Load Balancer

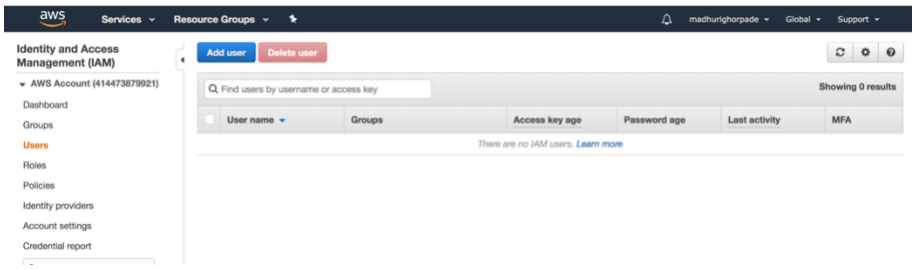
Elastic Load Balancing supports the following types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. Amazon ECS services can use either type of load balancer. Application Load Balancers are used to route HTTP/HTTPS (or Layer 7) traffic. Network Load Balancers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

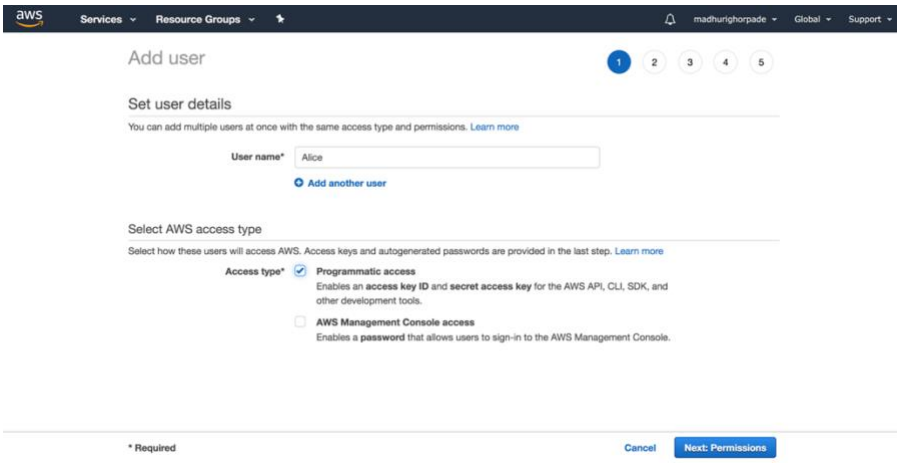
Steps to create an application load balancer

1. Create IAM user in AWS.

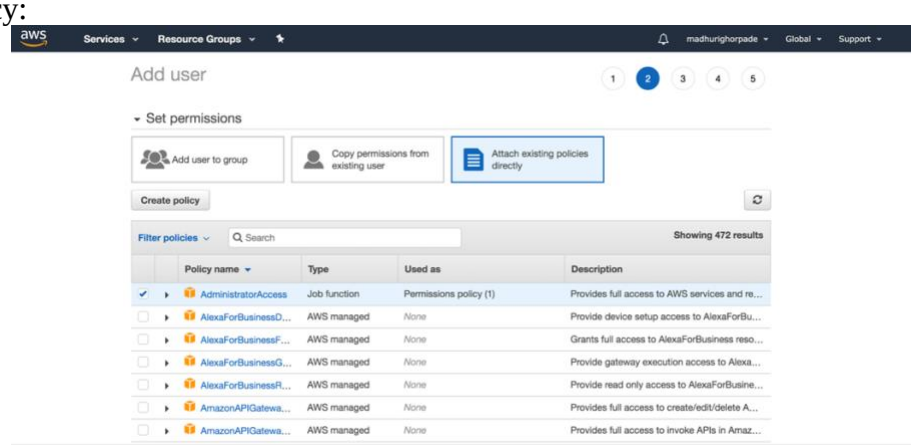
1.1. After login to AWS Management Console, navigate to the ‘Services’ menu and search for ‘IAM’ in the search bar.
Go to Users and click on the ‘Add user’ button:



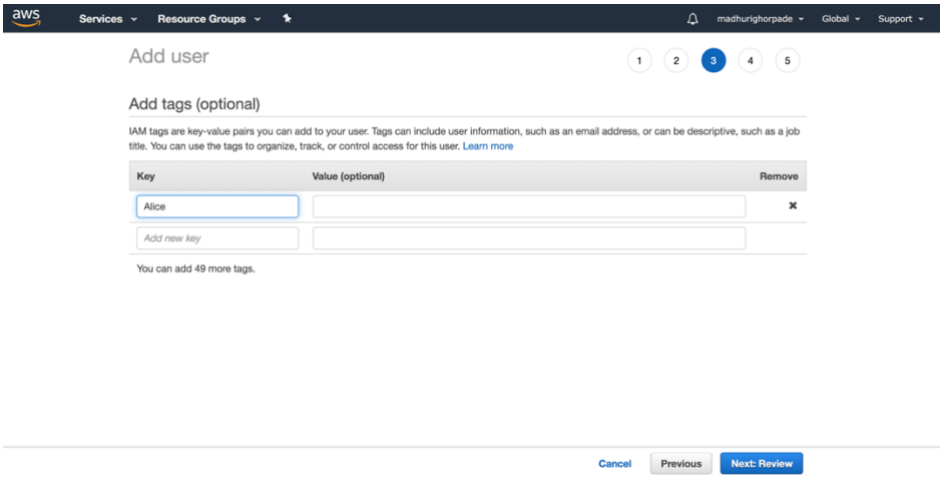
1.2. Provide a ‘User name’ and select ‘Programmatic access’



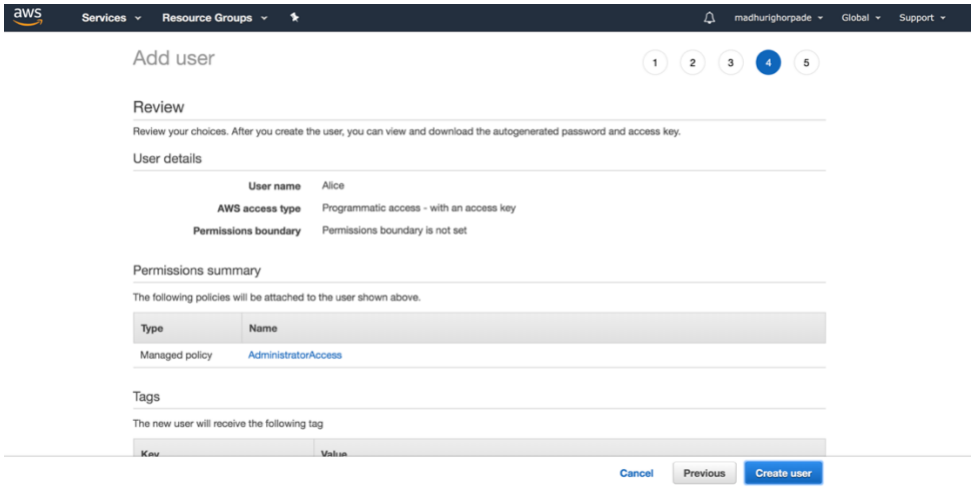
1.3. Click on ‘Attach existing policies directly’ options and select ‘AdministratorAccess’ policy:



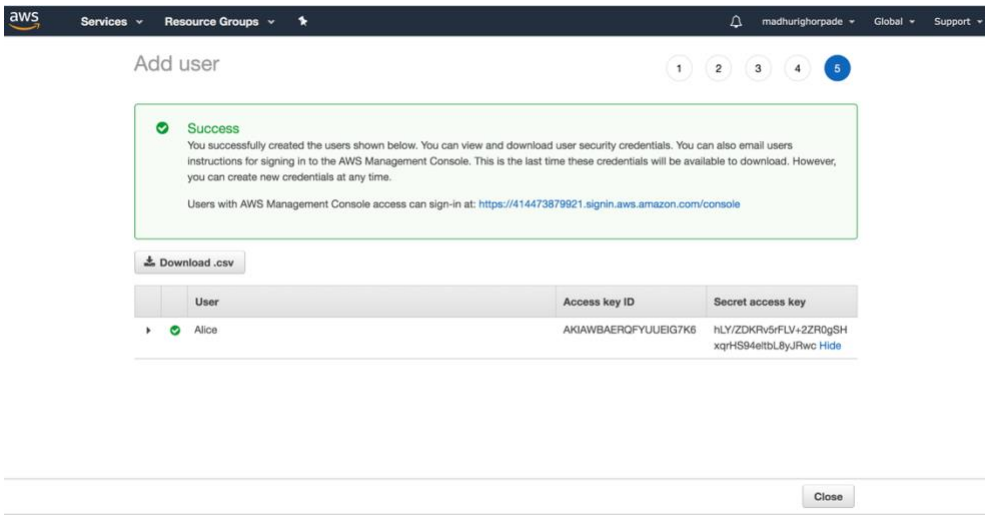
1.4. Adding tags is optional. Click ‘Review’ button



1.5. Review and click on ‘Create user’ button:



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]: AKIAWBAERQFYW5OF4SG2
AWS Secret Access Key [****************UBVq]: bLg/bcoGFHo55FzukLDZyZoOGR46Vj7PaJy0gQB
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_cer
tificate.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:

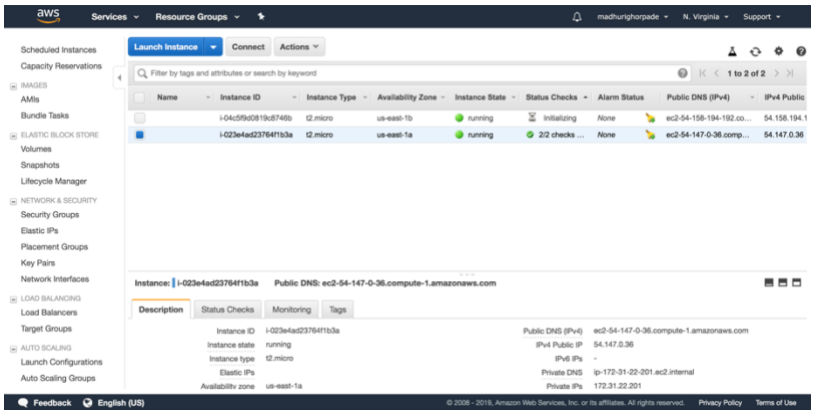
   aws iam list-server-certificates

```
(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"
        },
```

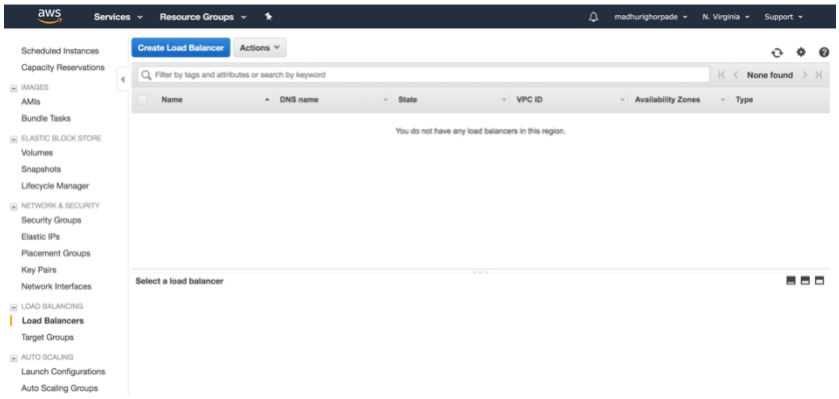
NOTE: Please create a replica of your EC2 instance in AWS (preferably in different availability zone than the previous EC2 instance) to utilize load balancer properties effectively.

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

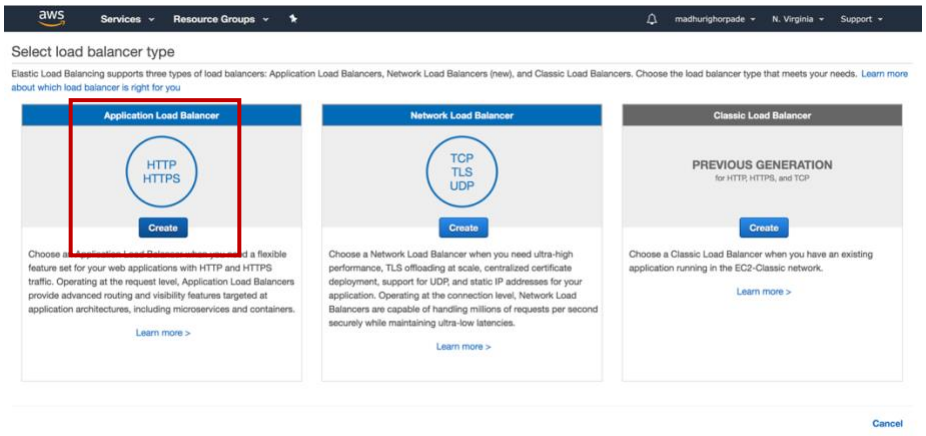
7.1.Navigate to EC2 services and locate ‘Load Balancing’ option from left column:



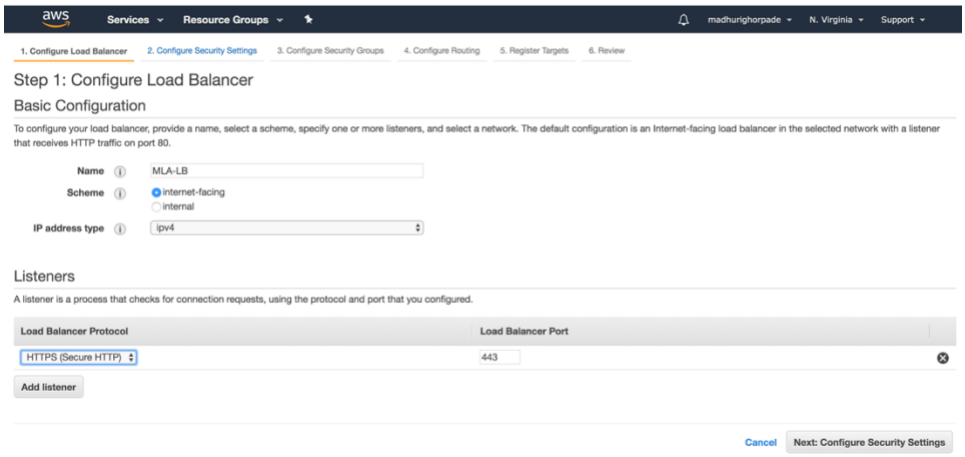
7.2.Click on ‘Create Load balancer’ button:



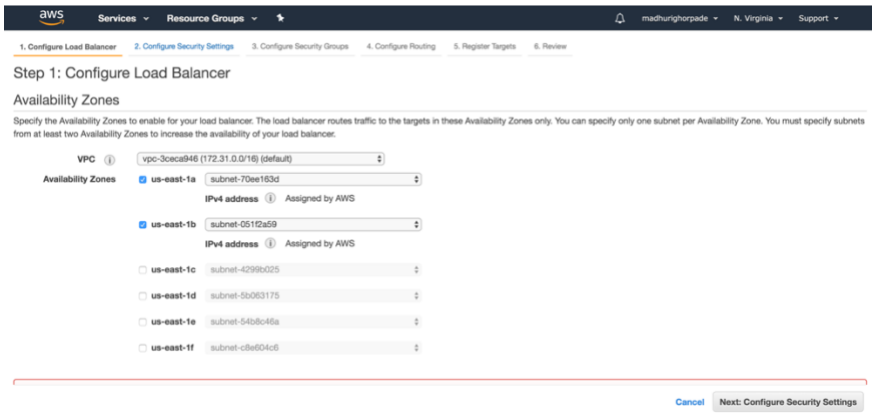
7.3. Select Application Load Balancer:



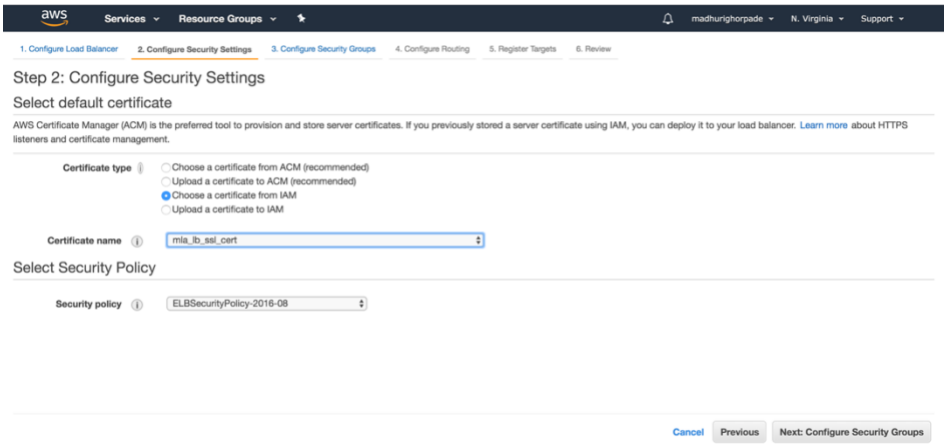
7.4. Provide a name to the newly created application load balancer. I have named it as ‘MLA-LB’. Select the Scheme as ‘internet-facing’ and add listener ‘HTTPS’ with port 443:



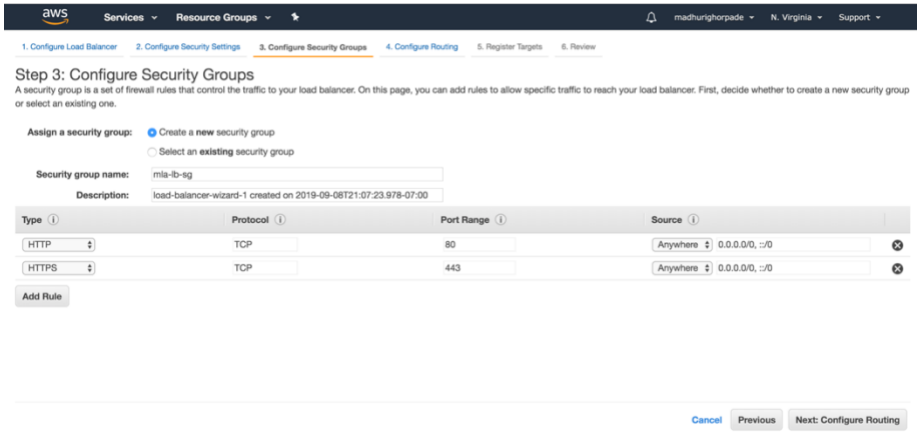
7.5. Select the two availability zones for two different EC2 instances. I have my one EC2 instance in ‘us-east-1a’ and the other EC2 instance in ‘us-east-1b’ zone. You can select your own EC2 instance zones:



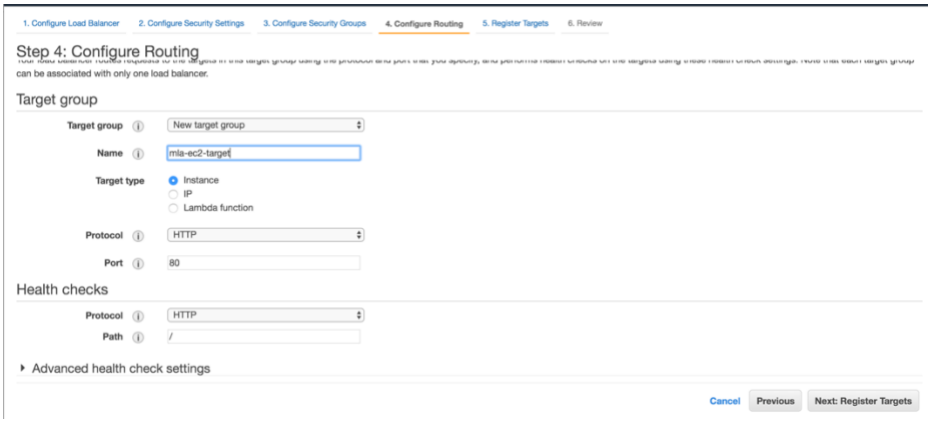
7.6. Since a certificate is already created and uploaded to IAM, select Certificate type as ‘Choose a certificate from IAM’ and select the certificate name from drop down menu, that you have uploaded in step#6:



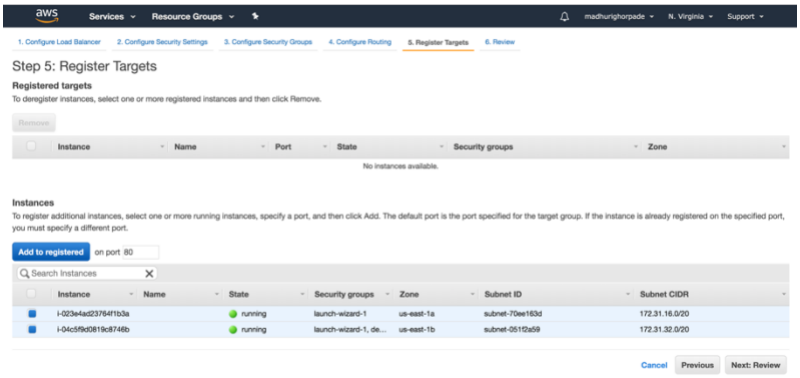
7.7. Click on ‘Create a new security group’ option. Provide a name for the security group, I have named it as ‘mla-lb-sg’. Add HTTP and HTTPS rules to it:

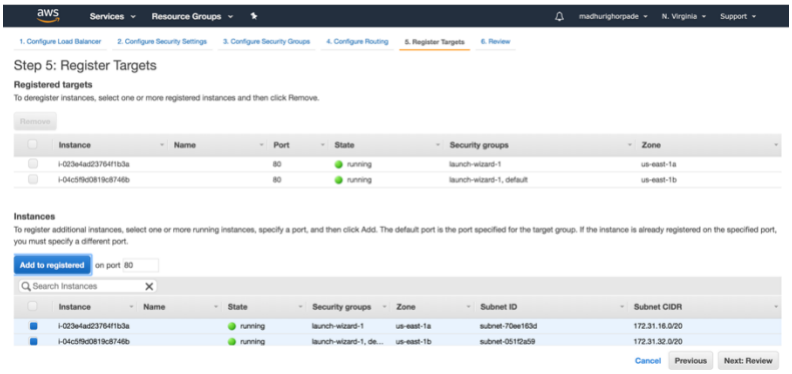


7.8. Create a new target group. I have named it as ‘mla-ec2-target’. Select target type as ‘Instance’. Leave the rest of the settings as it is:

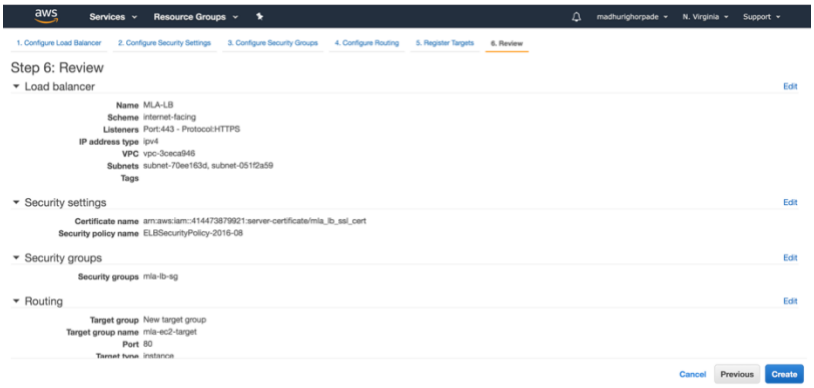


7.9. Select the two EC2 instances and add them to the register targets by clicking on ‘Add to registered’ button:

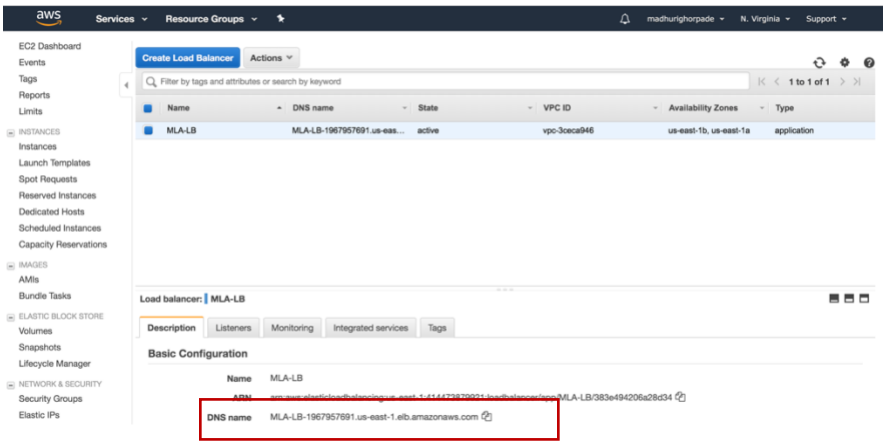




7.10. Review and click on ‘Create’ button:



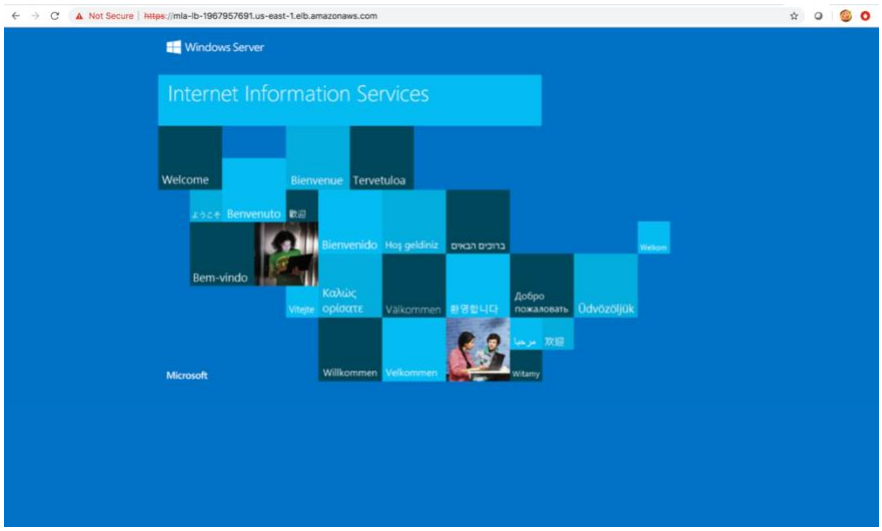
7.11. Wait till the ‘State’ of load balancer is active. Also copy the ‘DNS name’ of the load balancer to access it:



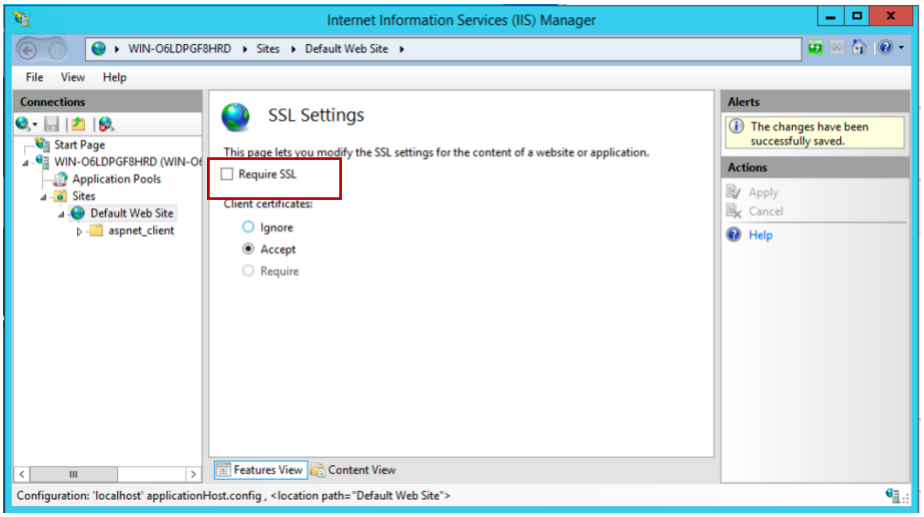
8. Now when the load balancer is ready, you can test the load balancer in following two different ways:

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:


Internet Information Services (IIS) Manager
WIN-O6LDPGF8HRD ▸ Sites ▸ Default Web Site ▸
File View Help
Connections
Start Page
WIN-O6LDPGF8HRD (WIN-O6
Application Pools
Sites
Default Web Site
aspnet_client
SSL Settings
This page lets you modify the SSL settings for the content of a website or application.
Require SSL
Client certificates:
Ignore
Accept
Require
Alerts
The changes have been successfully saved.
Actions
Apply
Cancel
Help
Features View
Content View
Configuration: 'localhost' applicationHost.config , <location path="Default Web Site">>