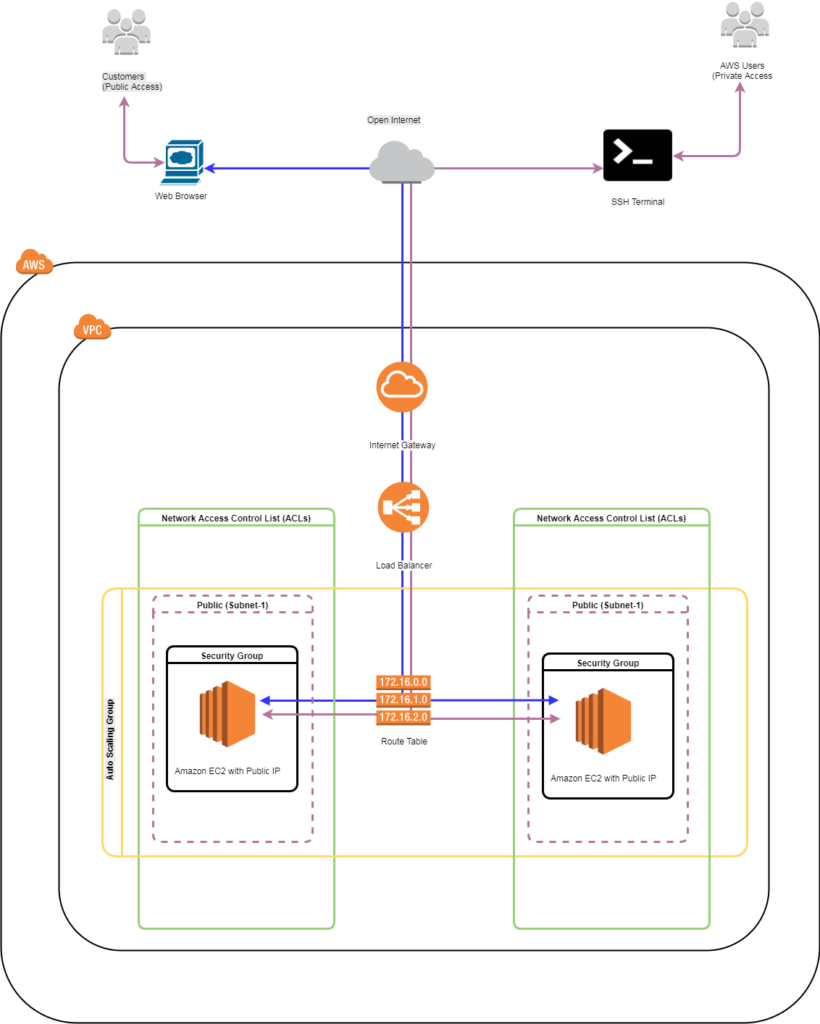
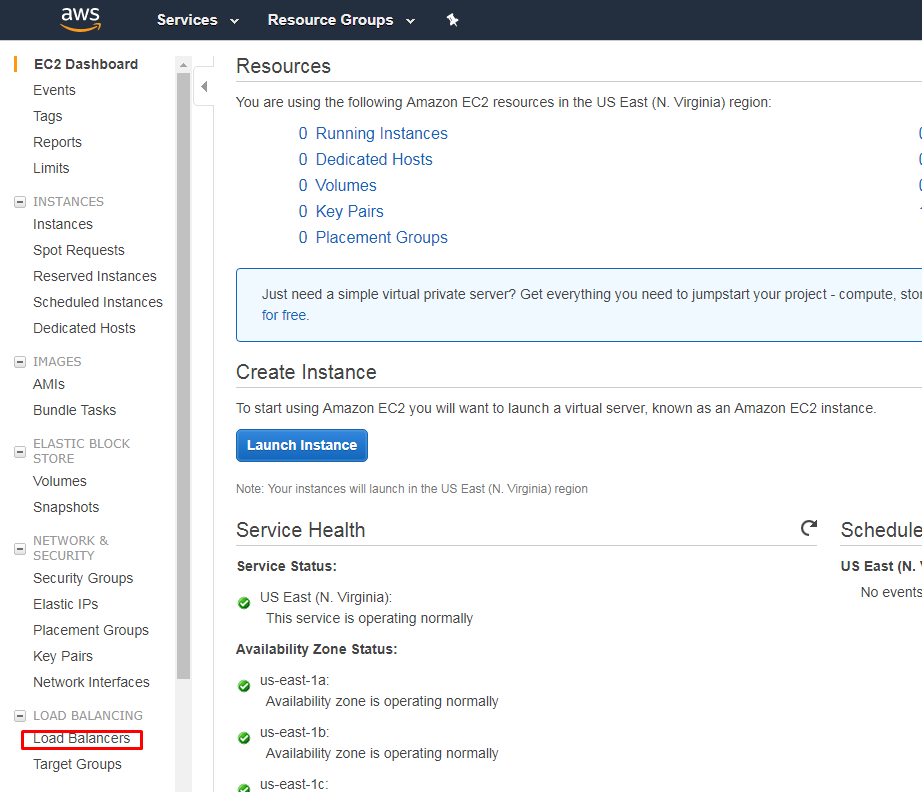
**Setting up a Classic ELB and Auto Scaling Group in AWS**

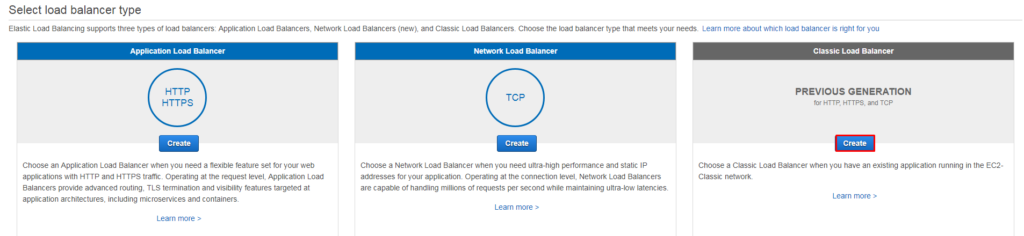
In this tutorial, we will be creating AWS Elastic Load Balancer for autoscaling and high availability of the application. For demonstration purposes, we will keep it simple.



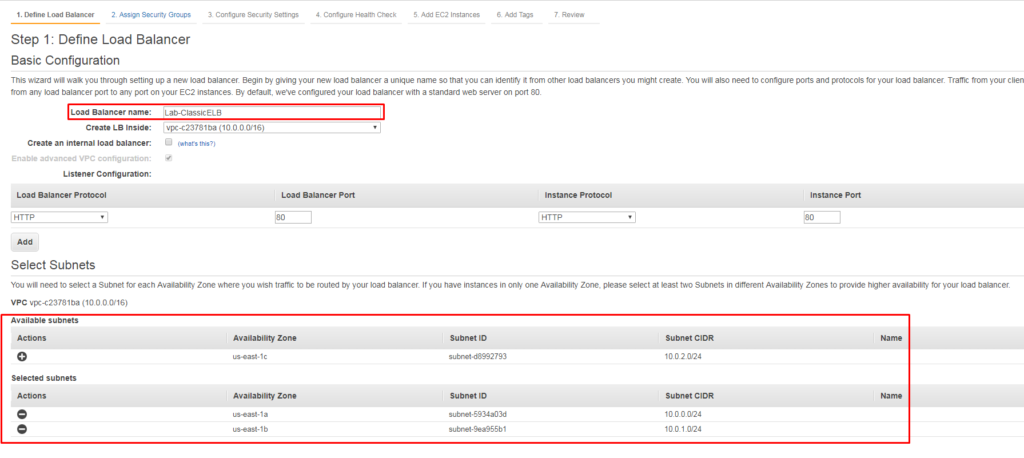
We will be following above architecture for this tutorial. First, we will go to EC2, in there we will find “Load Balancers”



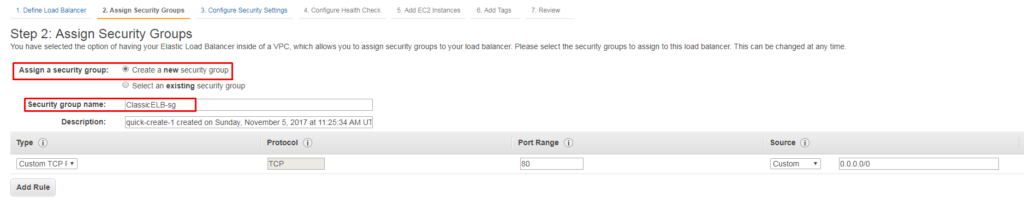
Next click on “Create Load Balancer”. In here you will find three types of load balance we will go first with “Classic Load Balancer”



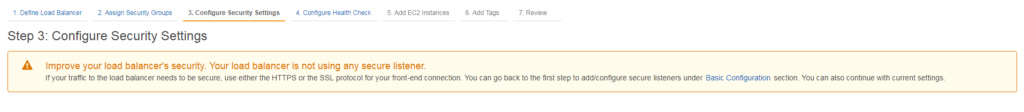
Next, you will find Basic Configuration for the Load Balancer. For simplicity, we will keep HTTP traffic as a “Load Balancer Protocol”. After that, we have select two availability zones and their subnets. Click next when you are done basic configuration.



In **Step 2** it will ask for Security Group. Click next after mentioning the settings for “ **Create A New Security Group** ” and “ **Security Group Name** “.

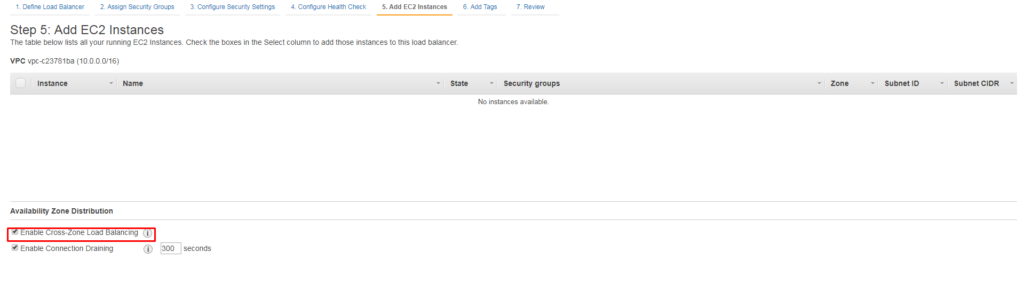


In **Step 3** it will prompt a warning that your load balance security listener is not set to secure listener, ignore that in our case. If you want to use https traffic for ELB then in the basic configuration (Step 1) you have to mention it.

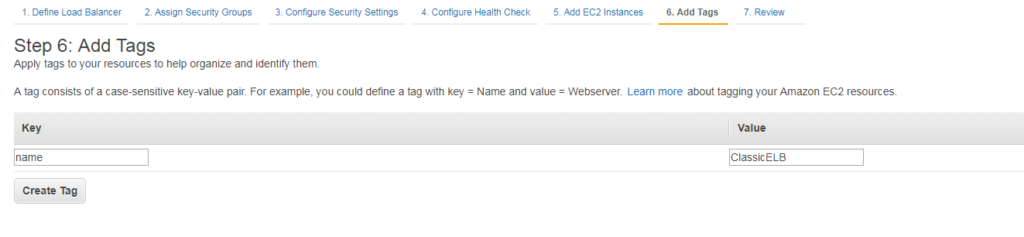


In **Step 4** it will require configuration for Health Check. Right now we have don’t have any instance created with “ **index.html** ” page, so we will use TCP protocol. By selecting TCP protocol ELB will ping port 80 to instances to make sure it is there and active. Advance details show healthy check of an instance time interval, you can configure according to your need but for this demonstration, we will use **Healthy Threshold = 2** . Click Next.

In **Step 4** , right now we don’t have any instances created so we will leave it by default. Just make sure that “ **Enable Cross-Zone Load Balancing** ” is checked. Click Next to Add Tags.



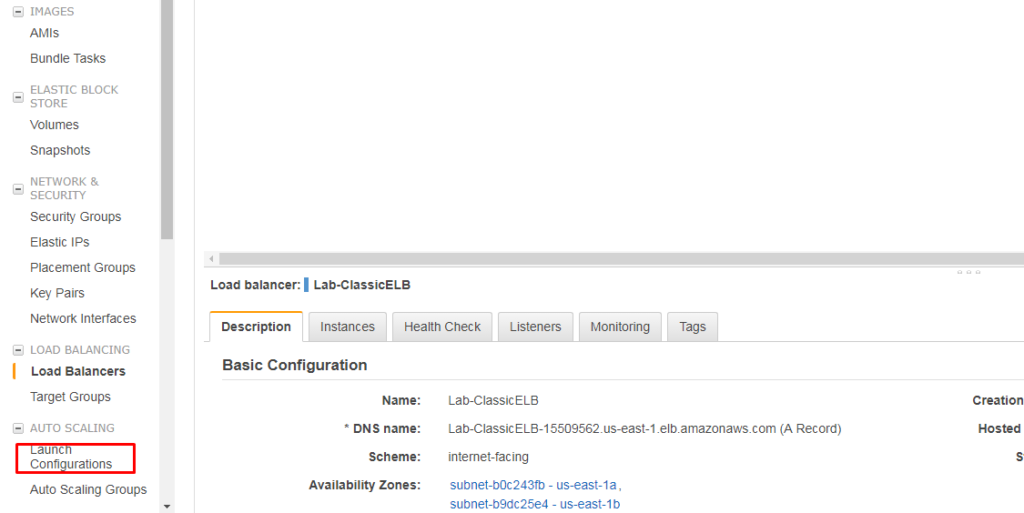
In **Step 6** , choose a name to tag your ELB and hit Review and Create



In **Step 7** , it will show the summary of the configuration click create. Voila you have successfully created an AWS Classic ELB

**Auto-Scaling Group**

Now we will move on to auto-scaling configuration and  auto-scaling group, so in EC2 console will be going to Auto Scaling in here you will find Launch Configuration

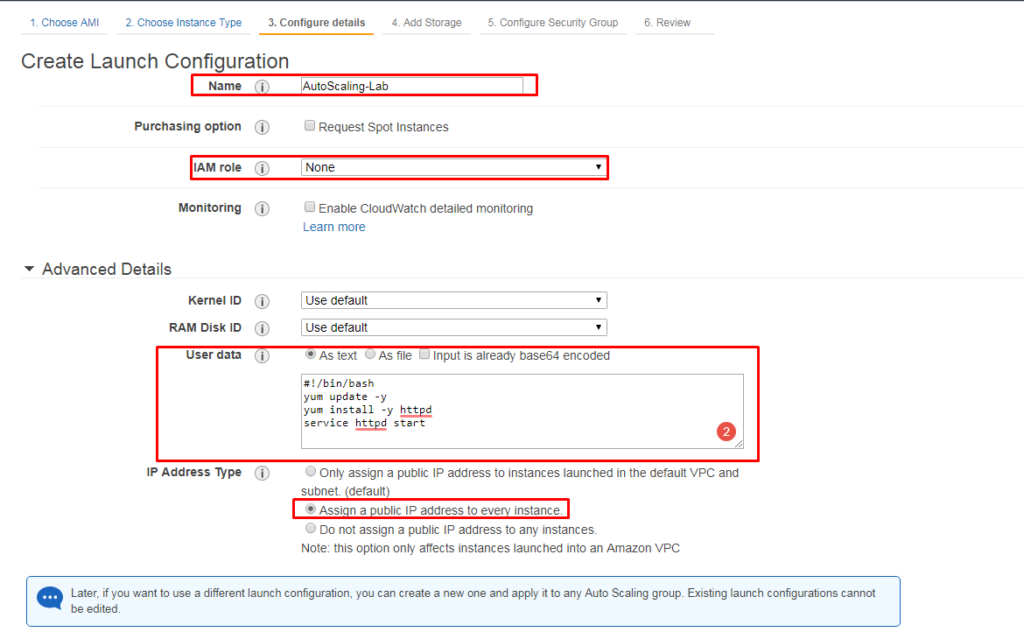


Click “ **Create Auto Scaling Group** ” and on next screen click on “ **Create Launch Configuration** “

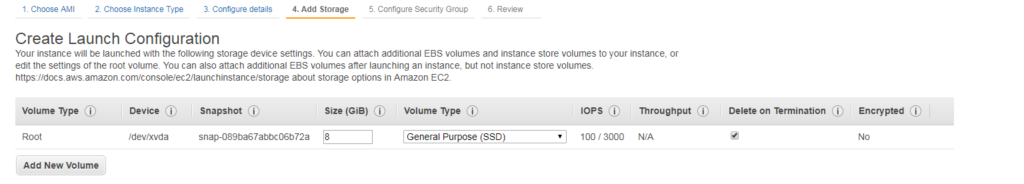
In **Step 1** we will choose the AMI

**Step 2** : We will choose t2.micro and click “Next: Configure Details”

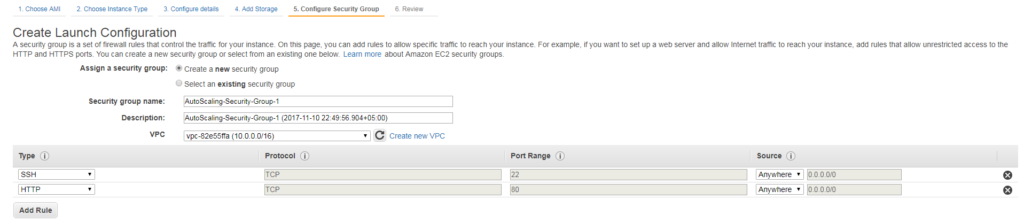
**Step 3:** We will configure the instance, in user data we have defined a bash script so when the instance will be provisioned it will automatically install the web server and start its service. Assign the Public IP address as we are not sure that what subnet will be assigned to that instance



**Step 4** : We will be adding storage, for the purpose of this demonstration leave it as default and click Next



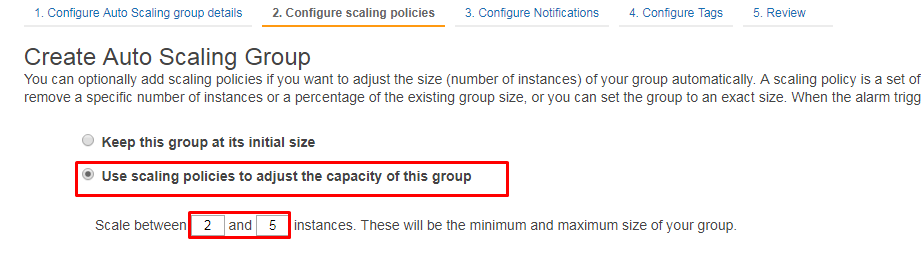
**Step 5** : We will configure security group



**Step 6** : Review your configurations and click on “ **Create launch configuration** “. After that, it will ask for the download key pair, download the key pair and click on “ **Create Launch Configuration** “

Now it will ask f0r auto-scaling policies

**Step 1:** We will create the autoscaling group and after making the changes click on Next

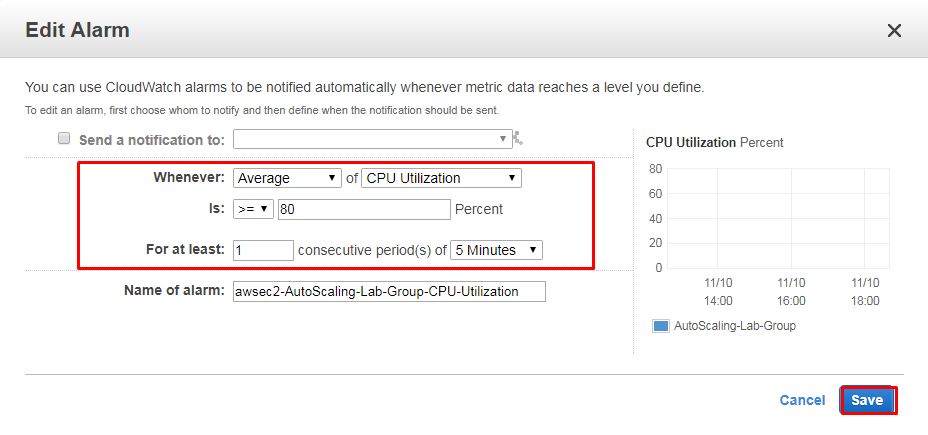


For decreasing the resources we will be using the below configuration, if the increased instances are utilizing below 40 percent usage then we need to remove these instances.

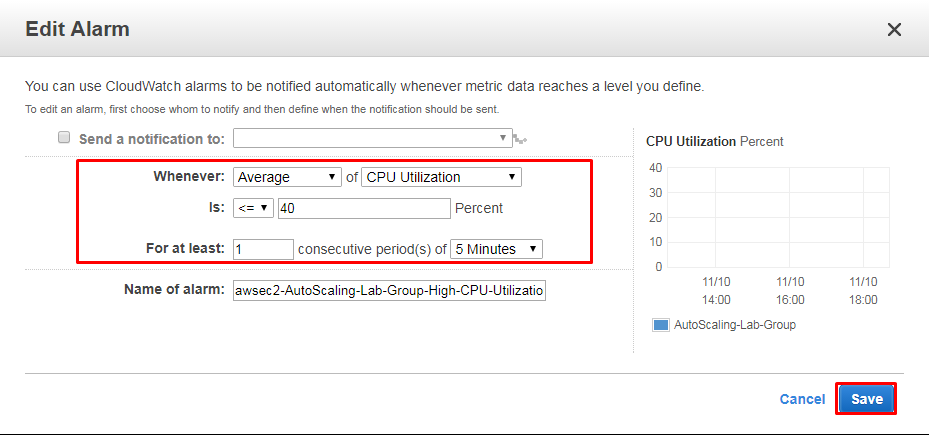
**Step2:** We will configure the policies for autoscaling, in “Increase Group Size” click add a new alarm. A new pop windows will open in here you have to define your trigger. We scale it between 2 to 5 instances.

**Note:** click on “Scale the Auto Scaling group using step or simple scaling policies” so that you can see Increase and decrease group size

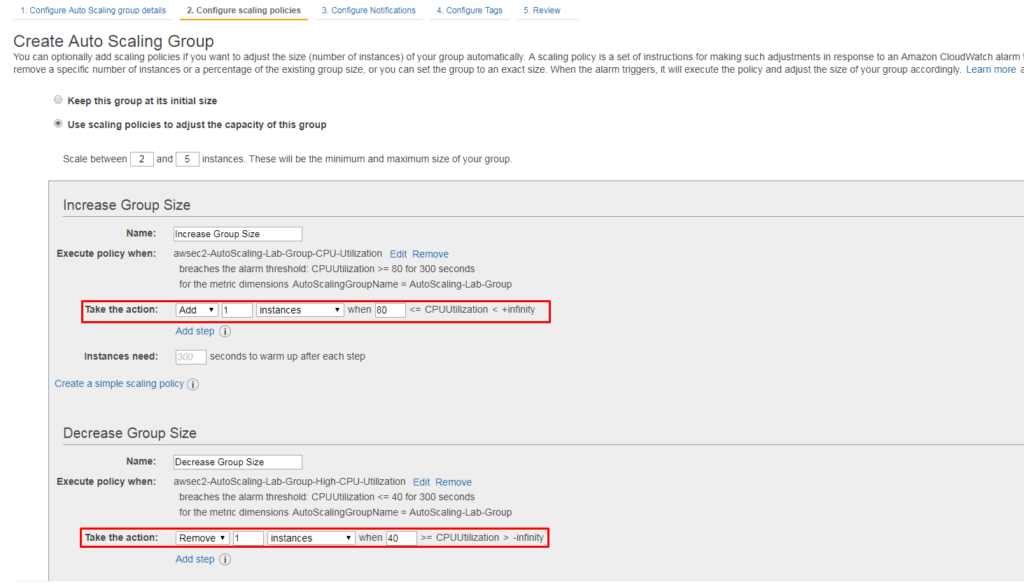
Now you will be seeing two options here one for the increase and the other for decrease group size, we need to edit these two alarms so we can trigger them according to our environment needs.



For decreasing the resources we will be using the below configuration, if the increased instances are utilizing below 40 percent usage then we need to terminate these instances.



Then finally we need to configure the actions for increasing and decreasing.

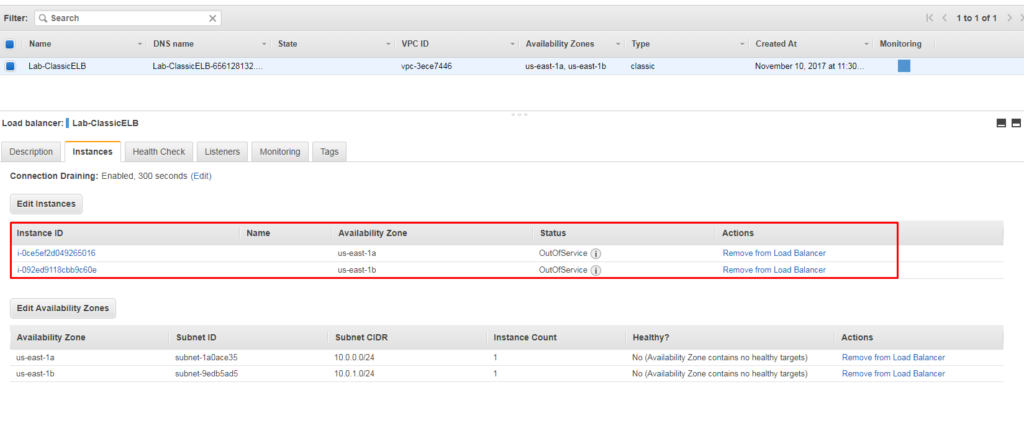


**Step 3:** it will ask for SNS notification, in our case we are not using it so leave it as default  (SNS service is used for notification to your mobile)

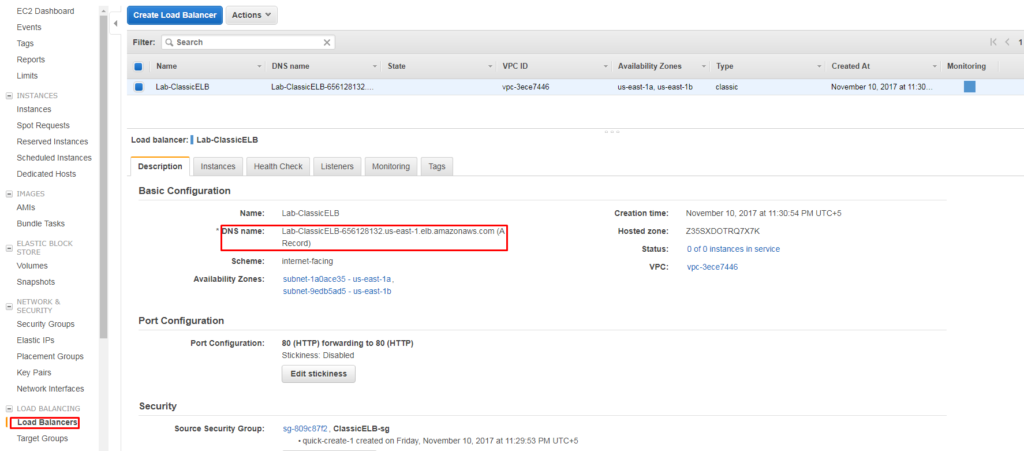
**Step 4:** In our case, we will leave it as default for demonstration purposes.

Step 5: Review your settings and click on Launch

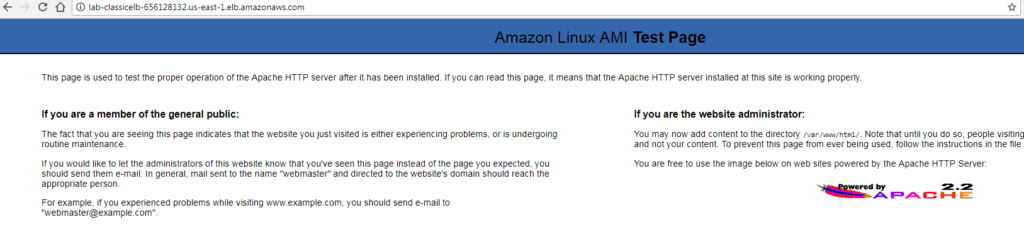
After the creation of two instances and auto-scaling group we will be going back to EC2 console and click on “Load Balancers”. We have to check the instances if they are added correctly to **Lab-** ClassicELB **.**



Now we will copy the DNS name of the load balancer and paste in our browser to see the results.If all the configurations are successfully made it will show the apache page on your browser.



After pasting the DNS name of ELB its showing the apache server default page



Now to check whether our auto-scaling and load balancing feature works or not, we will be going to terminate one of the instances and wait for the load balancer to check and provision new instance. Goto to instances in EC2 Console and terminate one of the instances. It will take few minutes to provision the new instance as soon as new instance will be provisioned it will automatically add it to load balancer.