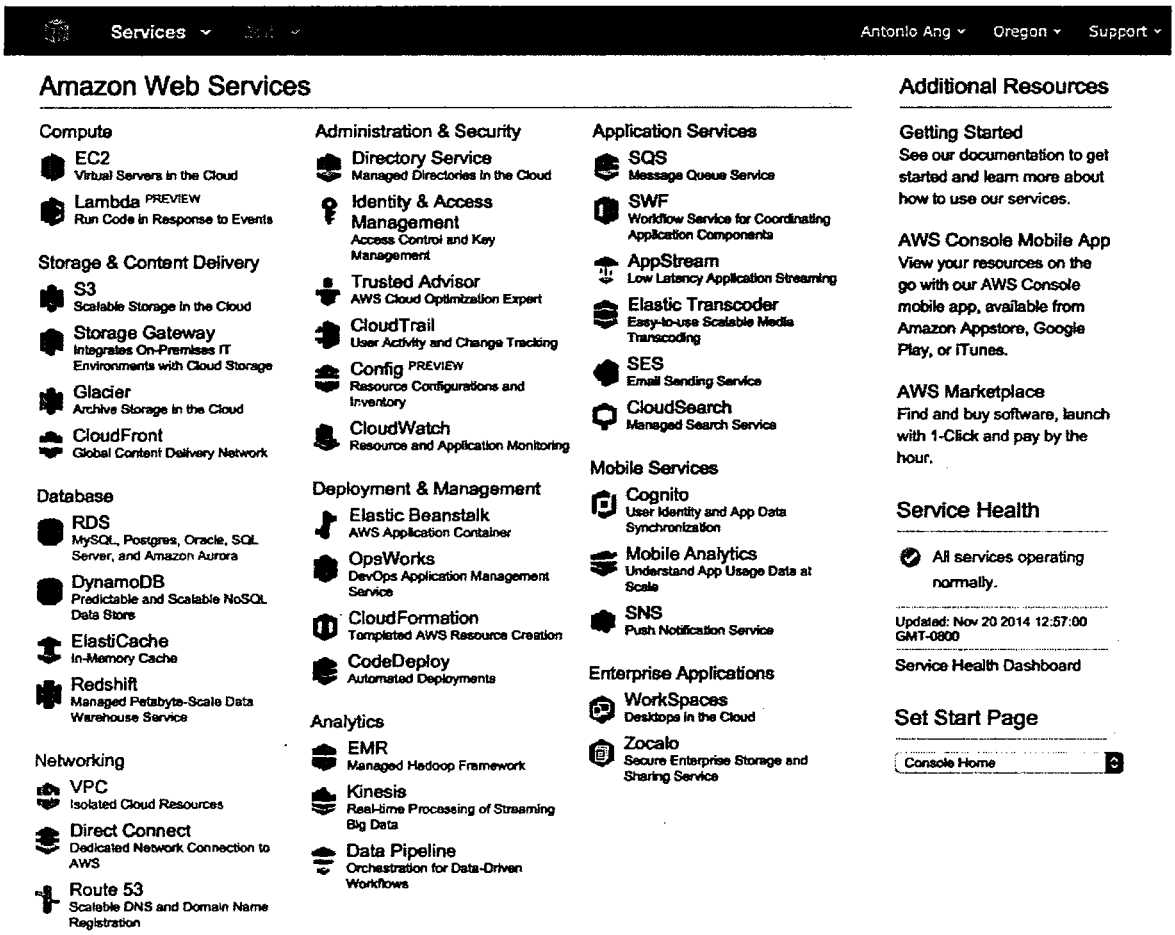
Lab 12

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 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:

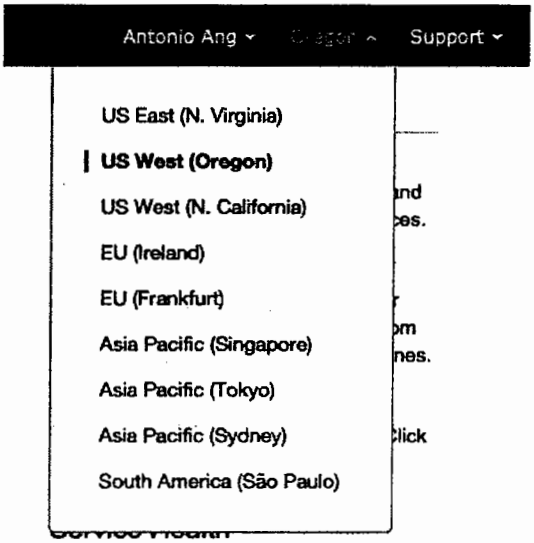
Open AWS Console

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

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.



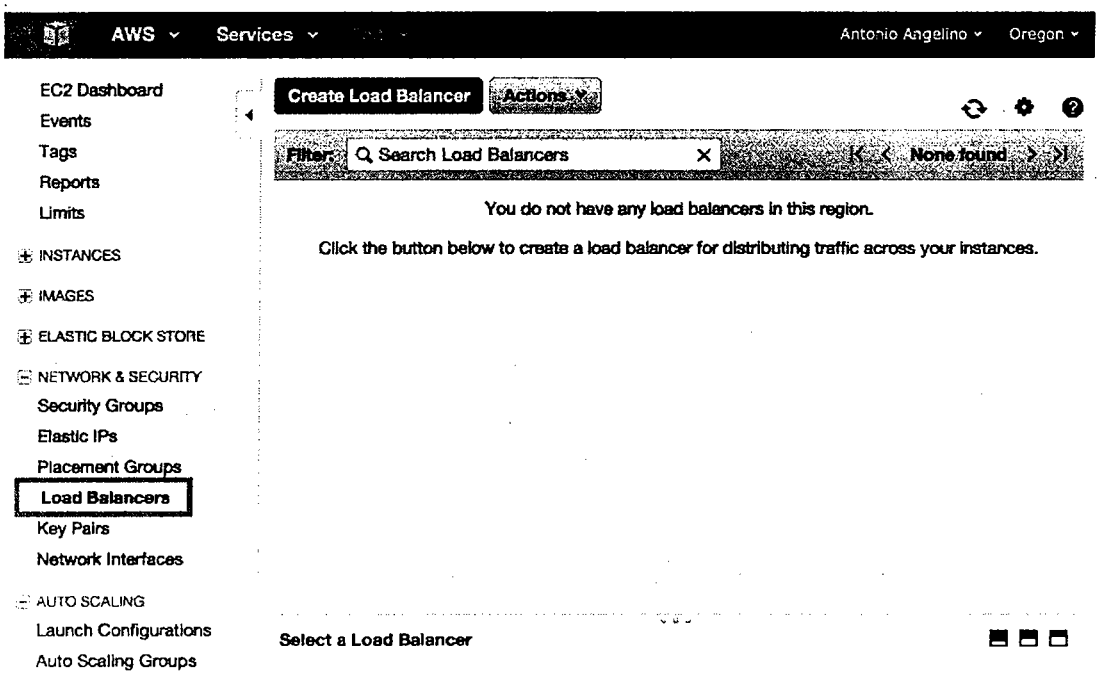
STEP 2: Create a load balancer using ELB

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:

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.

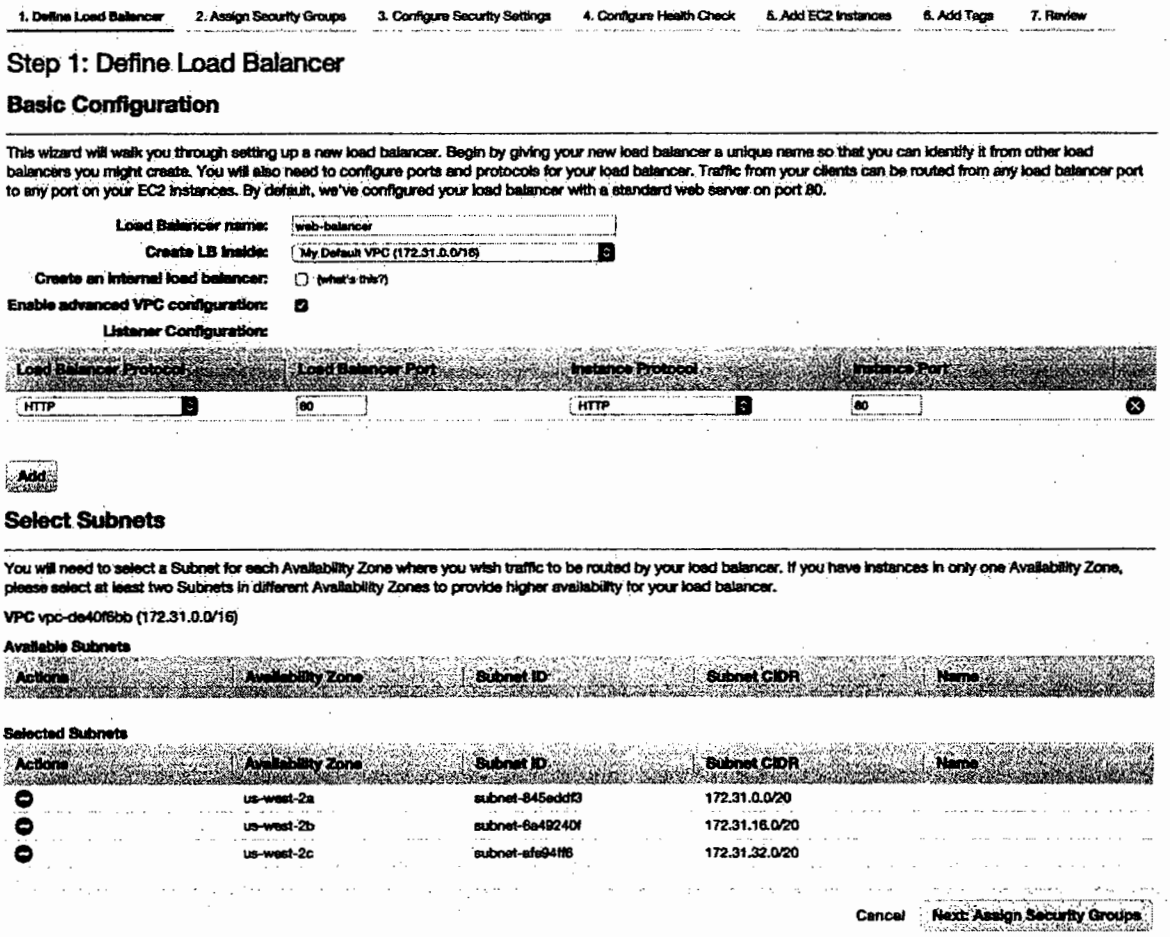
Use the following data for creating your load balancer:

- ✓ **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.



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: elb-security

Description: ELB security group

Type	Protocol	Port Range	Source	
HTTP	TCP	80	Anywhere 0.0.0.0/0	⊗
HTTPS	TCP	443	Anywhere 0.0.0.0/0	⊗

Add Rule

Cancel | Previous | Next: Configure Security Settings

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	HTTP
Ping Port	80
Ping Path	/

Advanced Details

Response Timeout	5	seconds
Health Check Interval	30	seconds
Unhealthy Threshold	2	
Healthy Threshold	10	

Cancel | Previous | Next: Add EC2 Instances

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)

Instance	Name	State	Security Groups	Zone	Subnet ID	Subnet CIDR
i-a8a0905e		running	launch-wizard-1	us-west-2a	subnet-845eddf3	172.31.0.0/20
i-a9a0905f		running	launch-wizard-1	us-west-2a	subnet-845eddf3	172.31.0.0/20

Availability Zone Distribution
2 instances in us-west-2a

☑ Enable Cross-Zone Load Balancing

☑ Enable Connection Draining 300 seconds

Cancel | Previous | Next: Add Tags

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

Edit load balancer definition

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

Configure Health Check

Edit health check

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

Add EC2 Instances

Edit instances

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances: i-a8a0905e, i-a9a0905f

VPC Information

Edit subnets

VPC: vpc-de40f8bb

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 websiteover HTTPS, your web browser will receive the ssl certificate for the website. It will examine the contents of the certificate to see that 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 websitecertificate 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/VAylioiY8IJtq6f+E8Vm2Bm7ksW2R2eAAQ3kFjxS

qH5zr8TdazY7FNEgvKkC40ASBRb9R8hwyCSJwpxltK49p+k2JuOp9BiUk8gElE7s

E59nYRT64VAHW6BE36CNk6ca2ZNx4RPt2OsPqiUf1blWMHiCgJ3gIxjcID0K/j4K

qbLUKxqBjcIbFMED376k+EwBgEWdpYV2PYNxA28rrV2LtDuAGO6RknFWFGtvgY7f

qkl6XYqMnMiji1b0PygudOQ7+oJACPCSkM4RP4wNzXg2UE8chEbStPtP9Si69qTL

8IJ7LxgnWjGzqZZGEn1ZByORZBSOL/KXvM5hVQIDAQABAoIBAQCSomv1dMEQI39b

ySveI+uTeZ7cvmUYGaBmJEiZ2bhbd/8sxFfif1YKaSGhN3tde9TfUbRwV2Iu7zmq

2P9Q2AcoIdeXOy9Qc2ON/78Cn7Ht3YmYewVpQQRk/Dd+WDOmnE/Eq/02mL9DokOM

JeCJl/bRX5awpMA5BdWIQyC87jaP3PmC62lgxFVgIXBqe1B+rb2JMOj4awgS4B/T

dQqsDNVs44q5kXwMdenXS7WRjqkCxo2NrhINIRd2E+besiWgjvIKYT0TqyHUWLVg

yzeFDaKdO9PKtc9Q595FFuRgjBvJkzvxmrJcQ/3TY/DLUYIe4wL5oYFztgAHWAfd

OlgSiz+BAoGBAP1//34AanGFE0xSsAYiEzohyfRgzGzXrsHl849X90CB5UW2e3dJ

uhgXz8gBg6WavqQWg1hCfzfwkrU2+L7XqkOxsTAcL/bN3CkN8D1ohwEgHb7qzJKZ

W6Y9X0xxJWK48PKPvx5KRUzbbp5tgGx1QXIVaLTvUVOWtTHB+yTcFc1AoGBAOKf

9hTJlAHlceVPX7UhjsQ06tGqqHxhzrUh3+Ejo5pTcRdtLrKDWcaSFG8iRcXwpc3i

vZ+9tl9vtDSd1rTa3REaVkRHYN9vsiY0JHHHYjk0fTSVpqvrFbu/qTS9MzZ/qaN2

2VDaAlmyhtALxM3/KTR/PNyaT1AfvpI9Y4/UlYWhAoGAYQKyy41tLrQ2hma+Zhp0

MTLtDlBc6uo/PoS5iImpXU5YZy1GYogcZ0v1gBzUPHPTsQfMi+ImvUmbWy4GU0JF

LlK59CdVU6XEMxHadiWiRJP9ziocz51QrXWfGqnSHM2Zp7nK8dSKxNLXkx0wwcVE

OEpKO128eOFbcUIZjjd+LmECgYBGd2qzCALnnAqQPOALmEWmKLYjP6doFZmKpN/S

R5ylbfCqUh7FDyapId8Mt2FurOdBX5GKzBibEEa+XZ3XWn6GxOOyE2fB0h9Y1bnH

TzHxi6qq4SWUK2L0oCHi7jmwZn2/AEOOalt0vJhCtIYbb43GbxHjnm+vAE/ndQMy
Q+3CQQKBgQDZZABWYtbg1ZKwXFUWlEI/AY5EEEk1u3qYbhng2bz72tW39dnbxJcz
fz2r4AvZ6ajpO2BTEdwzceGxlegQ8+I75aBbYgGLcEJiwrQZC0XCWbxwM1Umz5O6
OHhx4BxqnNObr4ZCWNgFy7UMmpqTYqhB5L/Zi6r6ae0/zAOY/HIbJg==
-----END RSA PRIVATE KEY-----

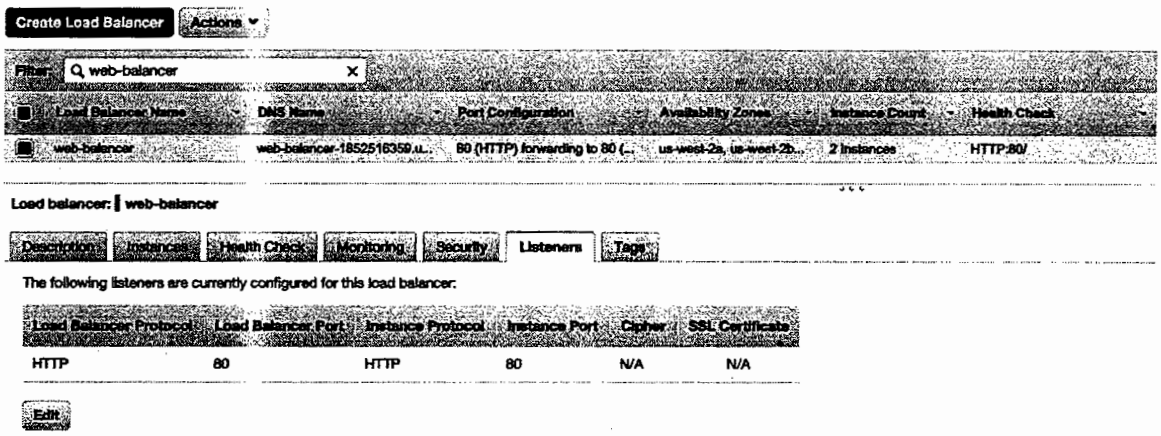
Self-signed certificate:

-----BEGIN CERTIFICATE-----
MIIDfjCCAmYCCQDG5+kYEGkrdTANBgkqhkiG9w0BAQUFADCBgDELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBGcmFuY2lzY28x
GjAYBgNVBAoTEUNsb3VkIEFjYWRlbXkgSW5jMQ0wCwYDVQQLEwRMYWJzMRkwFwYD
VQQDExBjbG91ZGFjYWRlbXkuY29tMB4XDTE2MDIyNjEwMTE0M1oXDTE3MDIyNTEw
MTE0M1owgYAxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRYwFAYD
VQQHEw1TYW4gRnJhbmNpc2NvMRowGAYDVQQKExFDbG91ZCBBY2FkZW15IEluYzEN
MAsGA1UECxMETGFiczEZMBcGA1UEAxMQY2xvdWRhY2FkZW15LmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAOBpZbqAv1QMpYqImPCCbaun/hPFZtgZ
u5LFtkdngAEN5BY8Uqh+c6/E3Ws2OxTRILypAuNAEgUW/UfIcMgkicKcZbSuPafp
NibjqfQYiZPIBJRO7BOfZ2EU+uFQB1ugRN+gjZOnGtmTceET7djrD6olH9WyFjB4
goCd4CMY3CA9Cv4+Cqmy1CsagY3CGxTBA9++pPhMAYBFnaWFdj2DcQNvK61di7Q7
gBjukZJxVhRrb4GO36pJel2KjJzIo4tW9D8oLnTkO/qCQAjwkpDOET+MDc14NlBP
HIRG0rT7T/Uouvaky/CCey8YJ1oxs6mWRhJ9WQcjkWQUji/yl7zOYVUCAwEAATAN
BgkqhkiG9w0BAQUFAAOCAQEACGdFpb5Np8SKAi0H5K5mObijbDyJITeep35DJ2Rb
tNxicXMYCXeXaXe3AUgAjcA+RC0YsaqLG4mHe/YV+WY7kLgvHLJ5tHEmHHoSa0Oo
JsPKYtidsyzHQvXO/JA6HjNgajuSqDj1h01s9+6/dXFzyUqzkINCi5+H9yEYpalX
S39M+LM21arpHLyQDLA+/wmNvLsIxKTZebqSW8COAgCZFxajA5APYfOgzTyif1Ng
lBj3sV4s9qh5PAI+9c5yHVSB2O0IuEiKT8eFXGBIPKisEfchsYiCzDwoM3VG9B+F
Ti5OEUR0s2sTJy/qEev/4idnrwQSfUHVpDNlCCMT0XF0qg==
-----END CERTIFICATE-----

STEP 4: Enable SSL support for ELB

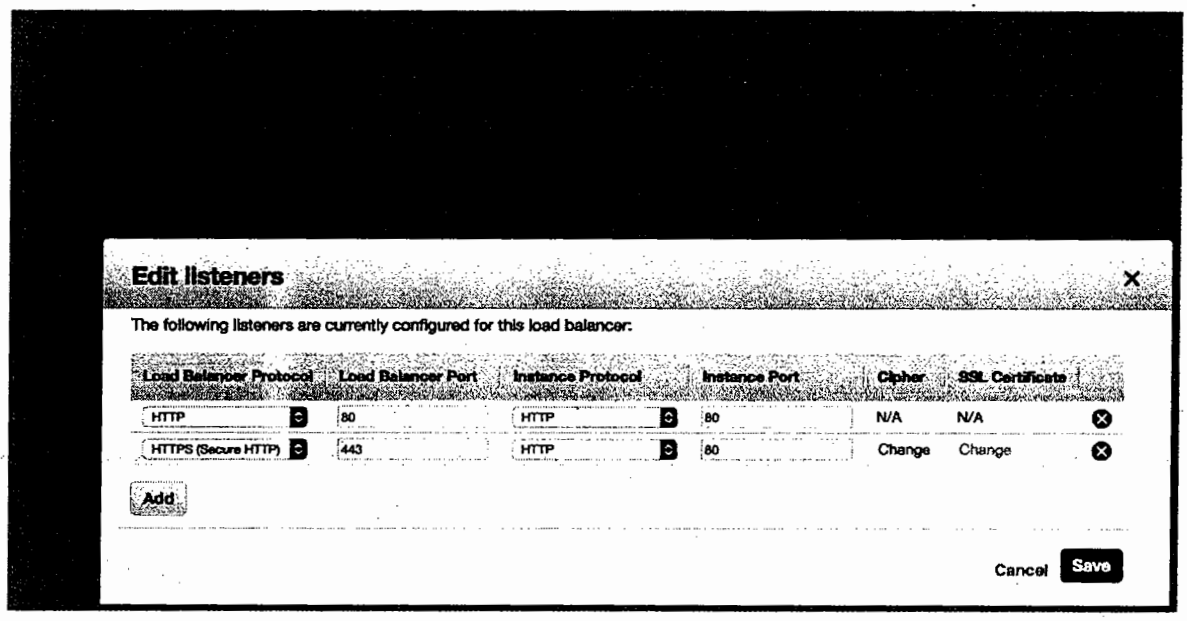
If you want to accept HTTPS connections using the load balancer, you need to add a new listener that acts as an SSL Terminator.

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



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

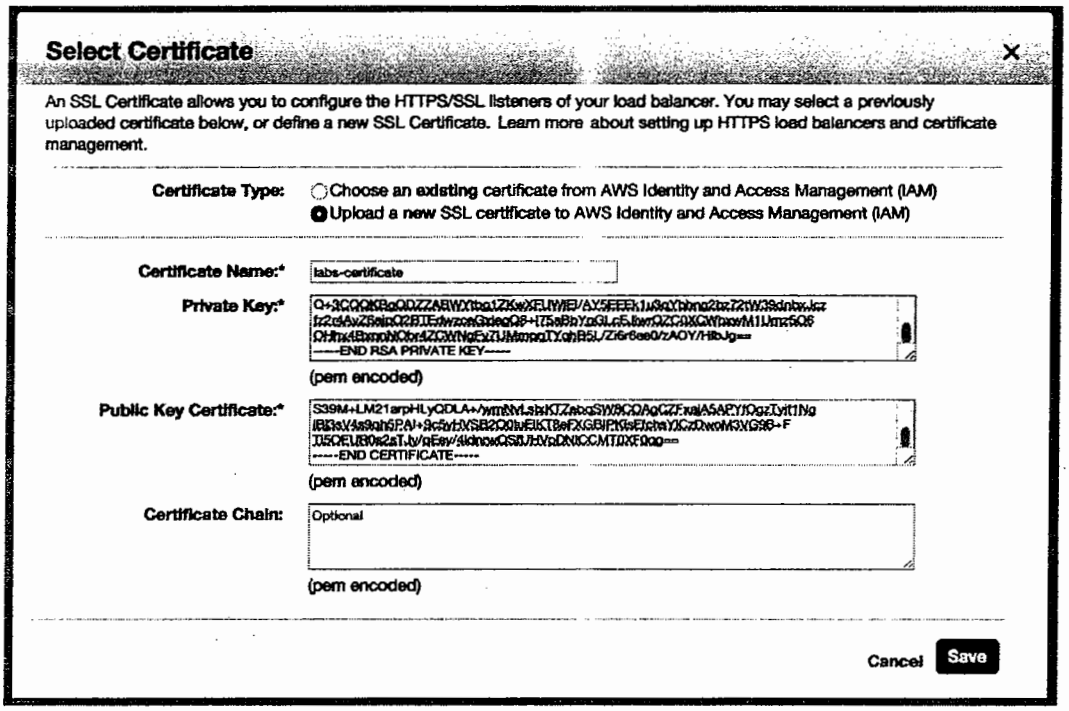
- ✓ LB Protocol: HTTPS - LB Port: 443 - Instance Protocol: HTTP - Instance Port: 80



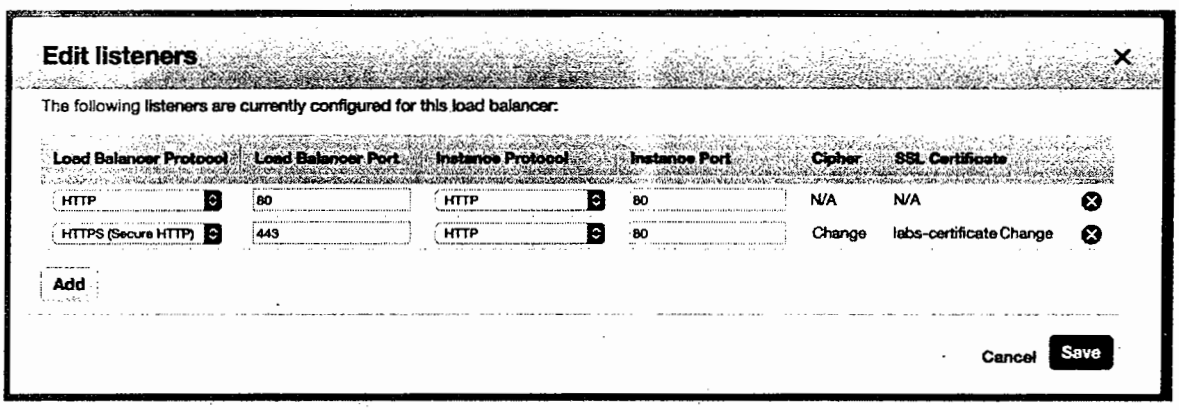
You must upload an SSL certificate and then assign it to the listener, so ELB will be able to use it for handling SSL requests.

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.

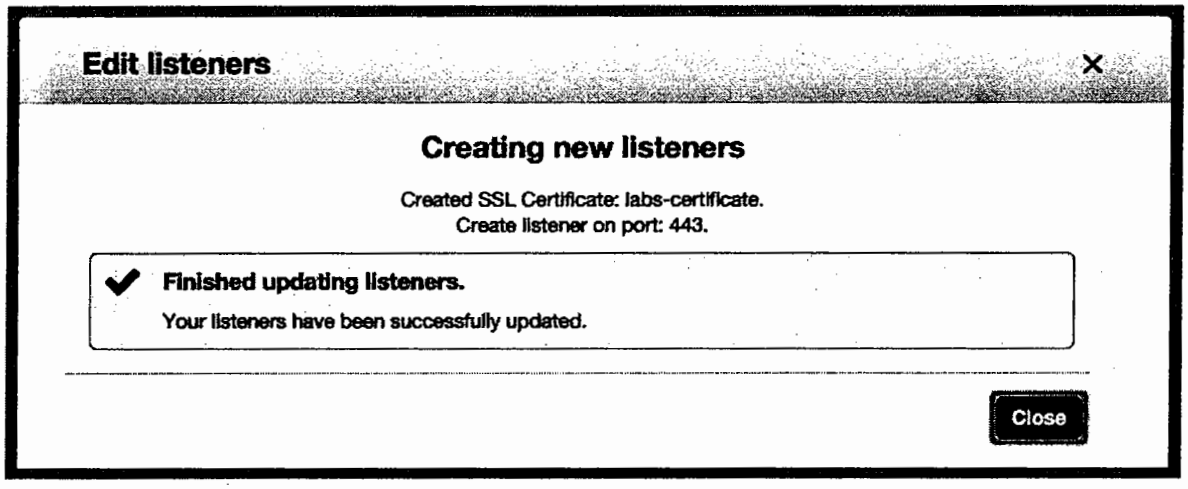
Choose a certificate name (e.g. labs-certificate) and then copy and paste both RSA Private and Public Certificate keys.



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



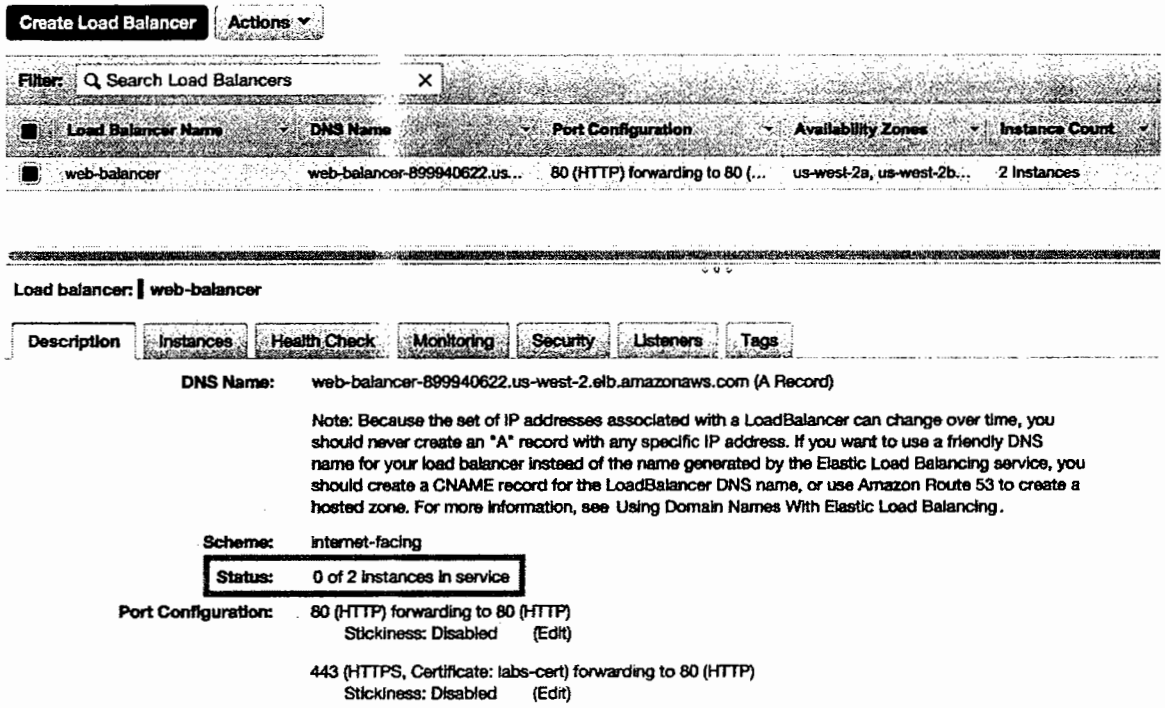
Click **Save** again and your load balancer will start serving content also using the HTTPs protocol.



STEP 5: Check if the Load Balancer is working properly

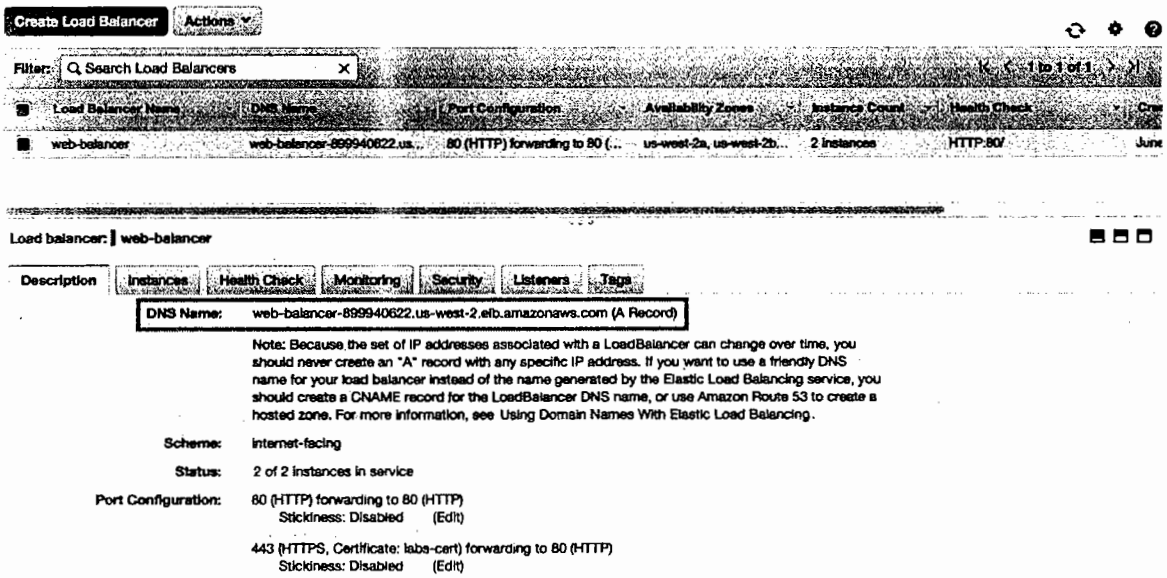
If you successfully create a Load Balancer, you should be able to see it in the Load Balancers listing page, even if it's still not working because the EC2 instances are not in service.

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.

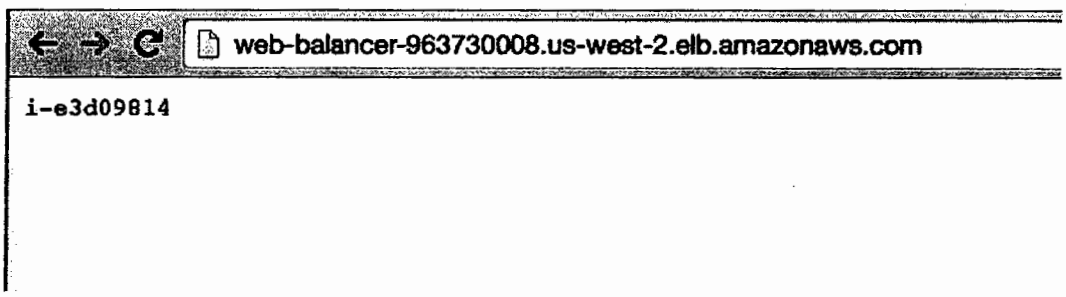


Your ELB instance is now ready and accessible from the web, so you can open the sample website hosted on backend instances.

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.



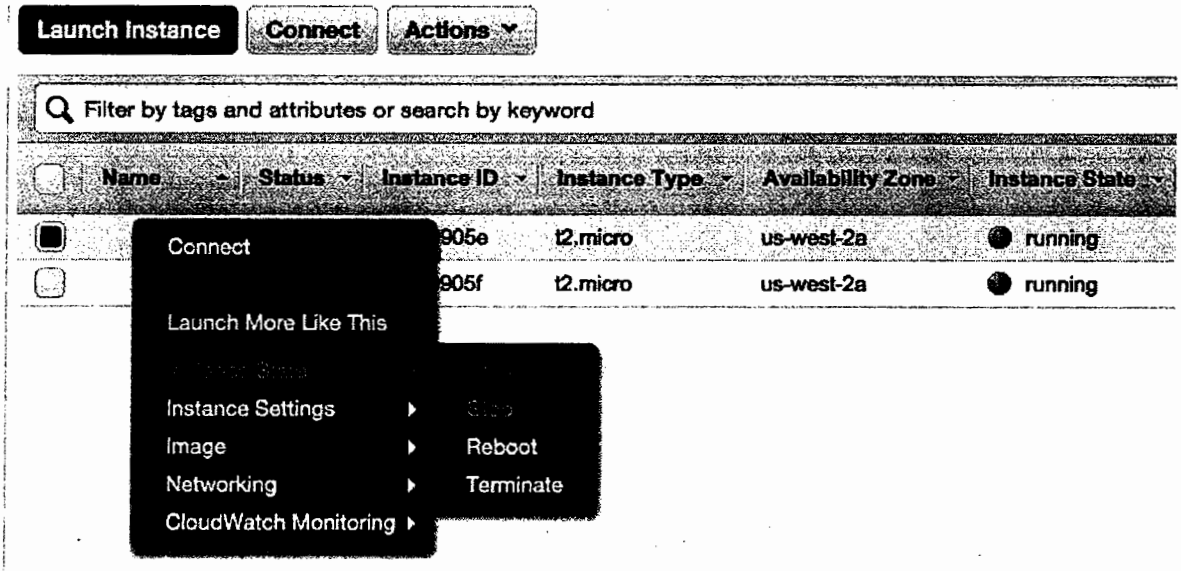
By opening the endpoint URL page, you will see a white page with the instance-ID of a specific backend node. Keep refreshing the page and you will see different instance IDs.



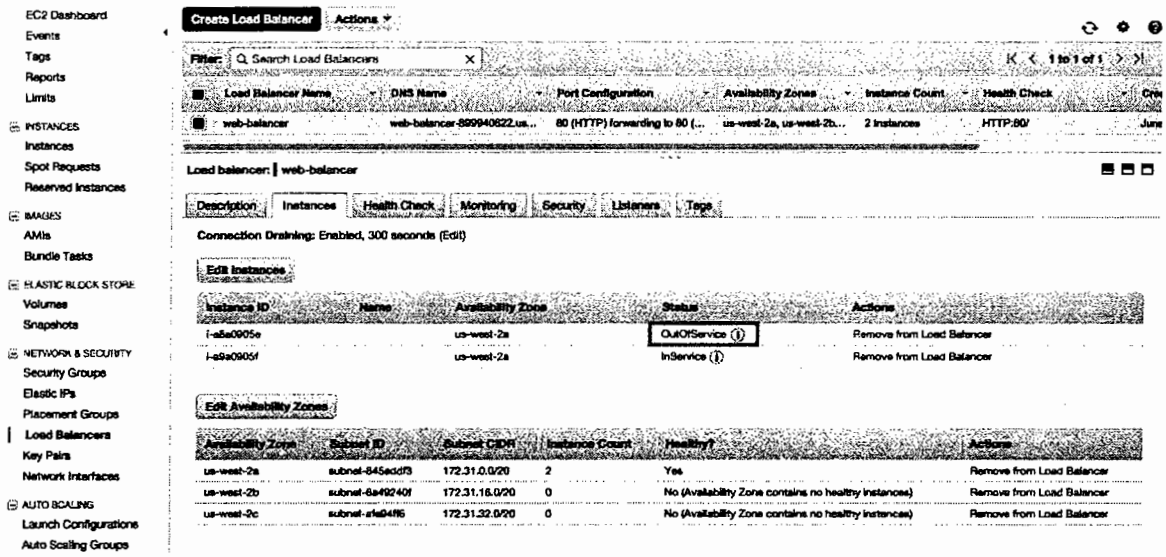
STEP 6: Check the ELB behavior during an instance failure

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

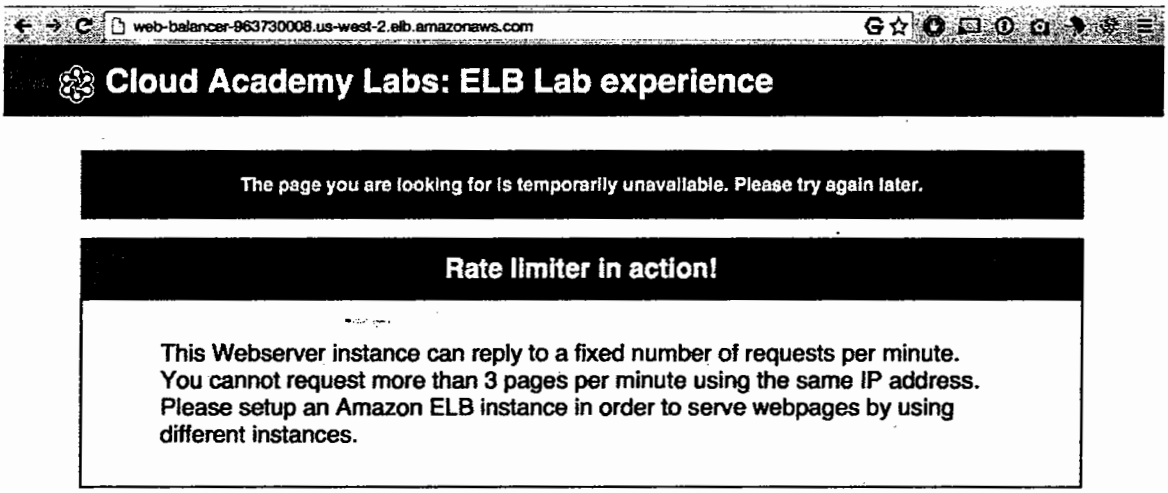
Click on **Instances** to open the EC2 instances dashboard. Select one of your instances by checking the instance ID, right-click on it and then click on the **Stop** action.



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.



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.

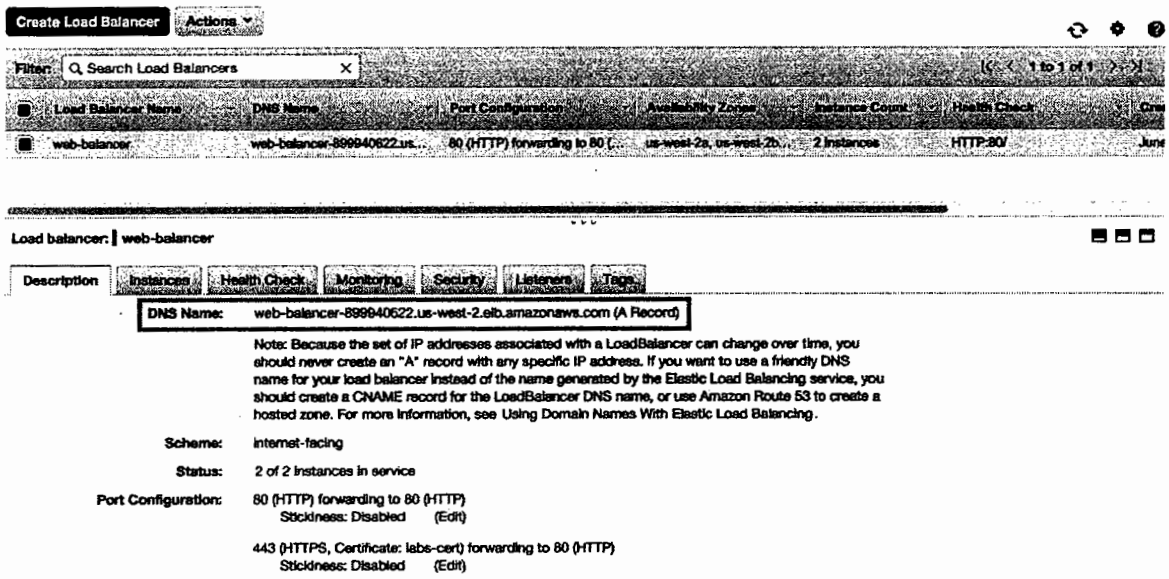


STEP 7: Check the ELB behavior after a successful node recovery

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 previosily stopped instance, right-click on it, and click on **Start**

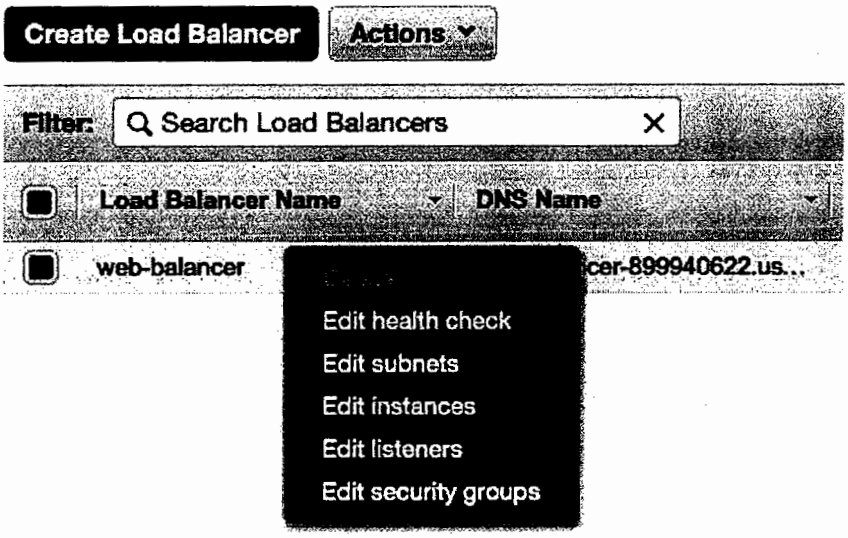
Amazon ELB constantly checks the backend instances status. After a few a minutes it will again start using both EC2 instances. You can check this by opening the ELB details pane.



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**.



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