AWS Basics: Elastic Load Balancing

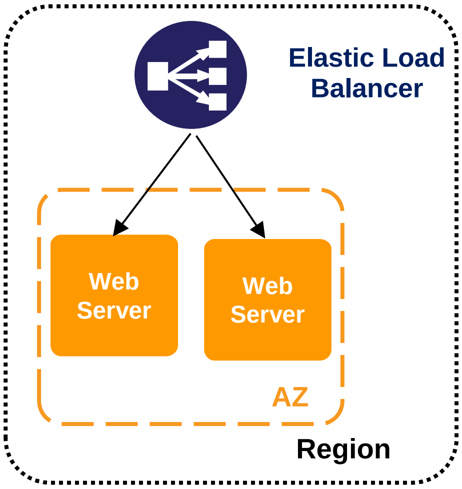
Table of Contents

Introduction

This lab will walk you through the process of creating an Elastic Load Balancer (ELB) to load balance traffic across multiple Amazon EC2 instances in a single Availability Zone. You will deploy a simple application on multiple Amazon EC2 instances and observe load balancing by viewing the application in your browser. The diagram below provides a high-level overview of the architecture you will implement in this exercise.

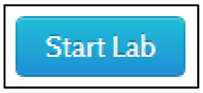
By completing this lab, you will achieve the following:

- Launch a multiple server web farm on Amazon EC2, using bootstrapping techniques to configure Linux instances with Apache, PHP and a simple PHP application downloaded from Amazon S3
- Create and configure an Elastic Load Balancer (ELB) to front your Amazon EC2 web server instances
- Explore Amazon CloudWatch metrics for the ELB

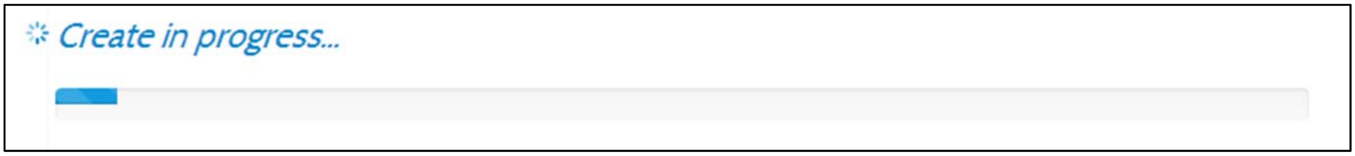


Start your *qwikLAB™*

1. Start your *qwikLAB™*
   Use the ‘Start Lab’ button to start your lab.
   (Hint: If you are prompted for a token, please use one you’ve been given or have purchased.)



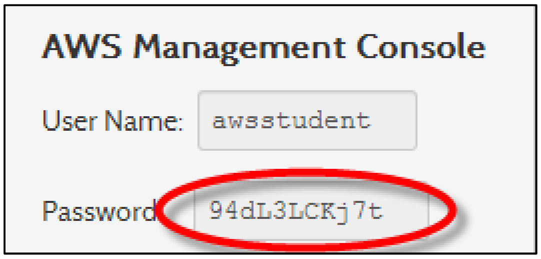
You will see the lab creation in progress.



Create in progress...

2. Note a few properties of the lab.
   a. **Duration -** The time the lab will run for before shutting itself down.
   b. **Setup Time -** The estimated lab creation time on starting the lab.
   c. **AWS Region** - The AWS Region the lab resources are being created in.

3. Copy the Password provided.
   Hint: selecting the value shown and using Ctrl+C works best

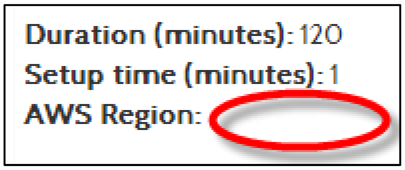


AWS Management Console

User Name: awsstudent

Password 94dL3LCKj7t

4. Note the AWS Region set for your lab in *qwikLAB™*



Duration (minutes): 120
Setup time (minutes): 1
AWS Region:

5. Click the ‘Open Console’ button.

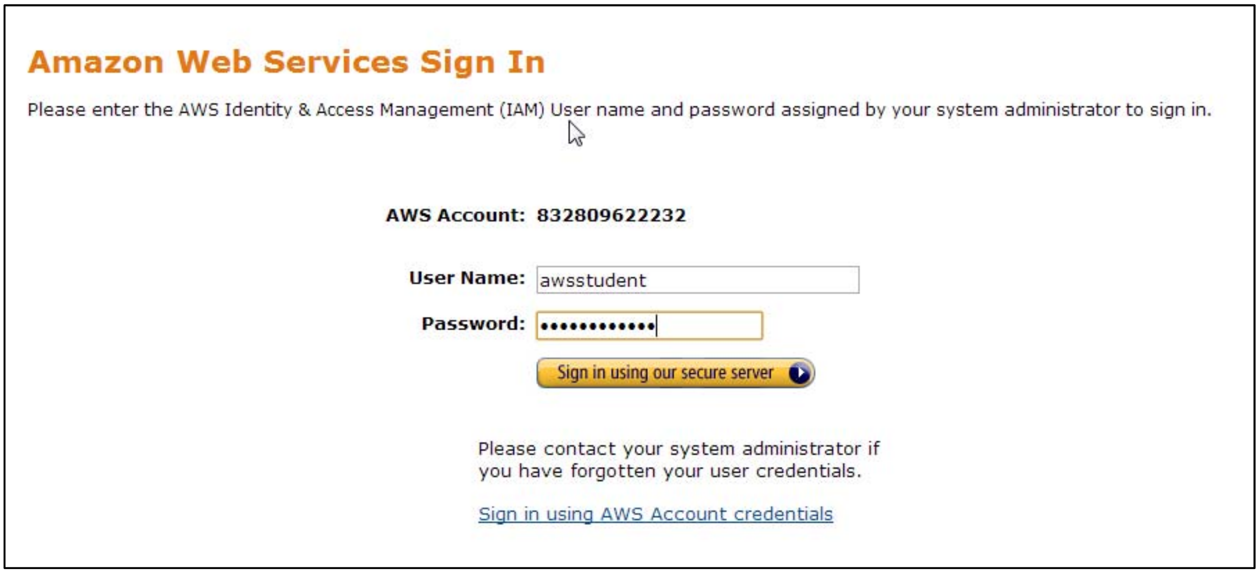
6. Make sure that you are not logged into any other instances of the AWS console (in a student account or your own account), as this may cause conflicts when you open the console and log in below for this lab.

7. Login to the AWS Management Console

   Enter the User Name '**awsstudent**' and paste the password you copied from the lab details in *qwikLAB*™ into the Password field.

   Click on the 'Sign in using our secure server' button.

   In this step you logged into the AWS Management Console using login credentials for a user provisioned via AWS Identity Access Management in an AWS account by *qwikLAB*™.



Amazon Web Services Sign In

Please enter the AWS Identity & Access Management (IAM) User name and password assigned by your system administrator to sign in.

AWS Account: 832809622232

User Name: awsstudent

Password: ••••••••••••

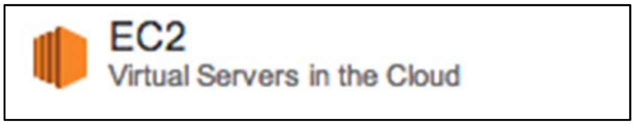
Sign in using our secure server

Please contact your system administrator if you have forgotten your user credentials.

Sign in using AWS Account credentials

AWS Management Console

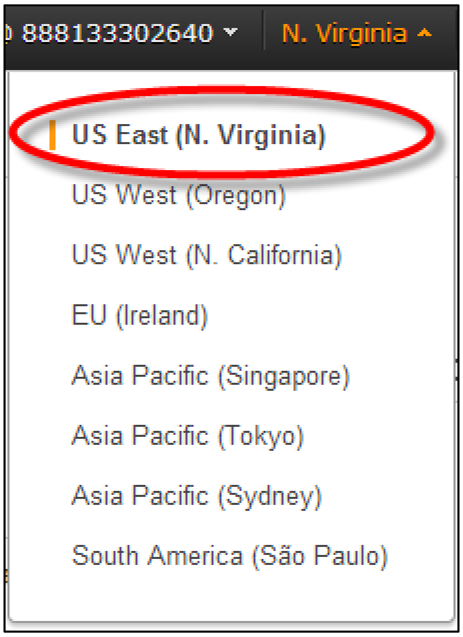
8. Select "EC2" from the Console Home



EC2
Virtual Servers in the Cloud

Confirm your AWS Region

9. Select or confirm that the same AWS Region is already set in the AWS Management Console

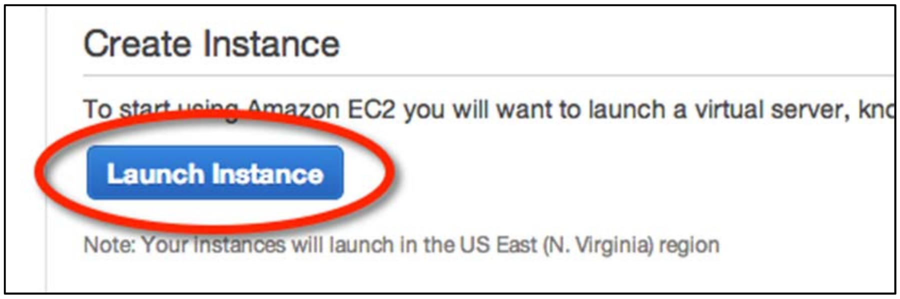


Launch Web Servers

In this step, we will launch 2 Amazon Linux EC2 instances, with an Apache PHP web server and basic application installed on initialization. We also demonstrate a simple example of bootstrapping instances using the Amazon EC2 metadata service.

To start your instances, navigate to the Amazon EC2 Dashboard in the AWS Console.

1. Click on **Launch Instance**

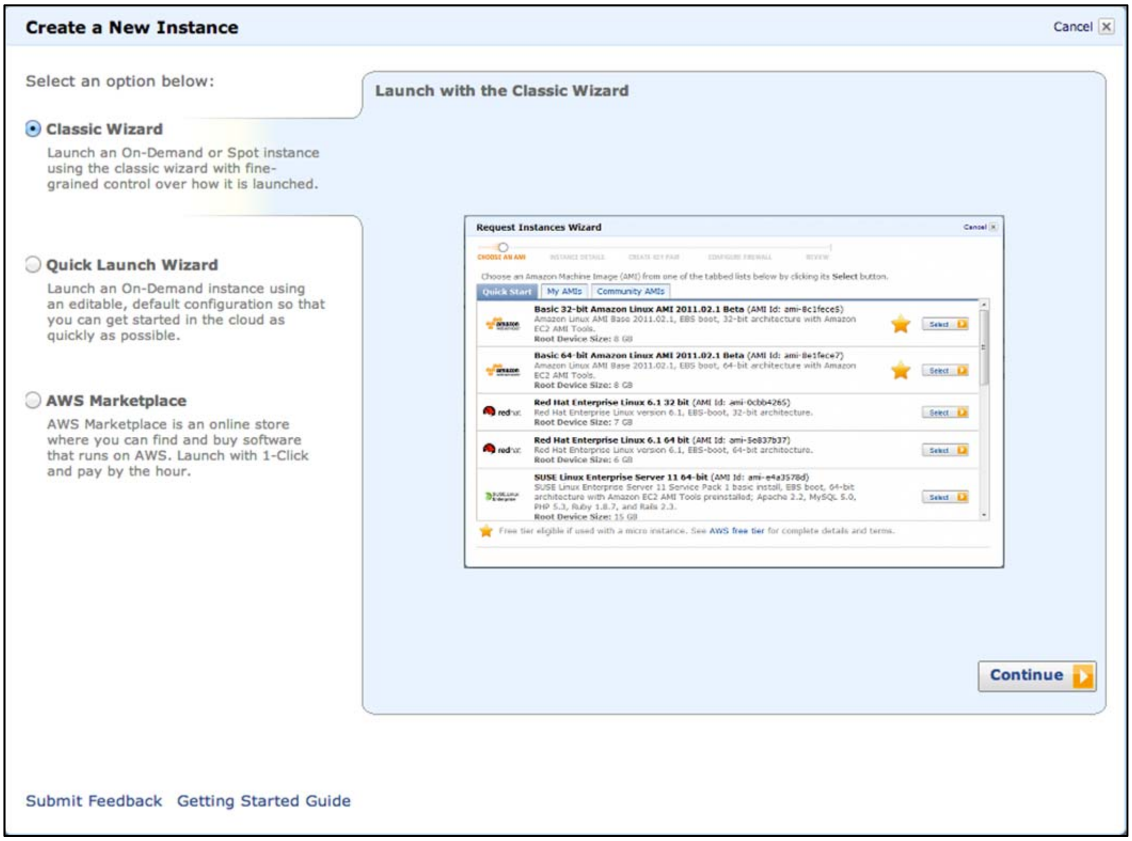


Create Instance

To start using Amazon EC2 you will want to launch a virtual server, kno

Note: Your Instances will launch in the US East (N. Virginia) region

2. Next, select **Classic Wizard** and click **Continue.**
   It is possible to start your instances using the Quick Launch Wizard, but for the purposes of this lab, we want to see all the settings step by step, which is the way the Classic Wizard captures instance configuration.



● **Classic Wizard**
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

○ **Quick Launch Wizard**
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

○ **AWS Marketplace**
AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Basic 32-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8c1fece5)
Amazon Linux AMI Base 2011.02.1, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Basic 64-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8e1fece7)
Amazon Linux AMI Base 2011.02.1, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Red Hat Enterprise Linux 6.1 32 bit (AMI Id: ami-0cbb4265)
Red Hat Enterprise Linux version 6.1, EBS-boot, 32-bit architecture.
Root Device Size: 7 GB

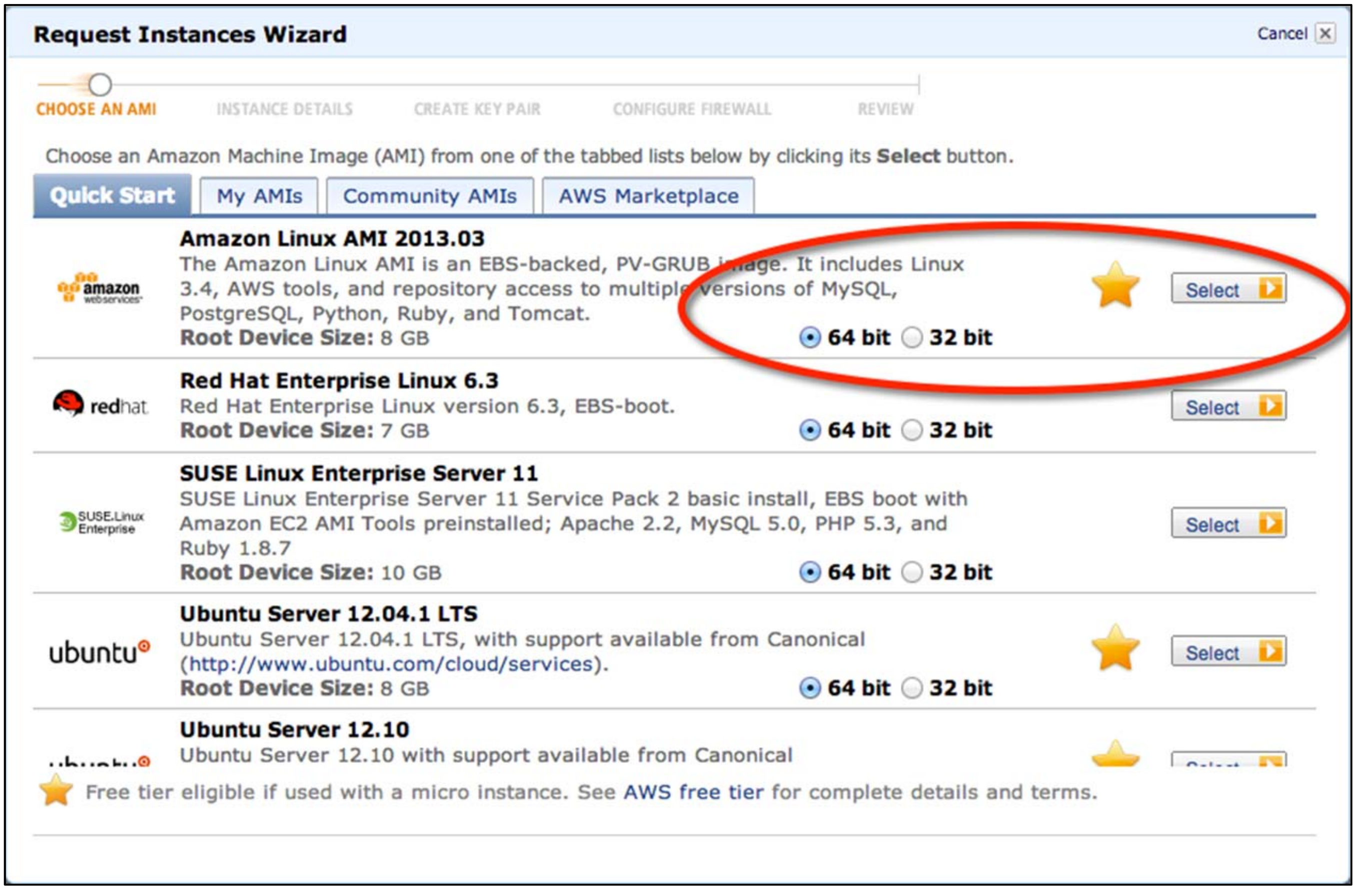
Red Hat Enterprise Linux 6.1 64 bit (AMI Id: ami-5e837b37)
Red Hat Enterprise Linux version 6.1, EBS-boot, 64-bit architecture.
Root Device Size: 6 GB

SUSE Linux Enterprise Server 11 64-bit (AMI Id: ami-e4a3578d)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3.
Root Device Size: 15 GB

★ Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

.

3. Now select the basic 64-bit Amazon Linux AMI.
   This is a machine image from which our instances will be created. In this case, we are selecting an Amazon maintained Linux distribution with access to the repositories needed to install software for this lab (such as Apache and PHP).



Request Instances Wizard

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Amazon Linux AMI 2013.03
The Amazon Linux AMI is an EBS-backed, PV-GRUB image. It includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.
Root Device Size: 8 GB ● **64 bit** ○ **32 bit**

Red Hat Enterprise Linux 6.3
Red Hat Enterprise Linux version 6.3, EBS-boot.
Root Device Size: 7 GB ● **64 bit** ○ **32 bit**

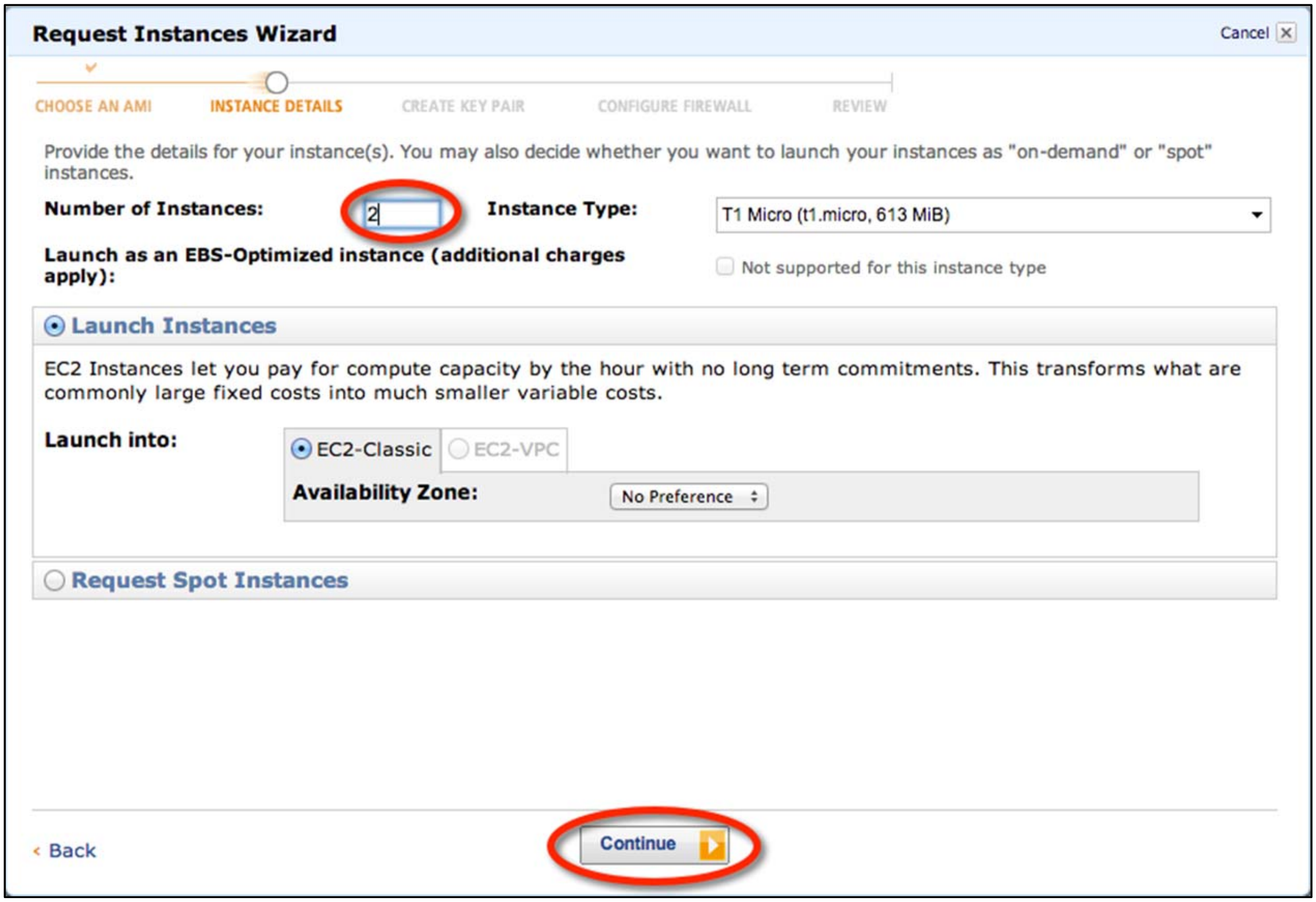
SUSE Linux Enterprise Server 11
SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7
Root Device Size: 10 GB ● **64 bit** ○ **32 bit**

Ubuntu Server 12.04.1 LTS
Ubuntu Server 12.04.1 LTS, with support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Size: 8 GB ● **64 bit** ○ **32 bit**

Ubuntu Server 12.10
Ubuntu Server 12.10 with support available from Canonical

Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

4. We want to start more than one instance for this lab, so change the number of instances to **2** and click **Continue**.



Request Instances Wizard

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

Number of Instances: 2

Instance Type: T1 Micro (t1.micro, 613 MiB)

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

● Launch Instances

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

Launch into: ● EC2-Classic ○ EC2-VPC

Availability Zone: No Preference

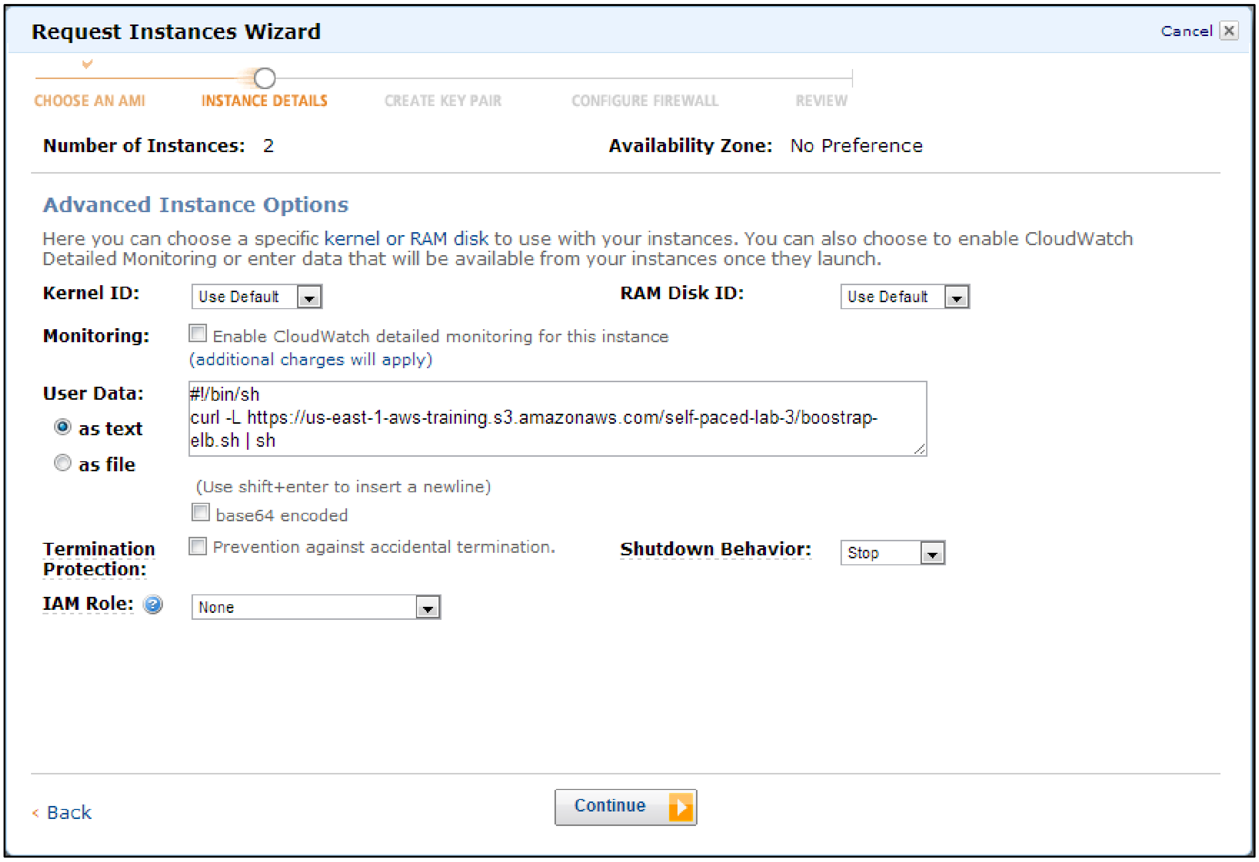
○ Request Spot Instances

On the Advanced Instance Options page, we will use the **User Data** field to bootstrap our instance, running a custom script to install the software packages (Apache and PHP) and sample code (PHP scripts) needed for this lab. User Data provides a mechanism to pass data or a script to the Amazon metadata service, which instances can access at launch time.

5. Copy and paste the following initialization script into the **User Data** field and click **Continue**:

```
#!/bin/sh
curl -L https://us-east-1-aws-training.s3.amazonaws.com/self-paced-lab-3/bootstrap-elb.sh | sh
```

This script will download and install various components, start them, and install our sample application. Note, if you experience problems, try pasting the script above into a text editor like notepad and then copy-paste in the User Data field.



Request Instances Wizard

Number of Instances: 2

Availability Zone: No Preference

Advanced Instance Options

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

Kernel ID: Use Default

RAM Disk ID: Use Default

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

User Data: ● as text ○ as file

```
#!/bin/sh
curl -L https://us-east-1-aws-training.s3.amazonaws.com/self-paced-lab-3/boostrap-elb.sh | sh
```

(Use shift+enter to insert a newline)

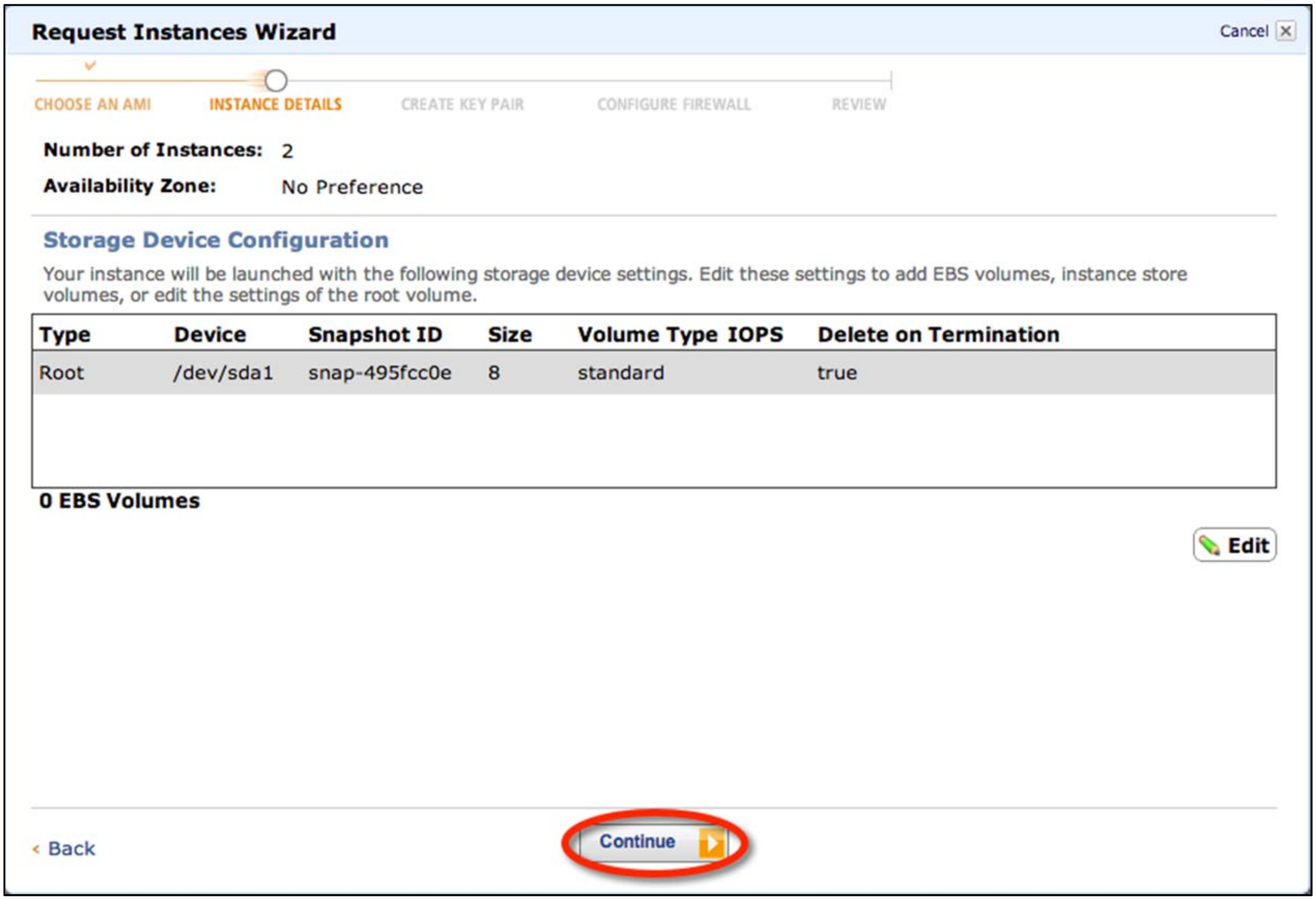
□ base64 encoded

Termination Protection: □ Prevention against accidental termination.

Shutdown Behavior: Stop

IAM Role: None

6. The Storage Device Configuration page allows you to configure disk volumes for your instance. For this exercise, we will keep the default settings, so click **Continue** to proceed.



Number of Instances: 2

Availability Zone: No Preference

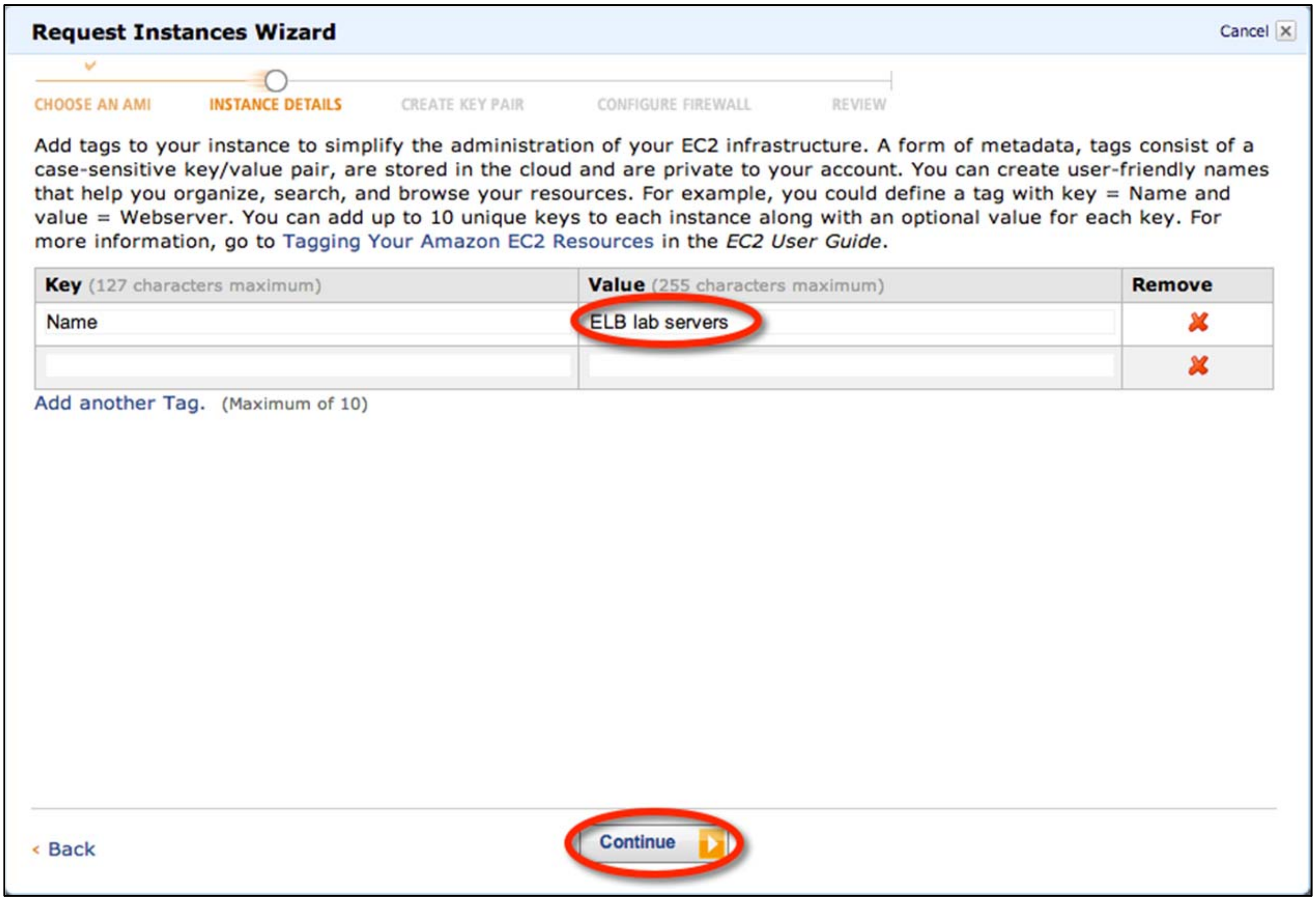
Storage Device Configuration

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

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-495fcc0e	8	standard		true

0 EBS Volumes

7. On the Tags page, give your new instances a readable **Name** like **ELB Lab Servers** and click **Continue.**

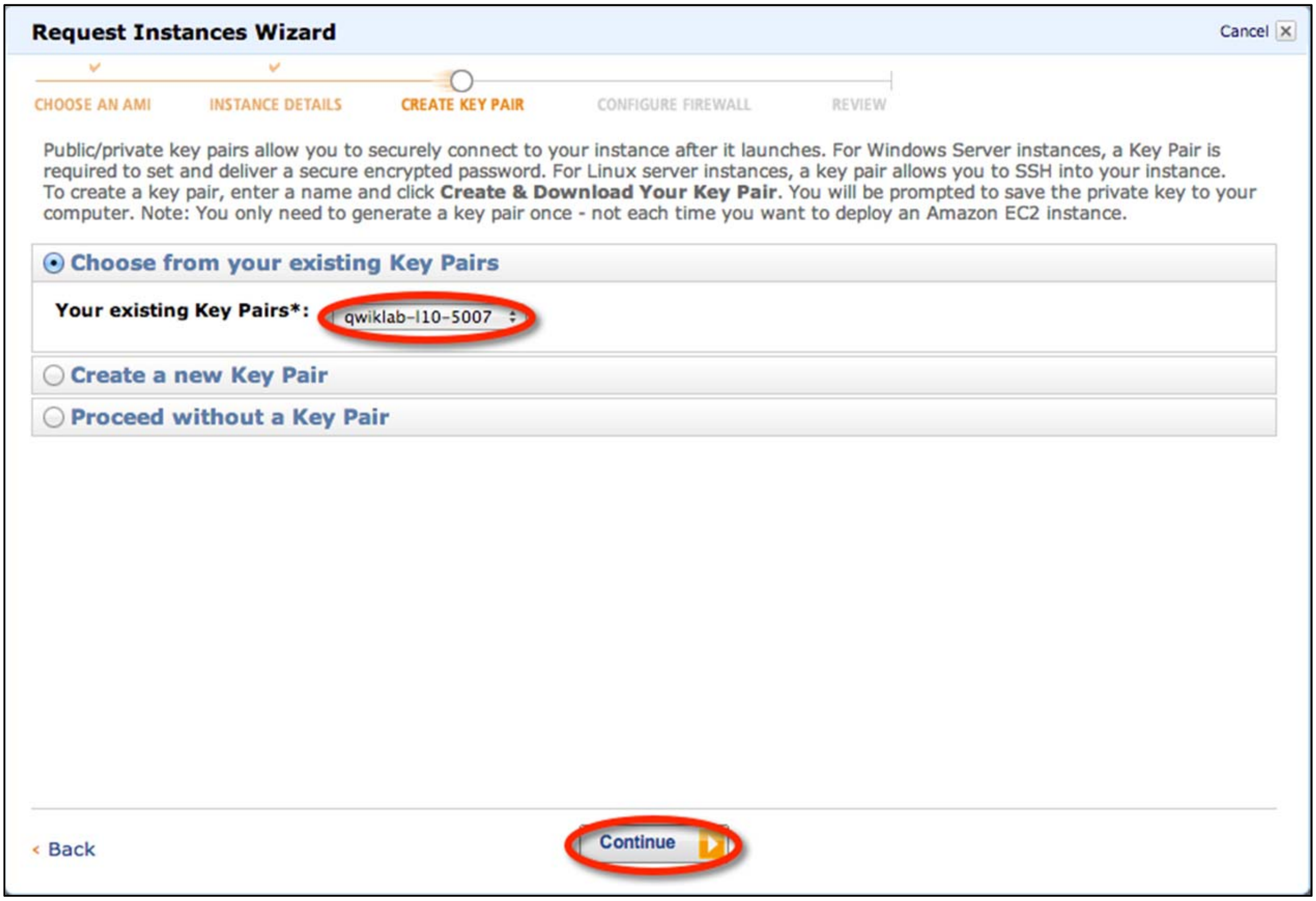


Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to Tagging Your Amazon EC2 Resources in the *EC2 User Guide*.

Key (127 characters maximum)	**Value** (255 characters maximum)	**Remove**
Name	ELB lab servers	✗
		✗

Add another Tag. (Maximum of 10)

8. On the Key Pairs page, you should see a *qwik*LAB™ generated Key Pair. Key Pairs can be used to access your instances with SSH. Make sure this Key Pair is selected, and click **Continue** to proceed. (Note: The following screen shot is just an example. The actual Key Pair name you see should match the name of the Key Pair you download from the *qwik*LAB™ start page.)



Request Instances Wizard

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

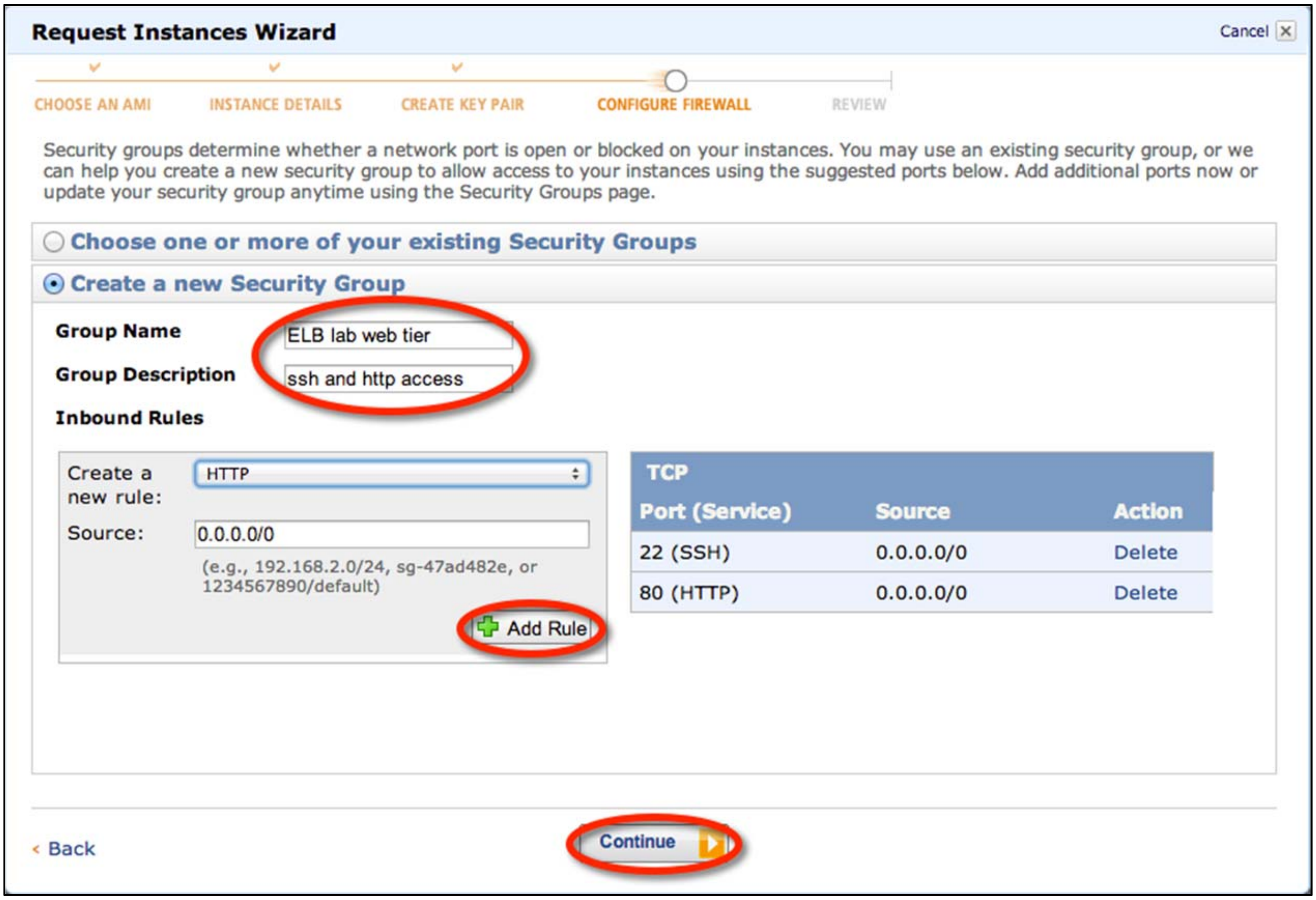
● **Choose from your existing Key Pairs**

Your existing Key Pairs*: qwiklab-l10-5007

○ **Create a new Key Pair**

○ **Proceed without a Key Pair**

9. Create a Security Group, which will contain your firewall rules. Since we are deploying a web server farm, you can name this “ELB lab web tier”, and open access to ports 22 and 80. Add a rule for both ports. This enables SSH and HTTP traffic to your web server instances. When you’re done, click **Continue** to proceed.



Request Instances Wizard

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

○ Choose one or more of your existing Security Groups

● Create a new Security Group

Group Name: ELB lab web tier

Group Description: ssh and http access

Inbound Rules

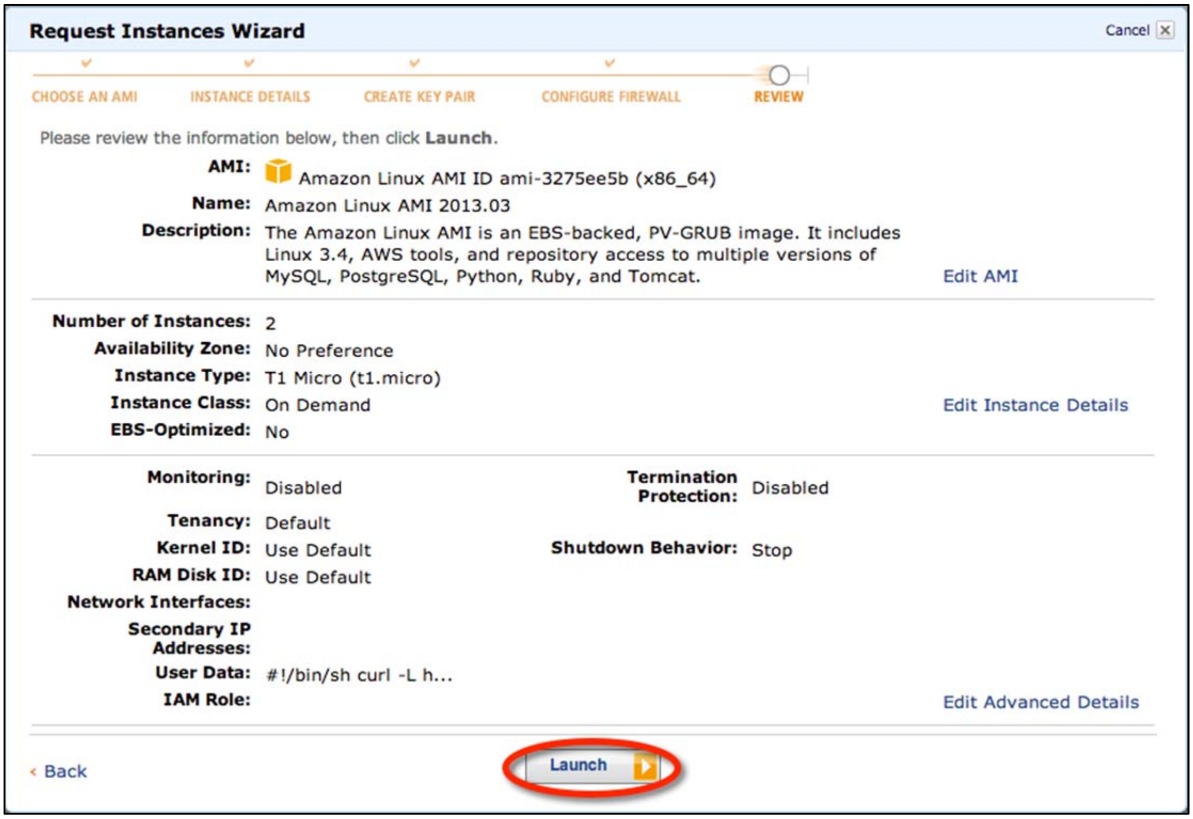
Create a new rule: HTTP

Source: 0.0.0.0/0

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

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

10. Review your choices, and then click **Launch**. Your instances will now start.
11.



Request Instances Wizard

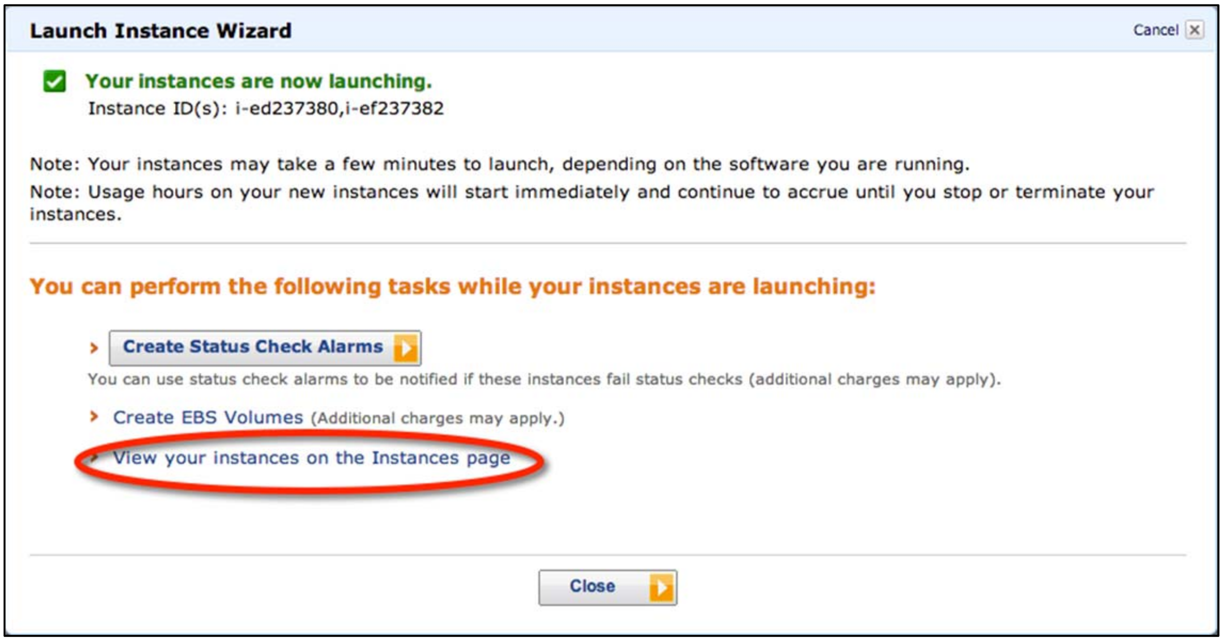
Please review the information below, then click **Launch**.

AMI: Amazon Linux AMI ID ami-3275ee5b (x86_64)
Name: Amazon Linux AMI 2013.03
Description: The Amazon Linux AMI is an EBS-backed, PV-GRUB image. It includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.

Number of Instances: 2
Availability Zone: No Preference
Instance Type: T1 Micro (t1.micro)
Instance Class: On Demand
EBS-Optimized: No

Monitoring: Disabled
Termination Protection: Disabled
Tenancy: Default
Kernel ID: Use Default
Shutdown Behavior: Stop
RAM Disk ID: Use Default
Network Interfaces:
Secondary IP Addresses:
User Data: #!/bin/sh curl -L h...
IAM Role:

12. Click **View your instances on the Instances page**.



Launch Instance Wizard

☑ **Your instances are now launching.**
Instance ID(s): i-ed237380,i-ef237382

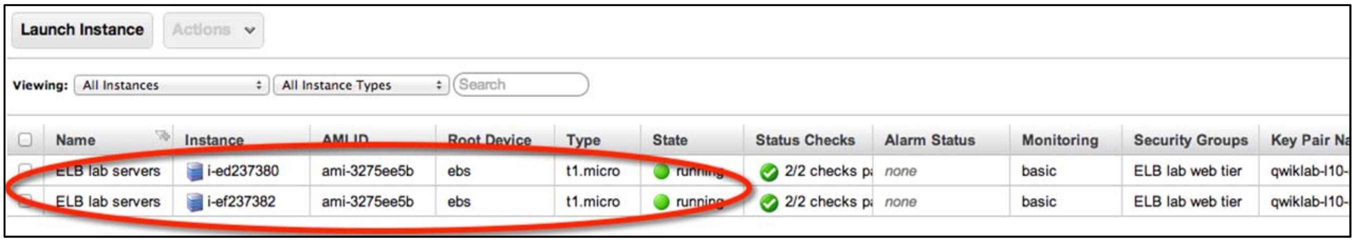
Note: Your instances may take a few minutes to launch, depending on the software you are running.
Note: Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

You can perform the following tasks while your instances are launching:

› Create Status Check Alarms
You can use status check alarms to be notified if these instances fail status checks (additional charges may apply).
› Create EBS Volumes (Additional charges may apply.)
› View your instances on the Instances page

Independently Connect to Each Web Server

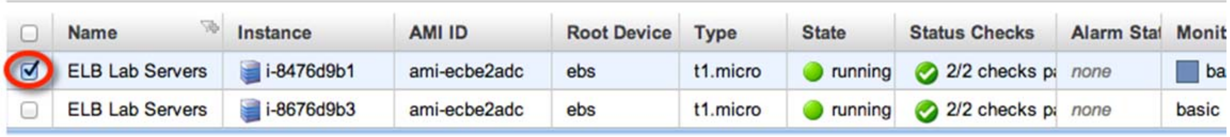
First, check that the instances we started have finished their creation cycle by monitoring to make certain they are running. You'll notice that the instances transition to a 'running' state with '2/2 checks passed'.



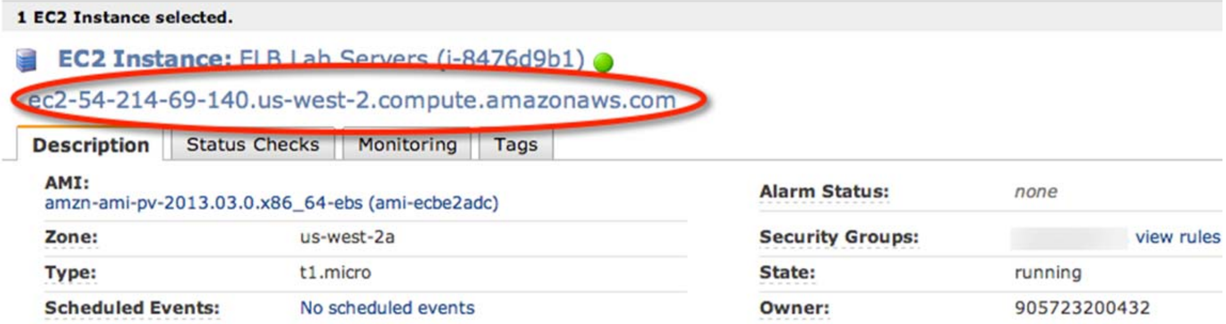
☐	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Security Groups	Key Pair Na
☐	ELB lab servers	i-ed237380	ami-3275ee5b	ebs	t1.micro	● running	✓ 2/2 checks pa	none	basic	ELB lab web tier	qwiklab-l10-
☐	ELB lab servers	i-ef237382	ami-3275ee5b	ebs	t1.micro	● running	✓ 2/2 checks pa	none	basic	ELB lab web tier	qwiklab-l10-

Once the servers are up and running, we will retrieve the public DNS entry allocated to each server, in order to access them from our web browser.

1. Click on the first EC2 instance, locate the server's DNS name, and select and copy the server's name.



☐	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Stat	Monit
☑	ELB Lab Servers	i-8476d9b1	ami-ecbe2adc	ebs	t1.micro	● running	✓ 2/2 checks pa	none	ba
☐	ELB Lab Servers	i-8676d9b3	ami-ecbe2adc	ebs	t1.micro	● running	✓ 2/2 checks pa	none	basic



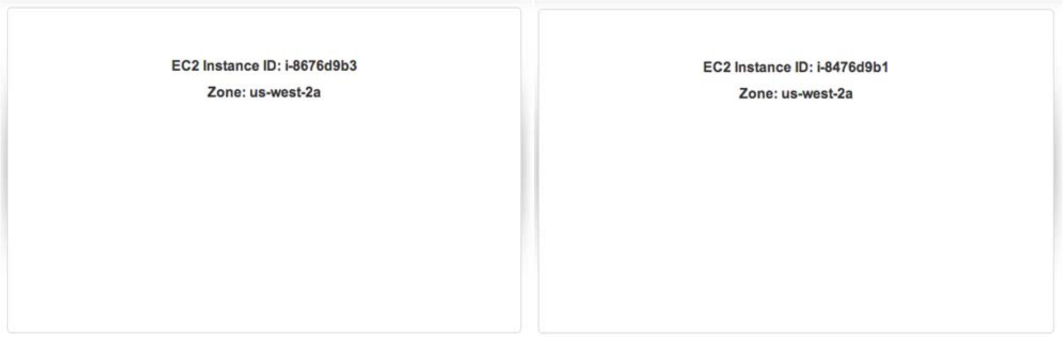
1 EC2 Instance selected.

EC2 Instance: ELB Lab Servers (i-8476d9b1)

ec2-54-214-69-140.us-west-2.compute.amazonaws.com

AMI: amzn-ami-pv-2013.03.0.x86_64-ebs (ami-ecbe2adc)		**Alarm Status:**	none
Zone:	us-west-2a	**Security Groups:**	view rules
Type:	t1.micro	**State:**	running
Scheduled Events:	No scheduled events	**Owner:**	905723200432

2. Paste the DNS name into a browser tab. Repeat for the second EC2 instance. You should see something like the following:



EC2 Instance ID: i-8676d9b3

Zone: us-west-2a

EC2 Instance ID: i-8476d9b1

Zone: us-west-2a

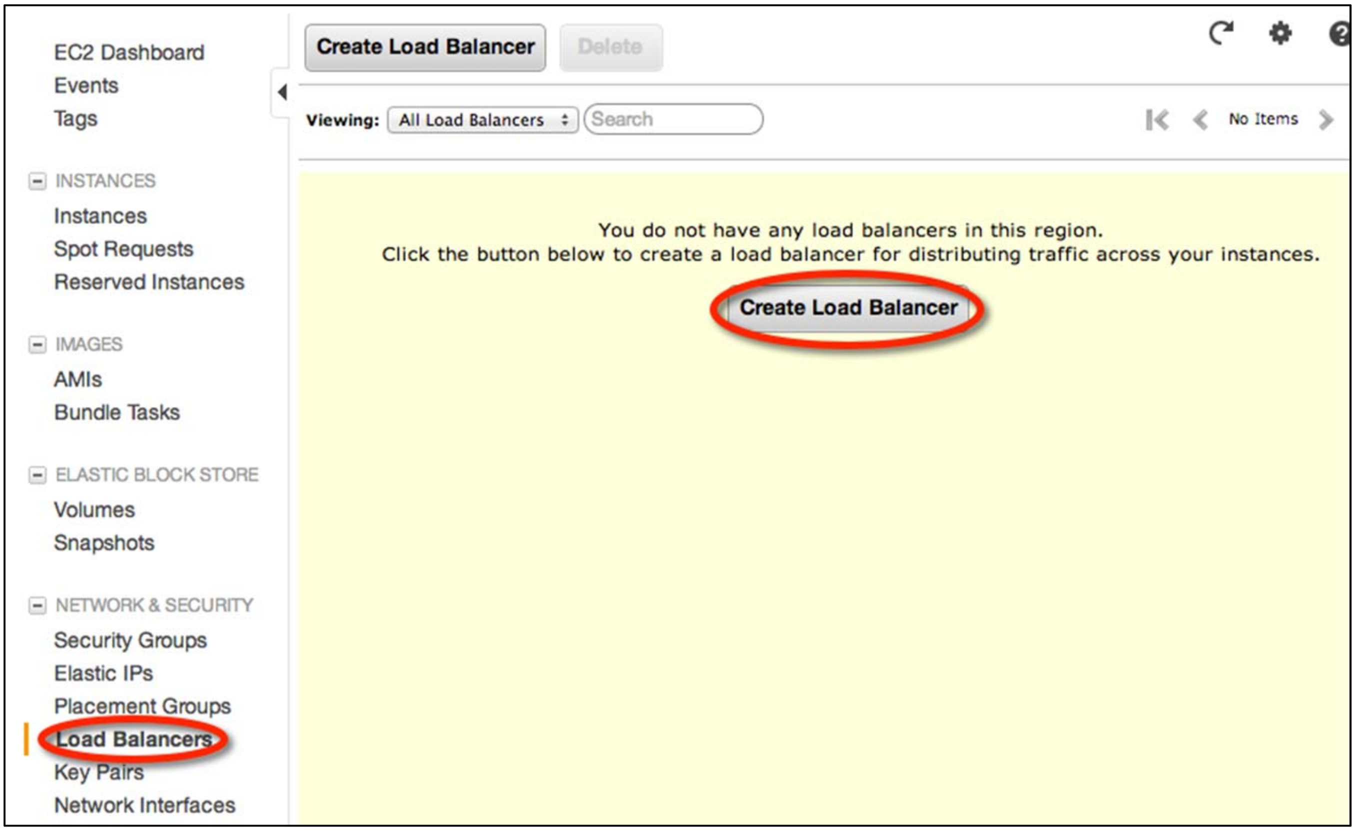
This is the web page returned by the PHP script that was installed when the instance was started. It is a simple script that interrogates the metadata service and returns the instance ID and in which availability zone it is running. Notice that each displays a different instance ID. This will help you identify which instance is processing your request when we put an Elastic Load Balancer in front of them.

Note: if you see an error instead of the instance ID and zone when you access the instances from the browser, try again for a couple of minutes. It’s possible that the bootstrapping script is still running, and has not yet completed installing and starting the web server and PHP application. If errors persist, verify that you entered the bootstrap script correctly in User Data when you launched your instances, and that the security group has port 80 open.

Create an Elastic Load Balancer (ELB)

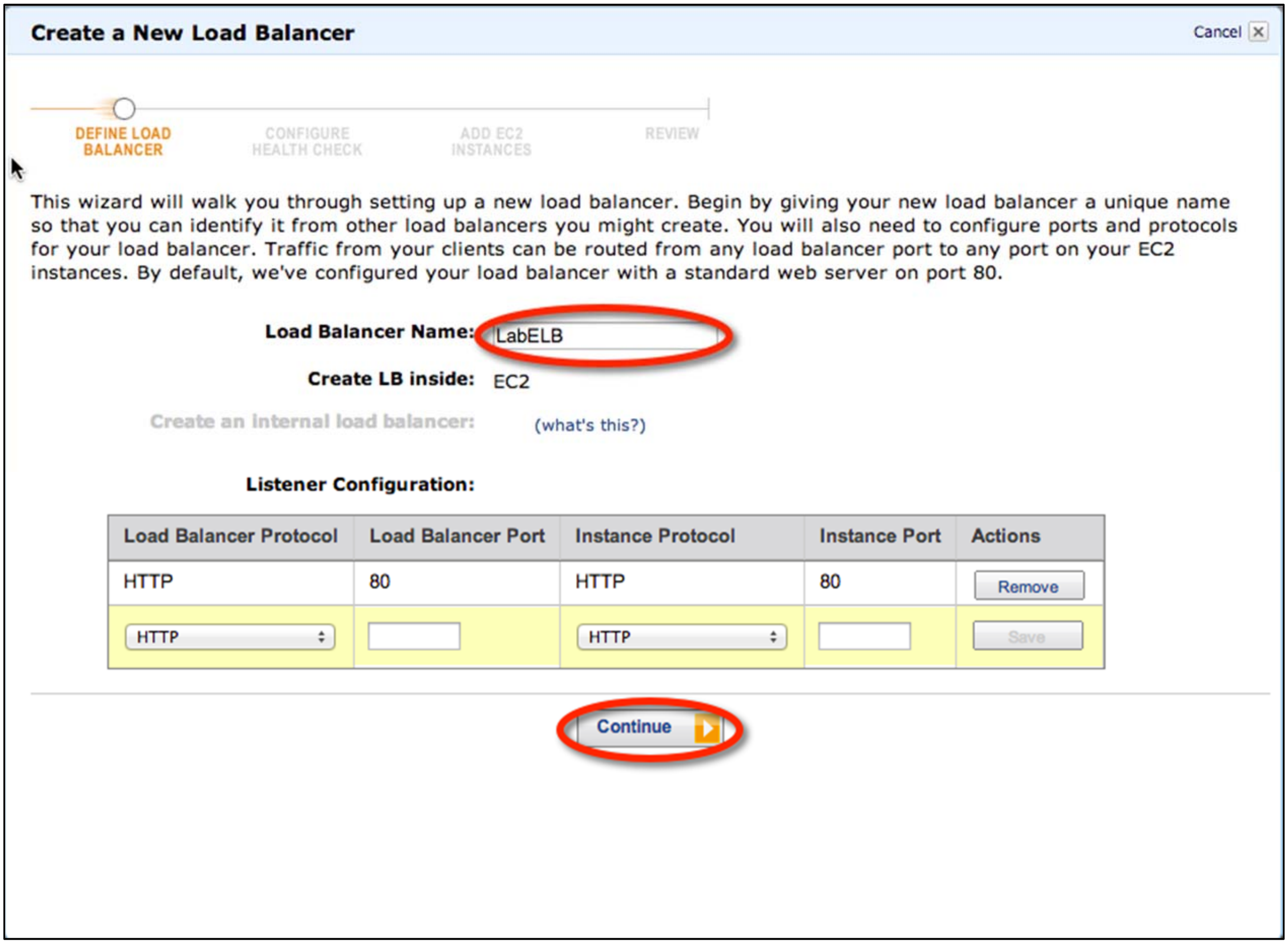
You now have two web servers, but you need a load balancer in front of these servers to give your users a single location for accessing both servers and to balance user requests across them.

1. In the **EC2 console**, click the **Load Balancers** link,
2. Then click the **Create Load Balancer** button.



You do not have any load balancers in this region.
Click the button below to create a load balancer for distributing traffic across your instances.

3. For this lab, we will be creating a simple HTTP load balancer, so give your ELB a new name like **LabELB**, accept the default listener, and click **Continue**.



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: LabELB

Create LB inside: EC2

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

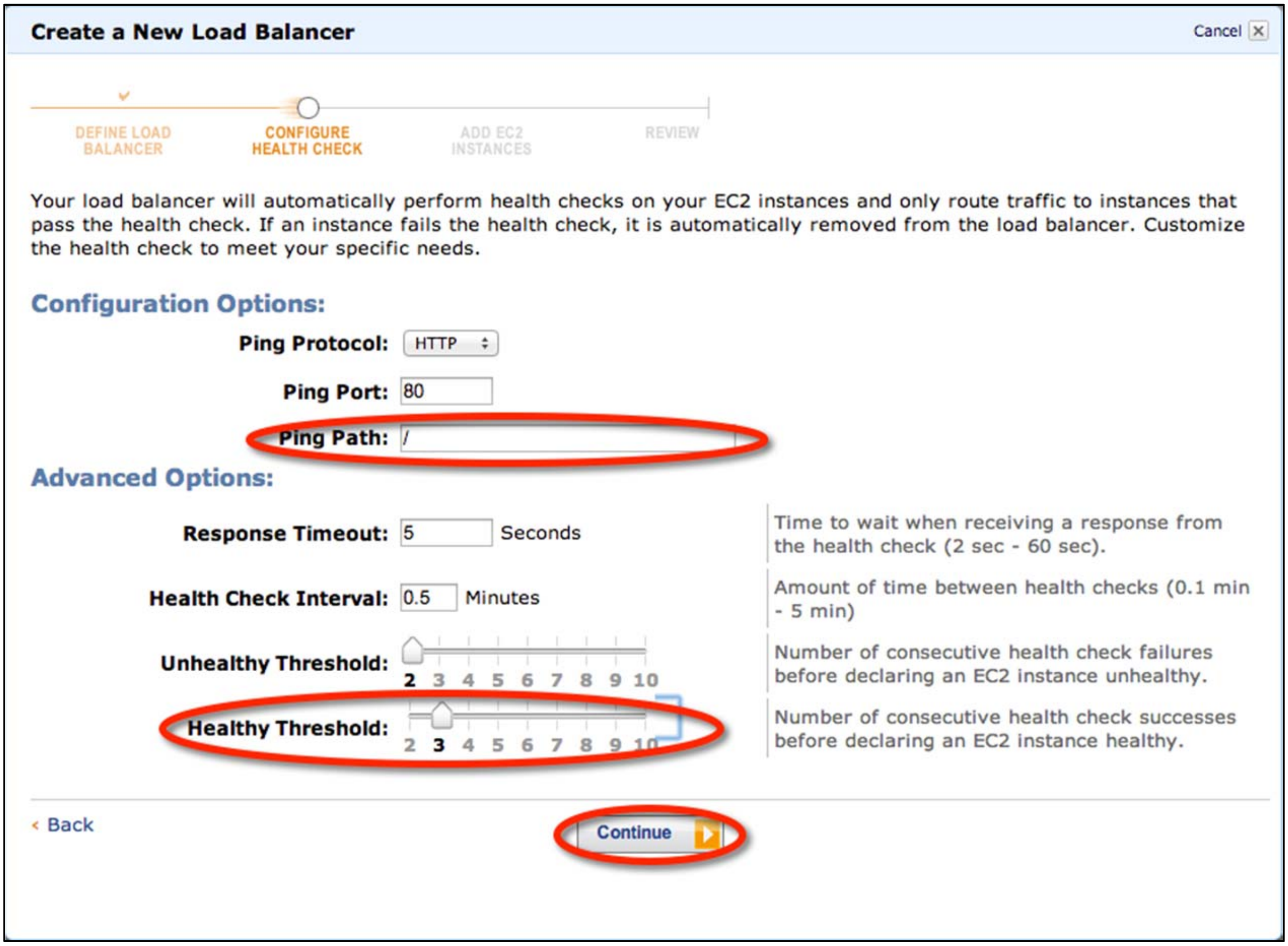
Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Actions
HTTP	80	HTTP	80	Remove
HTTP		HTTP		Save

4. On the next page, change **Ping Path** to **/** (delete the **index.html** part) and change the Healthy Threshold to 3. The ELB will periodically test the Ping Path URL on each of your web service instances to determine health – a 200 HTTP response code indicates healthy status, any other response code indicates unhealthy. If an instance is unhealthy and continues in that state for a successive number of checks (**Unhealthy Threshold**), the ELB will remove it from service until it recovers.

In our example, **/** will return the default page – the PHP generated page seen earlier. The Healthy Threshold is the number of successful checks the ELB expects to see in a row before bringing an instance into service behind the ELB. We are lowering this value to speed things up for this exercise.

5. Click **Continue** to proceed.



Create a New Load Balancer

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.

Configuration Options:

Ping Protocol: HTTP

Ping Port: 80

Ping Path: /

Advanced Options:

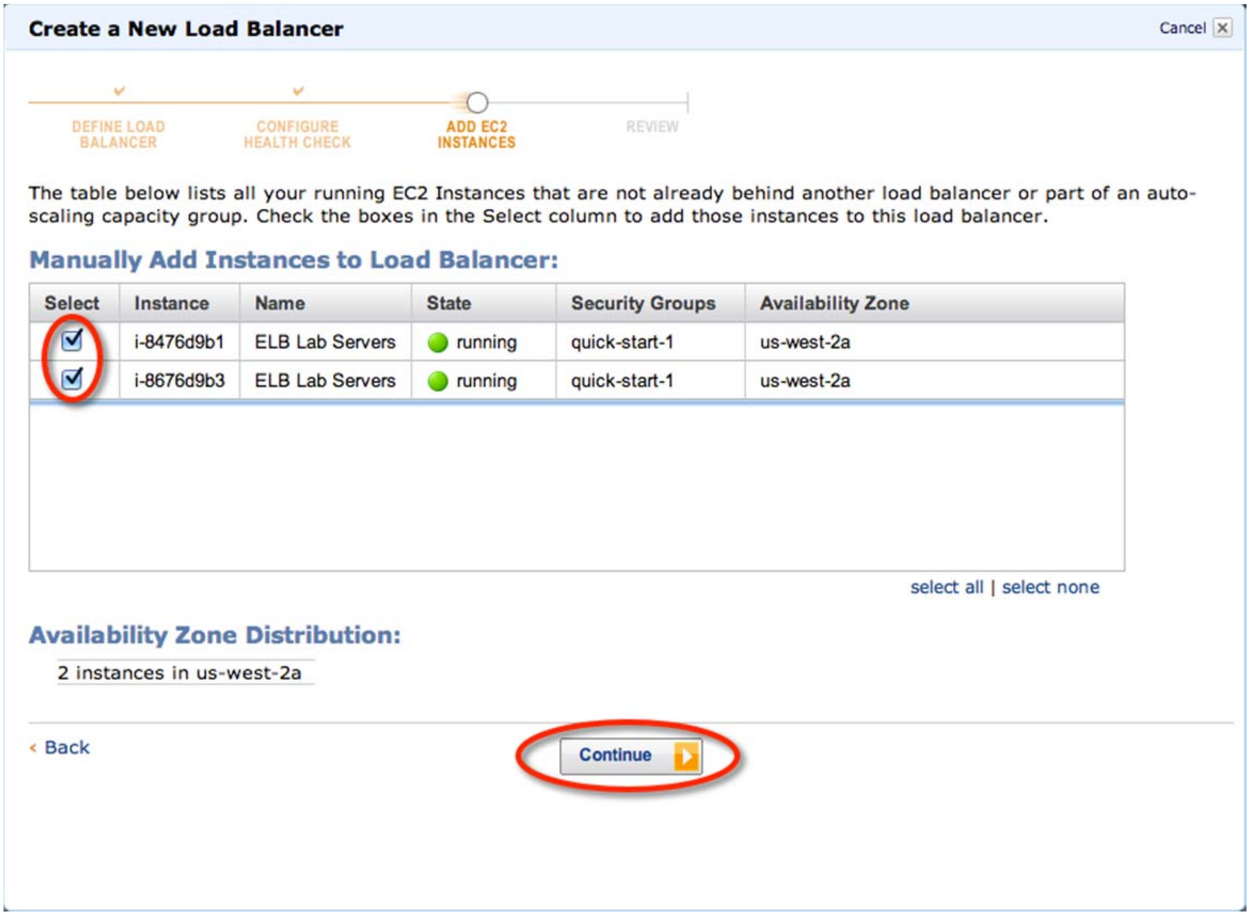
Response Timeout: 5 Seconds — Time to wait when receiving a response from the health check (2 sec - 60 sec).

Health Check Interval: 0.5 Minutes — Amount of time between health checks (0.1 min - 5 min)

Unhealthy Threshold: 2 — Number of consecutive health check failures before declaring an EC2 instance unhealthy.

Healthy Threshold: 3 — Number of consecutive health check successes before declaring an EC2 instance healthy.

6. Select your web server instances to add them to your ELB, and click **Continue**.



The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer.

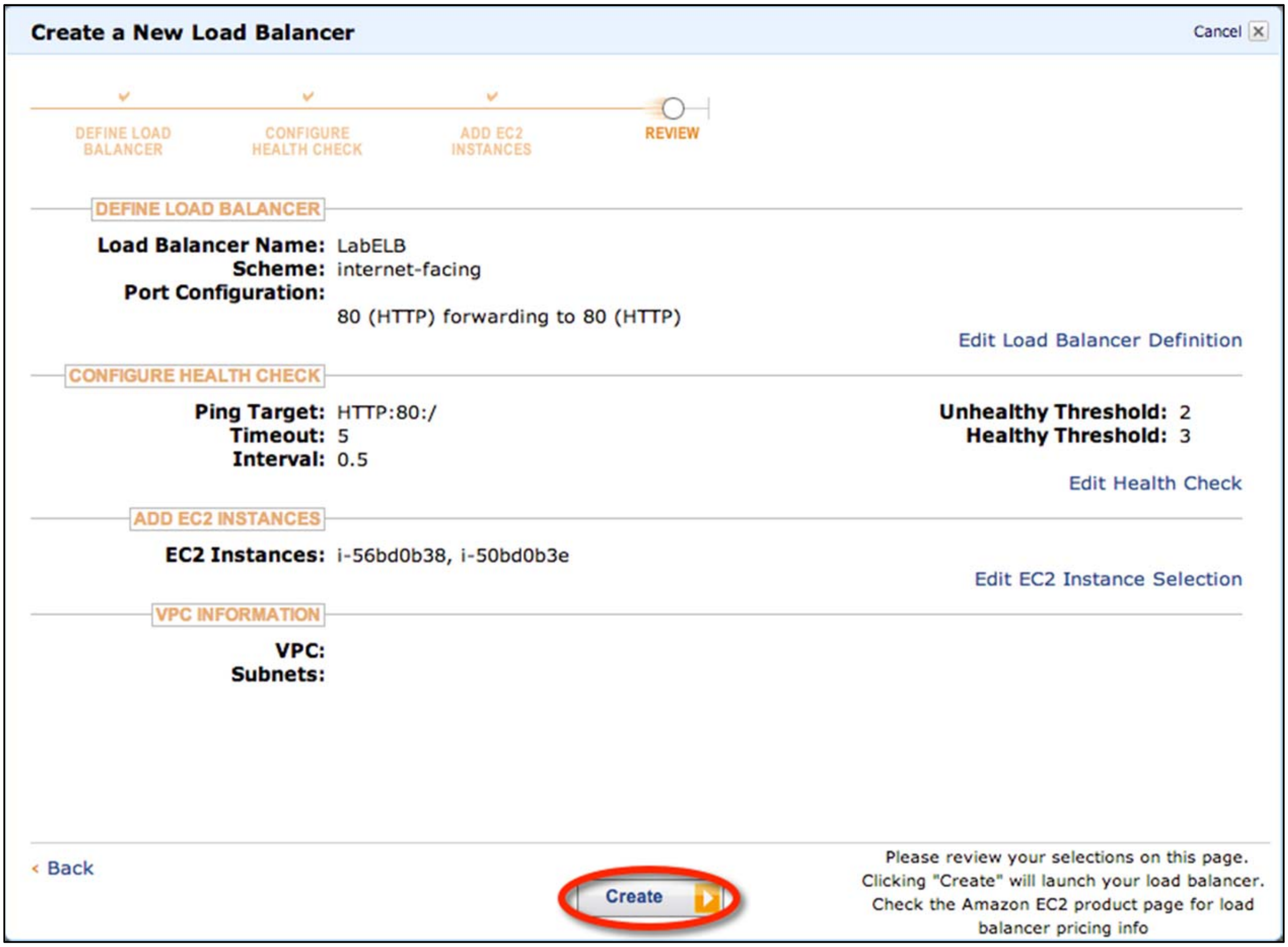
Manually Add Instances to Load Balancer:

Select	Instance	Name	State	Security Groups	Availability Zone
☑	i-8476d9b1	ELB Lab Servers	● running	quick-start-1	us-west-2a
☑	i-8676d9b3	ELB Lab Servers	● running	quick-start-1	us-west-2a

Availability Zone Distribution:

2 instances in us-west-2a

7. Review your ELB settings, and click **Create** (followed by **Close**).



DEFINE LOAD BALANCER

Load Balancer Name: LabELB
Scheme: internet-facing
Port Configuration:
80 (HTTP) forwarding to 80 (HTTP)

Edit Load Balancer Definition

CONFIGURE HEALTH CHECK

Ping Target: HTTP:80:/
Timeout: 5
Interval: 0.5
Unhealthy Threshold: 2
Healthy Threshold: 3

Edit Health Check

ADD EC2 INSTANCES

EC2 Instances: i-56bd0b38, i-50bd0b3e

Edit EC2 Instance Selection

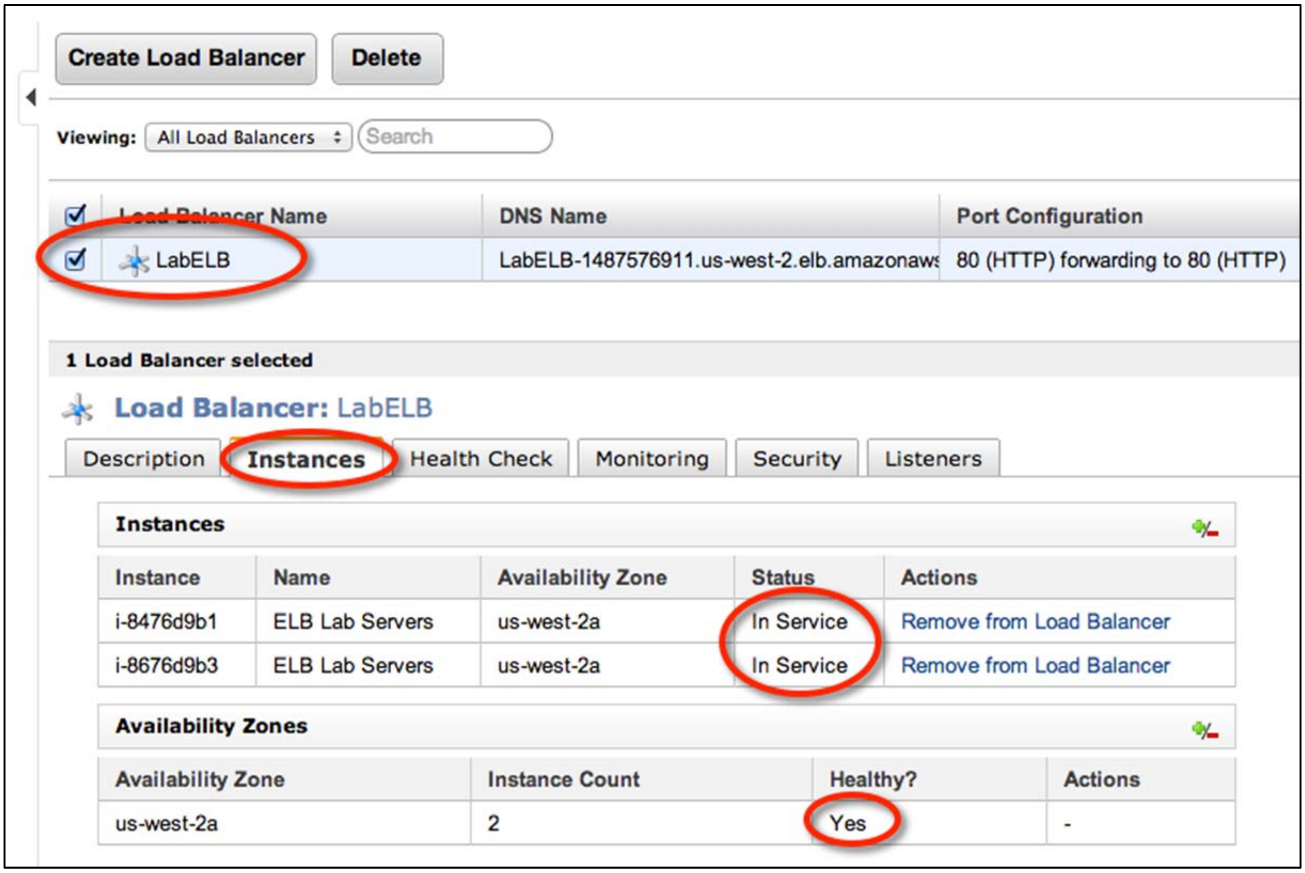
VPC INFORMATION

VPC:
Subnets:

Please review your selections on this page. Clicking "Create" will launch your load balancer. Check the Amazon EC2 product page for load balancer pricing info

AWS is now creating your ELB. It will take a couple of minutes to start up the load balancer, attach your web servers, and pass the health checks.

Click on your load balancer, select the **Instances** tab, and wait until the instance status changes from **Out of Service** to **In Service.** Also note that the overall **Healthy?** column changes from **No** to **Yes**. Your ELB is ready when this happens.



☑	Load Balancer Name	DNS Name	Port Configuration
☑	LabELB	LabELB-1487576911.us-west-2.elb.amazonaws	80 (HTTP) forwarding to 80 (HTTP)

1 Load Balancer selected

Load Balancer: LabELB

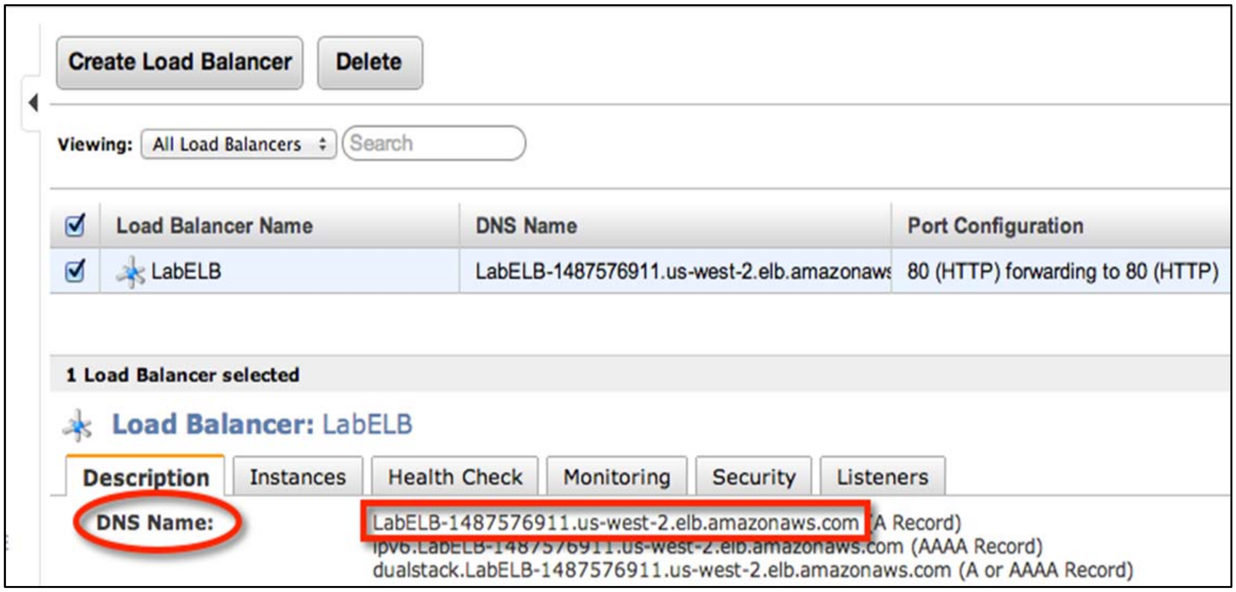
Instances

Instance	Name	Availability Zone	Status	Actions
i-8476d9b1	ELB Lab Servers	us-west-2a	In Service	Remove from Load Balancer
i-8676d9b3	ELB Lab Servers	us-west-2a	In Service	Remove from Load Balancer

Availability Zones

Availability Zone	Instance Count	Healthy?	Actions
us-west-2a	2	Yes	-

8. Once your ELB is healthy, click on the **Description** tab, select and copy the ELB's DNS name (A record).



☑	Load Balancer Name	DNS Name	Port Configuration
☑	LabELB	LabELB-1487576911.us-west-2.elb.amazonaws	80 (HTTP) forwarding to 80 (HTTP)

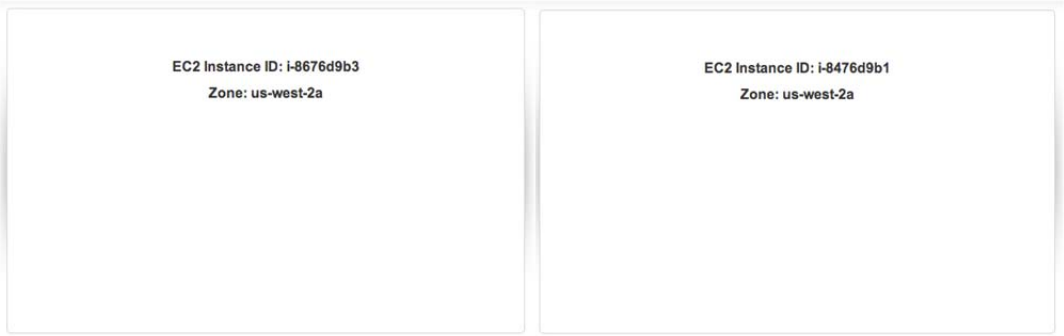
1 Load Balancer selected

Load Balancer: LabELB

DNS Name: LabELB-1487576911.us-west-2.elb.amazonaws.com (A Record)
ipv6.LabELB-1487576911.us-west-2.elb.amazonaws.com (AAAA Record)
dualstack.LabELB-1487576911.us-west-2.elb.amazonaws.com (A or AAAA Record)

ELBs can span availability zones, and they also scale elastically as needed to handle demand. Therefore, you should always access an ELB by DNS hostname, and not by IP address. An ELB may have multiple IP addresses associated with its DNS hostname.

9. Open the ELB URL in another browser tab. Click the browser refresh button a few times, and you should observe responses coming back from different web servers, like what is shown in the screenshots below.



EC2 Instance ID: i-8676d9b3
Zone: us-west-2a

EC2 Instance ID: i-8476d9b1
Zone: us-west-2a

View ELB CloudWatch Metrics

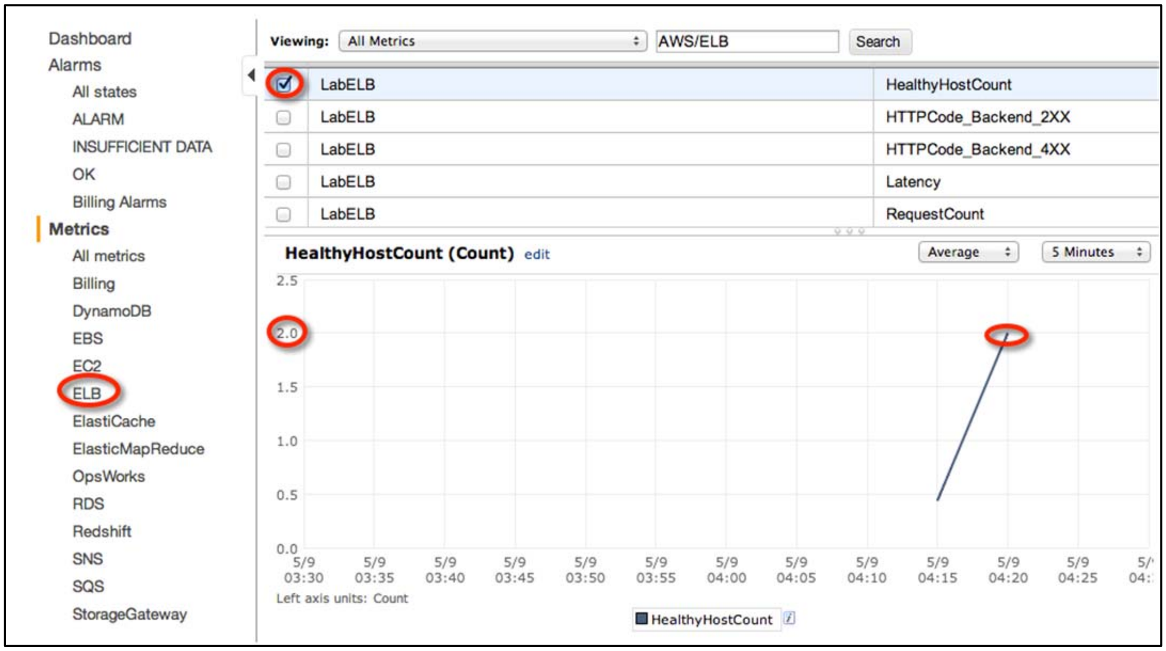
ELB automatically reports load balancer metrics to CloudWatch:

http://docs.amazonwebservices.com/ElasticLoadBalancing/latest/DeveloperGuide/US_MonitoringLoadBalancerWithCW.html

10. You can view these metrics by navigating to CloudWatch in the management console.

CloudWatch

11. Click the **ELB** link on the left, and select the metric you would like to view.



Viewing: All Metrics AWS/ELB

☑	LabELB	HealthyHostCount
□	LabELB	HTTPCode_Backend_2XX
□	LabELB	HTTPCode_Backend_4XX
□	LabELB	Latency
□	LabELB	RequestCount

HealthyHostCount (Count) Average 5 Minutes

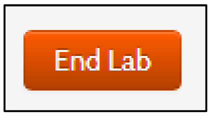
ELB metrics include latency, request count, healthy and unhealthy host counts, and many others. Metrics are reported as they are encountered and can take several minutes to show up in CloudWatch. The following screenshot shows a CloudWatch graph of the **HealthyHostCount metric**, which transitioned from zero healthy hosts to two shortly after the ELB was created for this lab.

Conclusion

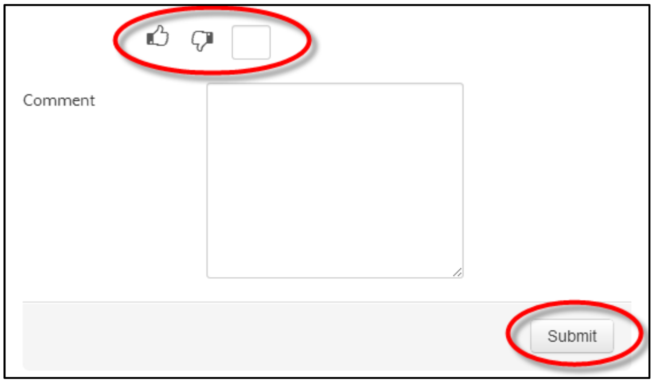
Congratulations! You have now been able to create an Elastic Load Balancer backed by multiple Amazon EC2 servers. In normal operation, we recommend that your Amazon EC2 instances be deployed across multiple availability zones to enable fault tolerance. You have also briefly seen how to monitor ELB metrics in CloudWatch.

End Lab

1. Sign-out of the AWS Management Console.
2. Click the End Lab button in *qwikLAB™*.



3. Give the lab a thumbs-up/down, or enter a comment and click Submit



Any errors in this lab may be reported to aws-course-feedback@amazon.com.