WS ELB | 3 Major Types of Load Balancer in AWS

## What is AWS ELB?

**AWS ELB (Amazon Elastic Load Balance)** helps to distribute the application traffic to various different targets such as EC2 instances. The vacant targets which are ready to collect the traffic are monitored by Amazon ELB whether they are healthy or not and the traffic is sent to the healthy one.

AWS ELB comes in three versions which perform different tasks.

* The Version 1 provides detailed instructions for using *Classic Load Balancers*.
* 2nd version provides detailed instructions for using *Application Load Balancers*.
* 3rd provides detailed instructions for using *Network Load Balancers*.

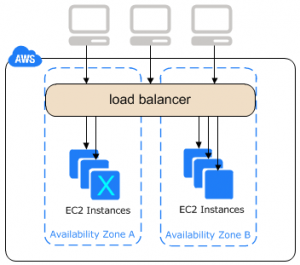
## ypes of Load Balancer in AWS

These are the unique Versions of AWS ELB, let’s discuss them one by one:

### a. Classic Load Balancers

Classic Load Balancers distribute upcoming traffic to different EC2 instances in multiple Availability Zones. During this process, there is a chance of the fault tolerance of your application. These Load Balancers detect healthy and unhealthy instance and direct the traffic towards only healthy ones. It also helps in a way such that without disrupting the flow of requests to your application you can add or remove instances from your load balancers as your need changes.

AWS ELB can calculate the majority of workloads automatically. Protocol and port which a person configures are used to detect the connection requests from clients it also forwards requests to one or more registered instances. The number of instances can be modified. Health checks can be monitored so that the load balancer only sends requests to the healthy instances.

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2018/07/Classic-Load-balancers.png)

*Learn AWS ELB – Classic Load Balancers*

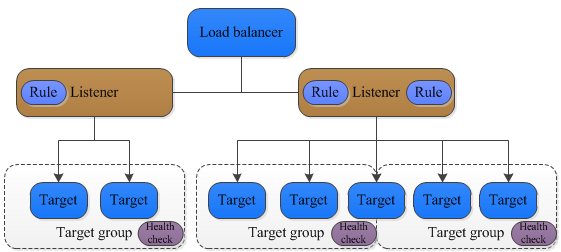
**Benefits of Load Balancers-**

* Provides Support to TCP and SSL listeners.
* Support to Sticky Session.
* Support to EC2- Classic.

### b. Application Load Balancers

After receiving the request Application Load Balancer analyzes the rules provide by the listener in priority order and determines the rule which has to apply. After that, it selects a target from the target group for the rule action. An Application Load Balancer functions at the Open Systems Interconnection (OSI) model which is the seventh layer of the OSI model.

The User can analyze the rules of the listener and can modify it by sending it to different target groups based on the content of the application traffic even when the target is associated with multiple target groups. Addition and removal of tags can do from the load balancers as per your needs. This can done without breaking the flow of your requests of the application.

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2018/07/APplication-Load-Balancer.png)

*AWS ELB – Application Load Balancers*

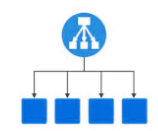
**Benefits of Application Load balancers-**

* Load Balancer’s performance improve in Application Load Balancer.
* Access logs containing information compress such that they may not require the additional space.
* Provides benefit for registering targets by IP address, including targets outside the VPC for the load balancer

### c. Network Load Balancers

It is the fourth layer of the Open System Interconnection Model. After the load balancer receives a connection request, it selects a target from the group which targets for the default rule.  
After enabling the availability zone Elastic Load Balancer creates the load balancer node in the availability zone. Each load balancer node automatically distributes traffic across the registered targets in its Availability Zone only.

Cross-zone Load Balancing enables to distribute traffic across the registered targets in all enabled Availability Zones. Enabling Multiple Availability Zone can cause harm by increasing the fault tolerance of the applications and it will happen if each target group has at least one target in each enabled Availability Zone. The problem can overcome in such a way that if one or more target groups do not have a healthy target in an Availability Zone, the IP address for the corresponding subnet from DNS is removed. If a person attempts again the request fails.

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2018/07/network-load-balancer.png)

*AWS ELB – NLB*

**Benefits of Network Load Balancers-**

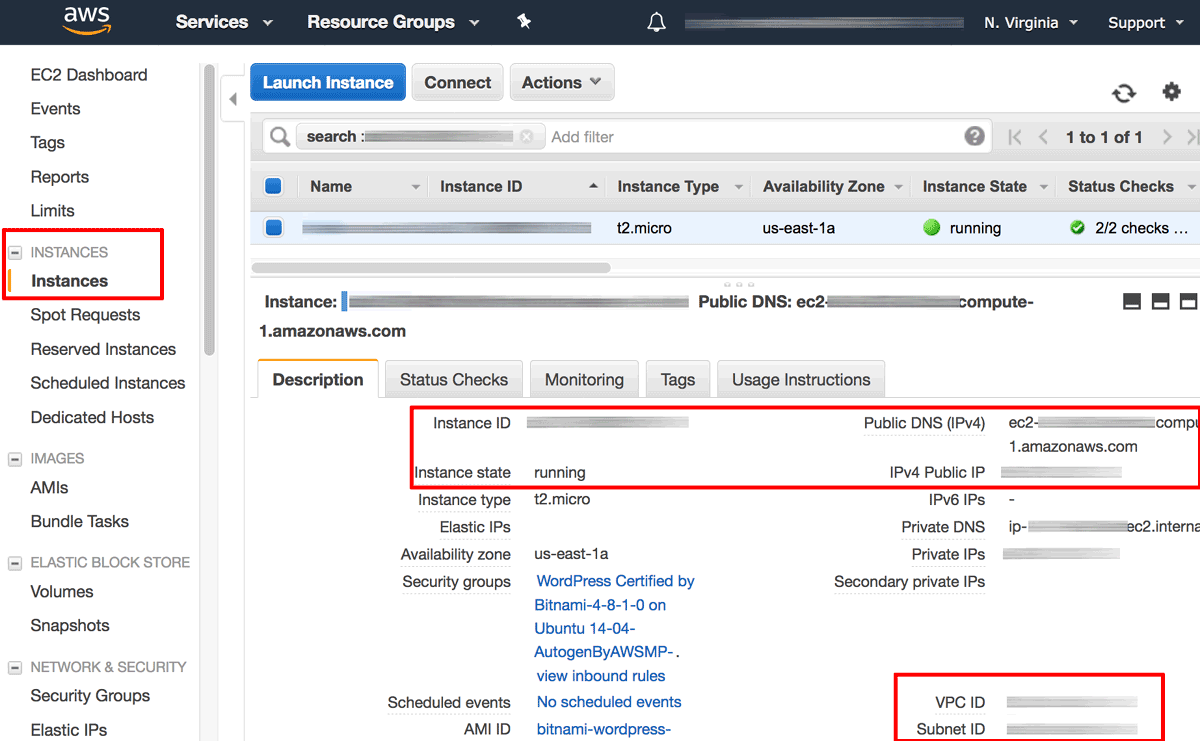
* NLB Provides the Support for static IP addresses for the load balancer.
* Provides support for registering targets by IP address which includes target outside the VPC for the Load Balancer.
* Provides support for monitoring the health of each service independently.

# Configure Elastic Load Balancing With SSL And AWS Certificate Manager For Bitnami Applications On AWS

### tep 1: Identify Your Bitnami Application Instance In The AWS EC2 Console

The first step is to identify your Bitnami application instance and collect various important bits of information, such as the instance ID, DNS name, public IP address, availability zone and Virtual Private Cloud (VPC) ID. To do this:

* Log in to the [AWS EC2 console](https://console.aws.amazon.com/ec2/).
* From the “Services” menu, select the “EC2” service.
* From the EC2 Dashboard, select the “Instances -> Instances” menu item.
* Search for and select your Bitnami application instance from the list of available instances.
* From the instance details pane, note the instance ID, DNS name, public IP address, availability zone and VPC ID. You will need these details in subsequent steps.

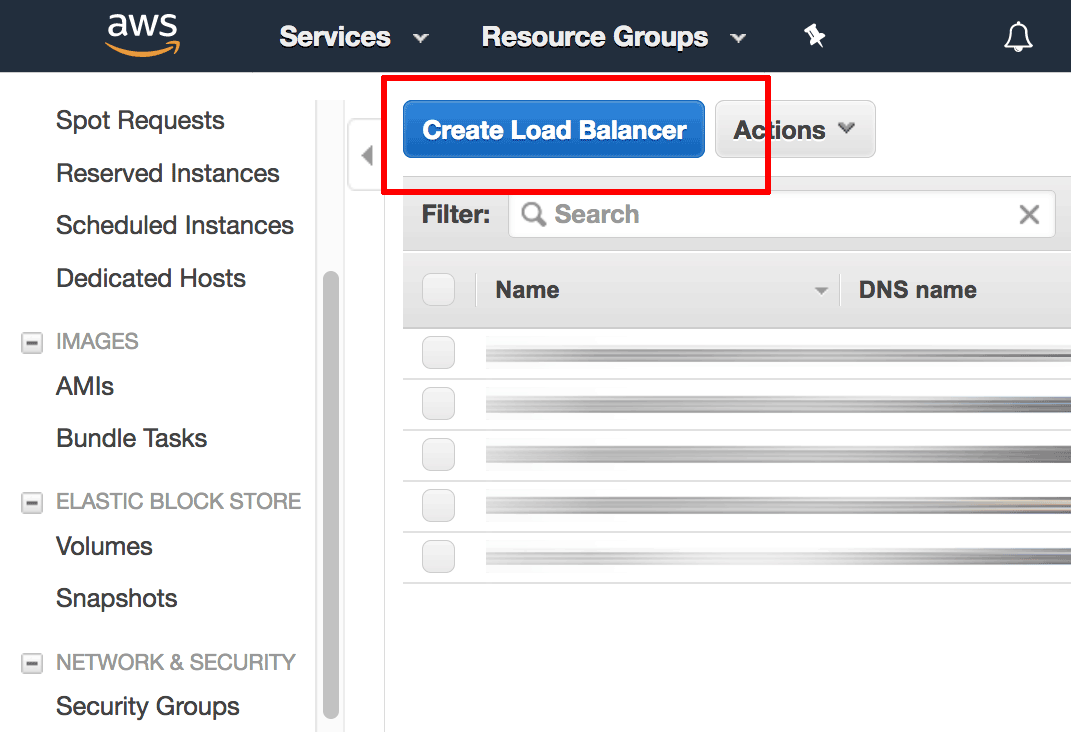
[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-1.png)

* Browse to the instance’s public IP address and confirm that you get a positive response, such as the welcome page of the Bitnami application. This response will be necessary to pass health checks performed by the load balancer.

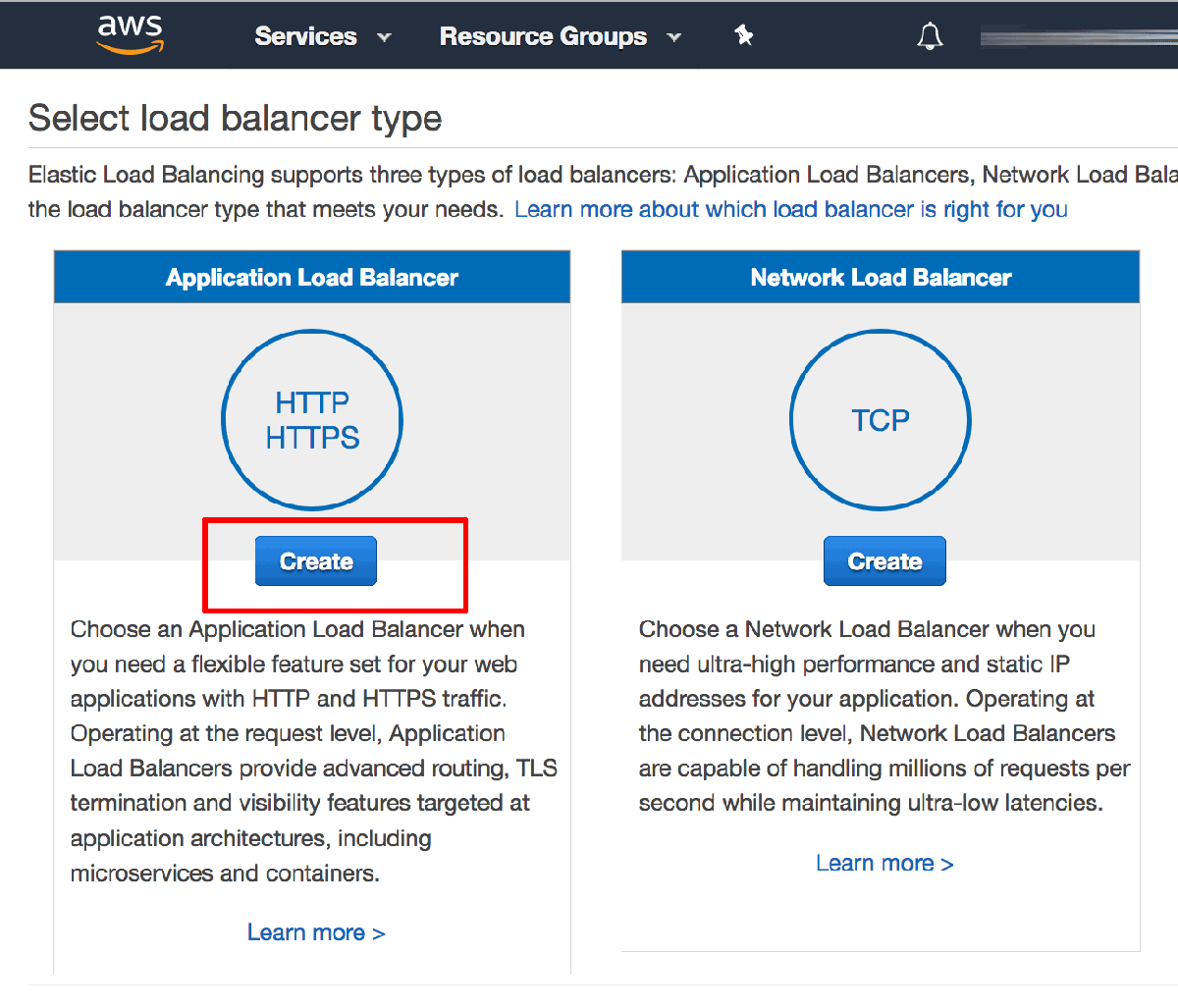
### Step 2: Create An Elastic Load Balancer With HTTP And HTTPS Support

Next, create an Elastic Load Balancer as follows:

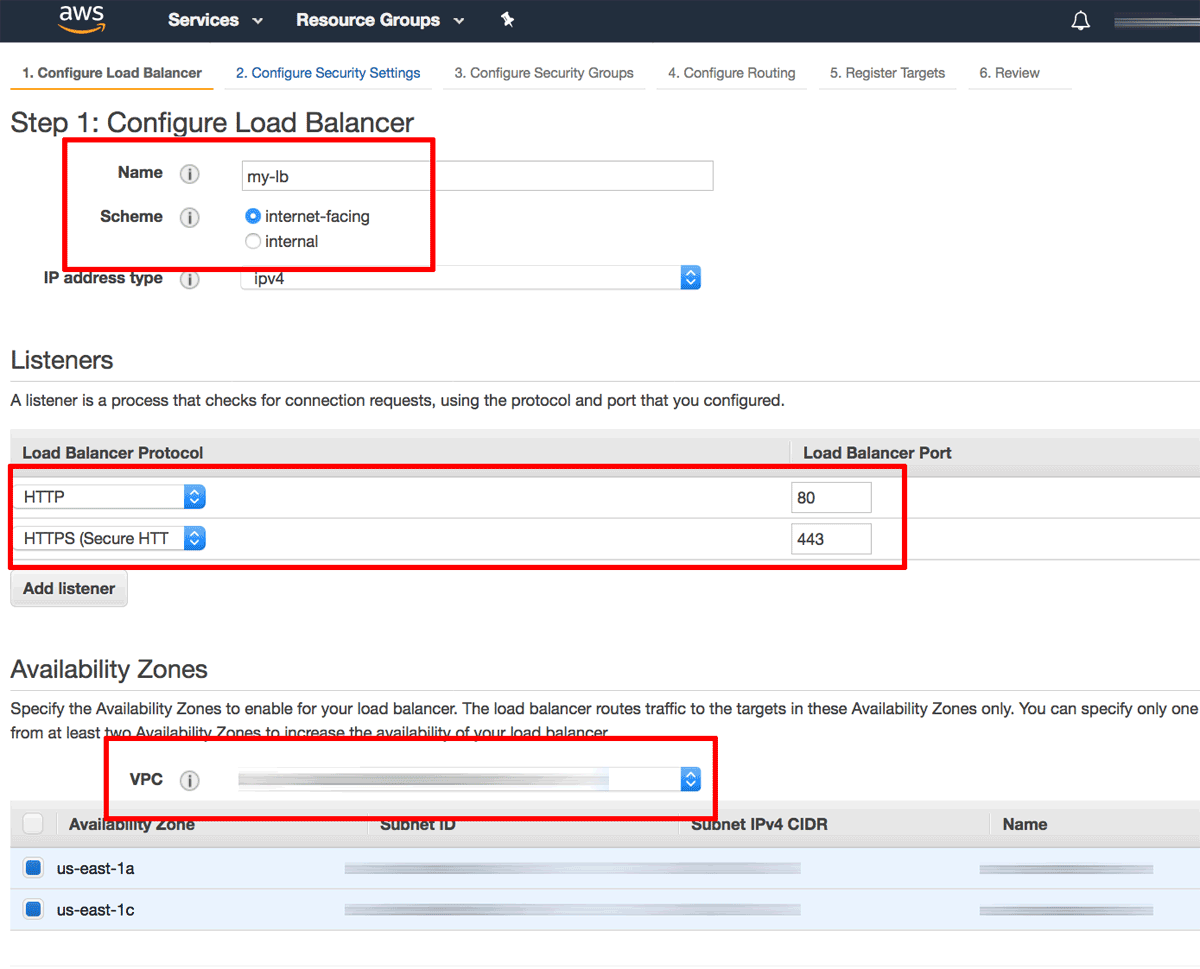
* From the EC2 Dashboard, select the “Load Balancing -> Load Balancers” menu item.
* Click the “Create Load Balancer” button.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-3.png)

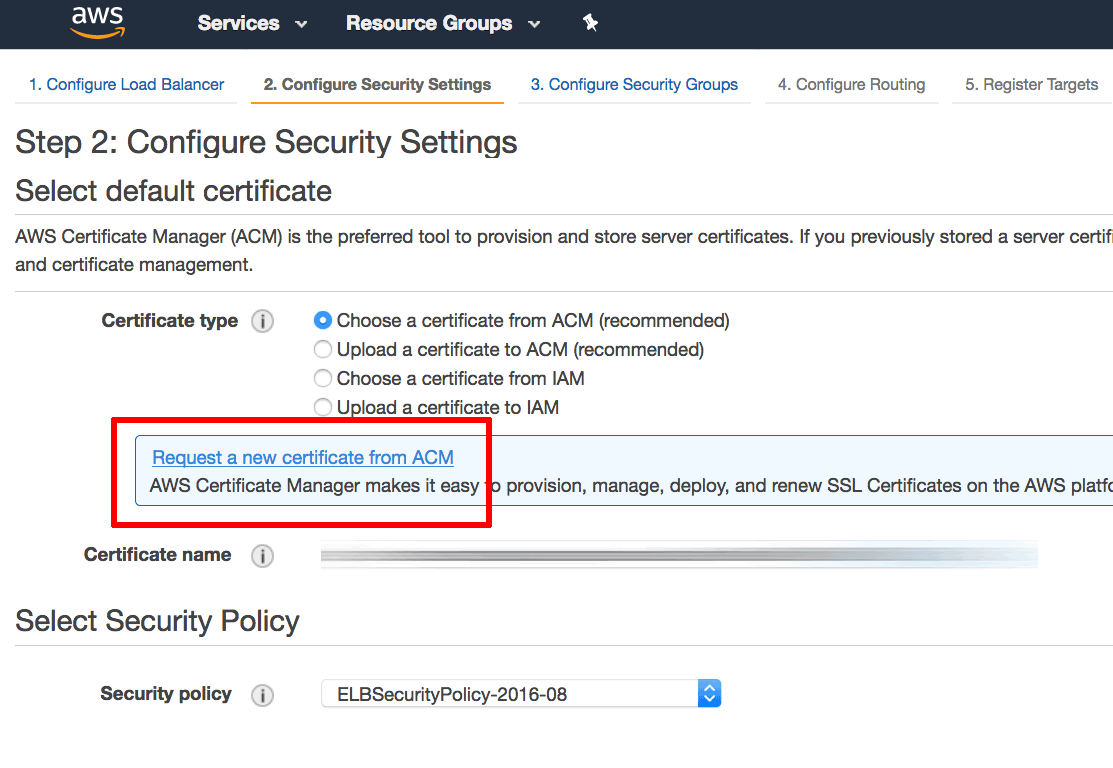
* On the “Select Load Balancer type” page, select the “Application Load Balancer” option and click “Create”.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-4.png)

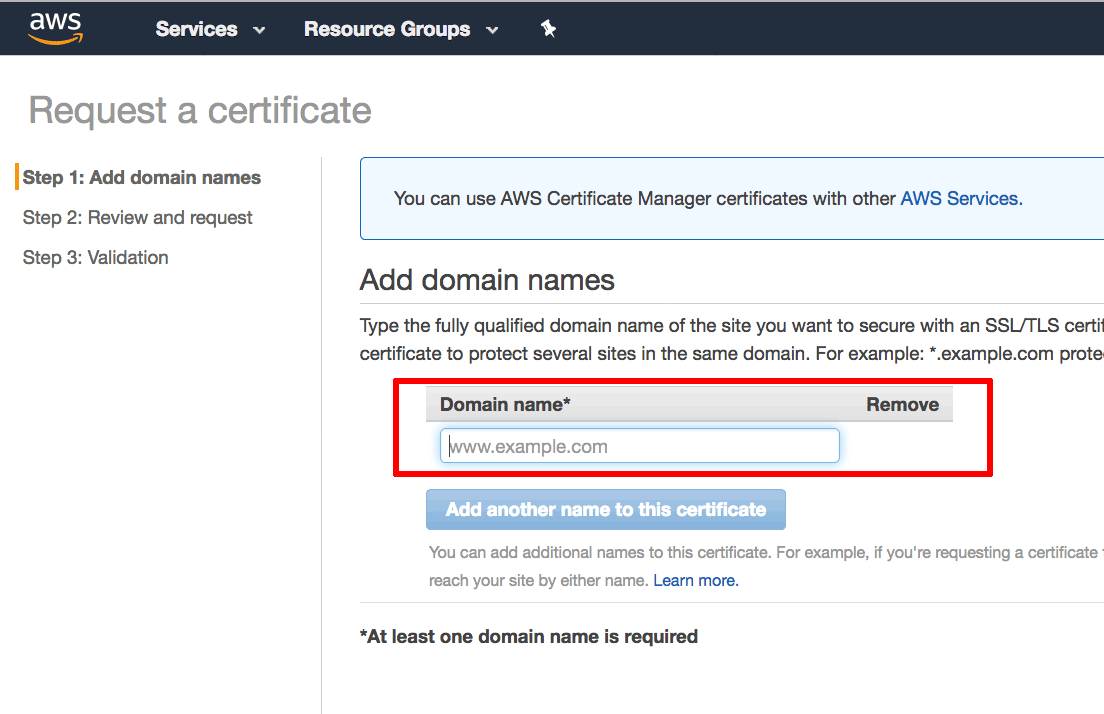
* On the subsequent “Configure Load Balancer” page:
  + Enter a name for the load balancer and specify the scheme as “Internet facing”.
  + In the “Listeners” section, ensure that there is an HTTP listener on load balancer port 80. Click the “Add listener” button and add a second HTTPS listener on load balancer port 443. This configures the load balancer to handle both HTTP and HTTPS requests from clients.
  + In the “Availability Zones” section, select the same VPC as the one used by your Bitnami application instance and select a subnet from each availability zone.
  + Click the “Next: Configure Security Settings” button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-5.png)

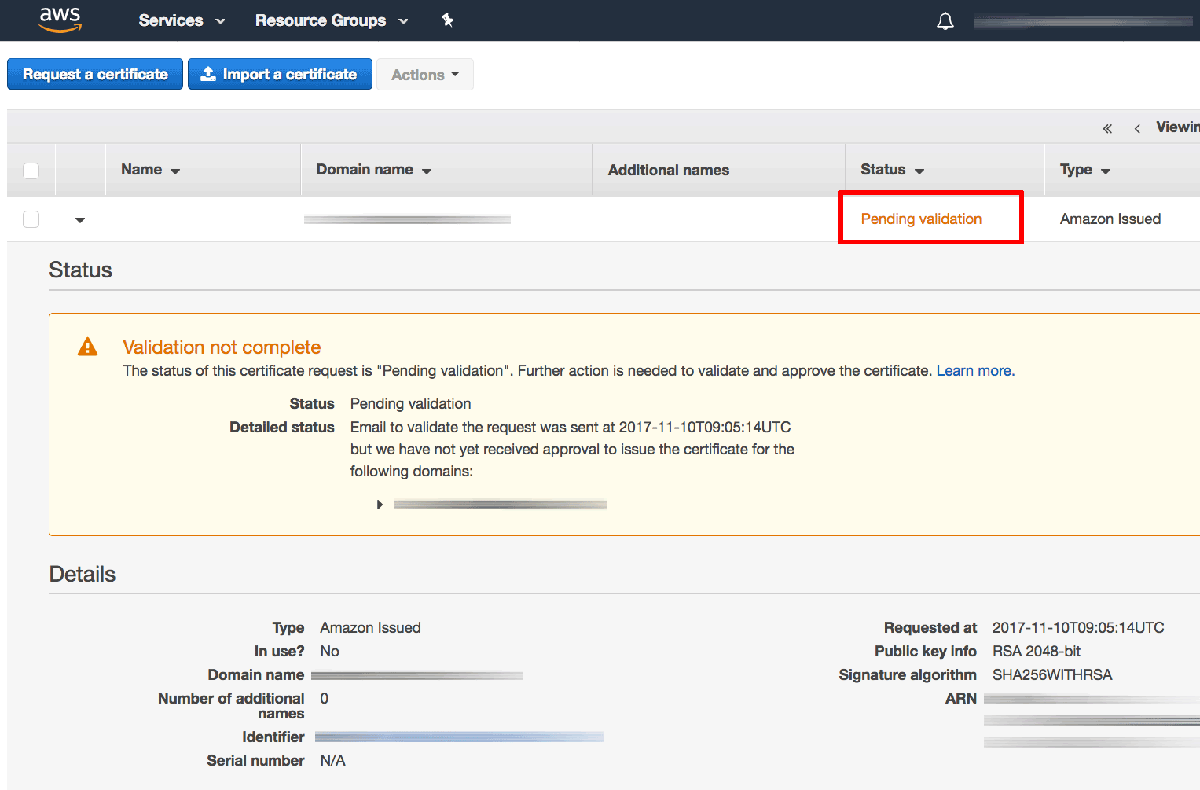
* On the “Configure Security Settings” page, select the “Request a new certificate from ACM” option to create a new SSL certificate for your load balancer. This will launch the [AWS Certificate Manager (ACM)](https://aws.amazon.com/certificate-manager/) in a new window.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-6.png)

* Within the AWS Certificate Manager, on the “Request a certificate” page, enter your domain name. Click “Review and request” to review and confirm the request.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-7.png)

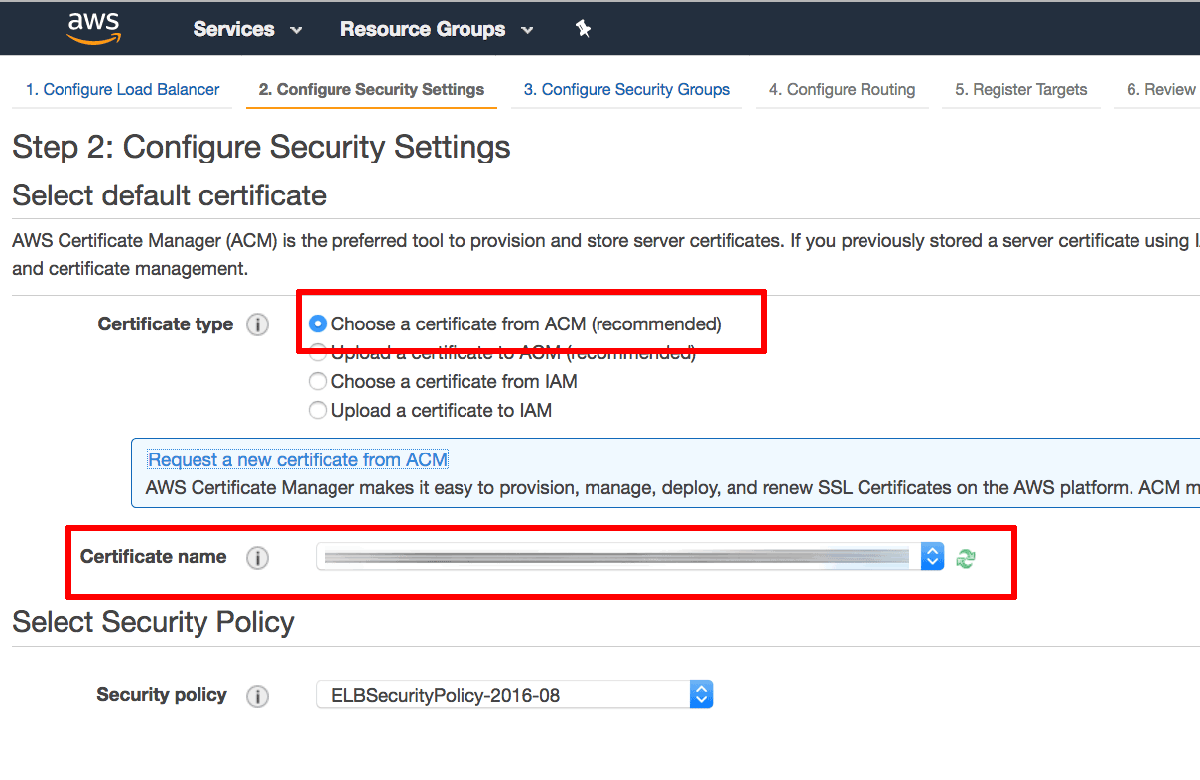
An email will now be sent to the registered owner of the domain with instructions to confirm the certificate request by validating the domain. The domain will appear in the AWS Certificate Manager with status set to “Pending validation”.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-8.png)

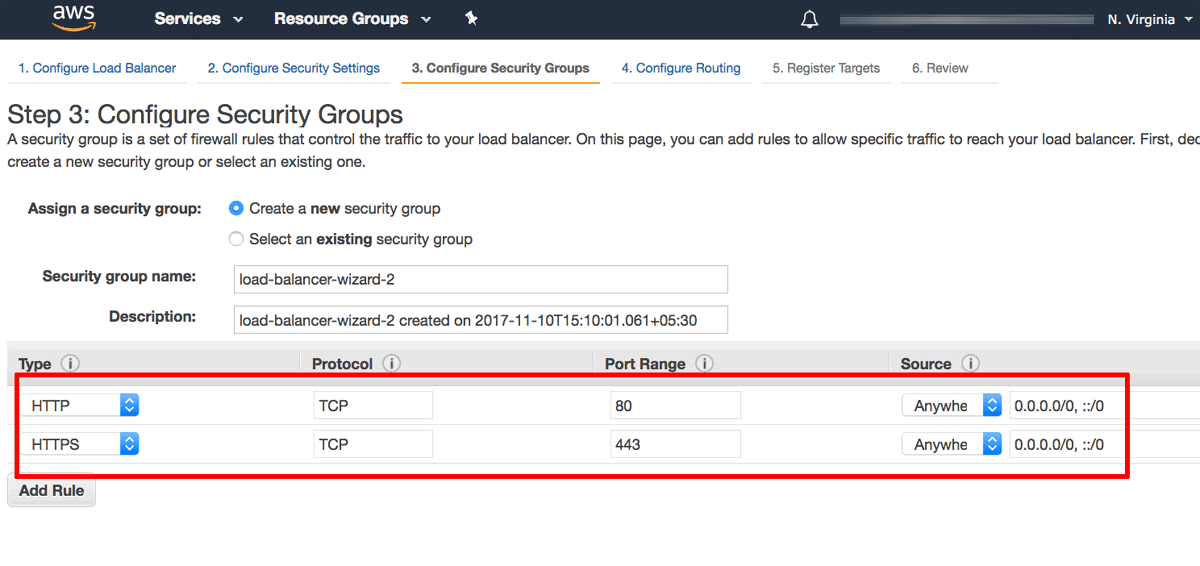
TIP: [Learn more about the AWS Certificate Manager domain validation process](https://aws.amazon.com/certificate-manager/faqs/).

Once the domain has been validated, the certificate will be issued and will appear in the AWS Certificate Manager with status set to “Issued”.

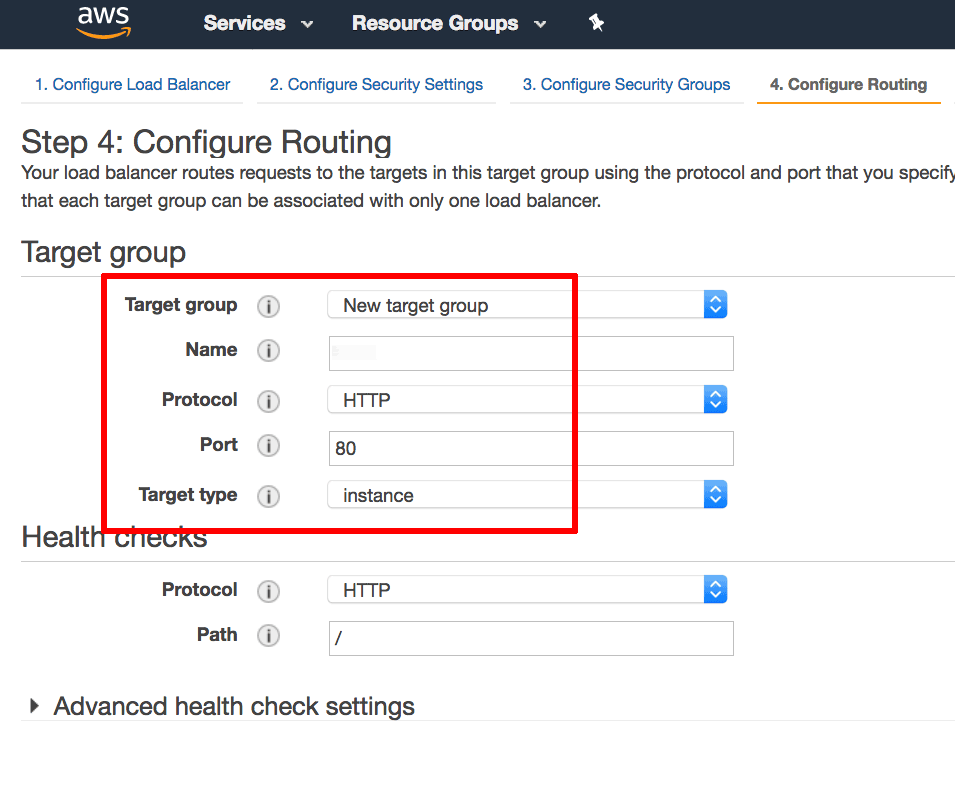
* Back on the “Configure Security Settings” page for the load balancer, the newly-issued certificate should now appear in the list of available certificates. Select it and click the “Next: Configure Security Group” button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-9.png)

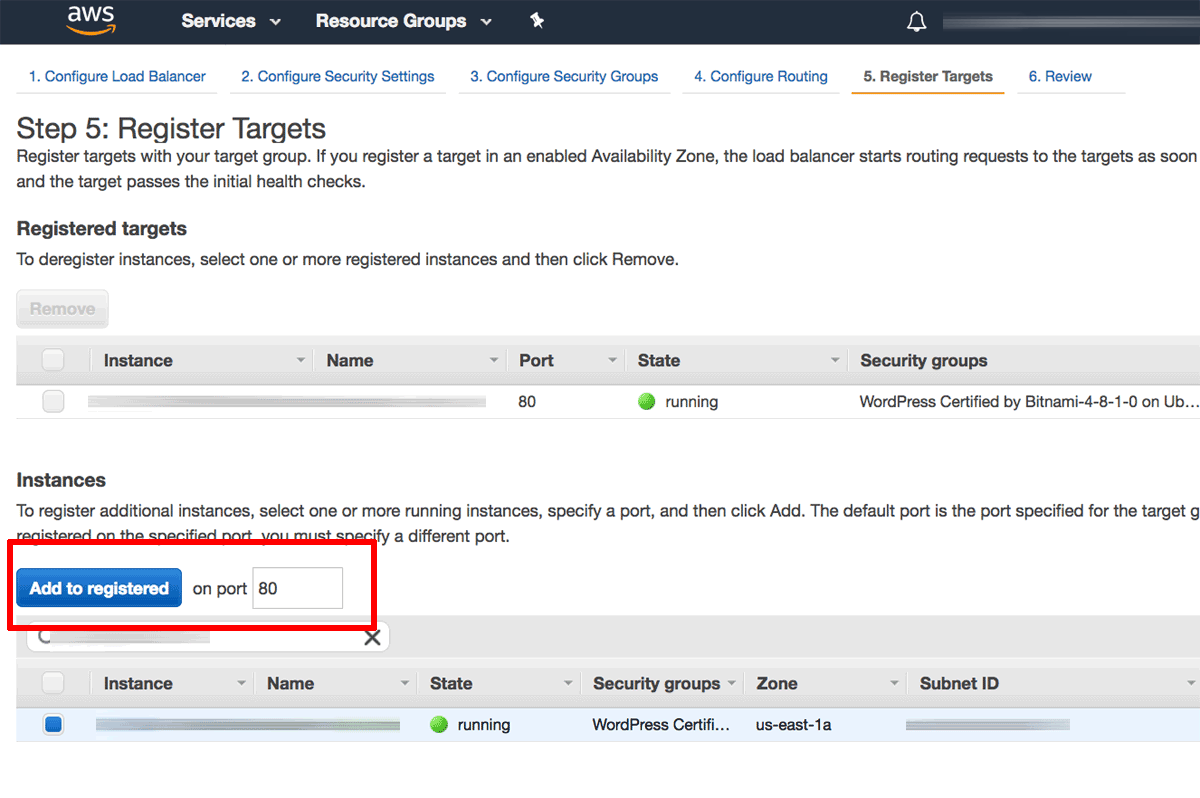
* On the “Configure Security Group” page:
  + Select the option to “Create a new security group”.
  + Add a security rule to allow inbound traffic on port 80 (the HTTP port) with source “Anywhere”.
  + Add a second security rule to allow inbound traffic on port 443 (the HTTPS port) with source “Anywhere”.
  + Click the “Next: Configure Routing” button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-10.png)

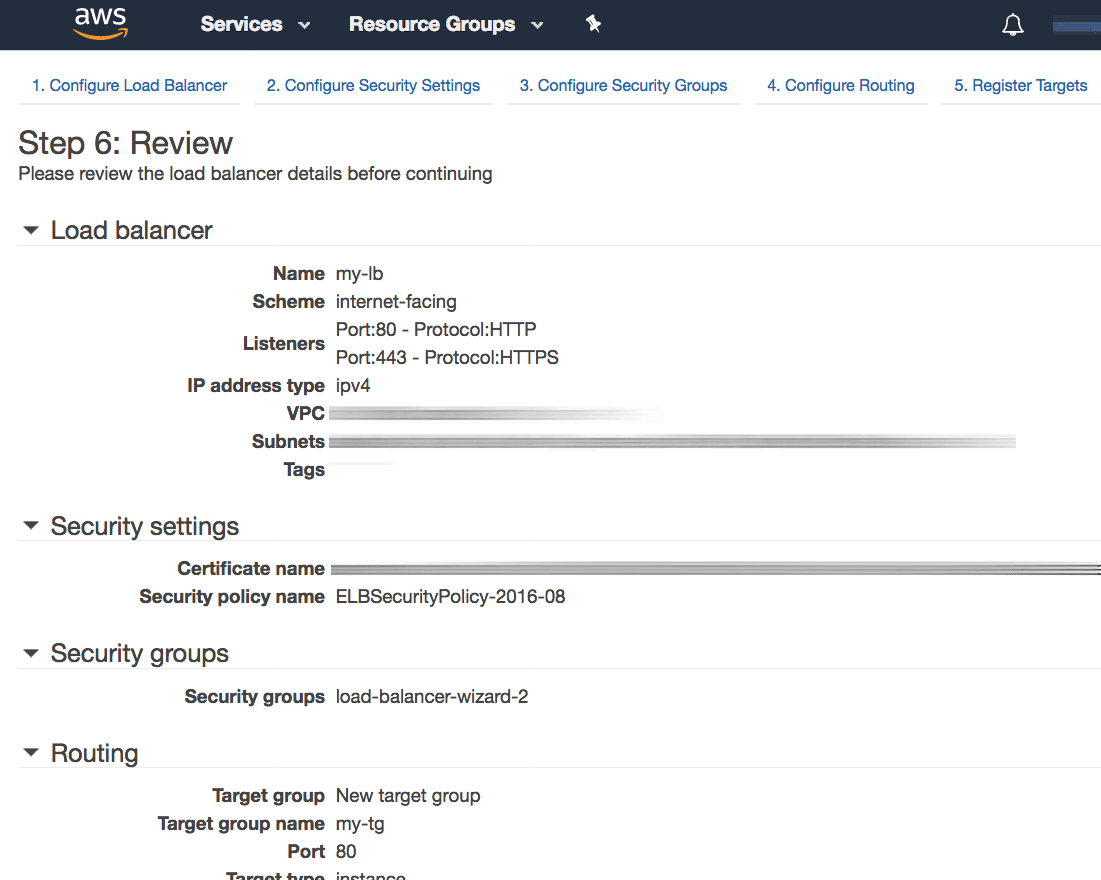
* On the “Configure Routing” page:
  + In the “Target group” section, create a new target group and assign it a name. Ensure that the protocol is set to “HTTP”, the port to “80” and the target type to “instance”. With this configuration, traffic between the load balancer and the instance will be transmitted using HTTP, even for HTTPS requests made by the client to the load balancer.
  + In the “Health checks” section, define the protocol as “HTTP” and the path to “/”.
  + Click the “Next: Register Targets” button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-11.png)

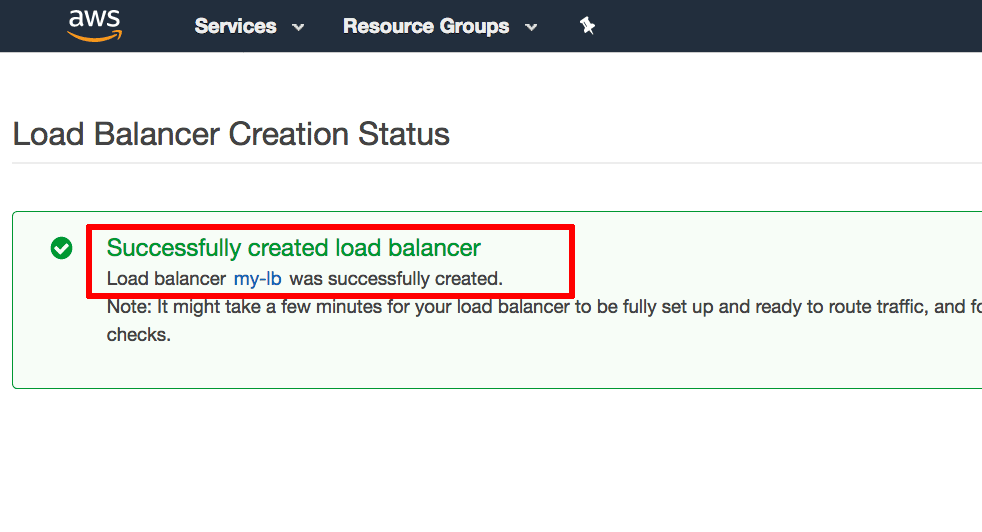
* On the “Register Targets” page, use the instance ID obtained in Step 1 to identify and select the Bitnami application instance. Click the “Add to registered” button to move the instance into the list of registered targets. Click the “Next: Review” button to proceed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-12.png)

* On the “Review” page, review the details of the configured load balancer. Click “Create” to confirm the configuration and deploy the load balancer.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-13.png)

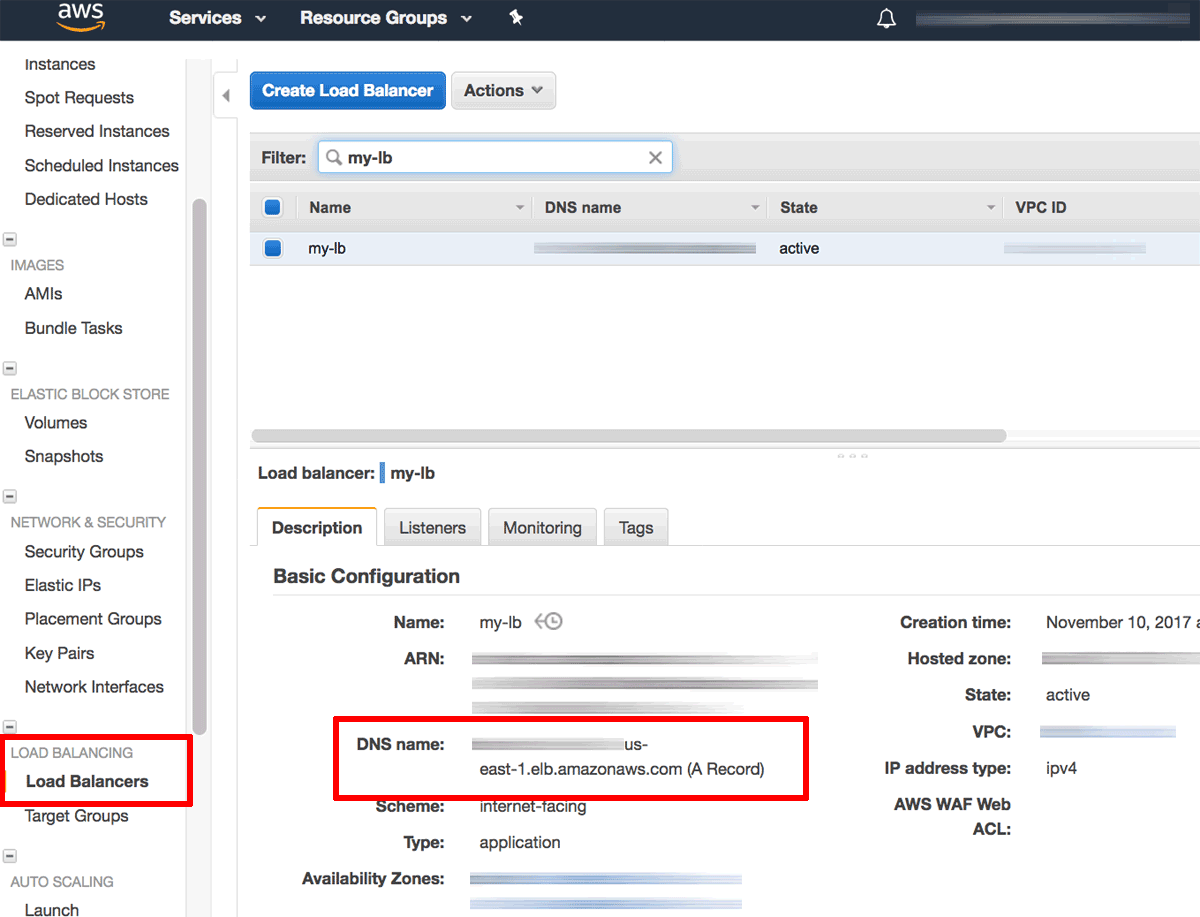
The load balancer will now be created and deployed.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-14.png)

### Step 3: Point Your Domain Name To The Load Balancer’s IP Address

Once the deployment is complete, perform the following steps:

* Visit the EC2 Dashboard and note the DNS name for the configured load balancer.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-15.png)

* Update your domain’s DNS settings by adding an A record that points to the public DNS name of the load balancer. To do this, you will usually need to log in to your domain name provider’s management console and make the necessary changes.

NOTE: Use the DNS name (and not the IP address) of the Elastic Load Balancer in the A record, because Elastic Load Balancer IP addresses are dynamic and can change without warning.

NOTE: Once you make the necessary changes, it can take up to 48 hours for the change to propagate across other DNS servers.

### Step 4: Modify The Web Server Configuration On The Bitnami Application Instance

The next step is to update the Apache server configuration on the Bitnami application instance to correctly handle SSL requests from the load balancer. To do this:

* [Log in to the application instance using SSH](https://docs.bitnami.com/aws/faq/get-started/connect-ssh/).
* Modify the /opt/bitnami/apps/APPNAME/conf/httpd-prefix.conf and add the following line at the top of the file. This line ensures correct operation of the HTTP connection between the load balancer and the instance, even for load-balanced HTTPS requests.
* SetEnvIf x-forwarded-proto https HTTPS=on

In some cases, it is also necessary to modify the Bitnami application configuration to automatically redirect all HTTP traffic to the HTTPS port. The procedure to do this varies per application: some applications may require you to manually edit a configuration file, others may provide an administration interface, and still others may not require any specific changes.

As an example, if you are using WordPress, you would need to edit the /opt/bitnami/apps/wordpress/htdocs/wp-config.php file and add the lines below before the WP\_HOME and WP\_SITEURL definitions:

if (strpos($\_SERVER['HTTP\_X\_FORWARDED\_PROTO'], 'https') !== false)

$\_SERVER['HTTPS']='on';

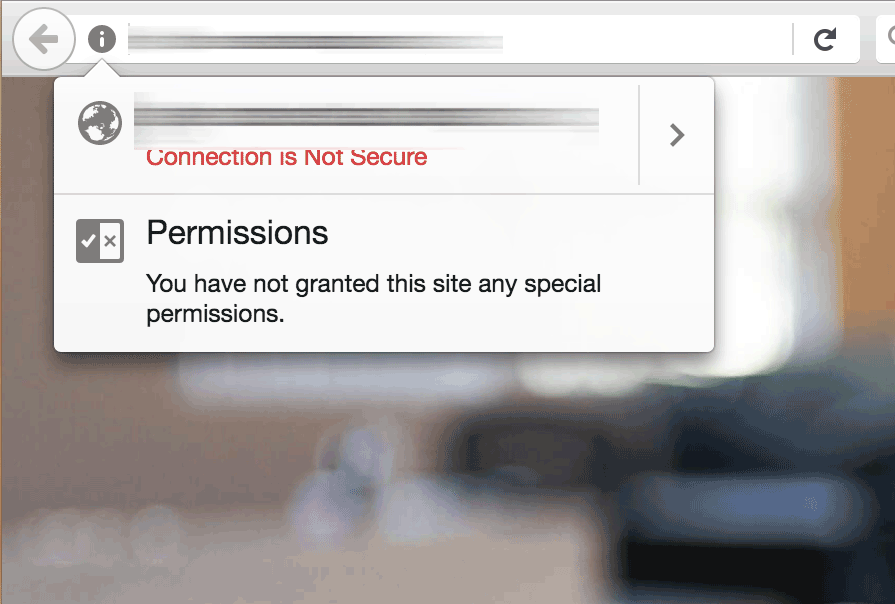
### Step 5: Force HTTPS

* Redirect all traffic to HTTPS by adding the following to /opt/bitnami/apache2/conf/bitnami/bitnami.conf
* <VirtualHost \*:80>
* ...
* RewriteEngine On
* RewriteCond %{HTTP:X-Forwarded-Proto} !https
* RewriteRule ^.\*$ https://%{SERVER\_NAME}%{REQUEST\_URI}
* ...
* Restart the server:
* sudo /opt/bitnami/ctlscript.sh restart apache

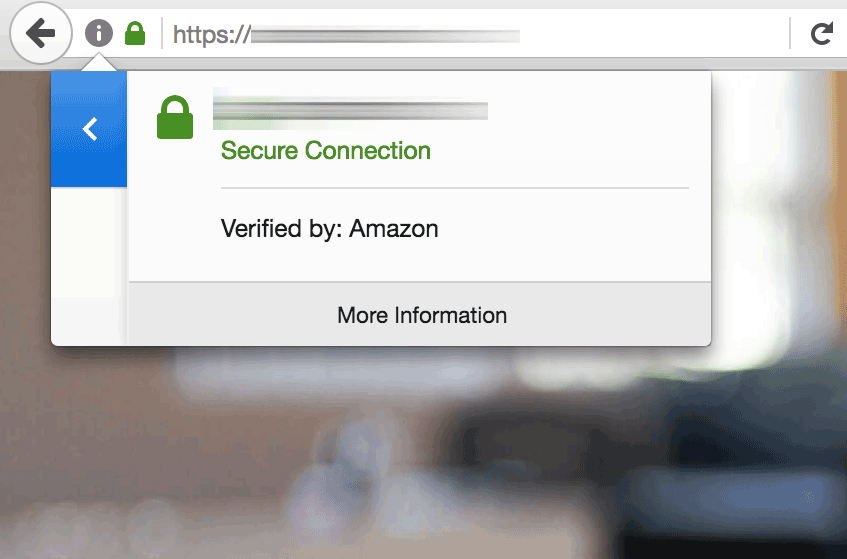
### Step 6: Test The Load Balancer

Verify that the changes to your domain name record have propagated by using the [Global DNS Propagation Checker](https://www.whatsmydns.net/) and entering your domain name into the search field. After confirming that the domain name now points to the public IP address of the load balancer, you can test it as follows (replace the DOMAIN placeholder with the correct domain name):

* Browsing to http://DOMAIN should result in the load balancer displaying the insecure welcome page of the Bitnami application.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-17.png)

* Browsing to https://DOMAIN should result in the load balancer displaying the secure welcome page of the Bitnami application. Clicking the padlock icon in the browser address bar should display the details of the domain and SSL certificate.

[](https://docs.bitnami.com/images/img/how_to_guides/configure-elb-ssl-aws/elb-aws-18.png)

In both cases, the request to the HTTP or HTTPS port of the load balancer is internally transferred to, and satisfied by, the Bitnami application instance.

Your Elastic Load Balancer has now been configured with an SSL certificate for your Bitnami application running on AWS.