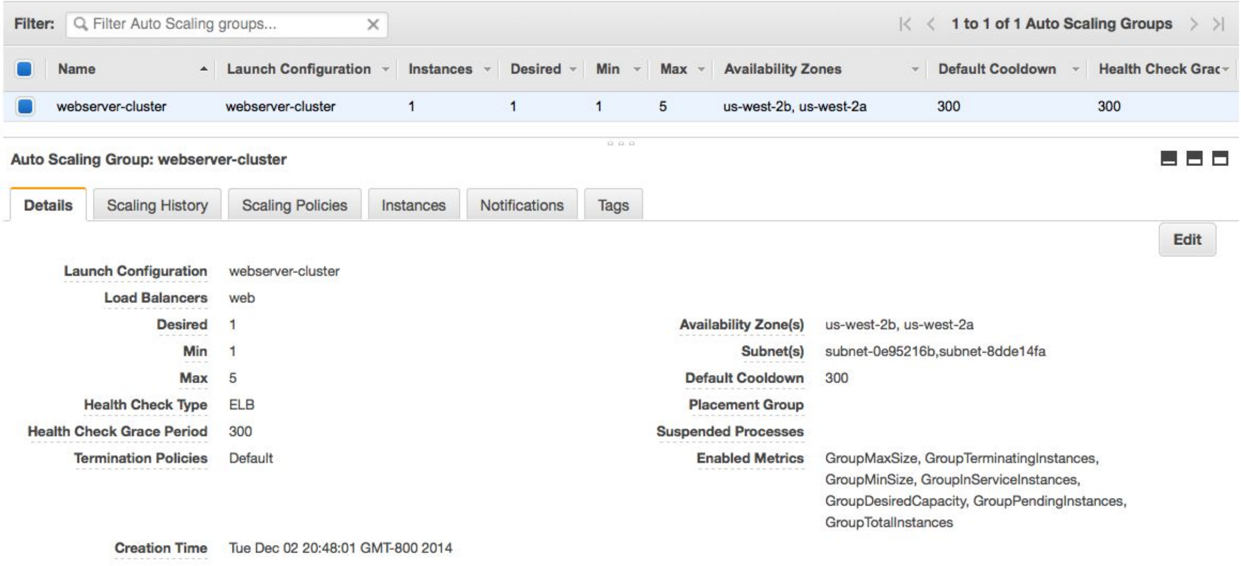
Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

Auto Scaling Group Details

Group name	webserver-cluster
Group size	1
Minimum Group Size	1
Maximum Group Size	5
Subnet(s)	subnet-0e95216b,subnet-8dde14fa
Load Balancers	web
Health Check Type	ELB
Health Check Grace Period	300
Detailed Monitoring	Yes

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

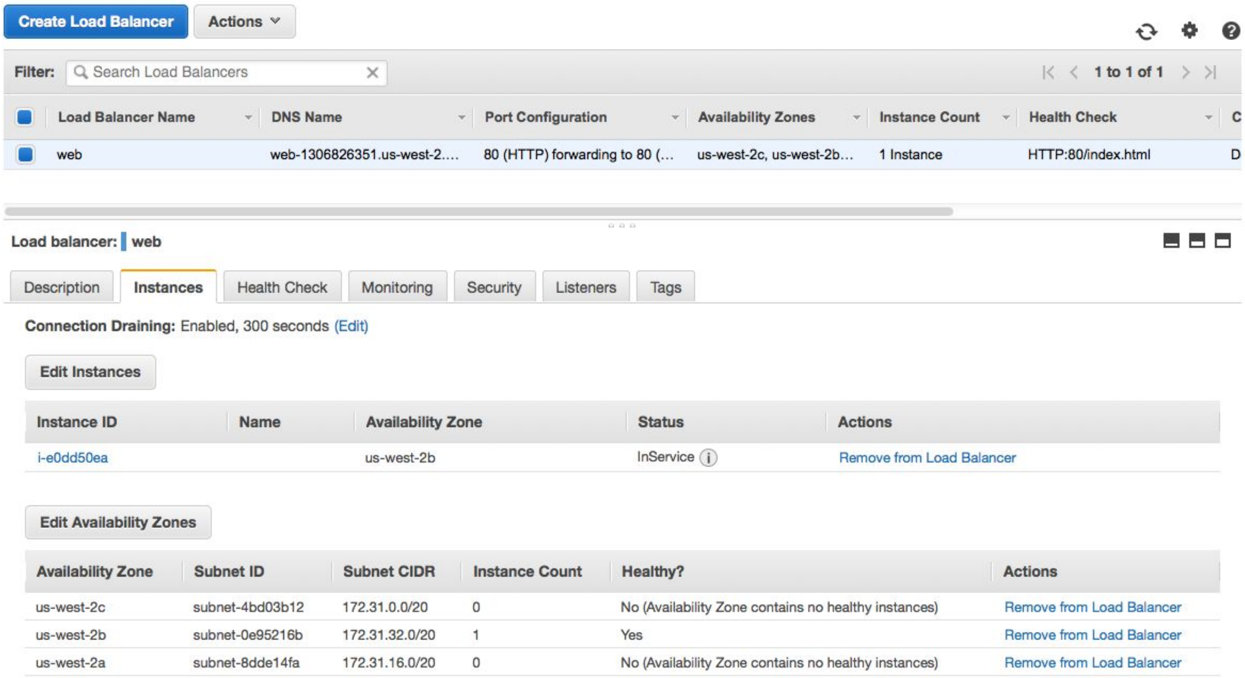

Filter: Filter Auto Scaling groups...
1 to 1 of 1 Auto Scaling Groups
Details
Scaling History
Scaling Policies
Instances
Notifications
Tags
Edit

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
webserver-cluster	webserver-cluster	1	1	1	5	us-west-2b, us-west-2a	300	300

Auto Scaling Group: webserver-cluster

Launch Configuration	webserver-cluster		
Load Balancers	web		
Desired	1	Availability Zone(s)	us-west-2b, us-west-2a
Min	1	Subnet(s)	subnet-0e95216b,subnet-8dde14fa
Max	5	Default Cooldown	300
Health Check Type	ELB	Placement Group	
Health Check Grace Period	300	Suspended Processes	
Termination Policies	Default	Enabled Metrics	GroupMaxSize, GroupTerminatingInstances, GroupMinSize, GroupInServiceInstances, GroupDesiredCapacity, GroupPendingInstances, GroupTotalInstances
Creation Time	Tue Dec 02 20:48:01 GMT-800 2014		

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.



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

Connection Draining: Enabled, 300 seconds (Edit)

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

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