Configure a Load Balancer on EC2 Instances

edureka!

Overview

This lab introduces the concept of Elastic Load Balancing. In this lab you will use Elastic Load Balancing to load balance traffic across multiple EC2 instances in a single Region. You will deploy a simple application on multiple EC2 instances and observe load balancing by viewing the application in your browser.

First, you will launch a pair of instances, bootstrap them to install web servers and content, and then access the instances independently using EC2 DNS records. Next, you will set up Elastic Load Balancing, add your instances to the load balancer, and then access the DNS record again to watch your requests load balance between servers.

Topics covered

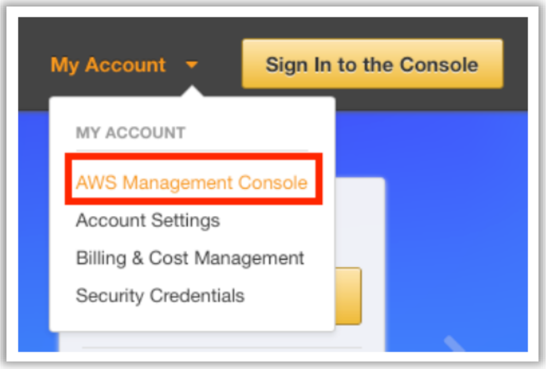
This lab will take you through:

- → Launching a multiple web server farm on EC2.
- → Using bootstrapping techniques to configure Linux instances with Apache, PHP, and a simple PHP application downloaded from Amazon Simple Storage Service (S3).
- → Creating and configuring a load balancer that will sit in front of your EC2 web server instances.

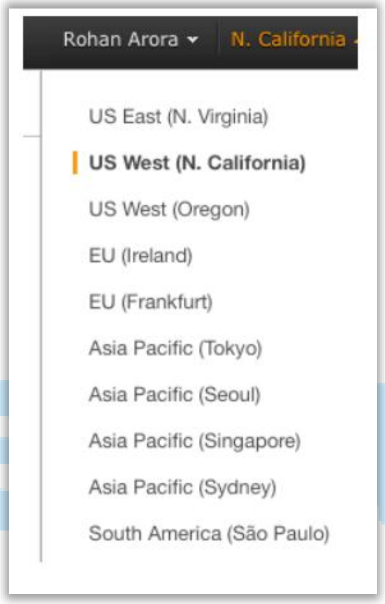
Launch a web server (Instance A) in one of the Availability Zones

In this section, you will launch two Amazon Linux EC2 instances, with an Apache PHP web server and basic application installed on initialization. You will also demonstrate a simple example of bootstrapping instances using Amazon EC2 metadata service.

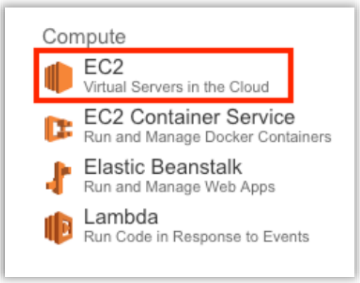
1. Login to AWS Management Console.



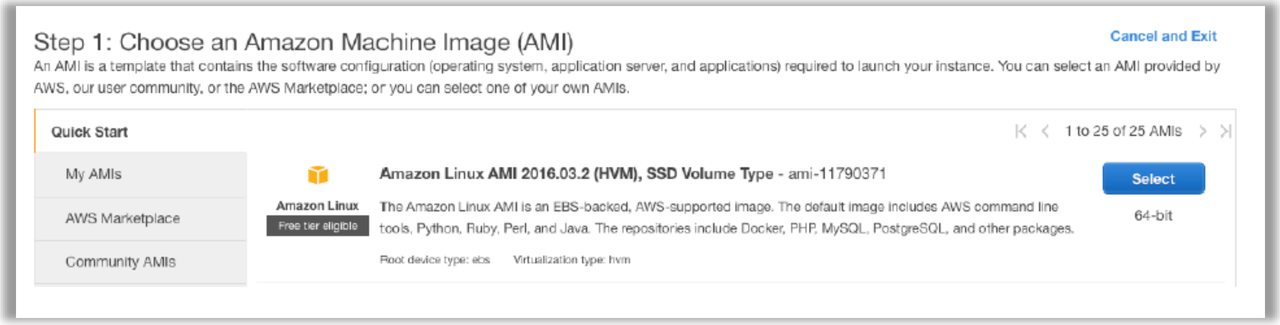
2. Select your preferred Region.



3. Click EC2 under Compute section. This will take you to EC2 dashboard.


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 in Response to Events

4. Click Launch Instance.
5. Because you require a Linux instance, in the row for the basic 64-bit Amazon Linux AMI, which will normally be the first option on the list, click Select.



Step 1: Choose an Amazon Machine Image (AMI)

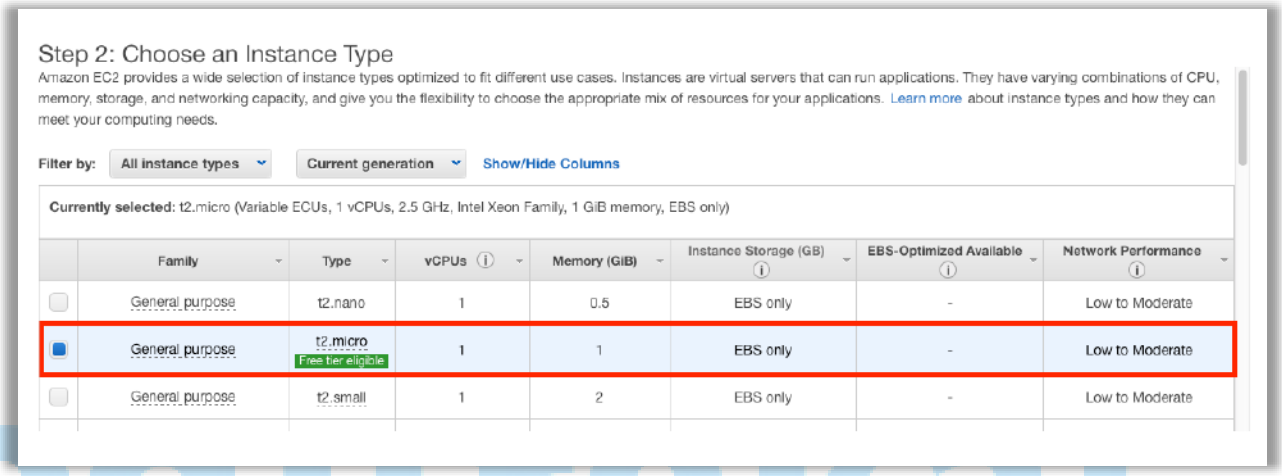
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2016.03.2 (HVM), SSD Volume Type - ami-11790371

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

6. On the Choose an Instance Type page, choose t2.micro, which is free tier eligible.



Step 2: Choose an Instance Type

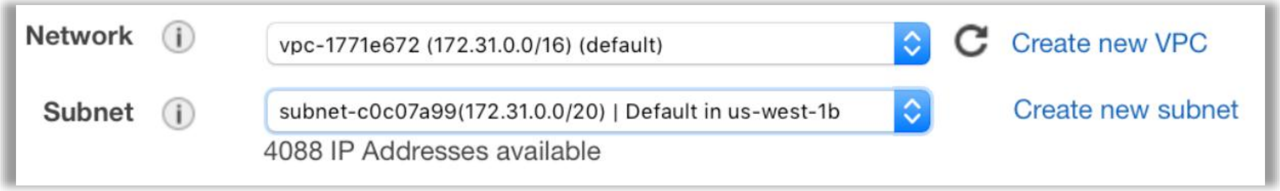
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources 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.nano	1	0.5	EBS only	-	Low to Moderate
■	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

7. Click Next: Configure Instance Details.
8. On the Configure Instance Details page, select VPC and one of the subnets associated with an Availability Zone. Over here, we are choosing us-west-1b as our preferred zone to launch this instance into.



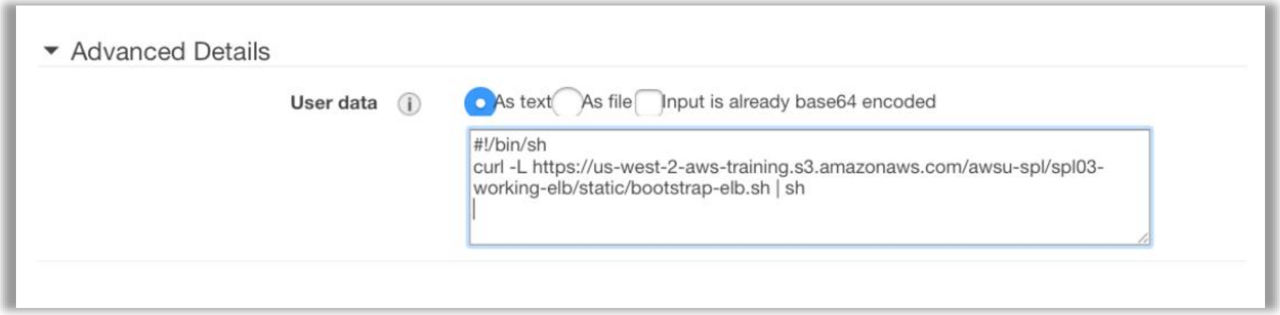
Network: vpc-1771e672 (172.31.0.0/16) (default)

Subnet: subnet-c0c07a99(172.31.0.0/20) | Default in us-west-1b

4088 IP Addresses available

9. Scroll down and expand Advanced Details section.
10. For User Data, select As Text.
11. Copy and paste following script into the User Data box with the As Text option as selected. This will automatically install and start the Apache Web Server and other components when instance is created and launched.

```
#!/bin/sh
curl -L https://us-west-2-aws-training.s3.amazonaws.com/awsu-spl/spl03-working-elb/static/bootstrap-elb.sh | sh
```

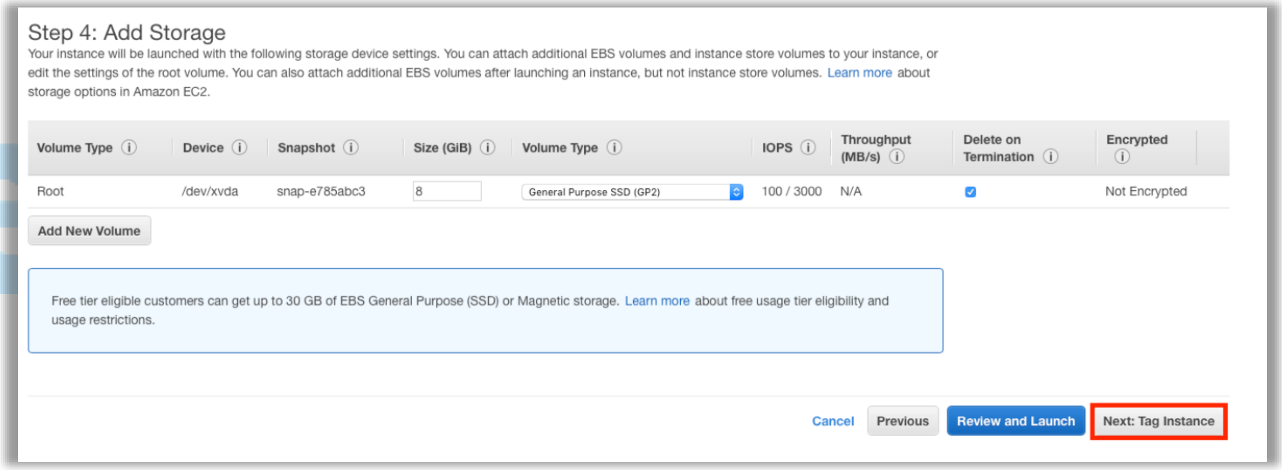


Advanced Details

User data ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/sh
curl -L https://us-west-2-aws-training.s3.amazonaws.com/awsu-spl/spl03-working-elb/static/bootstrap-elb.sh | sh
```

12. Click Next: Add Storage.
13. Click Next: Tag Instance to accept the default storage device configuration.



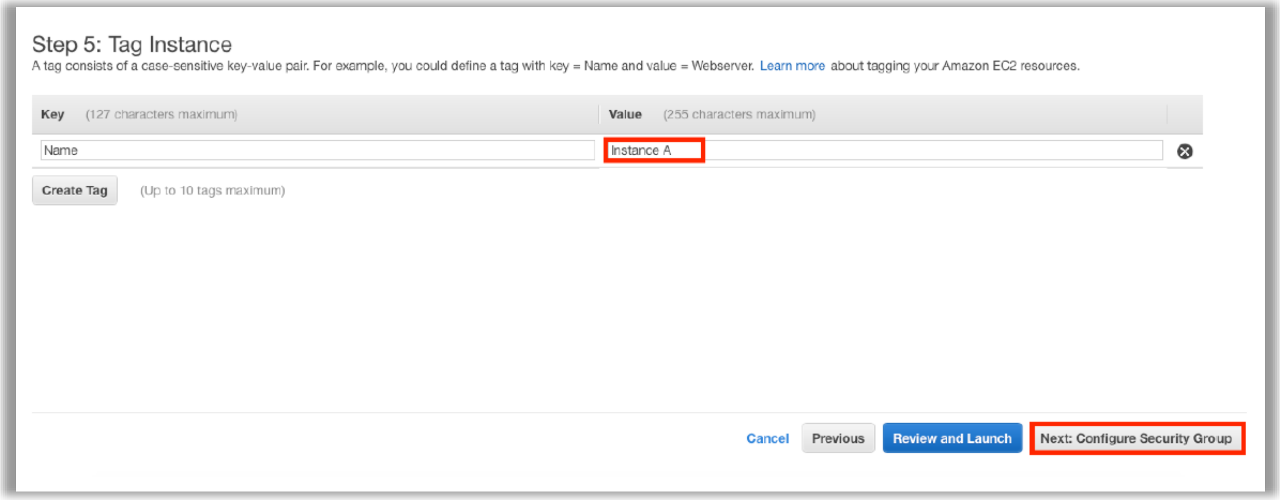
Step 4: Add Storage

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. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-e785abc3	8	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

14. In the Value box, type a name for your instance. Over here, we name it as Instance A.



Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Instance A

Create Tag (Up to 10 tags maximum)

15. Click Next: Configure Security Group.

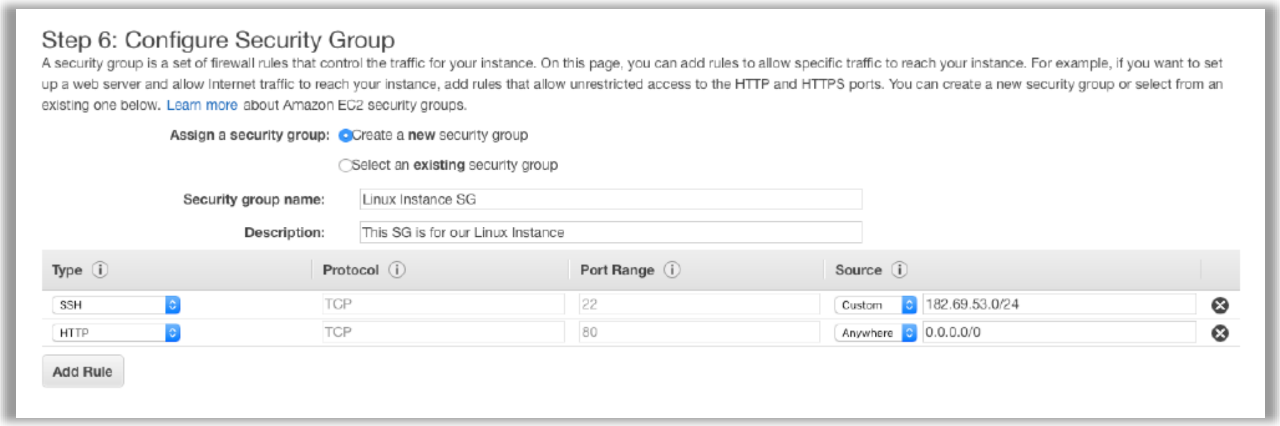
   → Now you will create security group. A security group acts as a firewall that controls the traffic allowed into a group of instances. When you launch an EC2 instance, you can assign it to one or more security groups. For each security group, you add rules that govern the allowed inbound traffic to instances in the group. All other inbound traffic is discarded. You can modify rules for a security group at any time. The new rules are automatically enforced for all existing and future instances in the group.

16. For Assign a security group, click Create a new Security group.
17. In the Security group name box, type a name that you would like to assign to this security group.
18. (Optional) type a description for your security group.

   → By default, AWS creates a rule that allows Secure Shell (SSH) access from any IP address. It is highly recommended that you restrict terminal access to the ranges of IP addresses (e.g., IPs assigned to machines within your company) that have a legitimate business need to administer to your EC2 instance.

19. Click Add Rule to open a new port.
20. In the Type drop-down list, click HTTP.

   → This will add a default handler for HTTP that will allow requests from anywhere on the internet. Since you want this web server to be accessible to the general public, you can leave this rule as is without any further configuration.



Step 6: Configure Security Group

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

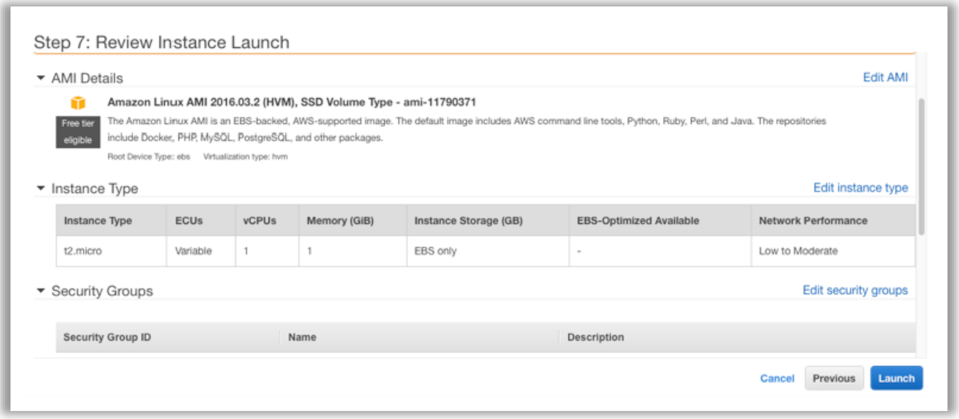
Security group name: Linux Instance SG

Description: This SG is for our Linux Instance

Type	Protocol	Port Range	Source	
SSH	TCP	22	Custom 182.69.53.0/24	⊗
HTTP	TCP	80	Anywhere 0.0.0.0/0	⊗

21. Click Review and Launch.

22. Review your choices, and then click Launch.



Step 7: Review Instance Launch

AMI Details

Amazon Linux AMI 2016.03.2 (HVM), SSD Volume Type - ami-11790371

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

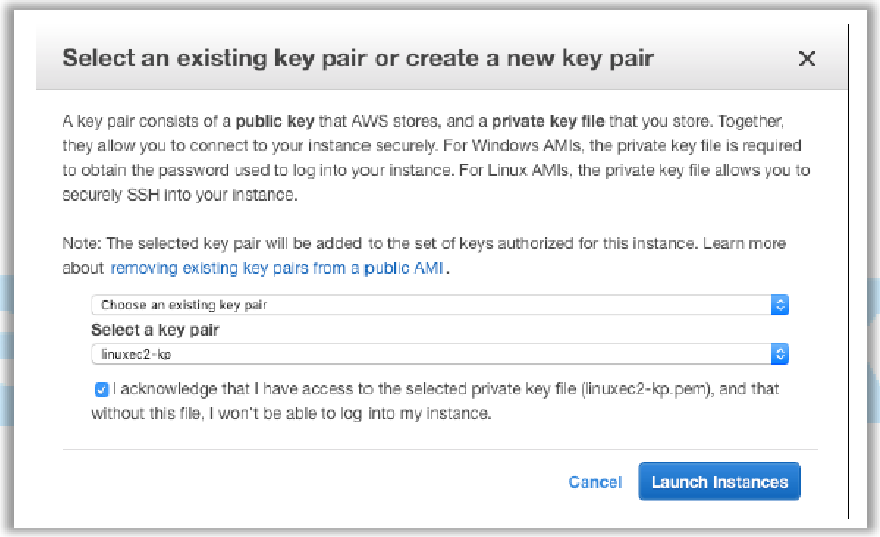
Instance Type

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

Security Groups

Security Group ID	Name	Description

23. Choose an existing key pair and select the acknowledgement check 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.

Choose an existing key pair

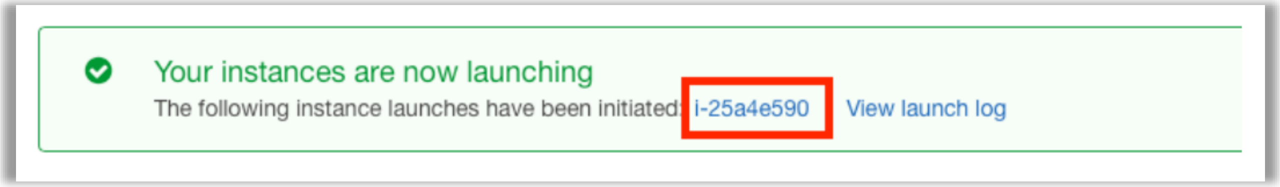
Select a key pair

linuxec2-kp

☑ I acknowledge that I have access to the selected private key file (linuxec2-kp.pem), and that without this file, I won't be able to log into my instance.

24. Click Launch Instances.

25. On the status page, which notifies you that your instances have been initiated, click on instance ID.



Your instances are now launching

The following instance launches have been initiated: i-25a4e590 View launch log

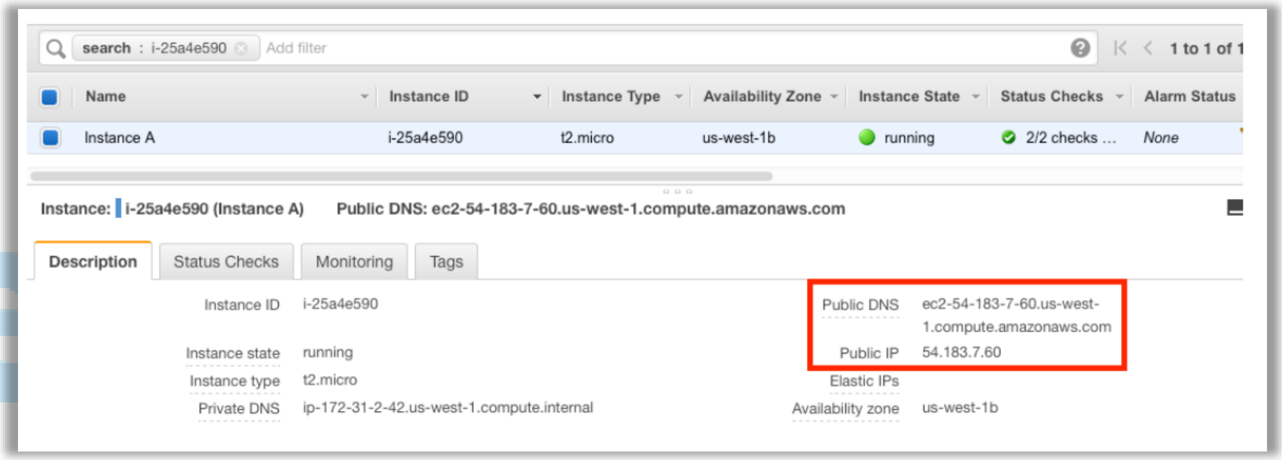
26. Before proceeding to the next step, check that the instance your started has finished its creation cycle. When it creation cycled is finished, you'll notice that the instances transition to a running state with 2/2 checks passed. This indicates you that this instance is now fully available to us.

→ Note: This may take a few minutes. You can refresh the status of your instances by clicking the circular arrow icon in the upper-right hand corner of the page.

Connect to Instance A

Now it's time for you to connect to this instance via its Public IP or Public DNS in order to access it from your web browser.

27. Select your first EC2 instance to display a list of details and a status up for your instance in the bottom pane of the console.
28. Copy either the Public DNS or Public IP value to your clipboard.



search : i-25a4e590 Add filter 1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Instance A	i-25a4e590	t2.micro	us-west-1b	running	2/2 checks ...	None

Instance: i-25a4e590 (Instance A) Public DNS: ec2-54-183-7-60.us-west-1.compute.amazonaws.com

Instance ID i-25a4e590

Instance state running

Instance type t2.micro

Private DNS ip-172-31-2-42.us-west-1.compute.internal

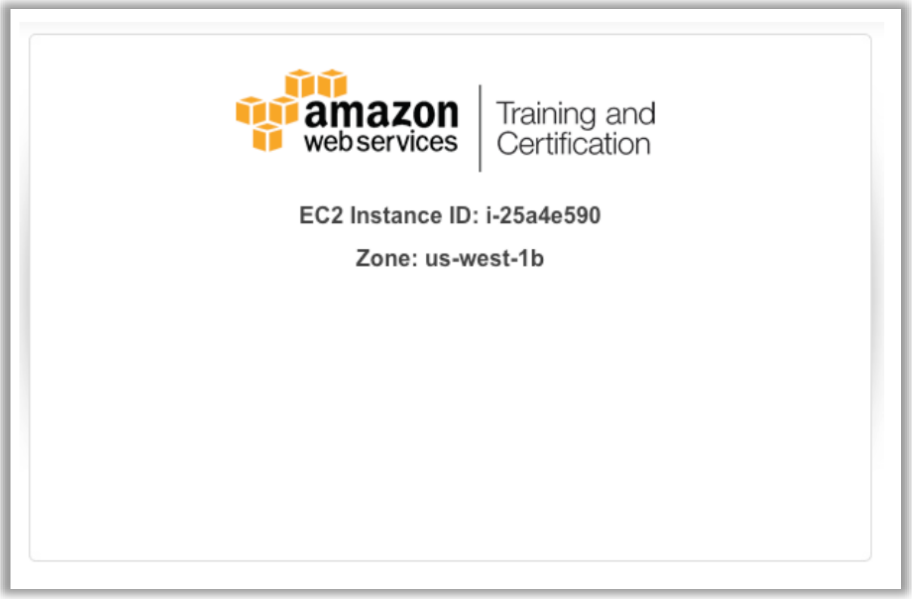
Public DNS ec2-54-183-7-60.us-west-1.compute.amazonaws.com

Public IP 54.183.7.60

Elastic IPs

Availability zone us-west-1b

29. Open a new browser window, paste the Public DNS value in address bar, and press ENTER. Your browser will display a screen like this:



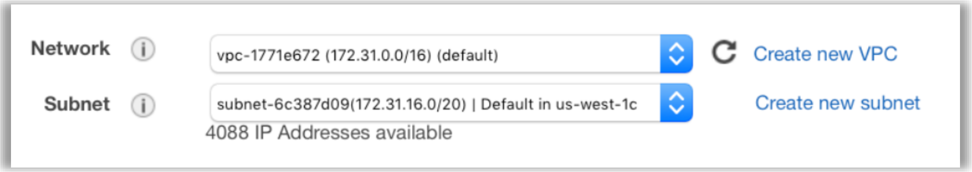
EC2 Instance ID: i-25a4e590

Zone: us-west-1b

→ 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 on each machine and returns the instance ID and the name of Availability Zone in which the instance is running.

Launch a duplicate web server (Instance B) in another Availability Zone

Repeat steps 4 to 29 to launch a duplicate instance in another Availability Zone. Over here, we are selecting us- west-1c to launch our duplicate instance into.

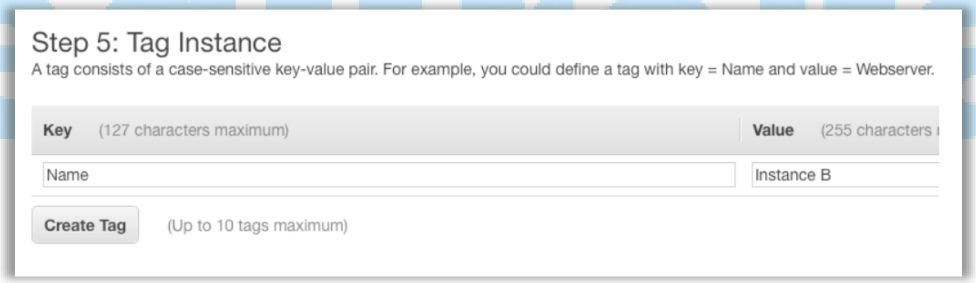


Network vpc-1771e672 (172.31.0.0/16) (default)

Subnet subnet-6c387d09(172.31.16.0/20) | Default in us-west-1c

4088 IP Addresses available

Also, this web server is being tagged as Instance B. This will help us to differentiate between these two instances launched in different Availability Zones.

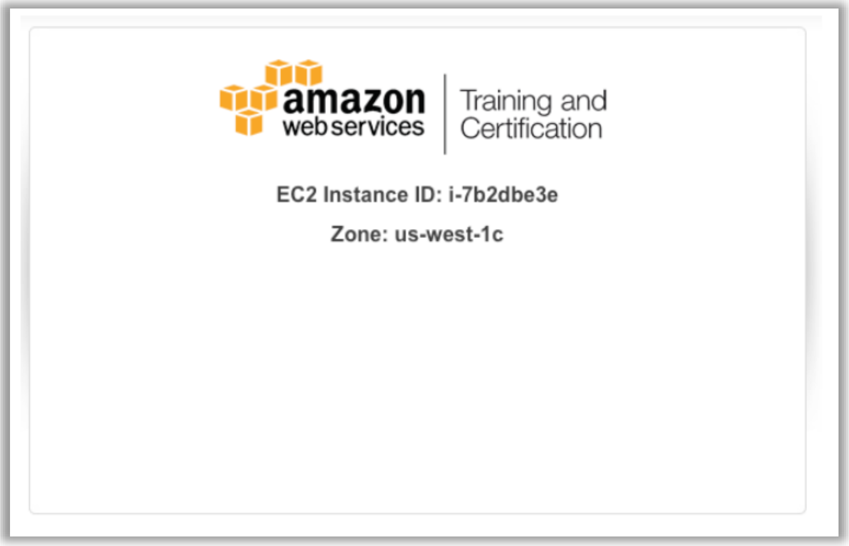


Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

Key (127 characters maximum)	Value (255 characters
Name	Instance B

Consequently, we get following output while accessing this second instance through our web browser:



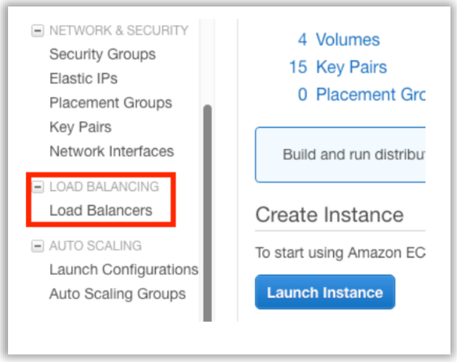
EC2 Instance ID: i-7b2dbe3e

Zone: us-west-1c

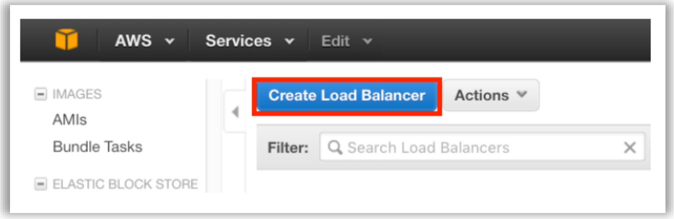
Create Load Balancer

You have now two web servers running in different AZs. Now you need a load balancer in front of these web servers to give your users a single location for accessing both and to balance user requests across them. For this lab, you will be creating a simple HTTP load balancer.

30. Return to the AWS Management Console.
31. In the console's left navigation pane, click Load Balancers. You may need to scroll down to see the link.



32. Click Create Load Balancer.



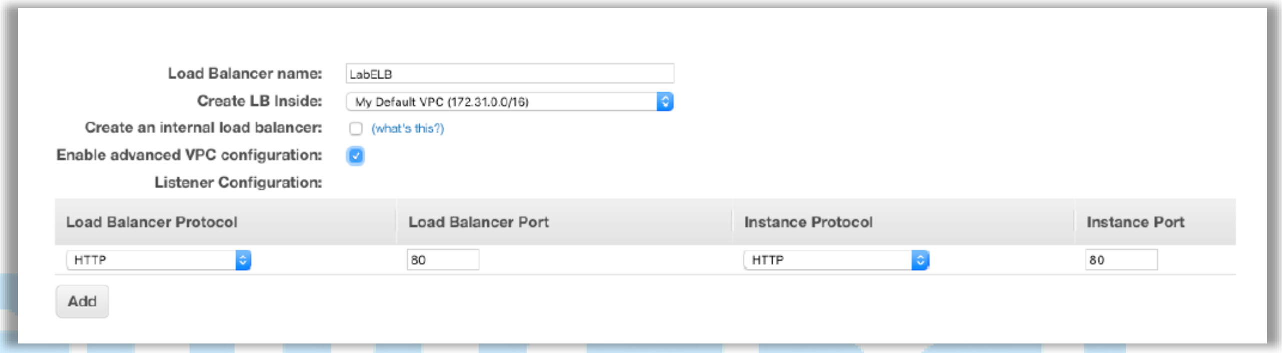
33. In the Load Balancer name box, type a new name like LabELB.

    → The name of your load balancer must be unique within your set of load balancers for the region, can have a maximum of 32 characters, can contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen.

34. For Create LB inside, select the same network that you selected for your instances: EC2-Classic or a specific VPC.

35. [Default VPC] If you selected a default VPC and would like to choose the subnets for your load balancer, select Enable advanced VPC configuration.

36. Leave the default listener configuration.



Load Balancer name: LabELB

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

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

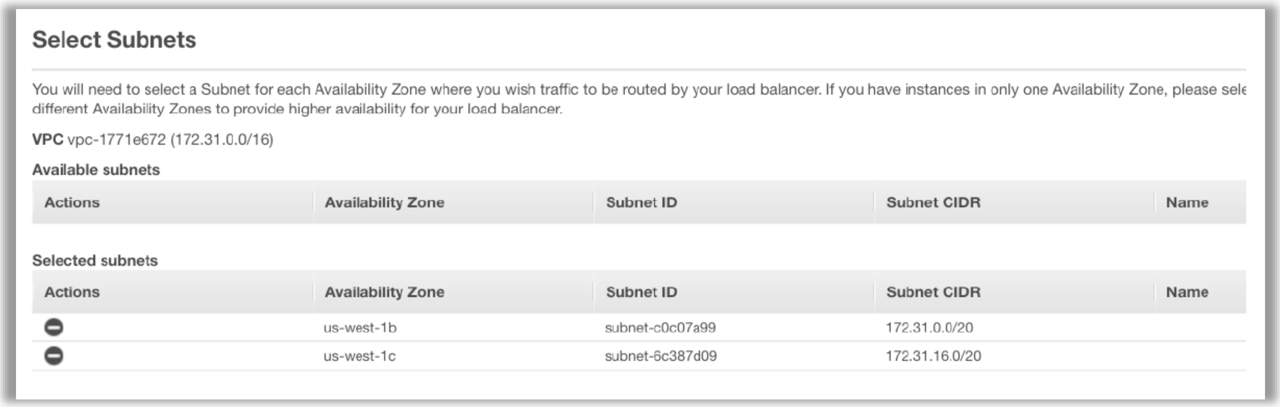
Enable advanced VPC configuration: ☑

Listener Configuration:

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

37. Under Select Subnets, all the subnets in which our web servers have been launched into.

    → The available subnets for the VPC for your load balancer are displayed under Available Subnets. Select public subnets that are in the same Availability Zones as your instances. Click the icon in the Action column for each subnet to attach. These subnets are moved under Selected Subnets. You can select at most one subnet per Availability Zone. If you select a subnet from an Availability Zone where there is already a selected subnet, this subnet replaces the currently selected subnet for the Availability Zone.



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 sele different Availability Zones to provide higher availability for your load balancer.

VPC vpc-1771e672 (172.31.0.0/16)

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name

Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-west-1b	subnet-c0c07a99	172.31.0.0/20	
⊖	us-west-1c	subnet-6c387d09	172.31.16.0/20	

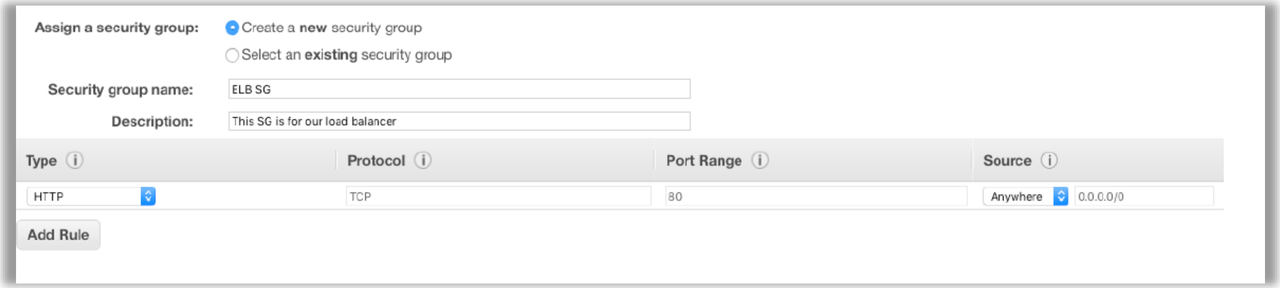
38. Click Next: Assign Security Groups.

Assign Security Groups to Your Load Balancer in a VPC

If you selected a VPC as your network, you must assign your load balancer a security group that allows inbound traffic to the ports that you specified for your load balancer and the health checks for your load balancer.

To assign security group to your load balancer

39. On the Assign Security Groups page, select Create a new security group.
40. Enter a name and description for your security group, or leave the default name and description. This new security group contains a rule that allows traffic to the port that you configured your load balancer to use. Over here, we would be allowing HTTP traffic from anywhere i.e. 0.0.0.0/0.



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

Security group name: ELB SG

Description: This SG is for our load balancer

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

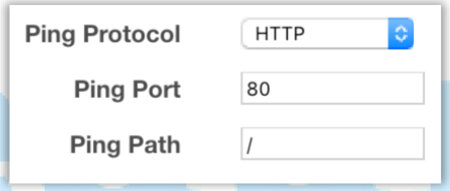
41. Click Next: Configure Security Settings.
42. Skip Step 3: Configure Security Settings by clicking on Next: Configure Health Check.

Configure Health Checks for Your EC2 Instances

Elastic Load Balancing automatically checks the health of the EC2 instances for your load balancer. If Elastic Load Balancing finds an unhealthy instance, it stops sending traffic to the instance and reroutes traffic to healthy instances. In this step, you customize the health checks for your load balancer.

To configure health checks for your instances

43. On the Configure Health Check page, do the following:
    - → Leave Ping Protocol set to its default value, HTTP.
    - → Leave Ping Port set to its default value, 80.
    - → In the Ping Path field, replace the default value with a single forward slash ("/"). Delete the text index.html.
    - → Leave the other fields set to their default values.



Ping Protocol HTTP

Ping Port 80

Ping Path /

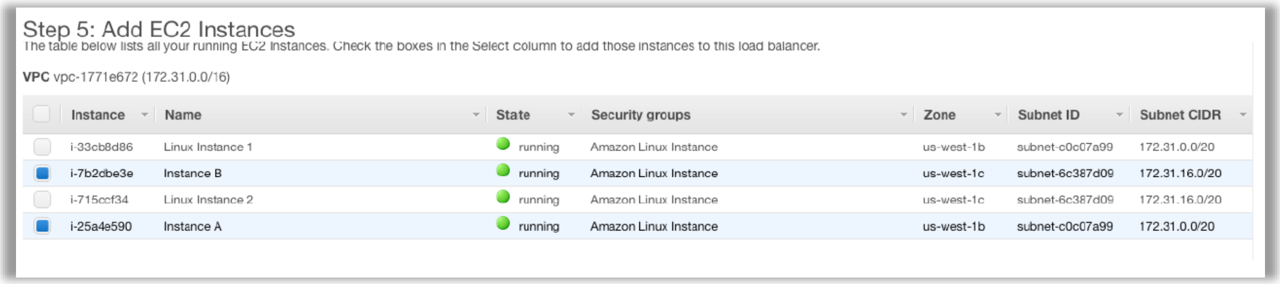
44. Click Next: Add EC2 Instances.
45. Select both of your web server instances to add them to your load balancer, and then click Next: Add Tags.

Register EC2 Instances with Your Load Balancer

Your load balancer distributes traffic between the instances that are registered to it.

To register EC2 instances with your load balancer

46. On the Add EC2 Instances page, select the instances to register with your load balancer.



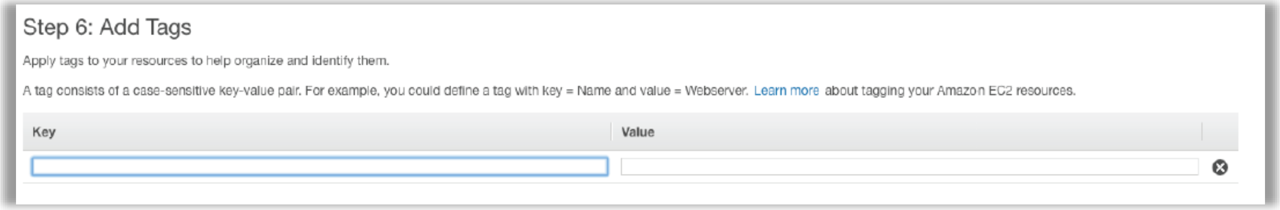
Step 5: Add EC2 Instances

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-1771e672 (172.31.0.0/16)

□	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
□	i-33cb8d86	Linux Instance 1	● running	Amazon Linux Instance	us-west-1b	subnet-c0c07a99	172.31.0.0/20
☑	i-7b2dbe3e	Instance B	● running	Amazon Linux Instance	us-west-1c	subnet-6c387d09	172.31.16.0/20
□	i-715ccf34	Linux Instance 2	● running	Amazon Linux Instance	us-west-1c	subnet-6c387d09	172.31.16.0/20
☑	i-25a4e590	Instance A	● running	Amazon Linux Instance	us-west-1b	subnet-c0c07a99	172.31.0.0/20

47. Click Next: Add Tags.
48. Here is where you could add tags and data to your tags. For this lab, tags are not necessary. Leave these fields empty and click Review and Create.



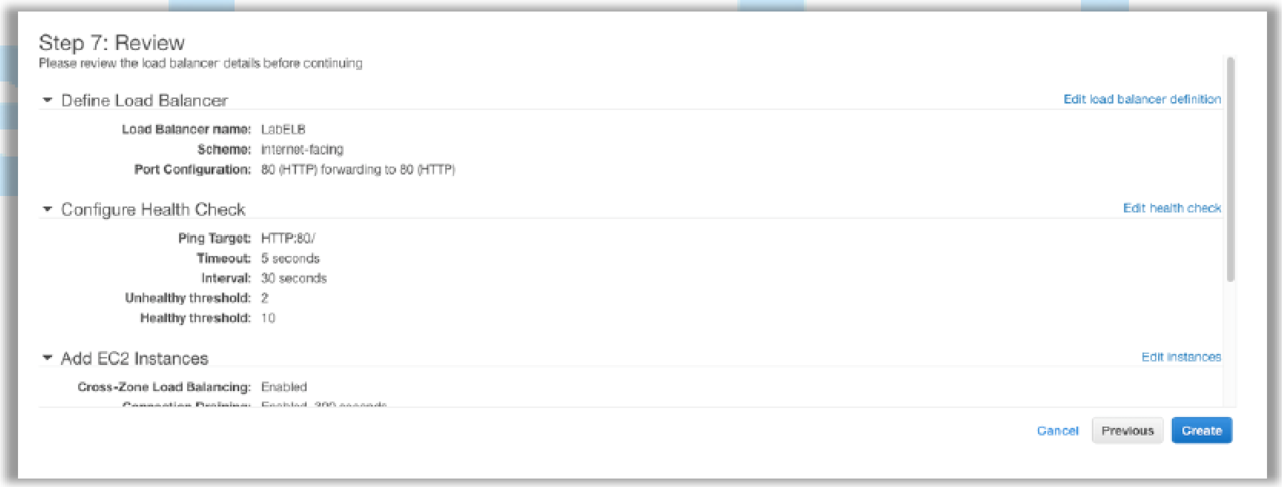
Step 6: Add Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	Value

49. Review your settings, and then click Create.



Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer

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

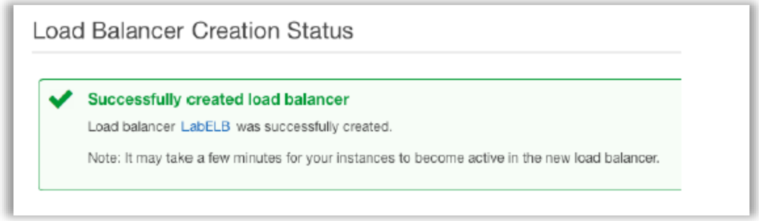
Configure Health Check

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

50. AWS is now creating your load balancer. It will take a couple of minutes to start up the load balancer, attach your web servers, and pass the health checks. Click on LabELB to monitor its progress.



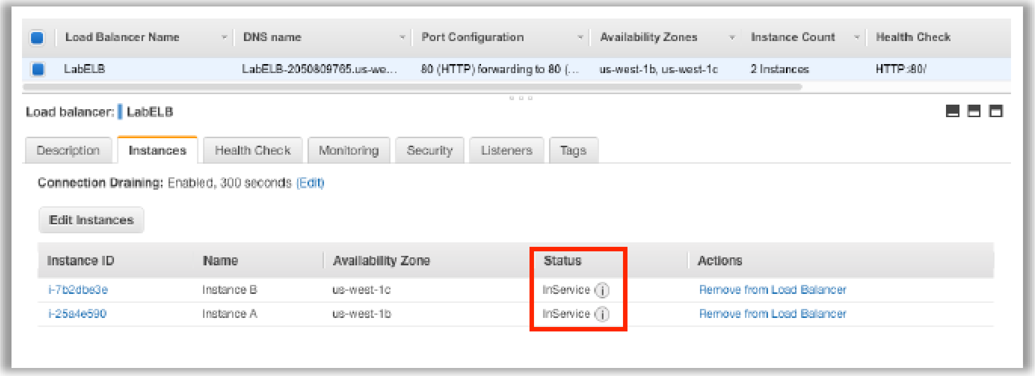
Load Balancer Creation Status

✓ **Successfully created load balancer**

Load balancer LabELB was successfully created.

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

51. Select the load balancer you just created, click the Instances tab, and wait for the status of both instances to change to InService. To refresh the status, click the circular arrow icon in the upper right.



Load Balancer Name	DNS name	Port Configuration	Availability Zones	Instance Count	Health Check
LabELB	LabELB-2050809765.us-we...	80 (HTTP) forwarding to 80 (...	us-west-1b, us-west-1c	2 Instances	HTTP:80/

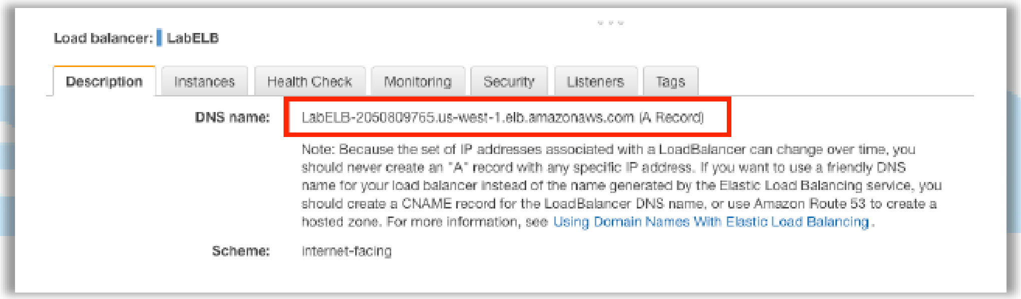
Load balancer: LabELB

Connection Draining: Enabled, 300 seconds (Edit)

Instance ID	Name	Availability Zone	Status	Actions
i-7b2dbe3e	Instance B	us-west-1c	InService	Remove from Load Balancer
i-25a4e590	Instance A	us-west-1b	InService	Remove from Load Balancer

52. When the status of both instances is InService, click the Description tab.

53. Copy the DNS name value to your clipboard. It will look like LabELB-2050809765.us-west- 1.elb.amazonaws.com. Do not copy the “(A Record)” text.



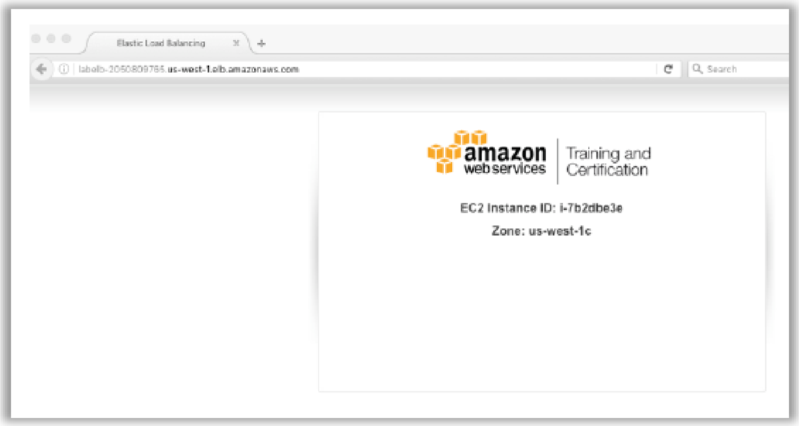
Load balancer: LabELB

DNS name: LabELB-2050809765.us-west-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see Using Domain Names With Elastic Load Balancing.

Scheme: internet-facing

54. Open a new browser window, paste the DNS name value in the address bar, and press ENTER.

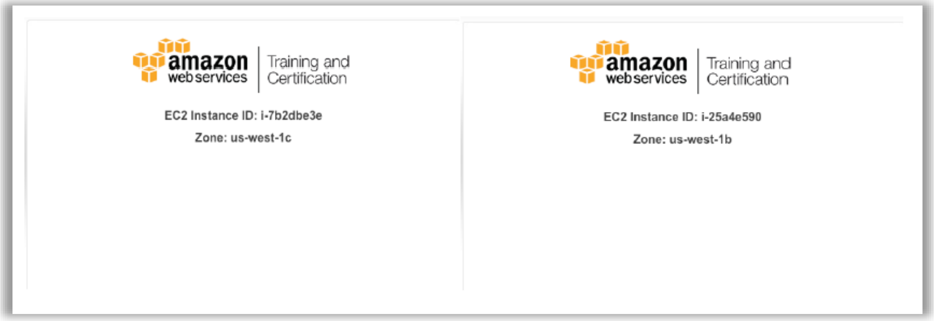


Elastic Load Balancing
labelb-2050809765.us-west-1.elb.amazonaws.com
Search
amazon web services
Training and Certification

EC2 Instance ID: i-7b2dbe3e

Zone: us-west-1c

55. Refresh your browser a few times, and you should see the EC2 Instance IDS changing. This means that the repeated responses are coming back through your different web servers.



EC2 Instance ID: i-7b2dbe3e
Zone: us-west-1c

EC2 Instance ID: i-25a4e590
Zone: us-west-1b

amazon webservices Training and Certification
amazon webservices Training and Certification

Conclusion

Congratulations! You have now successfully:

→ Launched a multiple server web farm on Amazon EC2

→ Used bootstrapping techniques to configure Linux instances with Apache, PHP, and a simple PHP application downloaded from Amazon S3.

→ Created and configured a load balancer that sits in front of your EC2 web server instances.