

Load Balancer

Elastic Load Balancing supports the following types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. Amazon ECS services can use either type of load balancer. Application Load Balancers are used to route HTTP/HTTPS (or Layer 7) traffic. Network Load Balancers and Classic Load Balancers are used to route TCP (or Layer 4) traffic.

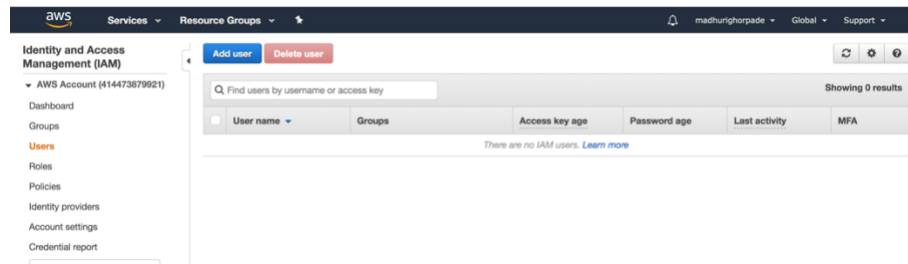
1. **Application Load Balancer:** An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each container instance in your cluster. Application Load Balancers support dynamic host port mapping. For example, if your task's container definition specifies port 80 for a NGINX container port, and port 0 for the host port, then the host port is dynamically chosen from the ephemeral port range of the container instance (such as 32768 to 61000 on the latest Amazon ECS-optimized AMI). When the task is launched, the NGINX container is registered with the Application Load Balancer as an instance ID and port combination, and traffic is distributed to the instance ID and port corresponding to that container. This dynamic mapping allows you to have multiple tasks from a single service on the same container instance.
2. **Network Load Balancer:** A Network Load Balancer makes routing decisions at the transport layer (TCP/SSL). It can handle millions of requests per second. After the load balancer receives a connection, it selects a target from the target group for the default rule using a flow hash routing algorithm. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration. It forwards the request without modifying the headers. Network Load Balancers support dynamic host port mapping. For example, if your task's container definition specifies port 80 for an NGINX container port, and port 0 for the host port, then the host port is dynamically chosen from the ephemeral port range of the container instance (such as 32768 to 61000 on the latest Amazon ECS-optimized AMI). When the task is launched, the NGINX container is registered with the Network Load Balancer as an instance ID and port combination, and traffic is distributed to the instance ID and port corresponding to that container. This dynamic mapping allows you to have multiple tasks from a single service on the same container instance.
3. **Classic Load Balancer:** A Classic Load Balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS). Classic Load Balancers currently require a fixed relationship between the load balancer port and the container instance port. For example, it is possible to map the load balancer port 80 to the container instance port 3030 and the load balancer port 4040 to the container instance port 4040. However, it is not possible to map the load balancer port 80 to port 3030 on one container instance and port 4040 on another container instance. This static mapping requires that your cluster has at least as many container instances as the desired count of a single service that uses a Classic Load Balancer.

Steps to create an application load balancer

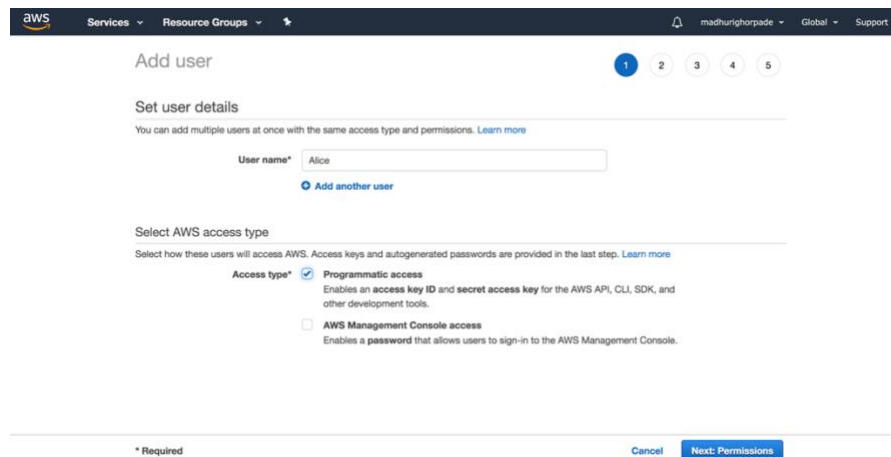
1. Create IAM user in AWS.

1.1. After login to AWS Management Console, navigate to the 'Services' menu and search for 'IAM' in the search bar.

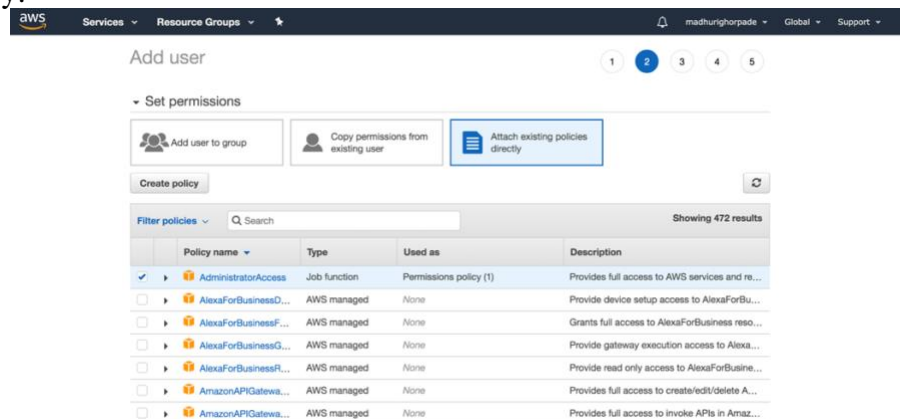
Go to Users and click on the 'Add user' button:



1.2. Provide a 'User name' and select 'Programmatic access'



1.3. Click on 'Attach existing policies directly' options and select 'AdministratorAccess' policy:



1.4. Adding tags is optional. Click 'Review' button

aws Services Resource Groups

madhurighorpade Global Support

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
Alice		X
Add new key		

You can add 49 more tags.

Cancel Previous **Next: Review**

1.5. Review and click on 'Create user' button:

aws Services Resource Groups

madhurighorpade Global Support

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Alice
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

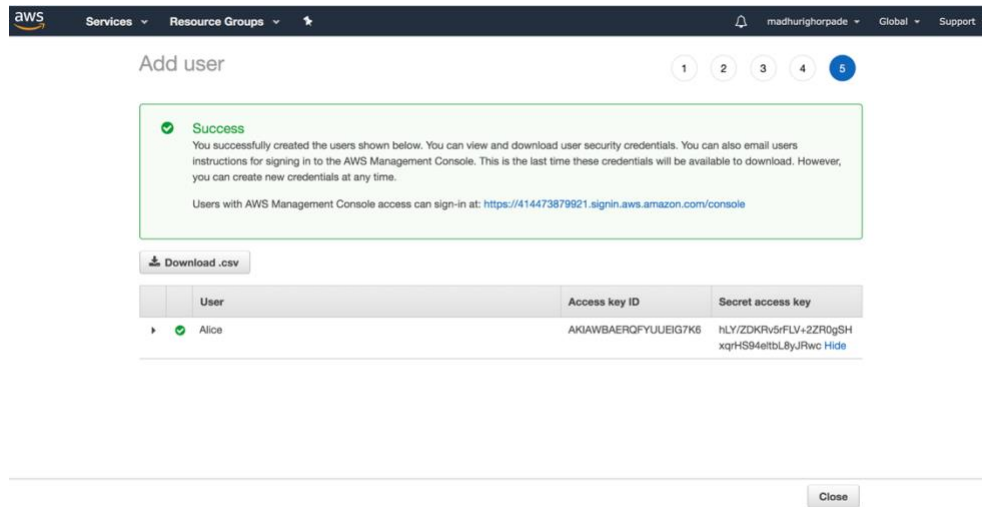
Tags

The new user will receive the following tag

Key	Value
-----	-------

Cancel Previous **Create user**

1.6. Once the user is created, make a note of 'Access key ID' and 'Secure access key'.
Alternatively, you can also download the 'Download.csv' which you will require later for configuring the AWS through CLI (Command Line Interface).



2. Install AWS CLI packages on your local machine.
If you do not have AWS CLI packages on your local machine, you can download using the link below:
For MAC OS: <https://docs.aws.amazon.com/cli/latest/userguide/install-macos.html>
For Windows OS: <https://docs.aws.amazon.com/cli/latest/userguide/install-windows.html>
3. Configure AWS on CLI using following steps:

aws configure

- ☐ Enter your own 'Access key ID' and 'Secret Access Key' (copied from step 1.6)
- ☐ Provide region name, I have specified 'us-east-1' but you can specify 'us-west-1'
- ☐ Leave output format as blank.

Below is the snippet:

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ aws configure
AWS Access Key ID [*****BWQG]: AKIAWBAERQFYW50F4SG2
AWS Secret Access Key [*****UBVq]: bLg/bcoGFHo55FzukLDZyZo0GR46Vj7PaJy0gQB
Default region name [us-east-1]: us-east-1
Default output format [None]:
(base) Madhuris-MacBook-Pro:mla madhurighorpade$
```

4. Follow the commands to create your own SSL certificate:
 - 4.1. Generate a private key. I have named it as 'mla_private.pem':

openssl genrsa -out mla_private.pem 2048

```
((base) Madhuris-MacBook-Pro:mla madhurighorpade$ openssl genrsa -out mla_private.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

- 4.2. Generate public key using above created private key (mla_private.pem). I have named it as 'mla_public.pem':

openssl rsa -in mla_private.pem -outform PEM -pubout -out mla_public.pem

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ openssl rsa -in mla_private.pem -outform PEM -pubout -out mla_public.pem
writing RSA key
```

- 4.3. Create a CSR (Certificate Signing Request) using above created private key (mla_private.pem). I have named it as 'mla_certificate.csr':

openssl req -new -key mla_private.pem -out mla_certificate.csr

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ openssl req -new -key mla_private.pem -out mla_certificate.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:SACRAMENTO
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CSUS
Organizational Unit Name (eg, section) []:Myorg
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

- 4.4. Verify if the private key, public key and certificate is created and placed in your local directory:

ls -la

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ ls -la
total 24
drwxr-xr-x  5 madhurighorpade  staff   160 Sep  8 20:49 .
drwx-----+ 283 madhurighorpade  staff  9056 Sep  8 20:43 ..
-rw-r--r--  1 madhurighorpade  staff   968 Sep  8 20:49 mla_certificate.csr
-rw-----  1 madhurighorpade  staff  1675 Sep  8 20:47 mla_private.pem
-rw-r--r--  1 madhurighorpade  staff   451 Sep  8 20:48 mla_public.pem
```

- 4.5. Create a Self-signed certificate using private key (mla_private.pem) and CSR(mla_certificate.csr) generated above . I have named it as 'mla_certificate.crt':

openssl x509 -req -days 365 -in mla_certificate.csr -signkey mla_private.pem -out mla_certificate.crt

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ openssl x509 -req -days 365 -in mla_certificate.csr -signkey mla_private.pem -out mla_certificate.crt
Signature ok
subject=C = US, ST = CA, L = SACRAMENTO, O = CSUS, OU = Myorg
Getting Private key
(base) Madhuris-MacBook-Pro:mla madhurighorpade$
```

5. Now the SSL certificate is created and needs to be uploaded to your AWS account using AWS IAM:

- 5.1. Provide a certificate name while uploading it. I have named it as 'mla_lb_ssl_cert'. You also need to provide the private key name (mla_private.pem) and self signed certificate name (mla_certificate.crt) created above:

aws iam upload-server-certificate --server-certificate-name mla_lb_ssl_cert --certificate-body file://mla_certificate.crt --private-key file://mla_private.pem

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ aws iam upload-server-certificate --server-certificate-name mla_lb_ssl_cert --certificate-body file://mla_certificate.crt --private-key file://mla_private.pem
{
  "ServerCertificateMetadata": {
    "Path": "/",
    "ServerCertificateName": "mla_lb_ssl_cert",
    "ServerCertificateId": "ASCAWBAERQFYXGK3PHUKG",
    "Arn": "arn:aws:iam::414473879921:server-certificate/mla_lb_ssl_cert",
    "UploadDate": "2019-09-09T03:55:03Z",
    "Expiration": "2020-09-08T03:50:57Z"
  }
}
(base) Madhuris-MacBook-Pro:mla madhurighorpade$
```

6. Verify if the certificate is uploaded:

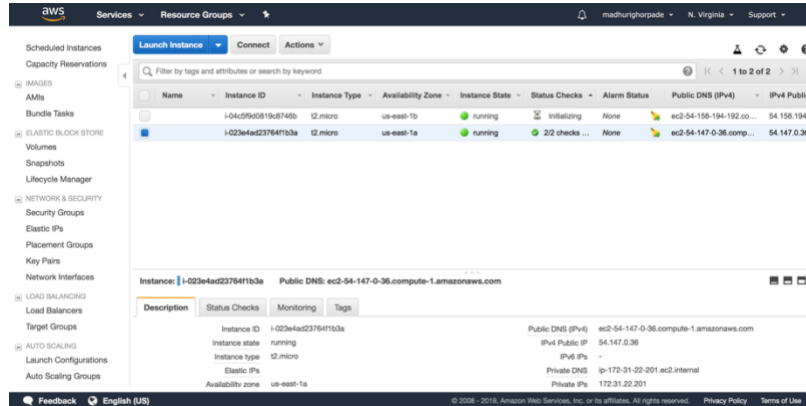
aws iam list-server-certificates

```
(base) Madhuris-MacBook-Pro:mla madhurighorpade$ aws iam list-server-certificates
{
  "ServerCertificateMetadataList": [
    {
      "Path": "/",
      "ServerCertificateName": "mla_lb_ssl_cert",
      "ServerCertificateId": "ASCAWBAERQFYXGK3PHUKG",
      "Arn": "arn:aws:iam::414473879921:server-certificate/mla_lb_ssl_cert",
      "UploadDate": "2019-09-09T03:55:03Z",
      "Expiration": "2020-09-08T03:50:57Z"
    }
  ],
}
```

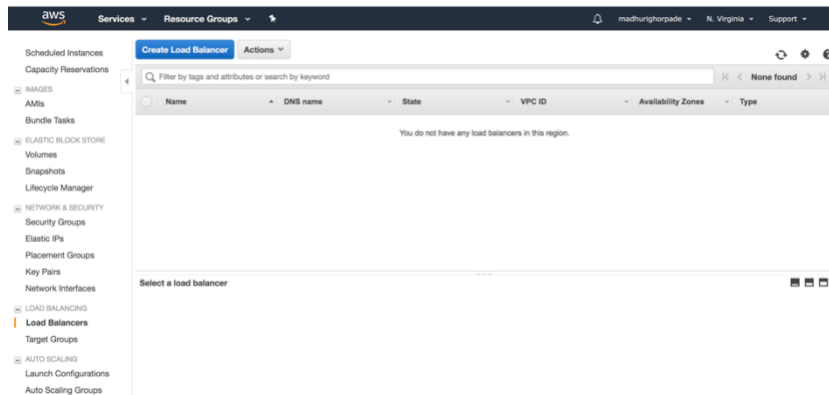
NOTE: Please create a replica of your EC2 instance in AWS (preferably in different availability zone than the previous EC2 instance) to utilize load balancer properties effectively.

- After the certificate is uploaded to your AWS account. Now you need to create a load balancer in AWS.

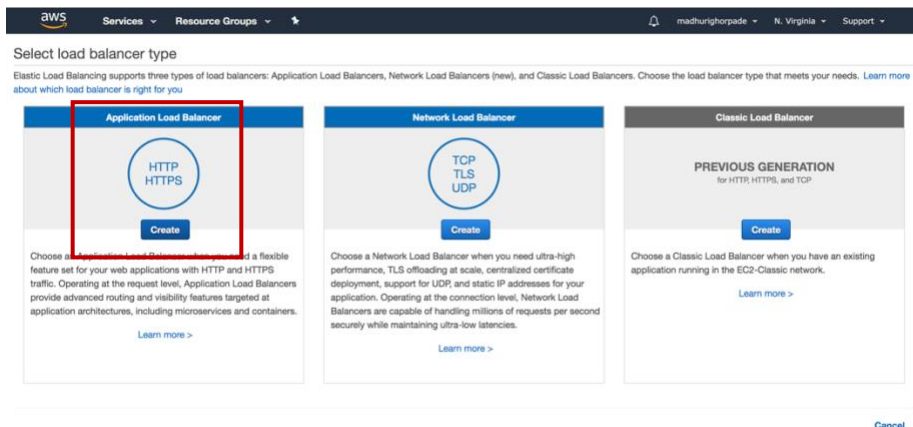
7.1. Navigate to EC2 services and locate 'Load Balancing' option from left column:



7.2. Click on 'Create Load balancer' button:



7.3. Select Application Load Balancer:



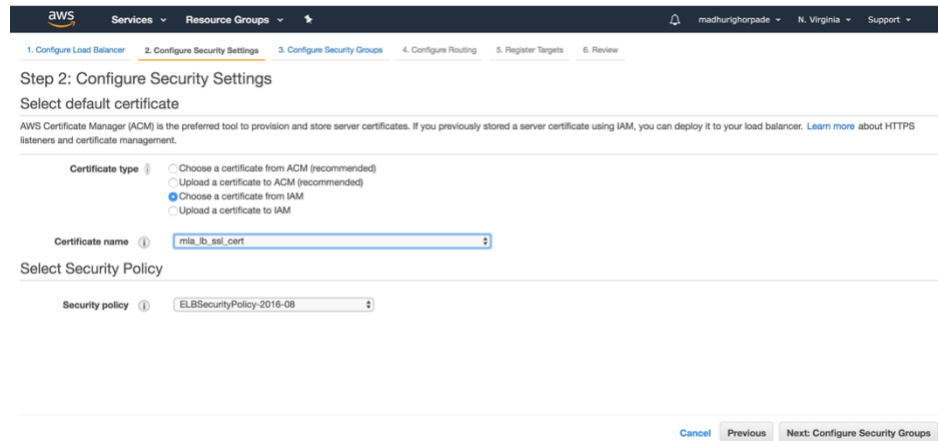
- 7.4. Provide a name to the newly created application load balancer. I have named it as ‘MLA-LB’. Select the Scheme as ‘internet-facing’ and add listener ‘HTTPS’ with port 443:

The screenshot shows the AWS Management Console interface for configuring a new Application Load Balancer. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The main content area is titled 'Step 1: Configure Load Balancer' and contains a 'Basic Configuration' section. In this section, the 'Name' field is set to 'MLA-LB', the 'Scheme' is set to 'internet-facing' (with 'internal' as an option), and the 'IP address type' is set to 'IPv4'. Below this is the 'Listeners' section, which includes a table with columns for 'Load Balancer Protocol' and 'Load Balancer Port'. A single listener is configured with 'HTTPS (Secure HTTP)' and port '443'. An 'Add listener' button is located below the table. At the bottom right, there are 'Cancel' and 'Next: Configure Security Settings' buttons.

- 7.5. Select the two availability zones for two different EC2 instances. I have my one EC2 instance in ‘us-east-1a’ and the other EC2 instance in ‘us-east-1b’ zone. You can select your own EC2 instance zones:

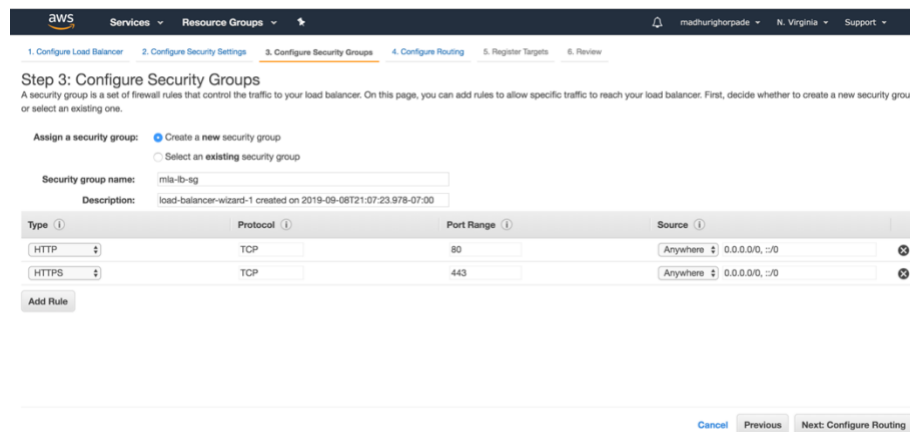
The screenshot shows the AWS Management Console interface for configuring a new Application Load Balancer, specifically the 'Availability Zones' section. The top navigation bar is the same as in the previous screenshot. The main content area is titled 'Step 1: Configure Load Balancer' and contains an 'Availability Zones' section. This section includes a description: 'Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.' Below this, the 'VPC' is set to 'vpc-30eca946 (172.31.0.0/16) (default)'. The 'Availability Zones' list shows two zones selected: 'us-east-1a' with subnet 'subnet-70ee163d' and 'us-east-1b' with subnet 'subnet-0512a59'. Other available zones like 'us-east-1c', 'us-east-1d', 'us-east-1e', and 'us-east-1f' are listed but not selected. At the bottom right, there are 'Cancel' and 'Next: Configure Security Settings' buttons.

- 7.6. Since a certificate is already created and uploaded to IAM, select Certificate type as ‘Choose a certificate from IAM’ and select the certificate name from drop down menu, that you have uploaded in step#6:



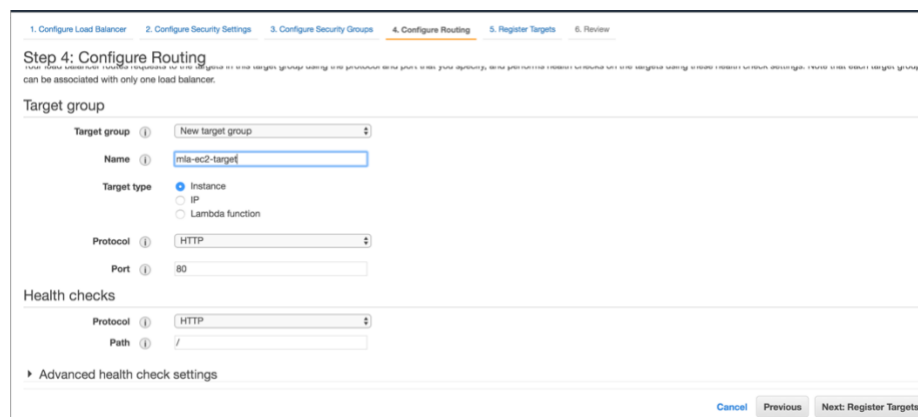
The screenshot shows the AWS console interface for Step 2: Configure Security Settings. The breadcrumb trail at the top indicates the sequence: 1. Configure Load Balancer, 2. Configure Security Settings (current), 3. Configure Security Groups, 4. Configure Routing, 5. Register Targets, and 6. Review. The main heading is "Step 2: Configure Security Settings" with the sub-heading "Select default certificate". Below this, a note states: "AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. [Learn more](#) about HTTPS listeners and certificate management." The "Certificate type" section has three radio buttons: "Choose a certificate from ACM (recommended)", "Upload a certificate to ACM (recommended)", and "Choose a certificate from IAM" (which is selected). Below this, the "Certificate name" dropdown menu is open, showing "mla_lb_ssl_cert" as the selected option. The "Select Security Policy" section shows "ELBSecurityPolicy-2016-08" selected in a dropdown. At the bottom right, there are three buttons: "Cancel", "Previous", and "Next: Configure Security Groups".

- 7.7. Click on ‘Create a new security group’ option. Provide a name for the security group, I have named it as ‘mla-lb-sg’. Add HTTP and HTTPS rules to it:



The screenshot shows the AWS console interface for Step 3: Configure Security Groups. The breadcrumb trail at the top indicates the sequence: 1. Configure Load Balancer, 2. Configure Security Settings, 3. Configure Security Groups (current), 4. Configure Routing, 5. Register Targets, and 6. Review. The main heading is "Step 3: Configure Security Groups" with the sub-heading "Assign a security group". Below this, a note states: "A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one." The "Assign a security group" section has two radio buttons: "Create a new security group" (which is selected) and "Select an existing security group". Below this, the "Security group name" text field contains "mla-lb-sg". The "Description" text field contains "load-balancer-wizard-1 created on 2019-09-08T21:07:23.978-07:00". Below this is a table with columns: Type, Protocol, Port Range, and Source. The table contains two rows: one for HTTP (Protocol: TCP, Port Range: 80, Source: Anywhere 0.0.0.0/0, ::/0) and one for HTTPS (Protocol: TCP, Port Range: 443, Source: Anywhere 0.0.0.0/0, ::/0). At the bottom left of the table is an "Add Rule" button. At the bottom right, there are three buttons: "Cancel", "Previous", and "Next: Configure Routing".

- 7.8. Create a new target group. I have named it as ‘mla-ec2-target’. Select target type as ‘Instance’. Leave the rest of the settings as it is:



The screenshot shows the AWS console interface for Step 4: Configure Routing. The breadcrumb trail at the top indicates the sequence: 1. Configure Load Balancer, 2. Configure Security Settings, 3. Configure Security Groups, 4. Configure Routing (current), 5. Register Targets, and 6. Review. The main heading is "Step 4: Configure Routing" with the sub-heading "Target group". Below this, a note states: "Your load balancer can route traffic to one or more target groups. Each target group can be associated with only one load balancer." The "Target group" section has a "Target group" dropdown menu with "New target group" selected. Below this, the "Name" text field contains "mla-ec2-target". The "Target type" section has three radio buttons: "Instance" (which is selected), "IP", and "Lambda function". Below this, the "Protocol" dropdown menu is set to "HTTP". The "Port" text field contains "80". Below this is the "Health checks" section, which has a "Protocol" dropdown menu set to "HTTP" and a "Path" text field containing "/". At the bottom left, there is a link "Advanced health check settings". At the bottom right, there are three buttons: "Cancel", "Previous", and "Next: Register Targets".

7.9. Select the two EC2 instances and add them to the register targets by clicking on ‘Add to registered’ button:

Step 5: Register Targets

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-023e4ad23764f1b3a		running	launch-wizard-1	us-east-1a	subnet-70ee1b3d	172.31.16.0/20
i-04c59d5819d3748b		running	launch-wizard-1, de...	us-east-1b	subnet-0512a5d9	172.31.32.0/20

Cancel Previous Next: Review

Step 5: Register Targets

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-023e4ad23764f1b3a		80	running	launch-wizard-1	us-east-1a
i-04c59d5819d3748b		80	running	launch-wizard-1, default	us-east-1b

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-023e4ad23764f1b3a		running	launch-wizard-1	us-east-1a	subnet-70ee1b3d	172.31.16.0/20
i-04c59d5819d3748b		running	launch-wizard-1, de...	us-east-1b	subnet-0512a5d9	172.31.32.0/20

Cancel Previous Next: Review

7.10. Review and click on ‘Create’ button:

Step 6: Review

Load balancer

- Name: MJA-LB
- Scheme: internet-facing
- Listeners: Port 443 - Protocol HTTP
- IP address type: ipv4
- VPC: vpc-30eca946
- Subnets: subnet-70ee1b3d, subnet-0512a5d9
- Tags:

Security settings

- Certificate name: arn:aws:iam::414473879921:server-certificate/mja_lb_ssl_cert
- Security policy name: ELBSecurityPolicy-2016-08

Security groups

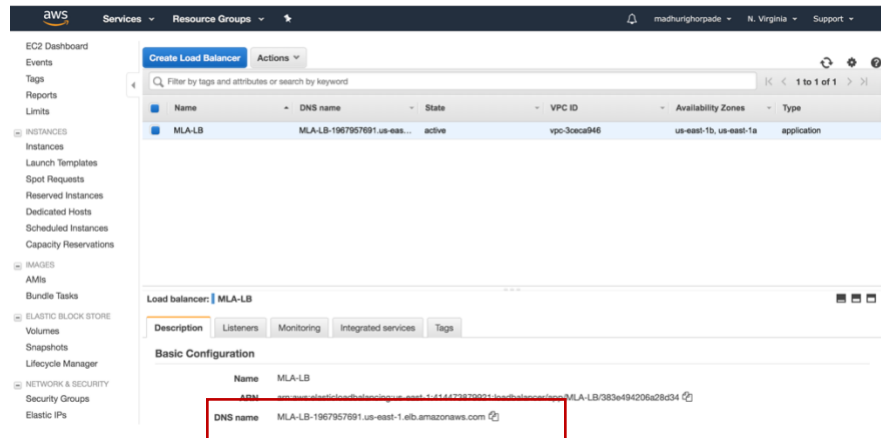
- Security groups: mja-lb-sg

Routing

- Target group: New target group
- Target group name: mja-ec2-target
- Port: 80
- Target type: Instance

Cancel Previous Create

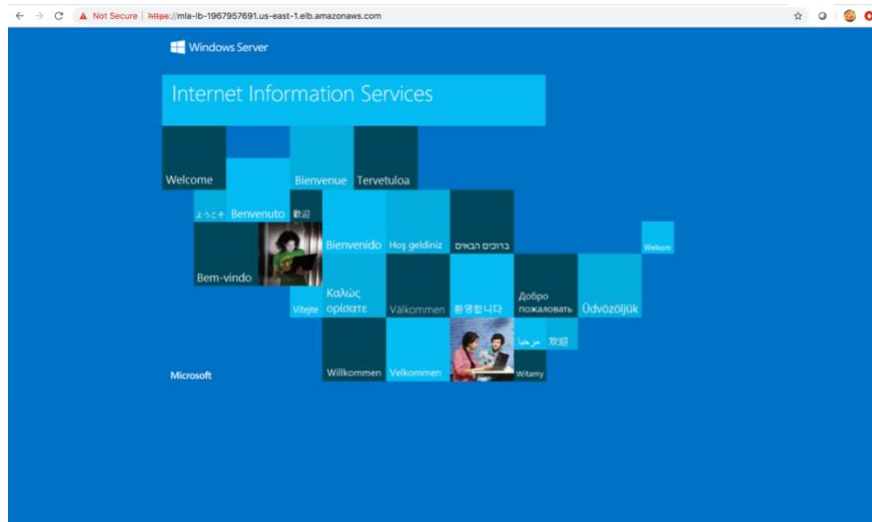
- 7.11. Wait till the 'State' of load balancer is active. Also copy the 'DNS name' of the load balancer to access it:



8. Now when the load balancer is ready, you can test the load balancer in following two different ways:

8.1. Open 'CommonUtils.Java', under 'util' folder. Replace 'MlaWebApi' string variable with 'DNS name' of load balancer. Once you change the url, clean build and run the project.

8.2. Run the DNS name in the web browser and check if you can access your EC2 instance:



Note: You may get an error while accessing your application if you have made your application secure. Since your EC2 instance is accepting only HTTPS traffic and now Load Balancer is using HTTP internally to connect to the instances.

To fix this, remote login into the EC2 instance, open IIS server -> SSL settings and uncheck the 'Require SSL' option:

