**[Overview](https://qwiklab.com/focuses/368?parent=catalog" \l "step1)**[**Start Lab**](https://qwiklab.com/focuses/368?parent=catalog#step2)

[Task 1: Launch Your Amazon EC2 Instance](https://qwiklab.com/focuses/368?parent=catalog#step3)

[Task 2: Monitor Your Instance](https://qwiklab.com/focuses/368?parent=catalog#step4)

[Task 3: Update Your Security Group and Access the Web Server](https://qwiklab.com/focuses/368?parent=catalog#step5)

[Task 4: Resize Your Instance: Instance Type and EBS Volume](https://qwiklab.com/focuses/368?parent=catalog#step6)

[Task 5: Explore EC2 Limits](https://qwiklab.com/focuses/368?parent=catalog#step7)

[Task 6: Test Termination Protection](https://qwiklab.com/focuses/368?parent=catalog#step8)

**Introduction to Amazon EC2**

[EC2](https://qwiklab.com/tags/EC2) [Elastic Web-Scale Computing](https://qwiklab.com/tags/Elastic%20Web-Scale%20Computing) [instances](https://qwiklab.com/tags/instances) [instance](https://qwiklab.com/tags/instance) [VMs](https://qwiklab.com/tags/VMs) [Compute](https://qwiklab.com/tags/Compute) [virtual servers](https://qwiklab.com/tags/virtual%20servers) [resizable compute capacity](https://qwiklab.com/tags/resizable%20compute%20capacity) [web-scale](https://qwiklab.com/tags/web-scale) [server instances](https://qwiklab.com/tags/server%20instances) [scale](https://qwiklab.com/tags/scale) [computing](https://qwiklab.com/tags/computing) [cloud](https://qwiklab.com/tags/cloud) [elastic](https://qwiklab.com/tags/elastic) [Linux](https://qwiklab.com/tags/Linux) [Windows](https://qwiklab.com/tags/Windows) [virtual machine](https://qwiklab.com/tags/virtual%20machine)

**Overview**

This lab provides you with a basic overview of **launching, resizing, managing, and monitoring** an Amazon EC2 instance.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud.

Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

**Topics covered**

Launch a web server with termination protection enabled

Monitor Your EC2 instance

Modify the security group that your web server is using to allow HTTP access

Resize your Amazon EC2 instance to scale

Explore EC2 limits

Test termination protection

Terminate your EC2 instance

**Task 1:** Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with termination protection. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

In the AWS Management Console on the Services menu, click EC2.

Click Launch Instance.

Choose an AMI

An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

A template for the root volume for the instance (for example, an operating system or an application server with applications)

Launch permissions that control which AWS accounts can use the AMI to launch instances

A block device mapping that specifies the volumes to attach to the instance when it is launched

The Quick Start list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

Click Select next to Amazon Linux AMI (at the top of the list).

Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

You will use a t2.micro instance which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory.

Click Next: Configure Instance Details.

Configure Instance Details

This page is used to configure the instance to suit your requirements. This includes networking and monitoring settings.

The Network indicates which Virtual Private Cloud (VPC) you wish to launch the instance into. You can have multiple networks, such as different ones for development, testing and production.

For Network, select Lab VPC.

The Lab VPC was created using a CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

For Enable termination protection, select Protect against accidental termination.

When an Amazon EC2 instance is no longer required, it can be terminated, which means that the instance is stopped and its resources are released. A terminated instance cannot be started again. If you want to prevent the instance from being accidentally terminated, you can enable termination protection for the instance, which prevents it from being terminated.

Scroll down, then expand Advanced Details.

A field for User data will appear.

When you launch an instance, you can pass user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Your instance is running Amazon Linux, so you will provide a shell script that will run when the instance starts.

Copy the following commands and paste them into the User data field:

#!/bin/bash

yum -y update

yum -y install httpd

chkconfig httpd on

service httpd start

echo "<html><h1>Hello From Your Web Server!</h1></html>" > /var/www/html/index.html

The script will:

Install system updates

Install an Apache web server (httpd)

Configure the web server to automatically start on boot

Activate the Web server

Create a simple web page

Click Next: Add Storage.

Add Storage

Amazon EC2 stores data on a network-attached virtual disk called Elastic Block Store.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

Click Next: Add Tags.

Add Tags

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define.

Click Add Tag, then configure:

Key: 

Value: 

Click Next: Configure Security Group.

Configure Security Group

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

On Step 6. Configure Security Group, configure the following:

Security group name: 

Description: 

In this lab, you will not log into your instance using SSH. Removing SSH access will improve the security of the instance.

Delete the existing SSH rule.

Click Review and Launch.

Review

The Review page displays the configuration for the instance you are about to launch.

Click Launch.

A Select an existing key pair or create a new key pair window will appear.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

In this lab you will not log into your instance, so you do not require a key pair.

Click the Choose an existing key pair drop-down and select Proceed without a key pair.

Select I acknowledge that ....

Click Launch Instances.

Your instance will now be launched.

Click View Instances.

The instance will appear in a pending state, which means it is being launched. It will then change to running, which indicates that the instance has started booting. There will be a short time before you can access the instance.

The instance receives a public DNS name that you can use to contact the instance from the Internet.

Your Web Server should be selected. The Description tab displays detailed information about your instance.

To view more information in the Description tab, drag the window divider upwards.

Review the information displayed in the Description tab. It includes information about the instance type, security settings and network settings.

Wait for your instance to display the following:

Instance State: running

Status Checks: 2/2 checks passed

Congratulations! You have successfully launched your first Amazon EC2 instance.

Task 2: Monitor Your Instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions.

Click the Status Checks tab.

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

Notice that both the System reachability and Instance reachability checks have passed.

Click the Monitoring tab.

This tab displays CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.

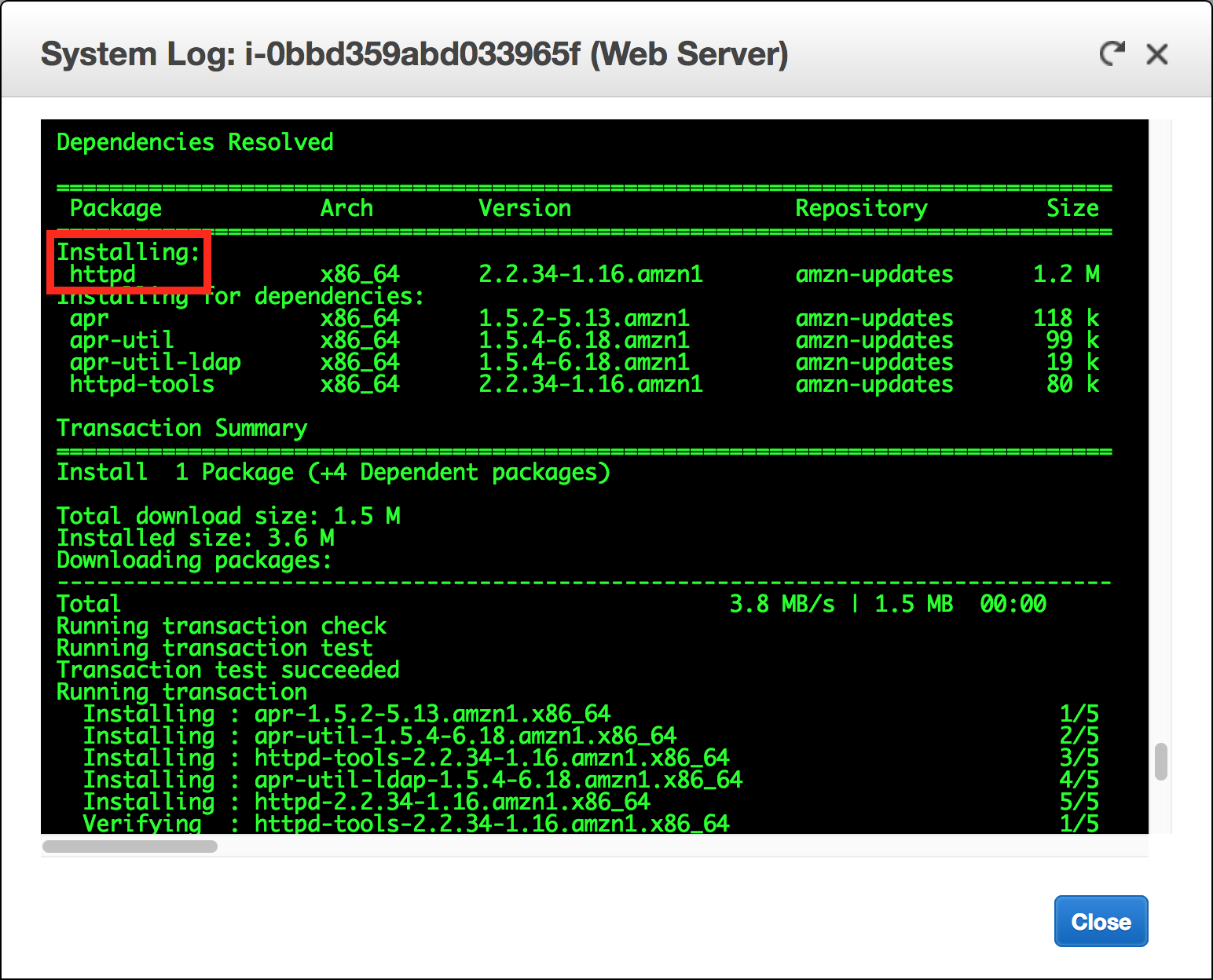
You can click on a graph to see an expanded view.

Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can enable detailed (one-minute) monitoring.

In the Actions menu, select Instance Settings Get System Log.

The System Log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

Scroll through the output and note that the HTTP package was installed from the user data that you added when you created the instance.



Click Close.

In the Actions menu, select Instance Settings Get Instance Screenshot.

This shows you what your Amazon EC2 instance console would look like if a screen were attached to it.



If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

Click Close.

Congratulations! You have explored several ways to monitor your instance.

Task 3: Update Your Security Group and Access the Web Server

When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you will access content from the web server.

Click the Description tab.

Copy the IPv4 Public IP of your instance to your clipboard.

Open a new tab in your web browser, paste the IP address you just copied, then press Enter.

Question: Are you able to access your web server? Why not?

You are not currently able to access your web server because the security group is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of using a security group as a firewall to restrict the network traffic that is allowed in and out of an instance.

To correct this, you will now update the security group to permit web traffic on port 80.

Keep the browser tab open, but return to the EC2 Management Console tab.

In the left navigation pane, click Security Groups.

Select Web Server security group.

Click the Inbound tab.

The security group currently has no rules.

Click Edit, then configure:

Type: HTTP

Source: Anywhere

Click Save

Return to the web server tab that you previously opened and refresh the page.

You should see the message Hello From Your Web Server!

Congratulations! You have successfully modified your security group to permit HTTP traffic into your Amazon EC2 Instance.

Task 4: Resize Your Instance: Instance Type and EBS Volume

As your needs change, you might find that your instance is over-utilized (too small) or under-utilized (too large). If so, you can change the instance type. For example, if a t2.micro instance is too small for its workload, you can change it to an m5.medium instance. Similarly, you can change the size of a disk.

Stop Your Instance

Before you can resize an instance, you must stop it.

When you stop an instance, it is shut down. There is no charge for a stopped EC2 instance, but the storage charge for attached Amazon EBS volumes remains.

On the EC2 Management Console, in the left navigation pane, click Instances.

Web Server should already be selected.

In the Actions menu, select Instance State Stop.

Click Yes, Stop.

Your instance will perform a normal shutdown and then will stop running.

Wait for the Instance State to display: stopped

Change The Instance Type

In the Actions menu, select Instance Settings Change Instance Type.

For Instance Type, select t2.small.

Click Apply.

When the instance is started again it will be a t2.small, which has twice as much memory as a t2.micro instance.

Resize the EBS Volume

In the left navigation menu, click Volumes.

In the Actions menu, select Modify Volume.

The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.

Change the size to: 

Click Modify.

Click Yes to confirm and increase the size of the volume.

Click Close.

Start the Resized Instance

You will now start the instance again, which will now have more memory and more disk space.

In left navigation pane, click Instances.

In the Actions menu, select Instance State Start.

Click Yes, Start.

Congratulations! You have successfully resized your Amazon EC2 Instance. In this task you changed your instance type from t2.micro to t2.small. You also modified your root disk volume from 8 GiB to 10 GiB.

Task 5: Explore EC2 Limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-region basis.

In the left navigation pane, click Limits.

Note that there is a limit on the number of instances that you can launch in this region. When launching an instance, the request must not cause your usage to exceed the current instance limit in that region.

You can request an increase for many of these limits.

Task 6: Test Termination Protection

You can delete your instance when you no longer need it. This is referred to as terminating your instance. You cannot connect to or restart an instance after it has been terminated.

In this task, you will learn how to use termination protection.

In left navigation pane, click Instances.

In the Actions menu, select Instance State Terminate.

Note that there is a message that says: These instances have Termination Protection and will not be terminated. Use the Change Termination Protection option from the Instances screen Actions menu to allow termination of these instances.

Also, the Yes, Terminate button is dimmed and cannot be clicked.

This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you will need to disable the termination protection.

Click Cancel.

In the Actions menu, select Instance Settings Change Termination Protection.

Click Yes, Disable.

You can now terminate the instance.

In the Actions menu, select Instance State Terminate.

Click Yes, Terminate.

Congratulations! You have successfully tested termination protection and terminated your instance.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

Return to the AWS Management Console.

On the navigation bar, click <yourusername>@<AccountNumber>, and then click Sign out.

Click END LAB.

Click OK.

(Optional) Select the applicable number of stars, type a comment, and then click SUBMIT.

Note: The number of stars indicates the following:

1 star = Very dissatisfied

2 stars = Dissatisfied

3 stars = Neutral

4 stars = Satisfied

5 stars = Very satisfied

You may close the dialog if you don't want to provide feedback.

For feedback, suggestions, or corrections, please email us at [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com).

Additional Resources

[Launch Your Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/LaunchingAndUsingInstances.html)

[Amazon EC2 Instance Types](https://aws.amazon.com/ec2/instance-types)

[Amazon Machine Images (AMI)](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html)

[Amazon EC2 - User Data and Shell Scripts](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html)

[Amazon EC2 Root Device Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/RootDeviceStorage.html)

[Tagging Your Amazon EC2 Resources](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html)

[Security Groups](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html)

[Amazon EC2 Key Pairs](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)

[Status Checks for Your Instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html?icmpid=docs_ec2_console)

[Getting Console Output and Rebooting Instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-console.html)

[Amazon EC2 Metrics and Dimensions](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ec2-metricscollected.html)

[Resizing Your Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-resize.html)

[Stop and Start Your Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Stop_Start.html)

[Amazon EC2 Service Limits](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html)

[Terminate Your Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html)

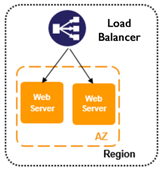
[Termination Protection for an Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html)

**Lab 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 Amazon Elastic Compute Cloud (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.

First, you will launch a pair of instances, bootstrap them to install web servers and content, and then access the instances independently using Amazon 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. Finally, you will view Elastic Load Balancing metrics in Amazon CloudWatch.

The following diagram provides a high-level overview of the architecture you will implement in this exercise.



Topics covered

This lab will take you through:

Launching 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 Simple Storage Service (S3).

Creating and configuring a load balancer that will sit in front of your Amazon EC2 web server instances.

Viewing Amazon CloudWatch metrics for the load balancer.

Technical knowledge prerequisites

To successfully complete this lab, you should be familiar with the AWS Management Console.

**Elastic Load Balancing**

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater levels of fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to distribute application traffic.

Achieve higher levels of fault tolerance for your applications by using Elastic Load Balancing to automatically route traffic across multiple instances and multiple Availability Zones. Elastic Load Balancing ensures that only healthy Amazon EC2 instances receive traffic by detecting unhealthy instances and rerouting traffic across the remaining healthy instances. If all of your Amazon EC2 instances in one Availability Zone are unhealthy, and you have set up Amazon EC2 instances in multiple Availability Zones, Elastic Load Balancing will route traffic to your healthy Amazon EC2 instances in those other zones.

Elastic Load Balancing automatically scales its request-handling capacity to meet the demands of application traffic. Additionally, Elastic Load Balancing offers integration with Auto Scaling to ensure that you have back-end capacity to meet varying levels of traffic levels without requiring manual intervention.

Elastic Load Balancing works with Amazon Virtual Private Cloud (VPC) to provide robust networking and security features. You can create an internal (non-Internet facing) load balancer to route traffic using private IP addresses within your virtual network. You can implement a multi-tiered architecture using internal and Internet-facing load balancers to route traffic between application tiers. With this multi-tier architecture, your application infrastructure can use private IP addresses and security groups, allowing you to expose only the Internet-facing tier with public IP addresses.

Elastic Load Balancing provides integrated certificate management and SSL decryption, allowing you to centrally manage the SSL settings of the load balancer and offload CPU-intensive work from your instances.

This lab guide explains basic concepts of Elastic Load Balancing in a step-by-step fashion. However, it can only give a brief overview of Elastic Load Balancing concepts. For further information, see <http://aws.amazon.com/elasticloadbalancing/>.

**Task 1: Launch Web Servers**

In this task, 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 the Amazon EC2 metadata service.

Amazon Machine Images (AMIs) and instances

Amazon EC2 provides templates known as Amazon Machine Images (AMIs) that contain a software configuration (for example, an operating system, an application server, and applications). You use these templates to launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

You can launch different types of instances from a single AMI. An instance type essentially determines the hardware capabilities of the virtual host computer for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. You can launch multiple instances from an AMI.

Your instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the AMI.

When you create an instance, you will be asked to select an instance type. The instance type you choose determines how much throughput and processing cycles are available to your instance.

On the Services menu, click EC2.

Click Launch Instance.

In the row for the Amazon Linux AMI, which will normally be the first option on the list, click Select.

The t2.micro instance type, which is the lowest-cost option, should be automatically selected. Click Next: Configure Instance Details.

In the Number of instances box, type 2.

Ensure Auto-assign Public IP is set to Enable.

Expand the Advanced Details section. In this section, you will use the User data field to bootstrap your 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.

Copy the following initialization script:

#!/bin/sh

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

Copy the above script and paste it into the User data box with the As text option selected. This will automatically install and start the Apache Web server and other components when the instance is created and launched. Tip If you type this text instead of copying it, press SHIFT+ENTER to create new lines in the text box.

Click Next: Add Storage.

Click Next: Add Tags to accept the default storage device configuration.

Click Add Tag.

For Key, enter 

For Value, 

This name, more correctly known as a tag, will appear in the console when the instance launches. It makes it easy to keep track of running machines in a complex environment. Use a name that you can easily recognize and remember.

Click Next: Configure Security Group.

Now you will create a new security group. A security group acts as a firewall that controls the traffic allowed into a group of instances. When you launch an Amazon 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.

Leave Create a new security group selected.

In the Security group name box, type the same name you used for your instance name (i.e, MyLBInstances).

Add a description for your security group. You may input any name and description that are meaningful to you.

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 your Amazon EC2 instance. Because the lab image you are using will be recycled within two hours, you can bypass this step for the lab.

Click Add Rule to open a new port.

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.

Click Review and Launch.

Review your choices, and then click Launch.

Note You may see a warning on this screen that “Your security group … is open to the world.” This is a result of not restricting SSH access to your machine, as described earlier. For the purposes of this lab only, you may ignore this warning.

In the key pair dialog box, select the acknowledgement check box, and then click Launch Instances.

A status page notifies you that your instances are launching.

Click View Instances.

Before proceeding to the next step, check that the instances you started have finished their creation cycle. When they have finished their creation cycle, you’ll notice that the instances transition to a running state with 2/2 checks passed. This indicates that you instance is now fully available for 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 corner of the page.

Task 2: Connect to Each Web Server

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

All Amazon EC2 instances are assigned two IP addresses at launch: a private IP address (RFC 1918) and a public IP address that are directly mapped to each other through Network Address Translation (NAT). Private IP addresses are only reachable from within the Amazon EC2 network. Public addresses are reachable from the Internet.

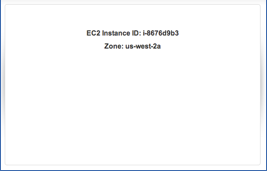
Amazon EC2 also provides an internal DNS name and a public DNS name that map to the private and public IP addresses, respectively. The internal DNS name can only be resolved within Amazon EC2. The public DNS name resolves to the public IP address outside the Amazon EC2 network and to the private IP address within the Amazon EC2 network.

Retrieve your host’s public DNS address

Select your first Amazon EC2 instance to display a list of details and a status update for your instance in the bottom pane of the console.

Copy the Public DNS (IPv4) value to your Clipboard. It will look something like ec2-54-84-236-205.compute-1.amazonaws.com.

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



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 the Availability Zone in which the instance is running.

Repeat the previous two steps for your second instance.

Notice that each machine displays a different instance ID. This will help you identify which instance is processing your request when you put a load balancer in front of them.

If you see an error instead of the instance ID and Availability Zone when you access the instances from the browser, try again after 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 when you launched your instances and that the security group has port 80 open.

Task 3: Create a Load Balancer

You now have two web servers. Now you need a load balancer in front of these 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.

Return to the AWS Management Console.

In the Console’s navigation pane, click Load Balancers. You may need to scroll down to see the link.

Click Create Load Balancer.

In Select load balancer type page, for Classic Load Balancer, click Create.

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

Leave all other default settings and click Next: Assign Security Groups.

For Assign a security group, make sure that Select an existing security group is selected.

Select the security group you created when you created your web server instances (i.e., MyLBInstances), and make sure that no other security groups are selected.

Click Next: Configure Security Settings.

Click Next: Configure Health Check.

Change the Ping Path value to / (delete the text index.html).

Change the Healthy threshold value to 3.

The ELB will periodically test the ping path on each of your web service instances to determine health: a 200 HTTP response code indicates a healthy status, and any other response code indicates an unhealthy status. If an instance is unhealthy and continues in that state for a successive number of checks (unhealthy threshold), the load balancer will remove it from service until it recovers.

In this configuration, making the ping path just a slash (/) will return the default page—the PHP-generated page seen earlier.

The healthy threshold is the number of successful checks the load balancer expects to see in a row before bringing an instance into service behind the load balancer. The lower value will speed things up for this exercise.

Click Next: Add EC2 Instances.

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

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.

Review your settings, and then click Create.

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.

Select the load balancer that 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.

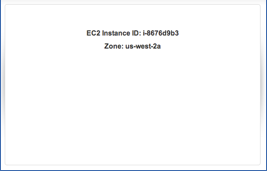
When the status of both Instances is InService, click the Description tab.

Copy the DNS name value to your Clipboard. It will look something like LabELB-325339205.us-west-2.elb.amazonaws.com. Do not copy the “(A Record)” text.

Load balancers can span Availability Zones, and they also scale elastically as needed to handle demand. Therefore, you should always access a load balancer by DNS hostname, and not by IP address. A load balancer may have multiple IP addresses associated with its DNS hostname.

Open a new browser window, paste the DNS Name value into the address bar, and press ENTER.

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



Task 4: View Elastic Load Balancing CloudWatch Metrics

Elastic Load Balancing automatically reports load balancer metrics to CloudWatch.

Amazon CloudWatch provides monitoring for AWS cloud resources and the applications customers run on AWS. Developers and system administrators can use it to collect and track metrics, gain insight, and react immediately to keep their applications and businesses running smoothly. CloudWatch monitors AWS resources such as Amazon EC2 and Amazon Relational Database Service (RDS) DB instances and can also monitor custom metrics generated by your applications and services. With CloudWatch, you gain system-wide visibility into resource use, application performance, and operational health.

Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, or scale your own monitoring systems and infrastructure. Using CloudWatch, you can easily monitor as much or as little metric data as you need. CloudWatch lets you programmatically retrieve your monitoring data, view graphs, and set alarms to help you troubleshoot, spot trends, and take automated action based on the state of your cloud environment.

For more information, see:

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

Return to the AWS Management Console and, on the Services menu, click CloudWatch.

In the navigation pane, click Metrics and in the All metrics tab, click ELB.

Tip: You could also use the search field to search for any metric you would like to view.

Scroll up and down to select the metric or metrics you would like to view.

Load balancing metrics include latency, request count, and healthy and unhealthy host counts. Metrics are reported as they are encountered and can take several minutes to show up in CloudWatch.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

Return to the AWS Management Console.

On the navigation bar, click <yourusername>@<AccountNumber>, and then click Sign out.

Click END LAB.

Click OK.

(Optional) Select the applicable number of stars, type a comment, and then click SUBMIT.

Note: The number of stars indicates the following:

1 star = Very dissatisfied

2 stars = Dissatisfied

3 stars = Neutral

4 stars = Satisfied

5 stars = Very satisfied

You may close the dialog if you don't want to provide feedback.

Conclusion

Congratulations! You now have 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 Amazon EC2 web server instances.

Viewed Amazon CloudWatch metrics for the load balancer.

What Next?

This lab is part of one or more \*qwik\*LABS Quests. If you have completed this lab successfully, you could be on your way to earning a Badge—all you need to do is enroll in the Quest and complete the remaining labs. Visit <https://amazon.qwiklabs.com/lab_catalogue> to explore Quests and enroll.

Additional Resources

More Self-Paced Labs can be found at: [https://amazon.qwiklabs.com](https://amazon.qwiklabs.com/)

For more information about Elastic Load Balancing and Elastic Load Balancing pricing, see: <http://aws.amazon.com/elasticloadbalancing/>

[AWS Training and Certification](http://aws.amazon.com/training/)

For feedback, suggestions, or corrections, please email: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com)

[Lab Overview](https://qwiklab.com/focuses/410?parent=catalog#step1)[Topics covered](https://qwiklab.com/focuses/410?parent=catalog#step2)[Technical knowledge prerequisites](https://qwiklab.com/focuses/410?parent=catalog#step3)[Start Lab](https://qwiklab.com/focuses/410?parent=catalog#step4)[Task 1: Launch Web Servers](https://qwiklab.com/focuses/410?parent=catalog#step5)[Task 2: Connect to Each Web Server](https://qwiklab.com/focuses/410?parent=catalog#step6)[Task 3: Create a Load Balancer](https://qwiklab.com/focuses/410?parent=catalog#step7)[Task 4: View Elastic Load Balancing CloudWatch Metrics](https://qwiklab.com/focuses/410?parent=catalog#step8)[End Lab](https://qwiklab.com/focuses/410?parent=catalog#step9)[Conclusion](https://qwiklab.com/focuses/410?parent=catalog#step10)[What Next?](https://qwiklab.com/focuses/410?parent=catalog#step11)[Additional Resources](https://qwiklab.com/focuses/410?parent=catalog#step12)

Working with Elastic Load Balancing

This lab introduces the concept of Elastic Load Balancing (ELB). In this lab you will use ELB to load balance a set of web servers in an Availability Zone. You will launch a pair of Amazon EC2 instances, bootstrap them to install web servers and content, and then access the instances independently using Amazon EC2 DNS records. Next, you will set up ELB, add your instances to the ELB, and then access the ELB DNS record to watch your requests load balance between servers. Finally, you will look at ELB metrics in CloudWatch. To successfully complete this lab, you should be familiar with the AWS Management Console.

This lab is included in these quests: [Compute & Networking](https://qwiklab.com/quests/6), [Websites & Web Apps](https://qwiklab.com/quests/3), [Digital Media](https://qwiklab.com/quests/2), [SysOps Administrator - Associate](https://qwiklab.com/quests/16), [Solutions Architect - Associate](https://qwiklab.com/quests/10). If you complete this lab you'll receive credit for it when you enroll in one of these quests.

Duration: 0m setup · 55m access · 30m completion

AWS Region: [us-west-2] US West (Oregon)

[load balancing](https://qwiklab.com/tags/load%20balancing) [dns](https://qwiklab.com/tags/dns) [cloudwatch](https://qwiklab.com/tags/cloudwatch) [spl03](https://qwiklab.com/tags/spl03) [webserver](https://qwiklab.com/tags/webserver) [php](https://qwiklab.com/tags/php) [activity tracking](https://qwiklab.com/tags/activity%20tracking) [elb](https://qwiklab.com/tags/elb)