

Lab 12

CREATE YOUR FIRST AMAZON ELASTIC LOAD BALANCING (ELB)

STEP 1: Log In to the Amazon Web Service Console

This laboratory experience is about Amazon Web Services and you will use the AWS Management Console in order to complete all the lab steps.

The screenshot shows the AWS Management Console interface. At the top, there's a header bar with the AWS logo, 'Services' dropdown, 'Tools' dropdown, and user information 'Antonio Ang', 'Oregon', and 'Support'. Below the header, the main content area is titled 'Amazon Web Services' and is divided into several columns of service categories. On the right side, there are 'Additional Resources' including 'Getting Started', 'AWS Console Mobile App', 'AWS Marketplace', 'Service Health', and 'Set Start Page'.

Amazon Web Services

- Compute**
 - EC2: Virtual Servers in the Cloud
 - Lambda PREVIEW: Run Code in Response to Events
- Storage & Content Delivery**
 - S3: Scalable Storage in the Cloud
 - Storage Gateway: Integrates On-Premises IT Environments with Cloud Storage
 - Glacier: Archive Storage in the Cloud
 - CloudFront: Global Content Delivery Network
- Database**
 - RDS: MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
 - DynamoDB: Predictable and Scalable NoSQL Data Stores
 - ElastiCache: In-Memory Cache
 - Redshift: Managed Petabyte-Scale Data Warehouse Service
- Networking**
 - VPC: Isolated Cloud Resources
 - Direct Connect: Dedicated Network Connection to AWS
 - Route 53: Scalable DNS and Domain Name Registration
- Administration & Security**
 - Directory Service: Managed Directories in the Cloud
 - Identity & Access Management: Access Control and Key Management
 - Trusted Advisor: AWS Cloud Optimization Expert
 - CloudTrail: User Activity and Change Tracking
 - Config PREVIEW: Resource Configurations and Inventory
 - CloudWatch: Resource and Application Monitoring
- Deployment & Management**
 - Elastic Beanstalk: AWS Application Container
 - OpsWorks: DevOps Application Management Service
 - CloudFormation: Templated AWS Resource Creation
 - CodeDeploy: Automated Deployments
- Analytics**
 - EMR: Managed Hadoop Framework
 - Kinesis: Real-time Processing of Streaming Big Data
 - Data Pipeline: Orchestration for Data-Driven Workflows
- Application Services**
 - SQS: Message Queue Service
 - SWF: Workflow Service for Coordinating Application Components
 - AppStream: Low Latency Application Streaming
 - Elastic Transcoder: Easy-to-use Scalable Media Transcoding
 - SES: Email Sending Service
 - CloudSearch: Managed Search Service
- Mobile Services**
 - Cognito: User Identity and App Data Synchronization
 - Mobile Analytics: Understand App Usage Data at Scale
 - SNS: Push Notification Service
- Enterprise Applications**
 - WorkSpaces: Desktops in the Cloud
 - Zocalo: Secure Enterprise Storage and Sharing Service

Additional Resources

- Getting Started**
See our documentation to get started and learn more about how to use our services.
- AWS Console Mobile App**
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
- AWS Marketplace**
Find and buy software, launch with 1-Click and pay by the hour.
- Service Health**
All services operating normally.
Updated: Nov 20 2014 12:57:00 GMT-0800
Service Health Dashboard
- Set Start Page**
Console Home

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

Log in to the AWS Management Console

In order to start the laboratory experience, open the Amazon Console by clicking this button:

[Open AWS Console](#)

Log in with the username **xxxx** and the password **xxxx**



Account:

User Name:

Password:

☐ I have an MFA Token ([more info](#))

Sign In

[Sign-in using root account credentials](#)

[Terms of Use](#) [Privacy Policy](#)
© 1996-2014, Amazon Web Services, Inc. or its affiliates.

Select the right AWS Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but you must use the region **US West (Oregon)** for this laboratory.

You can select the **US West (Oregon)** region using the upper right dropdown menu on the AWS Console page.

Antonio Ang ▾ Oregon ▾ Support ▾

- US East (N. Virginia)
- | **US West (Oregon)**
- US West (N. California)
- EU (Ireland)
- EU (Frankfurt)
- Asia Pacific (Singapore)
- Asia Pacific (Tokyo)
- Asia Pacific (Sydney)
- South America (São Paulo)

STEP 2: Create a load balancer using ELB

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances, enabling 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.

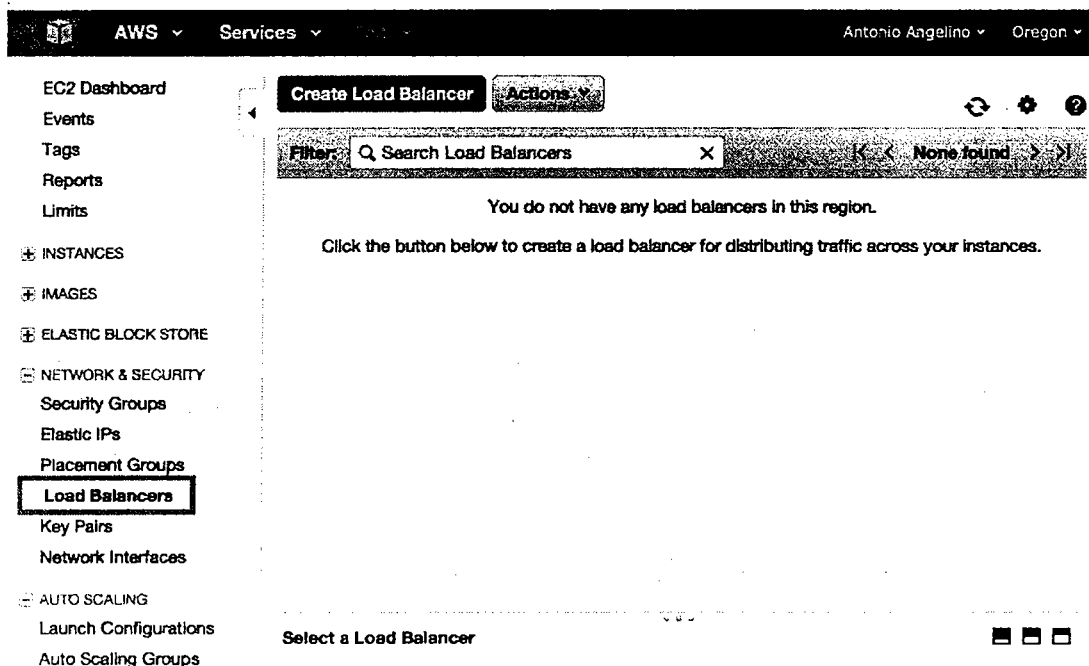
You can create a load balancer and register instances with the load balancer in one or more Availability Zones. The load balancer serves as a single point of contact for clients. This enables you to increase the availability of your application. You can add and remove EC2 instances from your load balancer as your needs change, without disrupting the overall flow of information. If an EC2 instance fails, Elastic Load Balancing automatically reroutes the traffic to the remaining running EC2 instances. If a failed EC2 instance is restored, Elastic Load Balancing restores the traffic to that instance. Elastic Load Balancing can also serve as the first line of defense against attacks on your network. You can offload the work of encryption and decryption to your load balancer so that your EC2 instances can focus on their main work.

Create an ELB using the EC2 dashboard. Select the EC2 service from the Management Console dashboard:

Compute



From the EC2 console dashboard, select **Load Balancers** and then click the **Create Load Balancer** blue button.



The ELB creation wizard is divided into 7 steps. You must choose the load balancer name, select which VPC, subnets, protocols, and ports should be used.

Use the following data for creating your load balancer:

- ✓ **Name:** web-balancer
- ✓ **Create LB inside:** Default VPC (172.31.0.0/16)
- ✓ **Create an internal load balancer:** False
- ✓ **Enable advanced VPC configuration:** Yes

You should enable the following protocol:

- ✓ **LB Protocol:** HTTP - **LB Port:** 80 - **Instance Protocol:** HTTP - **Instance Port:** 80

Select all available subnets and then click **Next: Assign Security Group** button.

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

Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name: web-balancer

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 |

Add

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-de40f8bb (172.31.0.0/16)

Available Subnets

| Actions | Availability Zone | Subnet ID | Subnet CIDR | Name |
|-------------------------------------|-------------------|-----------------|----------------|------|
| <input checked="" type="checkbox"/> | us-west-2a | subnet-845edd3 | 172.31.0.0/20 | |
| <input checked="" type="checkbox"/> | us-west-2b | subnet-8a48240f | 172.31.16.0/20 | |
| <input checked="" type="checkbox"/> | us-west-2c | subnet-afa94ff6 | 172.31.32.0/20 | |

Selected Subnets

| Actions | Availability Zone | Subnet ID | Subnet CIDR | Name |
|-------------------------------------|-------------------|-----------------|----------------|------|
| <input checked="" type="checkbox"/> | us-west-2a | subnet-845edd3 | 172.31.0.0/20 | |
| <input checked="" type="checkbox"/> | us-west-2b | subnet-8a48240f | 172.31.16.0/20 | |
| <input checked="" type="checkbox"/> | us-west-2c | subnet-afa94ff6 | 172.31.32.0/20 | |

Cancel **Next: Assign Security Groups**

Now create a **new security group** that exposes the ELB ports to the internet.

Use the following data for creating the S.G.:

- ✓ Name: elb-security
- ✓ Description: ELB security Group

Add the following rules:

- ✓ Type: HTTP - Source: Anywhere
- ✓ Type: HTTPS - Source: Anywhere

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

Step 2: Assign Security Groups

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

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

| Type ① | Protocol ① | Port Range ① | Source ① |
|--------|------------|--------------|--------------------|
| HTTP | TCP | 80 | Anywhere 0.0.0.0/0 |
| HTTPS | TCP | 443 | Anywhere 0.0.0.0/0 |

Click **Next: Configure Security Settings** and then click **Next: Configure Health Check** gray button for continuing the ELB configuration.

The default health check is almost sufficient for the web-server cluster you're going to use. But you must edit the **Ping Path** to / instead of /index.html.

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

Step 4: Configure Health Check

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

Ping Protocol 
 Ping Port
 Ping Path

Advanced Details

Response Timeout  seconds
 Health Check Interval  seconds
 Unhealthy Threshold  
 Healthy Threshold  

Cancel  

The 5th step of the wizard lists all available EC2 instances. Now select all the available EC2 instances so that they will be used for processing the incoming requests.

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

Step 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-de40f6bb (172.31.0.0/16)

| <input type="checkbox"/> | Instance | Name | State | Security Groups | Zone | Subnet ID | Subnet CIDR |
|-------------------------------------|------------|------|---------|-----------------|------------|-----------------|---------------|
| <input checked="" type="checkbox"/> | i-a8a0905e | | running | launch-wizard-1 | us-west-2a | subnet-845edd83 | 172.31.0.0/20 |
| <input checked="" type="checkbox"/> | i-a8a0905f | | running | launch-wizard-1 | us-west-2a | subnet-845edd83 | 172.31.0.0/20 |

Availability Zone Distribution

2 instances in us-west-2a

☒ Enable Cross-Zone Load Balancing 
☒ Enable Connection Draining  seconds

Cancel  

You can add one or more tags to the ELB instance in the 6th step. You may also skip this step and click the **Review and Create** blue button.

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

Step 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 |
|------|-----------|
| name | webserver |

Create Tag

Cancel Previous Review and Create

The last step allows you to review the ELB configuration before launching it. Click **Create** when you're ready to go.

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

Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer

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

Edit load balancer definition

Configure Health Check

Ping Target: HTTP:80/index.html
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy Threshold: 2
Healthy Threshold: 10

Edit health check

Add EC2 Instances

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances: i-a8a0805e, i-a9a0905f

Edit instances

VPC Information

VPC: vpc-de40f6bb

Edit subnets

Cancel Previous Create

You'll see a flash message after a few seconds that informs you of the ELB creation status.

Load Balancer Creation Status



Successfully created load balancer

Load balancer `web-balancer` was successfully created.

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

Close

STEP 3: Create a self-signed SSL certificate



SSL certificates are required in order to run websites using the **HTTPS** protocol. SSL certificates use a chain of trust, where each certificate is signed (trusted) by a higher, more credible certificate. At the top of the chain of trust are the root certificates, owned by Verisign and others. These certificates are typically shipped with your operating system or web browser.

Normal web traffic is sent unencrypted over the Internet. That is, anyone with access to the right tools can snoop through all of that traffic. This can lead to problems, especially where security and privacy are necessary, such as in credit card data and bank transactions. The **Secure Socket Layer** is used to encrypt the data stream between the web server and the web client (the browser).

SSL makes use of what is known as **asymmetric cryptography**, commonly referred to as **public key cryptography (PKI)**. With public key cryptography, two keys are created, one public, one private. Anything encrypted with either key can only be decrypted with its corresponding key. If a message or data stream were encrypted with the server's private key, it can only be decrypted using its corresponding public key. This ensures that only the data could have come from the server.

SSL Certificates serve a crucial role in the communication process. The certificate, signed by a trusted Certificate Authority (CA), ensures that the certificate holder is really who he/she claims to be. Without a trusted signed certificate, your data may be encrypted, however, the party you are communicating with may not be whom you think. Without certificates, impersonation attacks would be much more common.

When you visit a website over HTTPS, your web browser will receive the ssl certificate for the website. It will examine the contents of the certificate to see that it is valid for the domain name you are trying to visit. After that, it will verify the chain of trust. Your browser will look at who has signed the certificate. If that certificate is a root-certificate, it will compare it

When using a **self-signed certificate**, there is no chain of trust. The certificate has signed itself. The web browser will then issue a warning, telling you that the website certificate cannot be verified. Therefore, you should not use self-signed certificates for professional use, but they can be used to set up temporary ssl servers. You can use them for test and development servers where security is not a big concern.

How to generate a self-signed certificate

Generating a self-signed certificate is quite easy, you need to:

1. Generate a Private Key
2. Generate a Certificate Signing Request
3. Generate a self-signed Certificate

The **OpenSSL** toolkit is used to generate an **RSA Private Key** and **CSR (Certificate Signing Request)**. The first step is to create your RSA Private Key. This key is a 2048 bit RSA key which is encrypted using Triple-DES and stored in a PEM format so that it is readable as ASCII text. Issue the following commands for generating it:

```
openssl genrsa -out my-private-key.pem 2048
```

Once the private key is generated, a Certificate Signing Request can be generated. The CSR is then used for generating a self-signed certificate. During the generation of the CSR, you will be prompted for several pieces of information. These are the **X.509 attributes** of the certificate. One of the prompts will be for "Common Name (e.g., YOUR name)". It is important that this field be filled in with the fully qualified domain name of the server to be protected by SSL. If the website to be protected will be <https://vepsuntest.com>, then enter `vepsuntest.com` at this prompt. The command to generate the CSR is as follows:

```
openssl req -sha256 -new -key my-private-key.pem -out csr.pem
```

You need to generate a self-signed certificate now. Here's the command for generating a certificate valid for 10 years:

```
openssl x509 -req -days 3650 -in csr.pem -signkey my-private-key.pem -out my-certificate.pem
```

Remember that AWS requires a private key exported using the PEM format. Export it using the following command:

```
openssl rsa -in my-private-key.pem -outform PEM
```

You can simulate the whole flow for the generation of a self-signed certificate using our interactive shell. Click on the following button to start:

[Open the Interactive Shell](#)

If you want to skip the certificate generation, you can simply copy the following private key and the self-signed certificate:

-----BEGIN RSA PRIVATE KEY-----

```
MIIEpAIBAAKCAQEA4GlluoC/VAylioY8IItq6f+E8Vm2Bm7ksW2R2eAAQ3kFjxS
qH5zr8TdazY7FNEgvKkC40ASBRb9R8hwyCSJwpXltK49p+k2JuOp9BÜk8gEIE7s
E59nYRT64VAHW6BE36CNk6ca2ZNx4RPt2OsPqiUf1bIWMHiCgJ3glxjclD0K/j4K
qbLUKxqBjclbFMED376k+EwBgEWdpYV2PYNxA28rrV2LtDuAGO6RknFWFGtvgY7f
qkl6XYqMnMiji1b0PygudOQ7+oJACPCSkM4RP4wNzXg2UE8chEbStPtP9Si69qTL
8IJ7LxgnWjGzqZZGEn1ZByORZBSOL/KXvM5hVQIDAQABAoIBAQCSomv1dMEQI39b
ySvel+uTeZ7cvmUYGaBmJEiZ2bhbd/8sxFfif1YKaSGhN3tde9TfUbRwV2lu7zmq
2P9Q2AcoldeXOy9Qc2ON/78Cn7Ht3YmYewVpQQRk/Dd+WDOmnE/Eq/02mL9DokOM
JeCJl/bRX5awpMA5BdWlQyC87jaP3PmC62lgxFVglXBqe1B+rb2JMOj4awgS4B/T
dQqsDNVs44q5kXwMdenXS7WRjqkCxo2NrhINIRd2E+besiWgjuVKYT0TqyHUWLVg
yzeFDaKdO9PKtc9Q595FFuRgjBvJkzvXmrJcQ/3TY/DLUYle4wL5oYFztgAHWAfd
OlgSiz+BAoGBAP1//34AanGFE0xSsAYiEzohyfRgzGzXrsHl849X90CB5UW2e3dJ
uhgXz8gBg6WavqQWg1hCfzfwkrU2+L7XqkOxsTAcL/bN3CkN8D1ohwEgHb7qzJKZ
W6Y9X0xxJWK48PKPvx5KRUzbbp5tgGx1QXIVaLTvUVOWtTHB+yTcFc1AoGBAOKf
9hTJIAHlceVPX7UhjsQ06tGqqHxhzrUh3+Ejo5pTcRdtLrKDWcaSFG8iRcXwpc3i
vZ+9tl9vtDSd1rTa3REaVkrHYN9vsiY0JHHHYjk0fTSVpqvrFbu/qTS9MzZ/qaN2
2VDAAlmyhtALxM3/KTR/PNyAT1Afvpl9Y4/UIYWhAoGAYQKyy41tLrQ2hma+Zhp0
MTLtDLBc6uo/PoS5ilmpXU5YZy1GYogcZ0v1gBzUPHPTsQfMi+ImvUmbWy4GU0JF
LLK59CdVU6XEMxHadiWiRJP9ziocz51QrXWfGqnSHM2Zp7nK8dSKxNLXkx0wwwcVE
OEpkO128eOFbcUIZjjd+LmECgYBGd2qzCALnnAqQPOALmEWmKLYjP6doFZmKpN/S
R5ylbfCqUh7FDyapld8Mt2FurOdBX5GKzBibEEa+XZ3XWn6GxOOyE2fB0h9Y1bnH
```

TzHxi6qq4SWUK2L0oCHi7jmwZn2/AEOOalt0vJhCtIYbb43GbxHjnm+vAE/ndQMy
Q+3CQQKBgQDZZABWYtbG1ZKwXFUWlEI/AY5EEek1u3qYbhng2bz72tW39dnbxJcz
fz2r4AvZ6ajpO2BTedwzceGxlegQ8+I75aBbYgGLcEJiwrQZC0XCWbxwM1Umz5O6
OHhx4BxqnNObr4ZCWNgFy7UMmpqTYqhB5L/Zi6r6ae0/zAOY/HIbJg==
-----END RSA PRIVATE KEY-----

Self-signed certificate:

-----BEGIN CERTIFICATE-----

MIIDfjCCAmYCCQDG5+kYEGkrdTANBgkqhkiG9w0BAQUFADCgDELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCkNhbgGmb3JuaWEFJAUBGNVBAcTDVNHbiBGcmFuY2ZlY28x
GjAYBgNVBAoTEUNsb3VkiEFjYWRLbXkgSW5jMQ0wCwYDVQQLEwRMYWJzMRkwFwYD
VQQDExBjbG91ZGFjYWRLbXkuY29tMB4XDTE2MDIyNjEwMTE0M1oXDTE3MDIyNTEw
MTE0M1owgYAxCAJBGNVBAYTAIVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRYwFAFD
VQQHEw1TYW4gRnJhbG91ZGFjYWRLbXkuY29tMB4XDTE2MDIyNjEwMTE0M1oXDTE3MDIyNTEw
MTE0M1owgYAxCAJBGNVBAYTAIVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRYwFAFD
MA5GA1UECXMETGFicEZMBcGA1UEAxMQY2xvdWRhY2FkZW15LmNvbTCCASlwDQYJ
KoZIHvcNAQEBBQADggEPADCCAQoCggEBABOApZbqAv1QMpYqImPCCbaun/hPFZtgZ
u5LFtkdngAEN5BY8Uqh+c6/E3Ws2OxTRILypAuNAEgUW/UflcMgkicKcZbSuPafp
NibjqfQYiZPIBJRO7BOFZ2EU+uFQB1ugRN+gjZOnGtmTceET7djrD6olH9WyfJB4
goCd4CMY3CA9Cv4+Cqmy1CsagY3CGxTBA9++pPhMAYBFnaWFdj2DcQNvK61di7Q7
gBjukZJxVhRrb4GO36pJel2KjJzlo4tW9D8oLnTkO/qCQAjwkpDOET+MDc14NIBP
HIRG0rT7T/Uouvaky/CCey8YJ1oxs6mWRhJ9WQcjkWQUjiyl7zOYVUCAwEAATAN
BgkqhkiG9w0BAQUFAAOCAQEACGdFpb5Np8SKAi0H5K5mObijbDyJITeep35DJ2Rb
tNxicXMYCXeXaXe3AUgAja+RC0YsaqLG4mHe/YV+WY7klGvHLJ5tHEmHHoSa0Oo
JsPKYtidsyzHQvXO/JA6HjNgajuSqDj1h01s9+6/dXFzyUqzkINCi5+H9yEYpaX
S39M+LM21arpHLyQDLA+/wmNvLslxKTZebqSW8COAgCZFxajA5APYfOgzTyif1Ng
lBj3sV4s9qh5PAI+9c5yHVSb2O0luEiKT8eFXGBIPKisEfchsYiCzDwoM3VG9B+F
Ti5OEUR0s2sTjy/qEev/4idnrwQSfUHVpDNI CCMT0XF0qg==
-----END CERTIFICATE-----

STEP 4: Enable SSL support for ELB

If you want to accept HTTPS connections using the load balancer, you need to add a new listener that acts as an SSL Terminator.

Choose the **web-balancer** load balancer and then select the Listeners tab pane.

The screenshot shows the AWS Management Console interface for a load balancer. At the top, there's a 'Create Load Balancer' button and an 'Actions' dropdown. Below that is a search bar with 'web-balancer' entered. A table lists the load balancer details: 'web-balancer', 'web-balancer-1852516358.us...', '80 (HTTP) forwarding to 80 (...)', 'us-west-2a, us-west-2b...', '2 instances', and 'HTTP:80/'. Below the table, the 'Listeners' tab is selected. It shows a message: 'The following listeners are currently configured for this load balancer:'. A table lists the current listener: 'Load Balancer Protocol: HTTP', 'Load Balancer Port: 80', 'Instance Protocol: HTTP', 'Instance Port: 80', 'Cipher: N/A', and 'SSL Certificate: N/A'. An 'Edit' button is visible below the table.

| Load Balancer Name | DNS Name | Port Configuration | Availability Zones | Instance Count | Health Check |
|--------------------|-------------------------------|----------------------------------|---------------------------|----------------|--------------|
| web-balancer | web-balancer-1852516358.us... | 80 (HTTP) forwarding to 80 (...) | us-west-2a, us-west-2b... | 2 instances | HTTP:80/ |

Load balancer: web-balancer

Description Instances Health Check Monitoring Security Listeners Tags

The following listeners are currently configured for this load balancer:

| Load Balancer Protocol | Load Balancer Port | Instance Protocol | Instance Port | Cipher | SSL Certificate |
|------------------------|--------------------|-------------------|---------------|--------|-----------------|
| HTTP | 80 | HTTP | 80 | N/A | N/A |

Edit

Click **Edit** and then add the following listener:

- ✓ LB Protocol: HTTPS - LB Port: 443 - Instance Protocol: HTTP - Instance Port: 80

The screenshot shows the 'Edit listeners' dialog box. It has a title bar 'Edit listeners' with a close button. Below the title bar, it says 'The following listeners are currently configured for this load balancer:'. A table lists the current listeners: 'Load Balancer Protocol: HTTP', 'Load Balancer Port: 80', 'Instance Protocol: HTTP', 'Instance Port: 80', 'Cipher: N/A', and 'SSL Certificate: N/A'. Below this table, there's an 'Add' button. At the bottom right, there are 'Cancel' and 'Save' buttons.

| Load Balancer Protocol | Load Balancer Port | Instance Protocol | Instance Port | Cipher | SSL Certificate |
|------------------------|--------------------|-------------------|---------------|--------|-----------------|
| HTTP | 80 | HTTP | 80 | N/A | N/A |

Add

Cancel Save

You must upload an SSL certificate and then assign it to the listener, so ELB will be able to use it for handling SSL requests.

Click the **Change** link in the SSL Certificate column and then fill the Certificate fields. You can use the Self-signed SSL Certificate previously issued.

Choose a certificate name (e.g. labs-certificate) and then copy and paste both RSA Private and Public Certificate keys.

Select Certificate

An SSL Certificate allows you to configure the HTTPS/SSL listeners of your load balancer. You may select a previously uploaded certificate below, or define a new SSL Certificate. [Learn more](#) about setting up HTTPS load balancers and certificate management.

Certificate Type:

☐ Choose an existing certificate from AWS Identity and Access Management (IAM)
 ☒ Upload a new SSL certificate to AWS Identity and Access Management (IAM)

Certificate Name:

labs-certificate

Private Key:

```

O+3CQOKBoODZZARWYtbo1ZKw/FLWIE/AY5EEP1Jg3Ybno2bzZ2W39dnw4cz
fr2dAvZ6elnO2BTEdwonGde028+H758BbYnGLc5LmWZCAXGWpoxM1Umx5Q8
O7Hx4BmohObr4ZCWNoFzUAmppqTYqRSL/Z6r6ee0zAOY/HbJg==
-----END RSA PRIVATE KEY-----
          
```

(pem encoded)

Public Key Certificate:

```

S39M+LM21apHLyQDLA+ym8M1sbK7ZabqSWXQDAQZFaASAEYJ0gzUitNg
IR8vYas8n6PA+8c7dYSE2C0uEIK1BefXGBIPK6EdmYJCzDwom3YGS6-F
J5CEUR0n2aTy/rE8w/4knoxGSL7dyDNCCMTIXEdog==
-----END CERTIFICATE-----
          
```

(pem encoded)

Certificate Chain:

Optional

(pem encoded)

Cancel

Save

Click **Save** and you'll see the name of the uploaded certificate under the SSL Certificate column.

Edit listeners

The following listeners are currently configured for this load balancer:

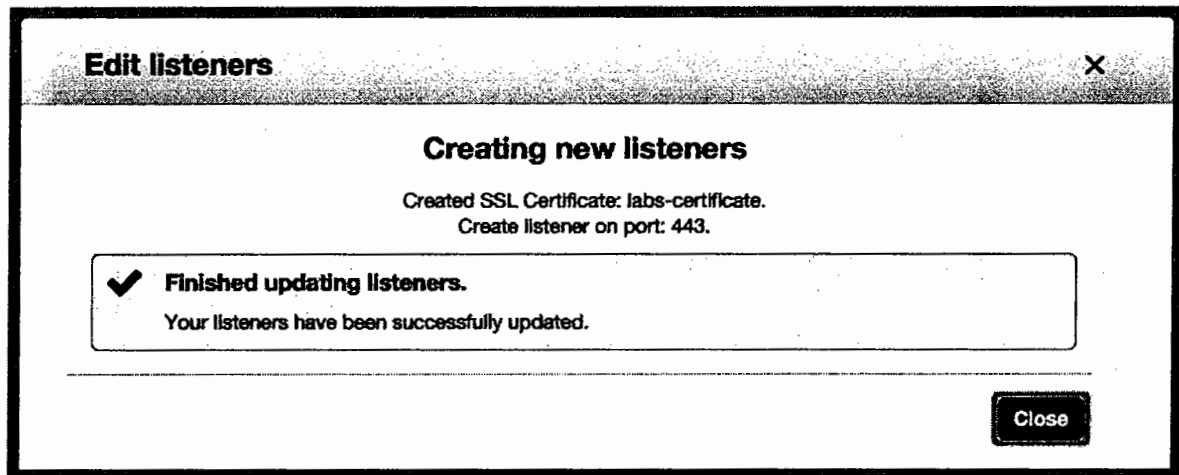
| Load Balancer Protocol | Load Balancer Port | Instance Protocol | Instance Port | Cipher | SSL Certificate | |
|------------------------|--------------------|-------------------|---------------|--------|-------------------------|---|
| HTTP | 80 | HTTP | 80 | N/A | N/A | ✕ |
| HTTPS (Secure HTTP) | 443 | HTTP | 80 | Change | labs-certificate Change | ✕ |

Add

Cancel

Save

Click **Save** again and your load balancer will start serving content also using the HTTPS protocol.



STEP 5: Check if the Load Balancer is working properly

If you successfully create a Load Balancer, you should be able to see it in the Load Balancers listing page, even if it's still not working because the EC2 instances are not in service.

Select the load balancer and check the **Status** field in the **Description** pane. After a couple of minutes, the load balancer will be ready to accept the incoming connections and balance the load between all selected EC2 instances.

Create Load BalancerActions

Filter: Q Search Load BalancersX

| <input type="checkbox"/> | Load Balancer Name | DNS Name | Port Configuration | Availability Zones | Instance Count |
|--------------------------|--------------------|------------------------------|----------------------------------|---------------------------|----------------|
| <input type="checkbox"/> | web-balancer | web-balancer-899940622.us... | 80 (HTTP) forwarding to 80 (...) | us-west-2a, us-west-2b... | 2 Instances |

Load balancer: web-balancer

Description

Instances

Health Check

Monitoring

Security

Listeners

Tags

DNS Name:

web-balancer-899940622.us-west-2.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

Status:

0 of 2 Instances In service

Port Configuration:

80 (HTTP) forwarding to 80 (HTTP)

Stickiness: Disabled (Edit)

443 (HTTPS, Certificate: labs-cert) forwarding to 80 (HTTP)

Stickiness: Disabled (Edit)

Your ELB instance is now ready and accessible from the web, so you can open the sample website hosted on backend instances.

You can find the ELB endpoint URL (check the **DNS Name** field) in the Description tab once you select your ELB instance in the load balancer listing page.

Create Load Balancer

Actions

| Filter: | Q Search Load Balancers | X | | | | | |
|--------------------|------------------------------|----------------------------------|---------------------------|----------------|--------------|------------|--|
| Load Balancer Name | DNS Name | Port Configuration | Availability Zones | Instance Count | Health Check | Created At | |
| web-balancer | web-balancer-899940622.us... | 80 (HTTP) forwarding to 80 (...) | us-west-2a, us-west-2b... | 2 Instances | HTTP:80/ | June | |

Load balancer: web-balancer

Description

Instances

Health Check

Monitoring

Security

Listeners

Tags

DNS Name:

web-balancer-899940622.us-west-2.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

Status:

2 of 2 Instances In service

Port Configuration:

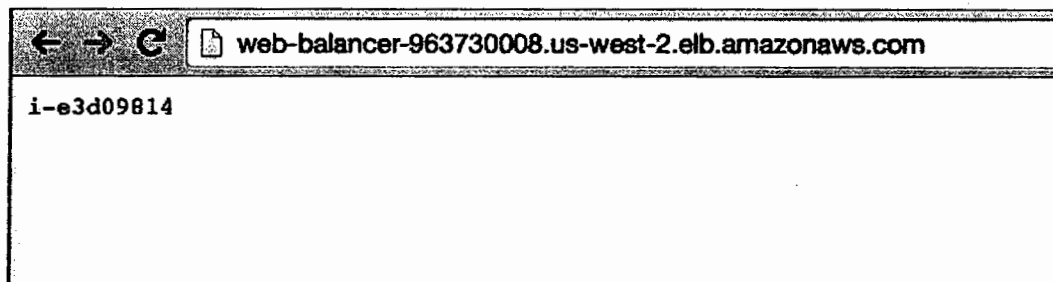
80 (HTTP) forwarding to 80 (HTTP)

Stickiness: Disabled (Edit)

443 (HTTPS, Certificate: labs-cert) forwarding to 80 (HTTP)

Stickiness: Disabled (Edit)

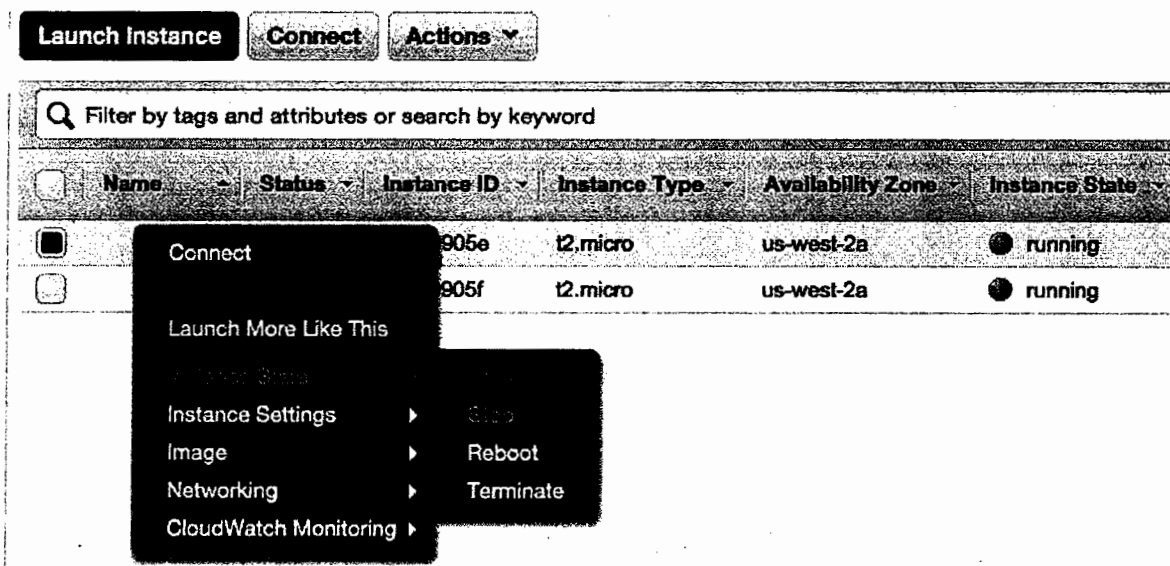
By opening the endpoint URL page, you will see a white page with the instance-ID of a specific backend node. Keep refreshing the page and you will see different instance IDs.



STEP 6: Check the ELB behavior during an instance failure

Let's stop one of our backend instances and check if our sample website continues working.

Click on **Instances** to open the EC2 instances dashboard. Select one of your instances by checking the instance ID, right-click on it and then click on the **Stop** action.



Go back to the Load Balancers listing page and you will see that one of backend EC2 instances is now marked as OutOfService, so all the incoming connections will be handled only using the alive instance.

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Load Balancers

Key Pairs

Network Interfaces

AUTO SCALING

Launch Configurations

Auto Scaling Groups

Create Load Balancer

Actions

Filter: Q Search Load Balancers

Load Balancer Name DNS Name Port Configuration Availability Zones Instance Count Health Check

web-balancer web-balancer-963730008.us-west-2.elb.amazonaws.com 80 (HTTP) forwarding to 80 (HTTP) us-west-2a, us-west-2b 2 Instances HTTP:80

Load balancer: web-balancer

Description Instances Health Check Monitoring Security Listeners Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

| Instance ID | Name | Availability Zone | Status | Actions |
|-------------|------|-------------------|------------------|---------------------------|
| i-8a0005e | | us-west-2a | OutOfService (1) | Remove from Load Balancer |
| i-9a0005f | | us-west-2a | InService (1) | Remove from Load Balancer |

Edit Availability Zones

| Availability Zone | Subnet ID | Subnet CIDR | Instance Count | Healthy? | Actions |
|-------------------|-----------------|----------------|----------------|--|---------------------------|
| us-west-2a | subnet-845edd73 | 172.31.0.0/20 | 2 | Yes | Remove from Load Balancer |
| us-west-2b | subnet-6a49240f | 172.31.16.0/20 | 0 | No (Availability Zone contains no healthy instances) | Remove from Load Balancer |
| us-west-2c | subnet-af4d4f6 | 172.31.32.0/20 | 0 | No (Availability Zone contains no healthy instances) | Remove from Load Balancer |

Now the load balancer is serving web pages by using only one EC2 instance. If you keep refreshing the page, the web server rate-limiter will return an error page.

web-balancer-963730008.us-west-2.elb.amazonaws.com

Cloud Academy Labs: ELB Lab experience

The page you are looking for is temporarily unavailable. Please try again later.

Rate limiter in action!

This Webserver instance can reply to a fixed number of requests per minute. You cannot request more than 3 pages per minute using the same IP address. Please setup an Amazon ELB instance in order to serve webpages by using different instances.

STEP 7: Check the ELB behavior after a successful node recovery

Let's restart the stopped instance in order to recover the full functionality of our balanced infrastructure:

1. Click on **Instances** t opening the EC2 instances dashboard
2. Select the previously stopped instance, right-click on it, and click on **Start**

Amazon ELB constantly checks the backend instances status. After a few minutes it will again start using both EC2 instances. You can check this by opening the ELB details pane.

The screenshot shows the AWS Management Console interface for a Load Balancer. At the top, there's a 'Create Load Balancer' button and an 'Actions' dropdown. Below is a table with columns: Load Balancer Name, DNS Name, Port Configuration, Availability Zones, Instance Count, Health Check, and Create Date. The table contains one entry: 'web-balancer' with DNS Name 'web-balancer-899940622.us-west-2.elb.amazonaws.com', Port Configuration '80 (HTTP) forwarding to 80', Availability Zones 'us-west-2a, us-west-2b', Instance Count '2 instances', Health Check 'HTTP-80', and Create Date 'June 1, 2015 1:10 PM UTC-07:00'. Below the table, the 'Load balancer: web-balancer' section is expanded, showing tabs for Description, Instances, Health Check, Monitoring, Security, Listeners, and Tags. The 'Description' tab is active, displaying the DNS Name, a note about IP addresses, Scheme (internet-facing), Status (2 of 2 instances in service), and Port Configuration (80 (HTTP) forwarding to 80 (HTTP) and 443 (HTTPS, Certificate: labs-cert) forwarding to 80 (HTTP)).

STEP 8: Destroy an ELB instance

Destroying an ELB instance is easy and fast.

Select your **Load Balancer instance** from the ELB instances list, click the **Action** button, then select **Delete**.

The screenshot shows the AWS Management Console interface for a Load Balancer. At the top, there's a 'Create Load Balancer' button and an 'Actions' dropdown. Below is a table with columns: Load Balancer Name, DNS Name, Port Configuration, Availability Zones, Instance Count, Health Check, and Create Date. The table contains one entry: 'web-balancer' with DNS Name 'web-balancer-899940622.us-west-2.elb.amazonaws.com'. The 'Actions' dropdown menu is open, showing options: Edit health check, Edit subnets, Edit instances, Edit listeners, and Edit security groups.

Click **Yes, Delete** for confirming the action and your load balancer will be permanently deleted.

