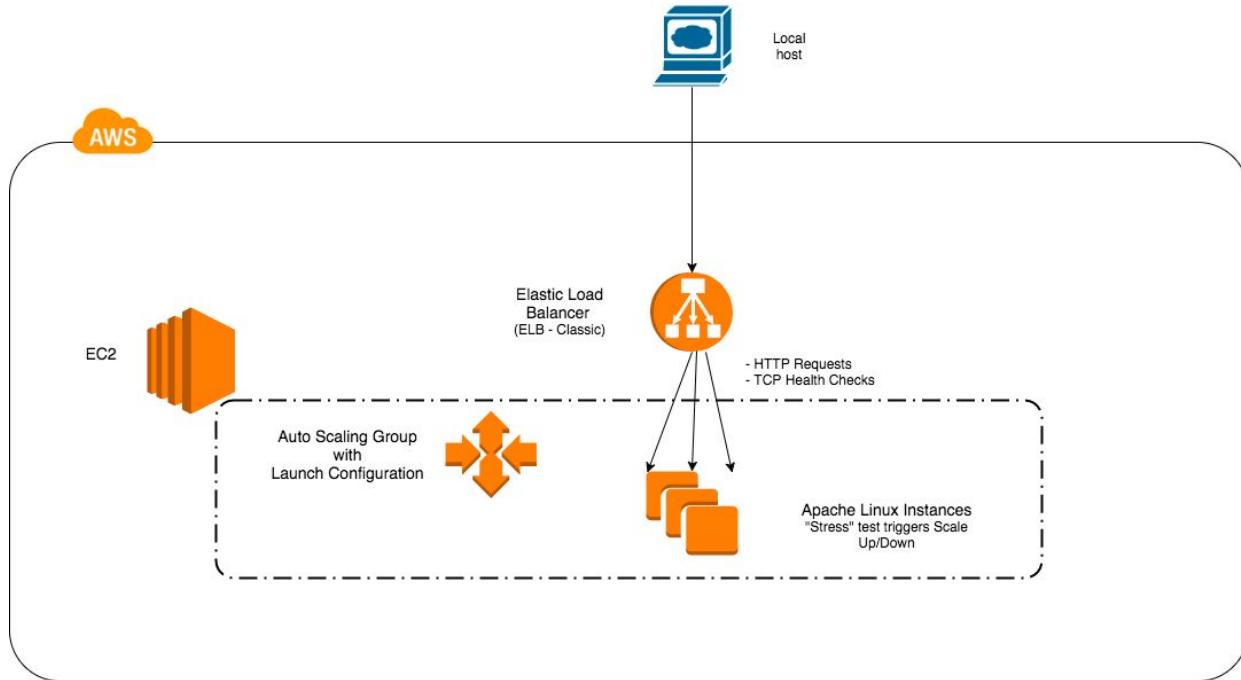


Lab Environment

After completing the lab instructions the environment will look similar to:



Logging in to the Amazon Web Services Console

Introduction

This Lab experience involves Amazon Web Services, and you will use the AWS Management Console to complete all the Lab Steps. Please note that you will have a space storage limit of 100GB for this lab, which will be more than sufficient to complete it.

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account details. Below the navigation is a search bar and a 'Recently visited services' section. It features several service icons: RDS, Lambda, API Gateway, EC2, and WorkSpaces. A 'Build a solution' section offers quick-start wizards for launching a virtual machine, building a web app, deploying a serverless microservice, hosting a static website, creating a mobile app backend, and registering a domain. To the right, there are 'Featured next steps' for managing spend and getting best practices, along with links to the AWS Marketplace and feedback submission. At the bottom, there are language and feedback links.

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.

Instructions

1. To start the Lab experience, open the Amazon Console by clicking this button:

[Open AWS Console](#)

2. Enter the following credentials created just for your Lab session, and click **Sign In**:

- **Account ID or alias:** Keep the pre-populated value
- **IAM user name:** student
- **Password:** \$passw0rd



Account ID or alias

IAM user name

Password

Sign In

3. Select the **US West (Oregon)** region using the upper right drop-down menu on the AWS Management Console:

The screenshot shows the AWS Management Console navigation bar. On the far left, there is a sidebar with various links like 'ful tips', 'Manage', 'Get real and usa', 'Create', 'Use AW manage now', 'ore AV', 'on Relati...', and 'anges and...'. To the right of the sidebar is a dark grey header bar containing the text 'student @ 1234-5678-9012' and three dropdown menus: 'Oregon', 'Support', and another one partially visible. Below the header is a large white area containing a list of AWS regions. The 'Oregon' region is highlighted with a blue border and the text 'US West (Oregon)' is bolded. Other regions listed include US East (N. Virginia), US East (Ohio), US West (N. California), Canada (Central), EU (Ireland), EU (Frankfurt), EU (London), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Seoul), Asia Pacific (Tokyo), Asia Pacific (Mumbai), and South America (São Paulo).

Region
US East (N. Virginia)
US East (Ohio)
US West (N. California)
US West (Oregon)
Canada (Central)
EU (Ireland)
EU (Frankfurt)
EU (London)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Seoul)
Asia Pacific (Tokyo)
Asia Pacific (Mumbai)
South America (São Paulo)

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 **US West (Oregon)** for this Lab.

Create a load balancer using ELB

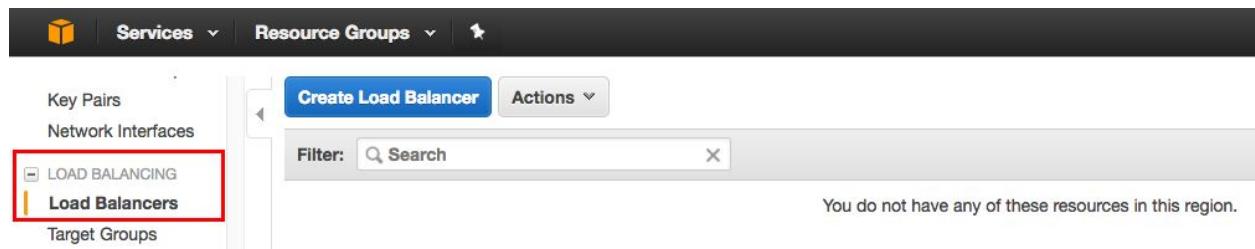
Introduction

Elastic Load Balancers automatically distribute incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater fault tolerance in your applications and seamlessly provide the correct amount of load balancing capacity needed in response to incoming application requests.

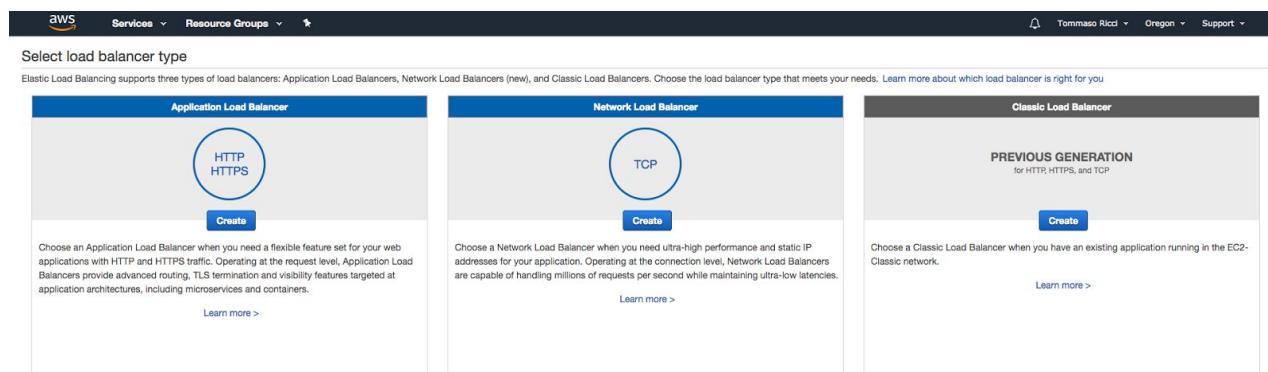
ELB detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. Elastic Load Balancers can be enabled within a single Availability Zone or across multiple zones for greater consistent application performance.

Instructions

1. Select **EC2** from the AWS Services menu.
2. In the left pane, click **Load Balancers** in the **Load Balancing** section:



3. Click the **Create Load Balancer** button.
4. Take a moment to read the information for both load balancer types. Click **Create** in the Classic Load Balancer square:



A multi-step wizard starts for creating a load balancer.

5. On the **Define Load Balancer** screen, configure it in the following manner:

- **Load Balancer name:** Enter *Web*
- **Create LB Inside:** Select the *vpc-<AlphaNumeric>* from the drop-down menu. Note the CIDR notation for the VPC. (For example, 172.31.0.0/16)
- **Enable advanced VPC configuration:** This field should be checked for you. If not, please check the box.
- **Listener Configuration:** Leave the default values (HTTP/80/HTTP/80)
- **Select Subnets > Available subnets:** Click the + action icon to select **us-west-2a and 2b Availability Zones**.

The screenshot shows the AWS CloudFormation 'Define Load Balancer' step. In the 'Available subnets' table, the row for 'us-west-2a' is highlighted with a red border, indicating it has been selected. The 'Selected subnets' table below it contains the same two rows, with 'us-west-2a' also highlighted with a red border.

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-west-2c	subnet-5fb21804	172.31.32.0/20	

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-west-2a	subnet-a25345c5	172.31.0.0/20	
	us-west-2b	subnet-f10821b8	172.31.16.0/20	

Notice when you select an **Available subnet** it transfers to the **Selected subnets** section.
Click **Next: Assign Security Groups** when ready.

6. On the **Assign Security Groups** page:

- Select the **Create a new security group** radio button
- Enter *elb-webserver* for the **Security group name**
- Enter *ELB for a webserver cluster* for the **Description**
- Create a single firewall rule of **Type HTTP; Protocol TCP; Port Range 80 and Source Anywhere (0.0.0.0/0)**.

AWS Services Edit student @ 3335-7500-3576 Oregon Support

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

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.

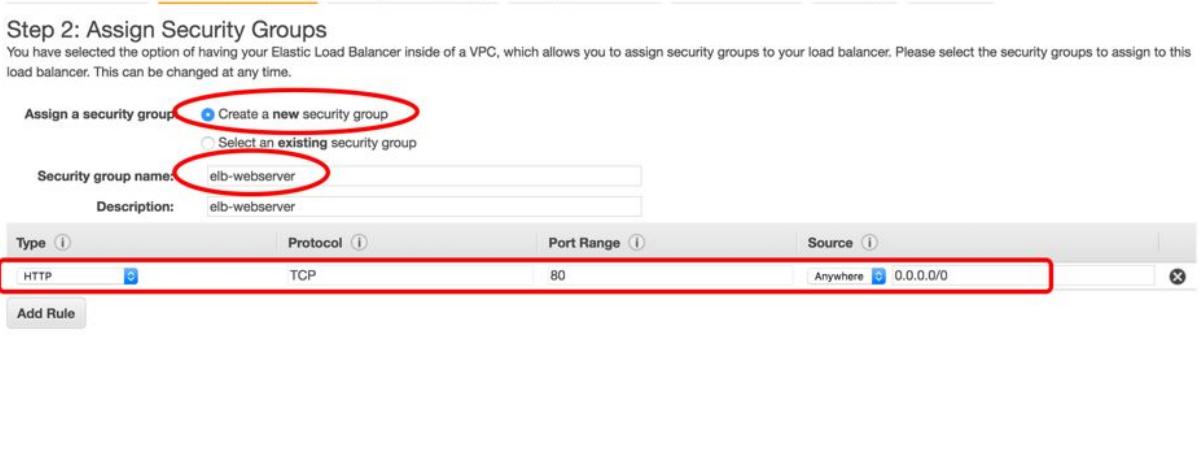
Assign a security group Create a new security group Select an existing security group

Security group name: elb-webserver

Description: elb-webserver

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

Add Rule Cancel Previous Next: Configure Security Settings



This will rule will allow inbound HTTP traffic on port 80 from any source IP address.
Click **Next: Configure Security Settings** when ready.

7. Ignore the warning in the **Configure Security Settings** page. You are only serving the HTTP protocol in this lab. Security is not a concern (implemented via HTTPS or SSL protocols) so these settings are not required:

AWS Services Edit student @ 3335-7500-3576 Oregon Support

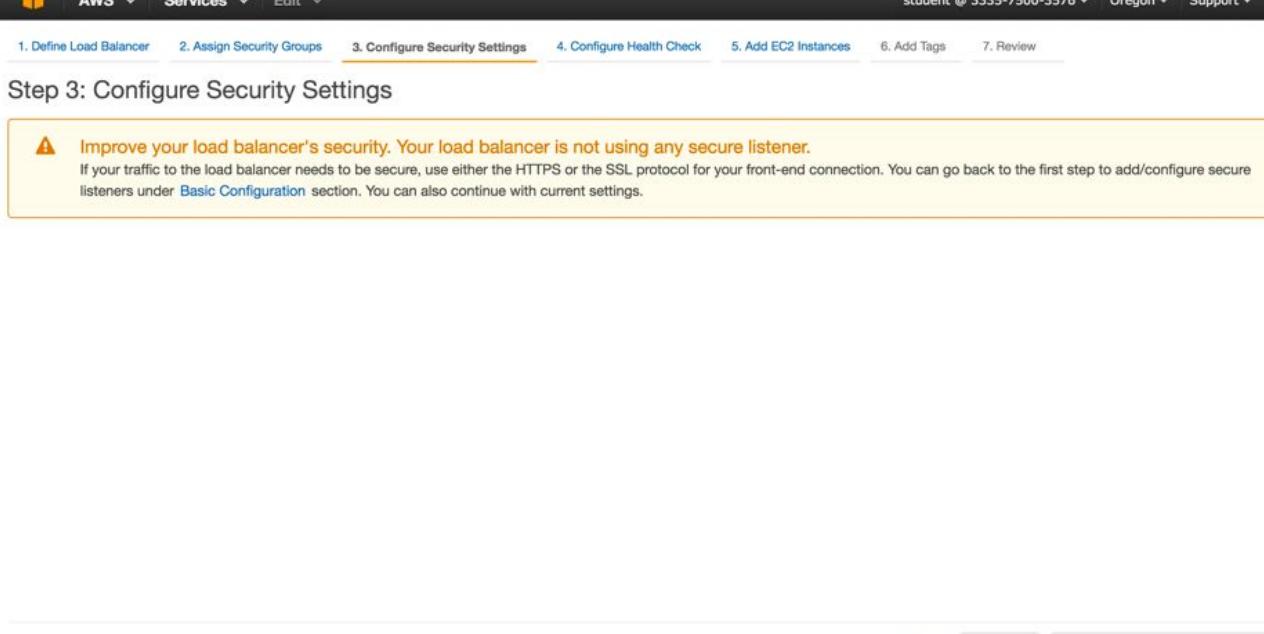
1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

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.

Cancel Previous Next: Configure Health Check



Click **Next: Configure Health Check**.

8. On the **Configure Health Check** page:

- **Ping Protocol:** Select **TCP**
- **Ping Port:** **80**

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: TCP (highlighted)

Ping Port: 80

Advanced Details

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

In a production environment you would likely set this to HTTP. However, this Lab is not concerned with how well your web server is doing serving HTML pages, it just wants to make sure it is up and running. Again, set the **Ping Protocol** to **TCP**.

Click **Next: Add EC2 Instances** when ready to proceed (the defaults for the advanced section will suffice).

9. On the **Add EC2 Instances** page of the wizard, notice the "No instances available" message:

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)

Select	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

Cancel Previous Next: Add Tags

The message is because you have not created and launched an Auto Scaling Group yet, which is used to launch and maintain one or more instances.

10. Click **Next: Add Tags** to continue. Leave the fields blank in the **Add Tags** page. Click **Review and Create**.

11. Review your settings for correctness and then click **Create** when ready:

The screenshot shows the AWS Step 7: Review page for creating a Load Balancer. The top navigation bar includes AWS Services, Edit, student @ 3335-7500-3576, Oregon, and Support. Below the navigation, a progress bar shows steps 1 through 7, with step 7 being the current active tab. The main content area is titled "Step 7: Review" and contains sections for "Define Load Balancer", "Configure Health Check", "Add EC2 Instances", and "VPC Information". Each section displays configuration details and provides "Edit" links. At the bottom right are "Cancel", "Previous", and "Create" buttons.

12. Wait for the **Load Balancer Creation Status** to display a successful creation message:

The screenshot shows the AWS Load Balancer Creation Status page. The top navigation bar is identical to the previous screenshot. The main content area displays a green box containing a success message: "Successfully created load balancer" followed by "Load balancer Web was successfully created." A note below states, "Note: It may take a few minutes for your instances to become active in the new load balancer." At the bottom right is a "Close" button.

13. Click **Close**.

Summary

ELB is ready to service requests! In this Lab Step you created an Elastic Load Balancer (ELB) to service HTTP requests on port 80 from any IP source address. This ELB will be used on the front-end to direct requests to several instances running a web server. This is a very common use case for ELBs.

Create a Launch Configuration

Introduction

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 that references the Launch Configuration. The Launch Configuration essentially contains the blueprint or DNA for the exact type of instance that should be launched. Hence, when auto scaling, each instance is guaranteed to be just like the last one. It's repeatable, scalable, and reliable.

When you create the Launch Configuration you will include information such as the Amazon machine image ID (AMI) 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.

First you will create a Launch Configuration, then the Auto Scaling group.

Instructions

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



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

The screenshot shows the AWS Management Console with the Auto Scaling service selected. On the left, there's a navigation menu with sections like 'IMAGES', 'ELASTIC BLOCK STORE', 'NETWORK & SECURITY', and 'AUTO SCALING'. Under 'AUTO SCALING', 'Launch Configurations' is highlighted with a red oval. In the main content area, the title 'Welcome to Auto Scaling' is displayed, followed by a note about managing EC2 capacity automatically. A prominent blue button labeled 'Create Auto Scaling group' is centered, also highlighted with a red oval. Below it, a note says 'Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.' There are three sections: 'Reusable Instance Templates' (with a gear and plus icon), 'Automated Provisioning' (with a checkmark and gear icon), and 'Adjustable Capacity' (with a gear and leaf icon). Each section has a 'Learn more' link.

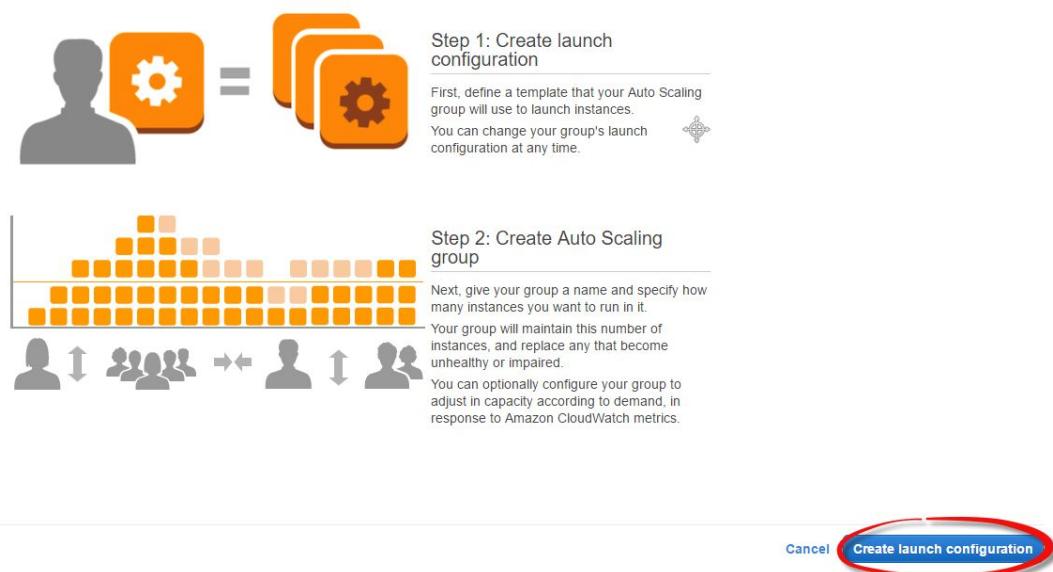
The Create Auto Scaling group wizard starts.

3. Take a moment to read the helpful **Create Auto Scaling Group** information that is part of the wizard. *Tip:* Reading the information within various AWS wizards helps with the learning curve and is a good habit to get into:

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.



Click the **Create Launch configuration** button. Creating a launch configuration is the first step in creating a Auto Scaling group. The wizard guides you through each required step and displays a graphical interface that is similar to the Launch Instance wizard.

4. On the **Choose AMI** page of the wizard you must select the AMI that will be used by all the EC2 instances of the Auto Scaling group. Select the Amazon Linux AMI:

Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-6df1e514
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm

SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-e4a30084
SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm

Click **Select** when ready. The next step is choosing the instance type.

5. On the **Choose Instance Type** page, select the **t2.micro** type and click **Next: Configure details**:

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

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

Cancel Previous Next: Configure details

6. On the **Configure details** page:

- Enter *webserver-cluster* for the **Name**
- Check **Enable CloudWatch detailed monitoring**
- Expand **Advanced Details**

- Select the **Assign a public IP address to every instance** radio button
- Paste the following bash snippet in the **User data** field:

```
#!/bin/bash
#Install and start Apache web server
yum install -y httpd24 php56
service httpd start
touch /var/www/html/index.html
chmod 644 /var/www/html/index.html
#Install CPU stress test tool
sudo yum install stress
```

The Configure details screen of the wizard should look similar to the following:

Create Launch Configuration

Name

Purchasing option Request Spot Instances

IAM role

Monitoring Enable CloudWatch detailed monitoring [Learn more](#)

Advanced Details

Kernel ID

RAM Disk ID

User data As text As file Input is already base64 encoded

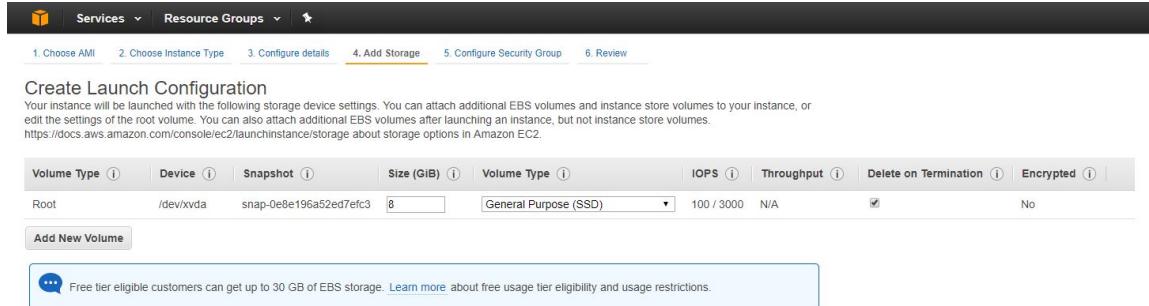
```
#!/bin/bash
#Install and start Apache web server
yum install -y httpd24 php56
service httpd start
#Install CPU stress test tool
sudo yum install stress
```

IP Address Type Assign a public IP address to every instance.

Note: this option only affects instances launched into an Amazon VPC

By default, CloudWatch monitors EC2 instances approximately every 5 minutes. Detailed monitoring enables monitoring more often (each minute). *Note:* Detailed monitoring does have an associated cost. Click **Next: Add Storage** when ready.

7. The **Add Storage** page of the wizard allows you to add or increment the size of any EBS volume attached to each EC2 instance started by the Auto Scaling group. Leave the defaults and do not add any EBS volumes.



Typically large EBS volumes are only needed if your software requires storage space to process the application data. Many applications store raw or processed data with Amazon S3, Redshift, DynamoDB or another storage/database service provided by Amazon. When that is the use-case, large EBS volumes are usually not required. This lab environment definitely does not need extra disk space. Click **Next: Configure Security** when ready.

8. On the **Configure Security Group** page there are several configurations for your Launch Configuration:

- Select **Create a new security group**
- Enter **Webserver-cluster** for the **Name**. Enter a **Description** as well.
- Add two rules. For the first rule configure:
 - **Type**=SSH
 - **Protocol**=TCP
 - **Port Range**=22
 - **Source**=Anywhere (0.0.0.0/0) *Warning:* In production the source should be more restrictive to account for corporate security policies. For example, your corporate external public IP range.
- For the second rule configure:
 - **Type**=HTTP
 - **Protocol**=TCP
 - **Port Range**=80
 - **Source**=Anywhere (0.0.0.0/0) *Warning:* In production the source should be more restrictive. For example, only the ELB should be able to connect to port 80 on the web servers. Then the ELB allows remote access from anywhere, but is the only component that can access the instances directly in the auto scaling group.

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

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: Webserver-cluster Launch Config security group

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere <input type="radio"/> 0.0.0.0/0 <input checked="" type="radio"/>
HTTP	TCP	80	Anywhere <input type="radio"/> 0.0.0.0/0 <input checked="" type="radio"/>

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review

Click **Review** when ready.

9. Once you have reviewed the details for accuracy, click **Create launch configuration**:

▼ Instance Type Edit 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 Edit details

Cancel Previous Create launch configuration

You will be presented with the **Select an existing key pair or create a new key pair** dialogue. 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. Always be sure to secure your key pair. Not doing so is a security risk.

10. In the **Select an existing key pair or create a new key pair** dialog:

- Select **Choose an existing key pair** from the first drop-down menu. If you did not need access to the instances, you could select **Proceed without a key pair** and acknowledge you will not be able to access the instance. (However, you will need SSH access later.)
- **Select a key pair:** Select the random number named key pair that is generated for you by the Cloud Academy platform
- Check the "I acknowledge that I have access to the selected private key file..." check box:

Select an existing key pair or create a new key pair

X

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](#).

Choose an existing key pair

Select a key pair

112446744799

Key pair with random number
name generated for you by the platform

acknowledge that I have access to the selected private key file (112446744799.pem), and
that without this file, I won't be able to log into my instance.

Cancel

Create launch configuration

Click **Create launch configuration** when ready to proceed.

Because the normal flow after creating a Launch configuration is to create an Auto Scaling group that will use the configuration, the next screen will present you with the option to do so. However, click **Cancel** as you will navigate directly in the console menus and create it from scratch in the next Lab Step.

Summary

You have created a Launch Configuration that can be used by an Auto Scaling group to launch identical instances every time. Note that you cannot modify a Launch Configuration. Why? It would impact the effectiveness and very purpose of having a launch configuration. For example, if you have multiple instances starting and terminating in accord with your scaling policy, then change the launch configuration, future instances would be different than the current production instances. This can be a nightmare to maintain and troubleshoot. (You can however create new Launch Configurations and have the Auto Scaling group associate with a new Launch Configuration.)

Create an Auto Scaling Group

Introduction

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, and 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 in an Auto Scaling group is the core value of the Auto Scaling service. It's what puts the "elastic" in EC2.

Instructions

1. Click **Auto Scaling Groups** in the left pane of the EC2 console, then click **Create Auto Scaling group**:

Instances
Spot Requests
Reserved Instances

- IMAGES
AMIs
Bundle Tasks

- ELASTIC BLOCK STORE
Volumes
Snapshots

- NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces

- AUTO SCALING
Launch Configurations
Auto Scaling Groups

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](#)

Auto Scaling Groups: 0 Launch Configuration: 1

Create Auto Scaling group [Create launch configuration](#)

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 **Automated Provisioning** **Adjustable Capacity**

Provision instances based on a reusable template you define, called a launch configuration. [Learn more](#)

Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000. [Learn more](#)

Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics. [Learn more](#)

Since you created a Launch configuration earlier, you will be able to associate it with the Auto Scaling group you create here.

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

The screenshot shows the 'Create Auto Scaling Group' wizard. The first step is to choose a launch configuration. Two options are available: 'Create a new launch configuration' (radio button not selected) and 'Create an Auto Scaling group from an existing launch configuration' (radio button selected). A red circle highlights the 'Create an Auto Scaling group from an existing launch configuration' radio button. Below it is a dropdown menu titled 'Filter launch configurations...' containing one item: '1 to 1 of 1 Launch Configurations'. The item 'webserver-ca' is listed with columns: Name (webserver-ca), AMI ID (ami-d1792ee1), Instance Type (t2.micro), and Security Groups (sg-a20117c5). A red circle highlights the 'webserver-ca' entry in the dropdown list. At the bottom right of the page, there are 'Cancel' and 'Next Step' buttons. The 'Next Step' button is highlighted with a red circle.

3. On the **Configure Auto Scaling group details** page, use the following settings:

- **Group name:** webserver-cluster
- **Group size:** Start with 1 instance
- **Network:** Select the **vpc-<AlphaNumeric>** from the drop-down menu
- **Subnet:** Select two subnets, one in **us-west-2a** and the other in **us-west-2b**.

While still on the first page of the wizard, expand **Advanced Details**, use the following settings:

- **Load Balancing:** Check **Receive traffic from one or more load balancers**
- Select **Web** for the **Classic Load Balancer** (the ELB you created earlier).
- **Health Check Type:** ELB
- **Monitoring:** Check **Enable CloudWatch detailed monitoring**

Create Auto Scaling Group

Subnet ⓘ

[Create new subnet](#)

A **No public IP addresses will be assigned**

None of the instances in this Auto Scaling group will be assigned a public IP address because you have not chosen to launch in your default VPC and subnet.

You can ensure a public IP address is assigned to instances launched with this configuration by selecting only default subnets of your default VPC.

[Learn more](#) about IP addressing in an Amazon VPC.

▼ Advanced Details

Load Balancing ⓘ Receive traffic from one or more load balancers [Learn about Elastic Load Balancing](#)

Classic Load Balancers ⓘ

Target Groups ⓘ

Health Check Type ⓘ ELB EC2

Health Check Grace Period ⓘ seconds

Monitoring ⓘ Enable CloudWatch detailed monitoring [Learn more](#)

Instance Protection ⓘ

It is OK to ignore the "No Public IP addresses ..." warning message. Other settings can be left at the default values.

Click **Next: Configure scaling policies** when ready. Scaling policies determine how and when your infrastructure will scale up and scale back down.

4. In the **Configure scaling policies** page, select the **Use scaling policies to adjust the capacity**

of this group radio button. Set it to scale between 1 and 5 instances:

The screenshot shows the AWS Auto Scaling Group creation wizard at Step 2: Configure scaling policies. The 'Use scaling policies to adjust the capacity of this group' option is selected. A yellow box highlights the 'Scale between 1 and 5 instances' input field. Below it, two policy configurations are shown: 'Increase Group Size' and 'Decrease Group Size'. Both policies have 'No alarm selected' and 'Add new alarm' options. The 'Increase Group Size' policy has 'Add 0 instances' and 'Instances need 300 seconds to warm up after each step'. The 'Decrease Group Size' policy has 'Remove 0 instances'.

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

Create Auto Scaling Group

Keep this group at its initial size

Use scaling policies to adjust the capacity of this group

Scale between and 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 instances

Add step

Instances need: 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 instances

Add step

Create a simple scaling policy

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

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. *Don't go to the next page of the wizard yet.*

5. Click **Add new alarm** under the **Increase Group Size** section. A **Create Alarm** dialogue is

displayed. You will configure this for a alarm-up action:

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.

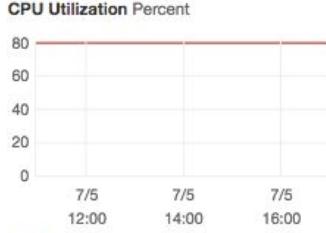
Send a notification to: [cancel](#)

With these recipients:

Whenever: of
Is: Percent

For at least: consecutive period(s) of

Name of alarm:



[Cancel](#) [Create Alarm](#)

If you want to receive a notification when the alarm is triggered, you need to set up an Simple Notification Service (SNS) topic.

- Check the **Send a notification** checkbox
- Click **create topic**. Enter *autoscaling-alarm-up* for the SNS topic name and then enter your valid email address in the recipients field.
- Set **Whenever** to **Average** and **CPU Utilization** as the type of metric
- Set the relationship to **>=** and enter **80** as the **Percent** (Note this will be shown as a horizontal line at 80 in the adjoining graph soon.)
- Set **For at least** to **1** consecutive period of **5 Minutes**
- Enter a name for the alarm (or use the default name), and then click **Create Alarm**

6. Create another alarm for the **Decrease Group Size**. Name it appropriately (autoscaling-alarm-down). Configure similar to the last alarm, but this time set it to **<= 10 Percent of CPU Utilization**. Although not critical, the default **Name of alarm** is misleading in this case. Change the "High" within the default string to "Low". The **Increase/Decrease Group Size** sections should look similar to the following:

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: [cancel](#)

With these recipients:

Whenever: of
Is: Percent

For at least: consecutive period(s) of

Name of alarm:

[Cancel](#) [Create Alarm](#)

Click **Next: Configure Notifications** when ready. **Configure Notifications** will set you up so you are notified whenever an Auto Scaling Group instance is launched or terminated, with or without success.

7. Click the **Add notification** button. You can use one of the same SNS topics previously created for the CloudWatch alarms. For the **autoscaling-alarm-up** select **launch** and **fail to launch**. Click **Add notification** to add a second notification. This time select an existing **autoscaling-alarm-down** along with **terminate** and **fail to terminate**. It should look similar to the following (but reflect your valid email address):

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: [create topic](#)

Whenever instances: launch
 terminate
 fail to launch
 fail to terminate

Send a notification to: [create topic](#)

Whenever instances: launch
 terminate
 fail to launch
 fail to terminate

[Add notification](#)

[Cancel](#) [Previous](#) [Review](#) [Next: Configure Tags](#)

Alternatively, you could create a new topic, or use a single existing topic. For example, use the existing scale up topic and leave all **Whenever instance** options. The advantage of adding a notification for scaling up and another for scaling down (as shown above) is it's a bit easier to read the email alarm at glance, rather than parsing the details within the email body to determine

if there was actually a scale up or down event. You won't need to **Configure Tags** for this lab, so click **Review** when ready.

8. Review all the selected options:

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

Edit details

Cancel **Previous** **Create Auto Scaling group**

Warning! It is very easy to make a mistake with the **Scaling Policies**. For example, mistakenly setting up the relationship incorrectly for increasing/decreasing the group size. To double-check your configuration, click **Previous** until at the **Configure Scaling Policies** section and verify it looks similar to the following:

Create Auto Scaling Group

Keep this group at its initial size

Use scaling policies to adjust the capacity of this group

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

Increase Group Size

Name:

Execute policy when: awsec2-webserver-cluster-CPU-Utilization [Edit](#) [Remove](#)
breaches the alarm threshold: CPUUtilization ≥ 80 for 300 seconds
for the metric dimensions AutoScalingGroupName = webserver-cluster

Take the action: instances $\leq \text{CPUUtilization} < +\infty$

[Add step](#) (i)

Instances need: seconds to warm up after each step

[Create a simple scaling policy](#) (i)

Decrease Group Size

Name:

Execute policy when: awsec2-webserver-cluster-High-CPU-Utilization [Edit](#) [Remove](#)
breaches the alarm threshold: CPUUtilization ≤ 10 for 300 seconds
for the metric dimensions AutoScalingGroupName = webserver-cluster

Take the action: instances $\geq \text{CPUUtilization} > -\infty$

[Add step](#) (i)

[Create a simple scaling policy](#) (i)

When you are satisfied, start the creation of your cluster by advancing the wizard and clicking **Create Auto Scaling group** on the **Review** screen of the wizard. Once created, viewing your **Scaling Policy** (**Auto Scaling Group** > **Scaling Policies** tab) should look similar to:

Scaling Policies

Decrease Group Size

Execute policy when: awsec2-webserver-cluster-Low-CPU-Utilization
breaches the alarm threshold: CPUUtilization <= 10 for 300 seconds
for the metric dimensions AutoScalingGroupName = webserver-cluster

Take the action: Remove 1 instances when 10 >= CPUUtilization > -infinity

Increase Group Size

Execute policy when: awsec2-webserver-cluster-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 80 for 300 seconds
for the metric dimensions AutoScalingGroupName = webserver-cluster

Take the action: Add 1 instances when 80 <= CPUUtilization < +infinity

Instances need: 300 seconds to warm up after each step

Important! Note that you can change values and parameters for your Scaling Policy with **Actions > Edit**. For example, to change the number of instances to scale up or down by was incorrect, you could fix it with **Actions > Edit**.

9. Navigate to the **EC2 Dashboard > Auto Scaling Groups**. In a minute or so your cluster will be deployed and your EC2 instance will be ready to use:

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

Auto Scaling Group: webserver-cluster

Details **Scaling History** **Scaling Policies** **Instances** **Notifications** **Tags** **Edit**

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

Note you can confirm this from the **EC2 Dashboard > Running Instances** too. Notice there should be one instance (singular) not multiple instances. Why? You set **Desired** and **Min** to *1* earlier.

10. Open the **Load Balancers** section, select your previously created ELB, and then open the **Instances** tab. You can see the new Auto Scaling instance(s) automatically added to the ELB

configuration:

The screenshot shows the AWS Load Balancer console. At the top, there's a search bar labeled 'Search Load Balancers'. Below it is a table with columns: Load Balancer Name, DNS Name, Port Configuration, Availability Zones, Instance Count, and Health Check. One row is visible for a load balancer named 'web' with the DNS name 'web-1306826351.us-west-2....'. The 'Instances' tab is selected in the navigation bar below the table. A note says 'Connection Draining: Enabled, 300 seconds (Edit)'. Below this, there's an 'Edit Instances' section with a table showing one instance: 'i-e0dd50ea' in 'us-west-2b' with 'InService' status. There's also an 'Edit Availability Zones' section with a table showing three zones: 'us-west-2c', 'us-west-2b', and 'us-west-2a', all with 'No (Availability Zone contains no healthy instances)' status.

End to end Testing (EC2 Auto Scaling Groups)

Introduction

Performing end-to-end tests to make sure everything is working as you think it should is very important. Although this may be an automated procedure performed by a QA (Quality Assurance) department, often a quick sanity test by other individuals and/or groups directly from the AWS console is also helpful. This Lab Step will point out a few ways to test that your Launch Configuration is working in conjunction with the Auto Scaling group and CloudWatch Alarm (which uses AWS Simple Notification Service (SNS)).

Note: This Lab Step assumes you are fairly comfortable at this point with the AWS console and can navigate to the various parts of the console already used within this Lab. The *Instructions* below are fairly terse based on this assumption.

Instructions

Quick Tests

- Confirm an instance is running and settles at the desired number of one. *Reminder:* You can do this from the **EC2 Dashboard > Running Instances** or **Auto Scaling Group > Instances** tab:

The screenshot shows the EC2 Instances page. It has a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. One instance is listed with the following details: Name 'i-05156edfd36e3941', Instance ID 'i-05156edfd36e3941', Instance Type 't2.micro', Availability Zone 'us-west-2b', Instance State 'running' (indicated by a green circle icon), and Status Checks 'Initializing' (indicated by a red square icon).

- Check the email for the address you entered for the SNS notifications earlier. You should have received a **AWS Notification - Subscription Confirmation** email from **AWS Notifications** for both

a **autoscaling-alarm-up** and **autoscaling-alarm-down** SNS topics. Click the **confirm subscription** link in the email body if so.

- Open up a new browser window or tab. Enter either the **Public DNS** or **Public IP** address of the running instance. You should receive the **Amazon Linux AMI test page** from the Apache web server you installed and started via **User data** in the Launch Configuration. Now you know the instance is up and running and the Apache web server installed OK when the instance started.
- Navigate to **EC2 > Load Balancers**, then select the **Instances** tab for your Load Balancer. Confirm the running instance is registered with the Load Balancer *and* the **Status** is **InService** (meaning the health checks are running correctly):

The screenshot shows the AWS EC2 Load Balancers Instances tab. At the top, there is a search bar and a header with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. A single instance is listed: Web-770731984.us-west-2.e... with a DNS name of Web-770731984.us-west-2.elb.amazonaws.com, in a VPC with ID vpc-38fb35f, in Availability Zones us-west-2a, us-west-2b, and Type classic. Below the table, it says "Load balancer: Web". Underneath the table, there are tabs for Description, Instances (which is selected), Health Check, Listeners, Monitoring, and Tags. A note says "Connection Draining: Enabled, 300 seconds (Edit)". Below the tabs, there is an "Edit Instances" button. A table shows the instance details: Instance ID i-01cec5e43c03eb0f4, Name (empty), Availability Zone us-west-2b, Status InService (with a help icon), and Actions Remove from Load Balancer. Below this is an "Edit Availability Zones" button. Another table shows the availability zones: us-west-2a with Subnet ID subnet-0d368e44, Subnet CIDR 172.31.32.0/20, and Instance Count 0 (Healthy? No, Actions Remove from Load Balancer); us-west-2b with Subnet ID subnet-cfdcbaa8, Subnet CIDR 172.31.16.0/20, and Instance Count 1 (Healthy? Yes, Actions Remove from Load Balancer).

- Tip:* If trouble shooting efforts are required, here is a very handy trick. If, for example, health checks are failing, then chances are the running instance will get terminated automatically. A new one will start, but five minutes later the health check will also fail and that new instance will get terminated. Rinse/repeat! To "break" the cycle while troubleshooting:
 - Navigate to **Auto Scaling Groups > Details** tab
 - Click **Edit** then click the **Suspended Processes** field
 - Select **Terminate** (This will prevent instances in your group from getting terminated. **Important!** This trick is for troubleshooting only! Don't forget to remove the configuration once the issue is resolved.)
 - Note:** **Launch** is also an option here. Temporarily setting this to **Launch** has the same effect as setting **Desired** and **Min** to **0**. After either of these settings and manually terminating an instance, a new instance will not be started. (A very helpful trick if you want to put a hold on everything until the next day, saving on potential instance usage charges.)
- From the **EC2 > Load Balancing > Load Balancers** page, copy the **DNS name**. (Example: Web-770731984.us-west-2.elb.amazonaws.com) Paste the **DNS name** into a new browser window or tab. You should see the default **Amazon Linux AMI Test Page** again. Note

that earlier you saw this page, but you had bypassed the load balancer, hence it only confirmed the instance was up and running Apache. This tests that the ELB will route requests to an instance in its instance pool. Notice the URL includes the ELB you created earlier. The request went through the ELB to the instance:

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

The Amazon Linux AMI is a supported and maintained Linux image provided by [Amazon Web Services](#) for use on [Amazon Elastic Compute Cloud \(Amazon EC2\)](#). It is designed to provide a stable, secure, and high performance execution environment for applications running on [Amazon EC2](#). It also includes packages that enable easy integration with [AWS](#), including launch configuration tools and many popular AWS libraries and tools. [Amazon Web Services](#) provides ongoing security and maintenance updates to all instances running the Amazon Linux AMI. The Amazon Linux AMI is provided at no additional charge to [Amazon EC2 users](#).

If you are the website administrator:
You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTTP servers. Thanks for using Apache and the Amazon Linux AMI!

Powered by **APACHE** 2.4 POWERED BY ELECTRICITY

Auto Scaling Tests

Note: This section assumes you know how to SSH into the Linux instance. Either from the EC2 console (on a Mac OS/Linux local host) or using Putty (for a local Windows host). The details are not included in this Lab Step.

- From the **EC2 > Running Instances**, select **Actions > Instance State > Terminate** on the sole running instance. It should relaunch automatically. Let the new instance launch and settle into a running state. This could take a minute or two:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-03a8acc1279ff741d	t2.micro	us-west-2b	running	Initializing	None	ec2-35-165-125-215.us...	35.165.125.215
	i-0647cff3aa593d04b	t2.micro	us-west-2a	terminated		None		-

- SSH into the running instance and stress test it.
 - The simplest way to do this is to download and then use the correct key pair file supplied for you by the Cloud Academy platform under the **YOUR LAB DATA** section (at the top). Two key pairs are available for use: PEM (Privacy-Enhanced Mail) and PPK (public-private key pair). PEM files can be used directly by a local Linux SSH client. Connecting from a local Windows host to a running Linux host requires a PPK file. (Although you can use PuTTYgen to convert a PEM file to a PPK, Cloud Academy has done that for you for convenience sake.)
 - Hint for Linux users:* From the **Running Instances** page click **Connect**. The SSH command for the selected instance using the local private key file you previously downloaded earlier can be copy/pasted and used from a terminal window. You may need to change the permissions using the `chmod` command as instructed to make sure the key file has read-only permissions.)
 - Hint for Windows users:* Download and use the PPK file supplied by Cloud Academy. As previously noted, it's simpler than creating your own key, converting it using PuTTYgen and then connecting via PuTTY.

- Install stress by using the yum package manager. *Note:* Although stress should already be installed from the User data in your Launch Configuration, this will guarantee the latest/greatest version, and that the install directory is in your environment. (\$PATH)

```
$ sudo yum install stress
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB     00:00
amzn-updates                                     | 2.3 kB     00:00

Resolving Dependencies
--> Running transaction check
---> Package stress.x86_64 0:1.0.4-4.2.amzn1 will be installed
---> Finished Dependency Resolution

Dependencies Resolved

=====
=
Package          Arch      Version       Repository      Size
=====
=
Installing:
stress           x86_64   1.0.4-4.2.amzn1    amzn-main      38 k

Transaction Summary
=====
=
Install 1 Package

Total download size: 38 k
Installed size: 89 k
Is this ok [y/d/N]: y
Downloading packages:
stress-1.0.4-4.2.amzn1.x86_64.rpm          | 38 kB     00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stress-1.0.4-4.2.amzn1.x86_64
1/1
  Verifying  : stress-1.0.4-4.2.amzn1.x86_64
1/1

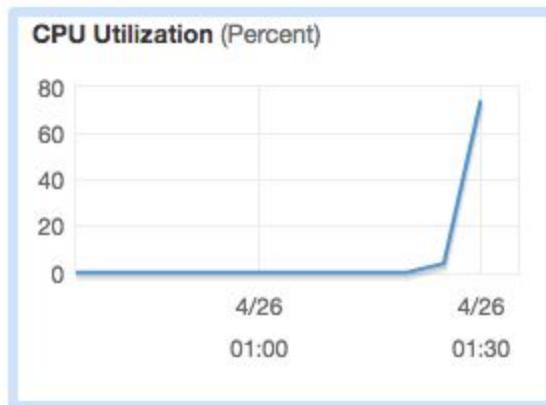
Installed:
stress.x86_64 0:1.0.4-4.2.amzn1
```

- Enter `which stress` to confirm the stress executable is in your path. (`/usr/bin/stress`)
- Finally, run stress to eat up CPU cycles for five minutes (`-t 5m` option):

```
$ stress -c 2 -i 1 -m 1 --vm-bytes 128M -t 5m
```

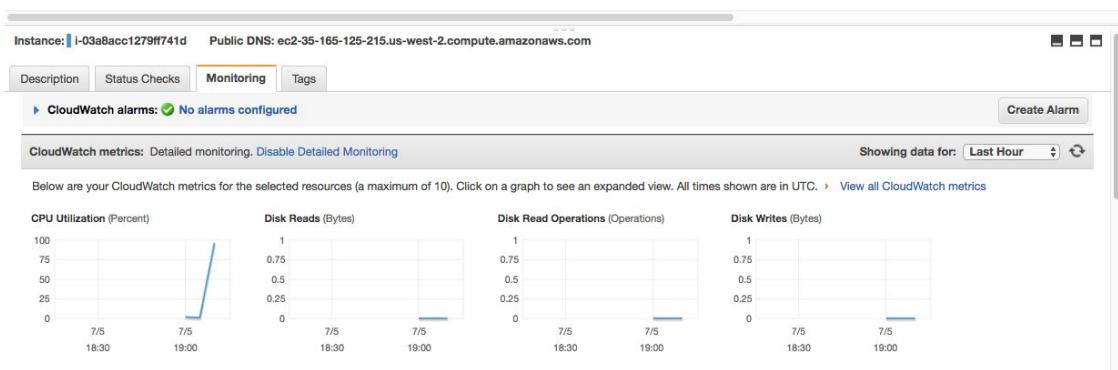
```
stress: info: [23710] dispatching hogs: 2 cpu, 1 io, 1 vm, 0 hdd
stress: info: [23710] successful run completed in 300s
```

- Navigate to **EC2 Dashboard > Running Instances > Monitoring tab**. Look at the **CPU Utilization**. Hint: You may want to click the refresh icon every 30 seconds or so on the bottom pane. It should quickly ramp up from nearly 0% to 100% within a few minutes:



- Recall the way that you configured the scaling policy in your Auto Scaling group. What do you think should happen as the CPU climbs on the running instance over the next 5+ minutes?
 - A new instance should automatically start:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-03a8acc1279ff741d	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-35-165-125-215.us...	35.165.125.215
	i-0647cff3aa593d04b	t2.micro	us-west-2a	terminated	0/2 checks ...	None	ec2-34-212-155-43.us-w...	-
	i-08eee4e64898989df	t2.micro	us-west-2a	pending	0/2 checks ...	None	ec2-34-212-155-43.us-w...	34.212.155.43



- Further, it should register with the ELB and pass health checks:

Load balancer: Web

Description Instances Health Check Listeners Monitoring Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-08eee4e64898989df		us-west-2a	InService ⓘ	Remove from Load Balancer
i-03a8acc1279ff741d		us-west-2b	InService ⓘ	Remove from Load Balancer

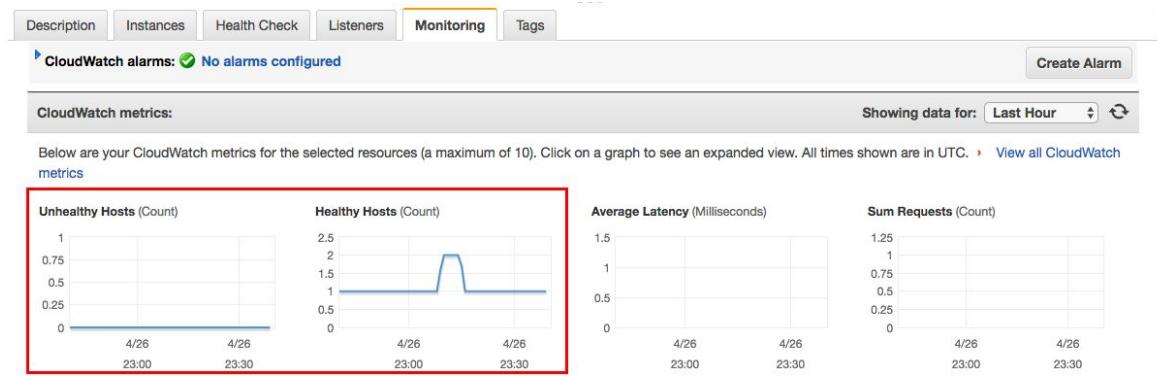
- Keep waiting for another ~5 minutes. Should anything else happen?
 - Yes! Since the stress test runs for only 5 minutes, the CPU utilization spread across both instances in your load balancer pool will quickly go back down to near zero. Since the scale-down policy relationship is <= 10, one of the two instances should be automatically shut down and then terminated:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-0fbed2e02d08909e3	t2.micro	us-west-2a	shutting-down	
	i-01cec5e43c03eb0f4	t2.micro	us-west-2b	running	2/2 checks ...

The **Instance State** transitions from **shutting-down** to **terminated** automatically. Congratulations... it's working! If you check your email, you should receive another notification as a CloudWatch Alarm is raised. For example, the email from **AWS Notifications** titled: **ALARM: "awsec2-webserver-cluster-Low-CPU-Utilization" in US West - Oregon**. Eventually, you should see one running instance, and two terminated instances (one you terminated manually, and another terminated based on the alarm triggered when the CPU dropped below 10% for 5 minutes):

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-08eee4e64898989df	t2.micro	us-west-2a	running	2/2 checks ...	None	ec2-34-212-155-43.us-w...	34.212.155.43
	i-03a8acc1279ff741d	t2.micro	us-west-2b	terminated		None		-
	i-0647cff3aa593d04b	t2.micro	us-west-2a	terminated		None		-

- Navigate to **Load Balancers > Monitoring** tab. This is another helpful page, where you can confirm no **Unhealthy Hosts**, and see the **Healthy Hosts** scale up from 1 to 2 and back down to 1:



When you are done looking at various screens in the AWS Management Console to check status, feel free to terminate any running instances. (Otherwise, we'll do it for you when the lab is completed.)

Summary

Although there is always more testing that could be done, or even automated testing implemented, the manual checks performed in this Lab Step are pretty thorough. Chances are very good that if all of the tests in this Lab Step work then your Auto Scaling Group, Launch Configuration, Elastic Load Balancer, CloudWatch alarm and SNS notification are all properly configured and working together. Congratulations!!