lab

lab title

AWS Elastic Load Balancers

V1.01

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

These lab notes are to support the hands on instructional videos of the Elastic Load Balancing section of the AWS Certified Associate Course.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

Implementing an Application Load Balancer

In this section we will create two EC2 servers running WordPress and Magento. We will then front these with an Application Load Balancer and implement path based routing.

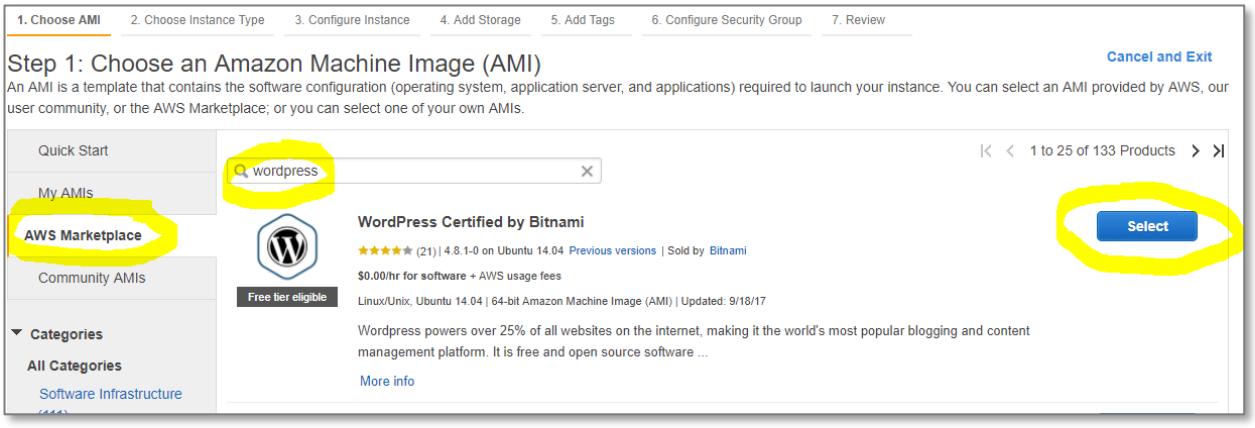
Creating a WordPress Server

Click on the services menu and select EC2.

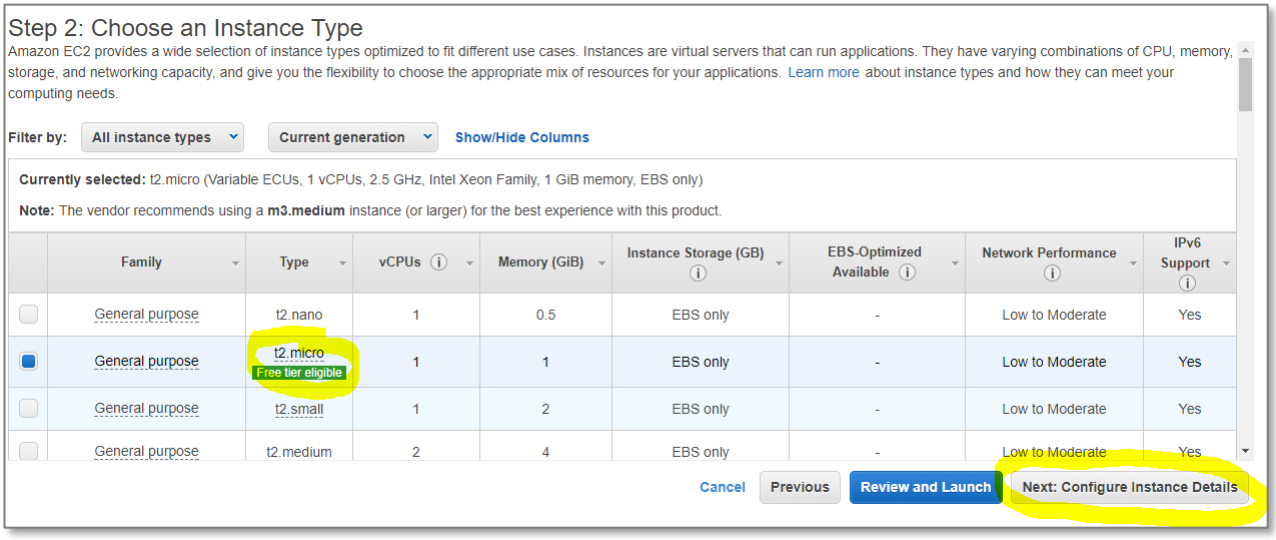
Click “Launch Instance”

Select “AWS Marketplace” and search for “Wordpress”

Select the Bitnami Wordpress AMI



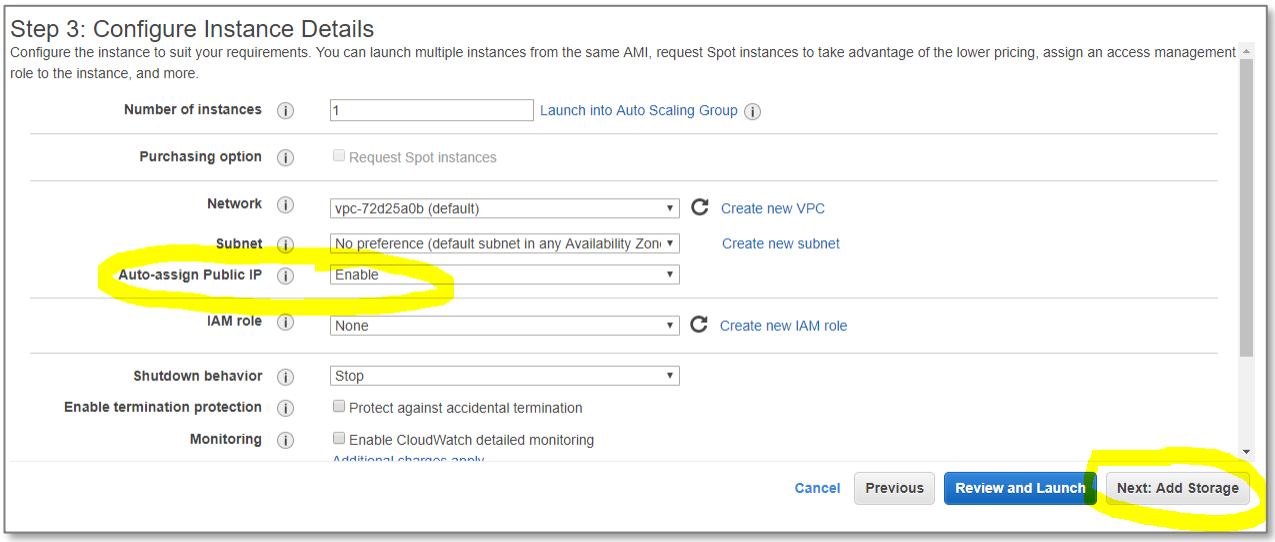
Select t2 micro instance type



Click “Next Configure Instance Details”

Select “Auto Assign Public IP”

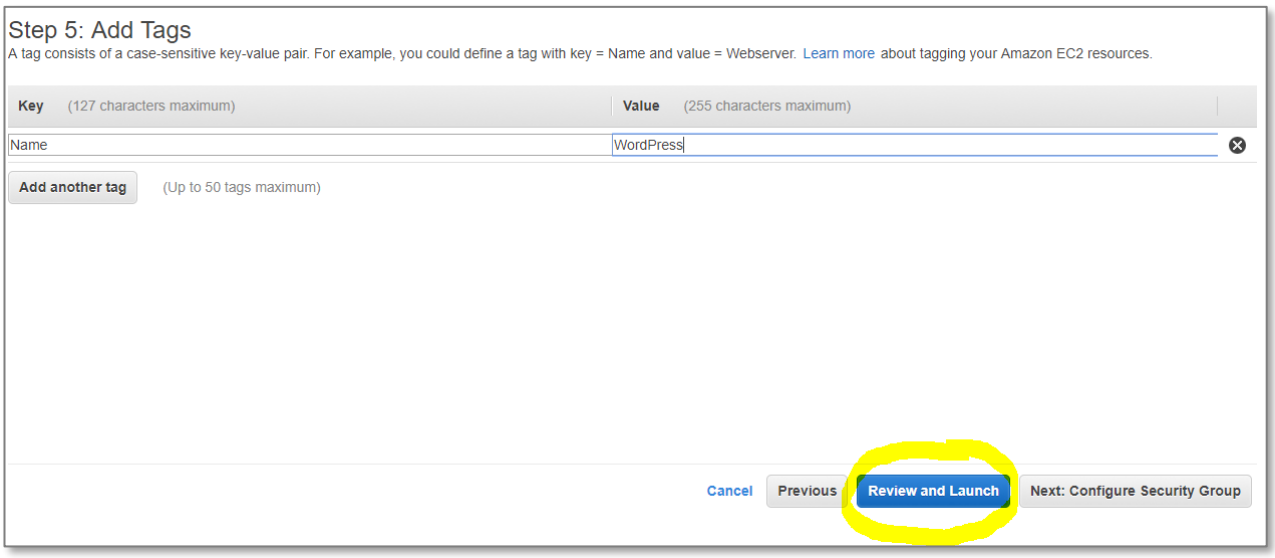
Click “Next Add Storage”



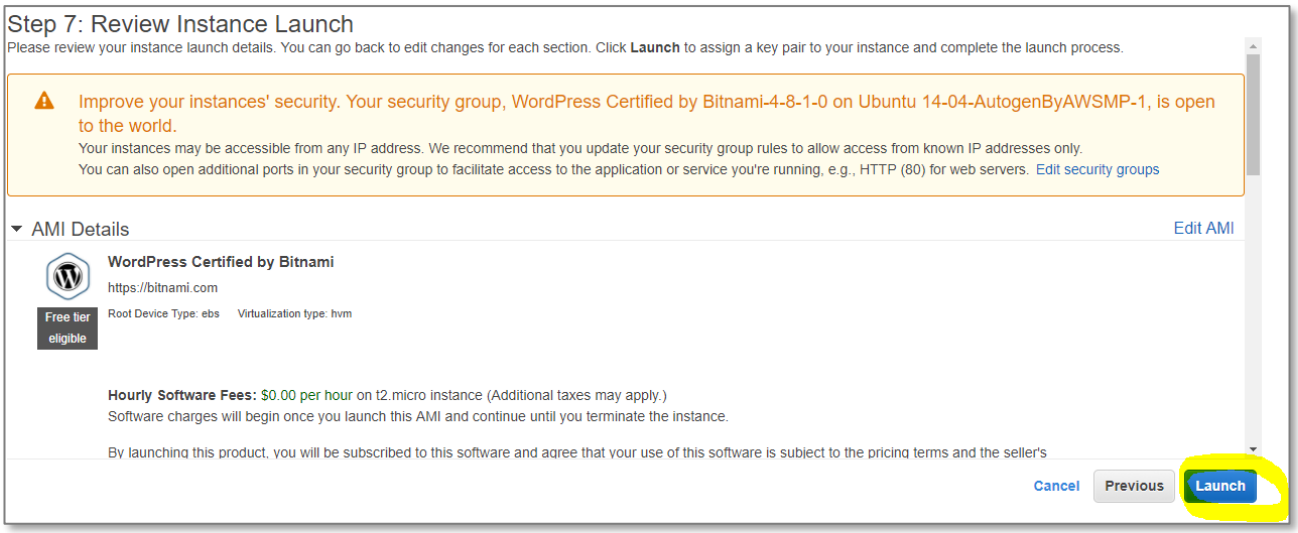
Click “Next Add Tags”

Add key “name” and value “WordPress”

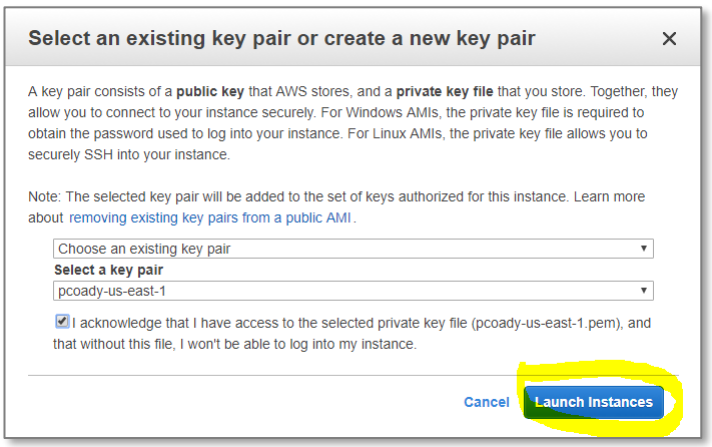
Click “Review and Launch”



Click “Launch”



Select a key pair and click “Launch Instances”



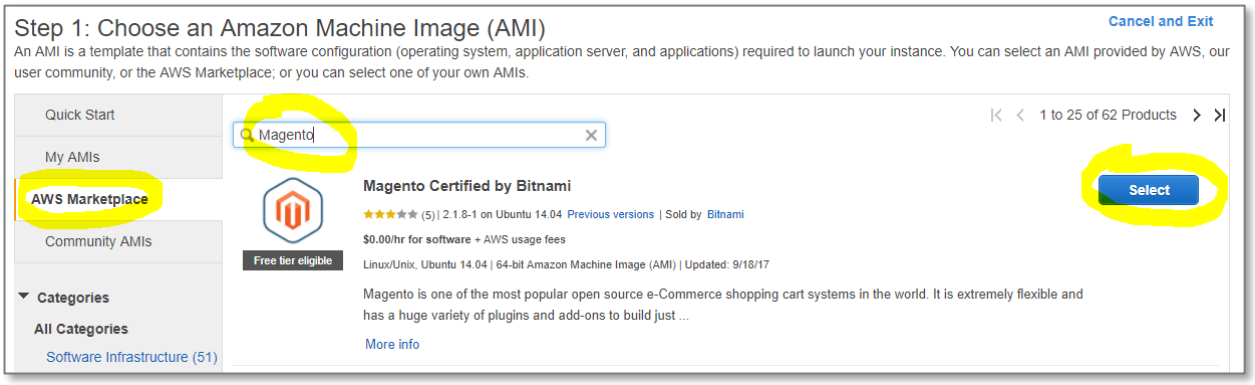
Creating a Magento Server

Click on the services menu and select EC2.

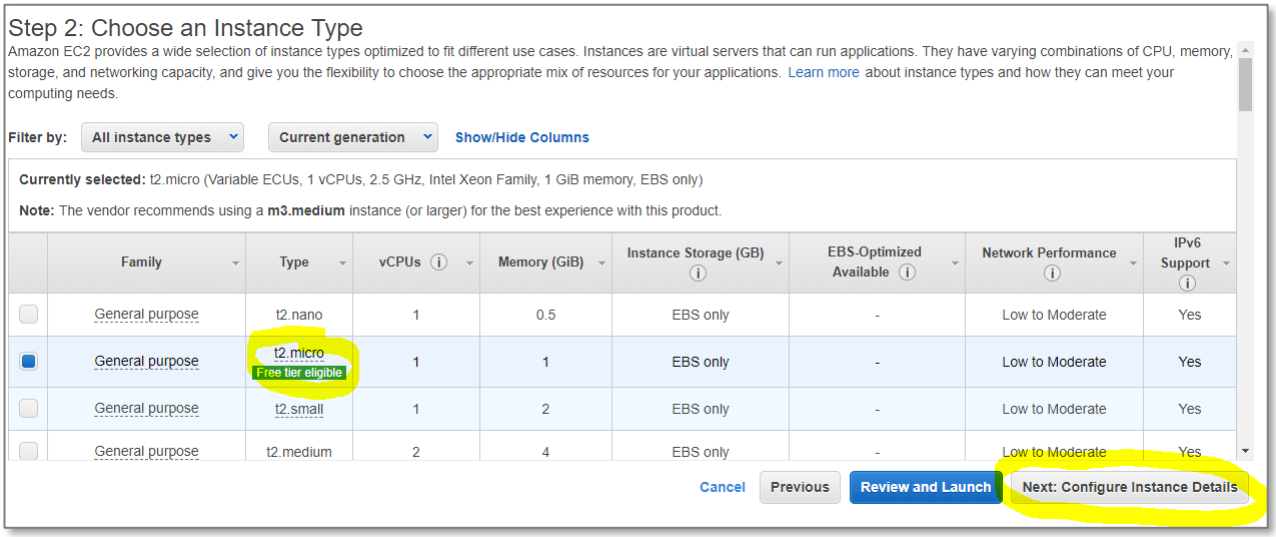
Click “Launch Instance”

Select “AWS Marketplace” and search for “Magento”

Select the Bitnami Magento AMI



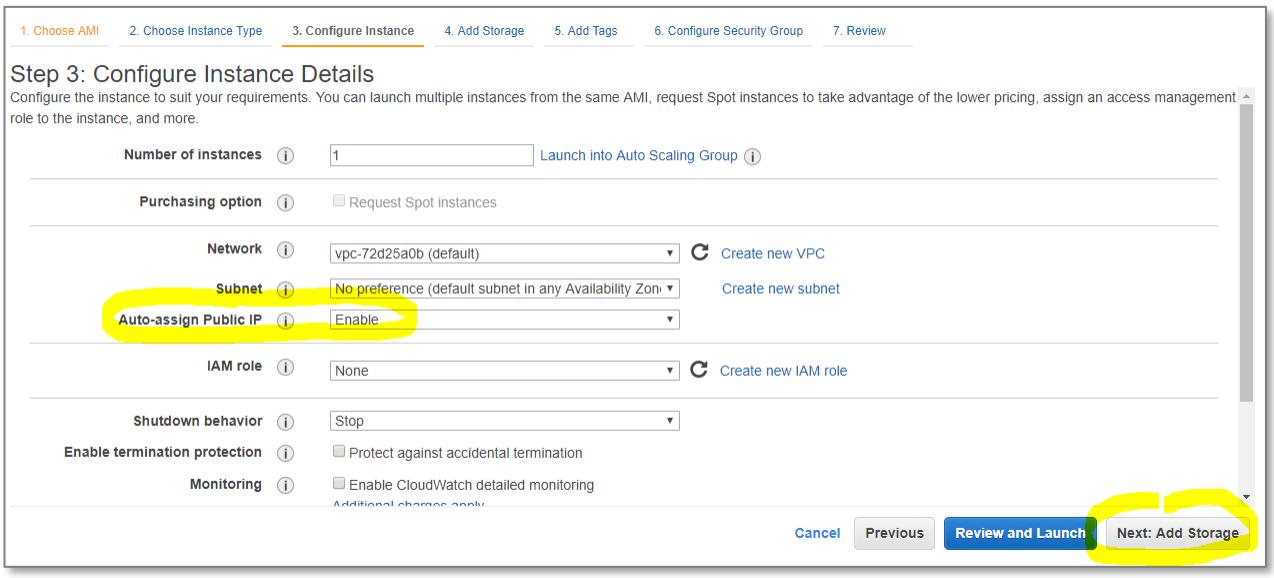
Select t2 micro instance type



Click “Next Configure Instance Details”

Select “Auto Assign Public IP”

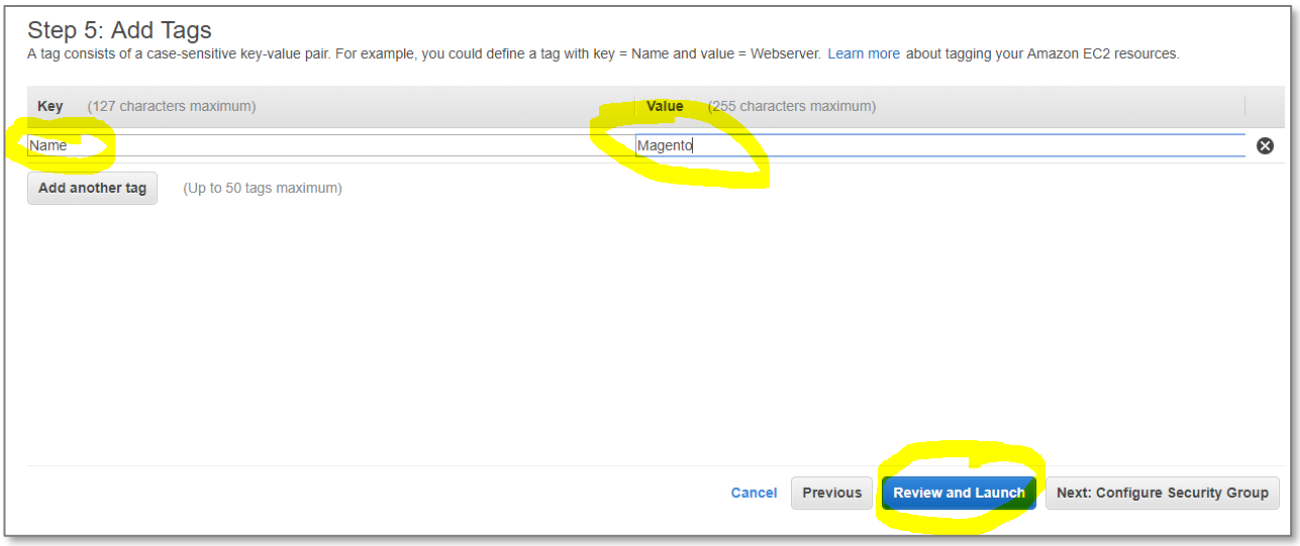
Click “Next Add Storage”



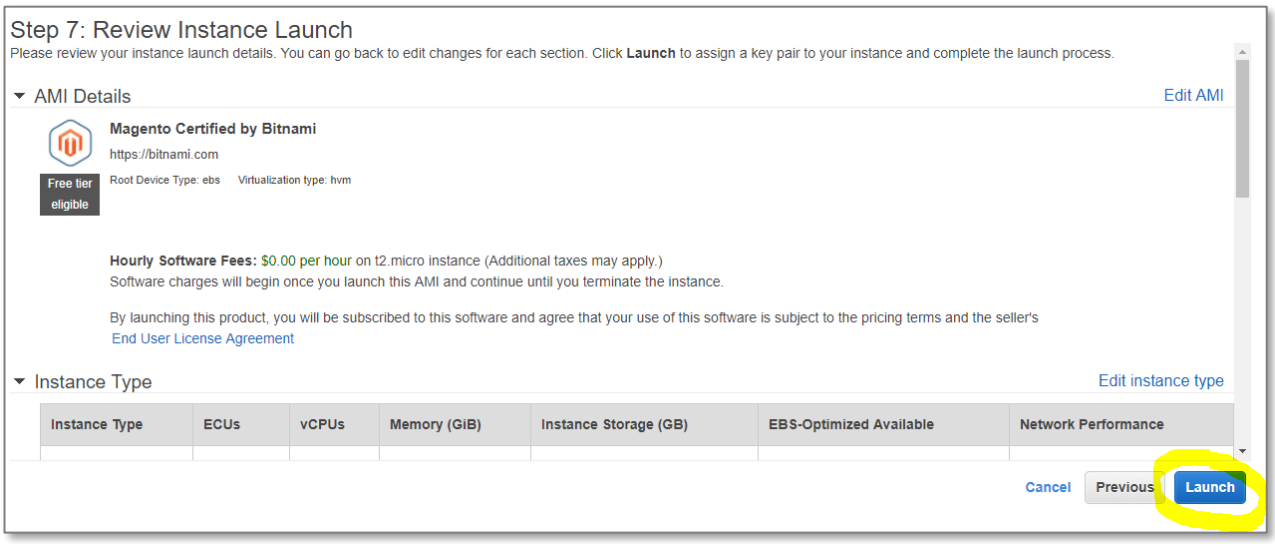
Click “Next Add Tags”

Add key “Name” and value “Magento”

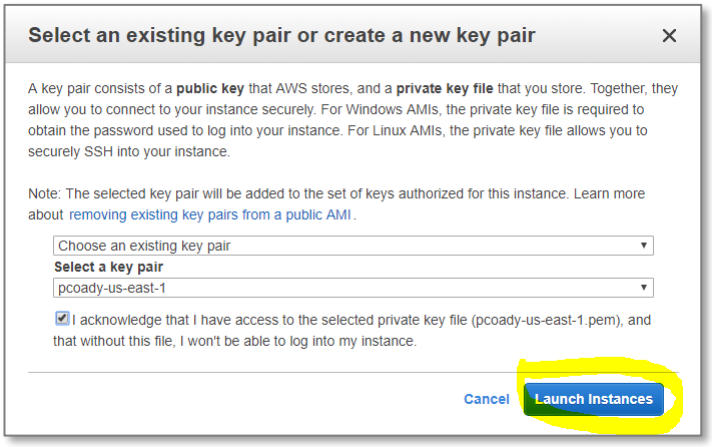
Click “Review and Launch”



Click “Launch”



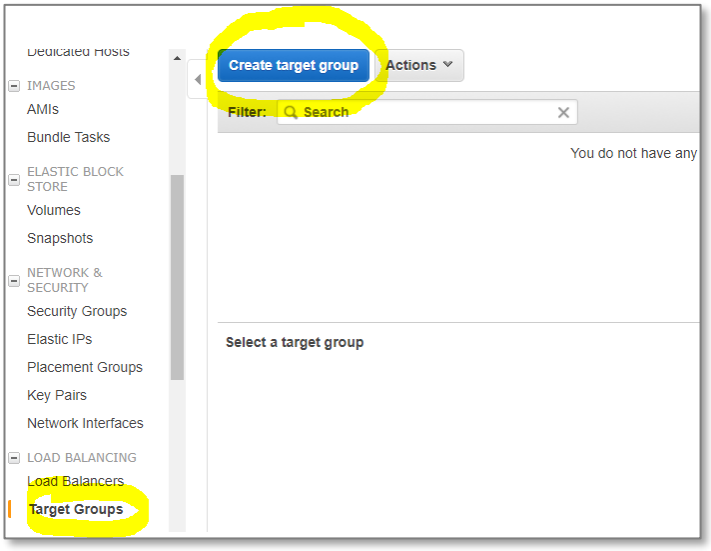
Select a key pair and click “Launch Instances”



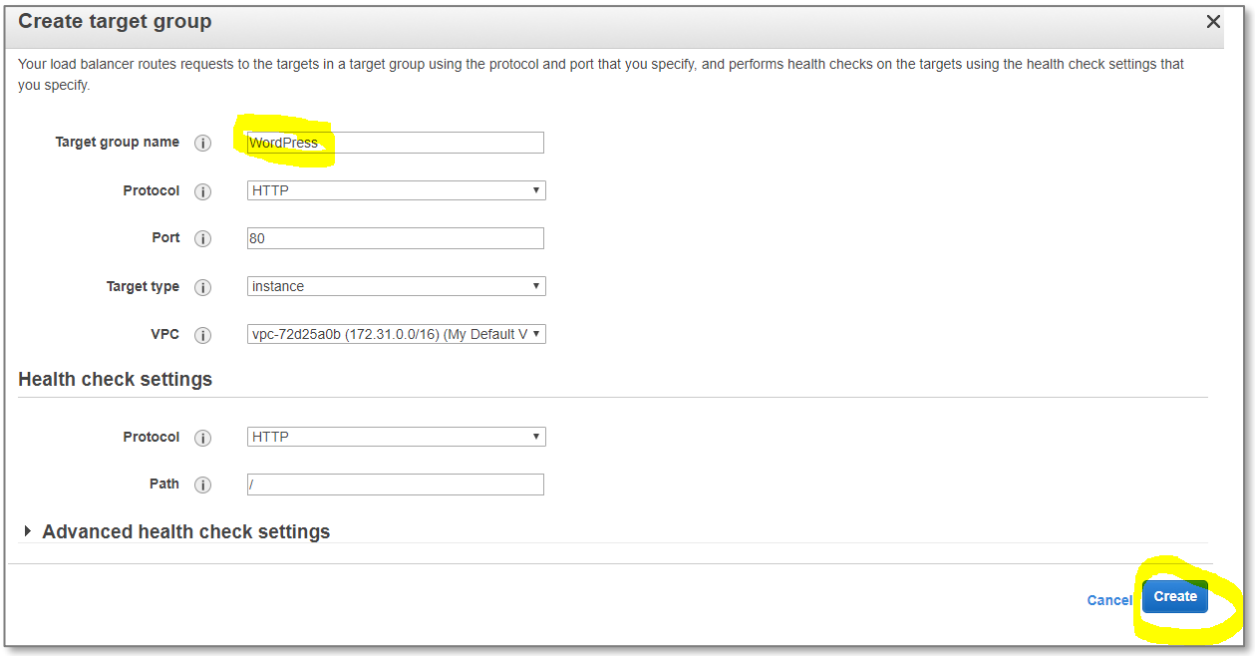
Creating Target Groups

Scroll down and select “Load Balancers”

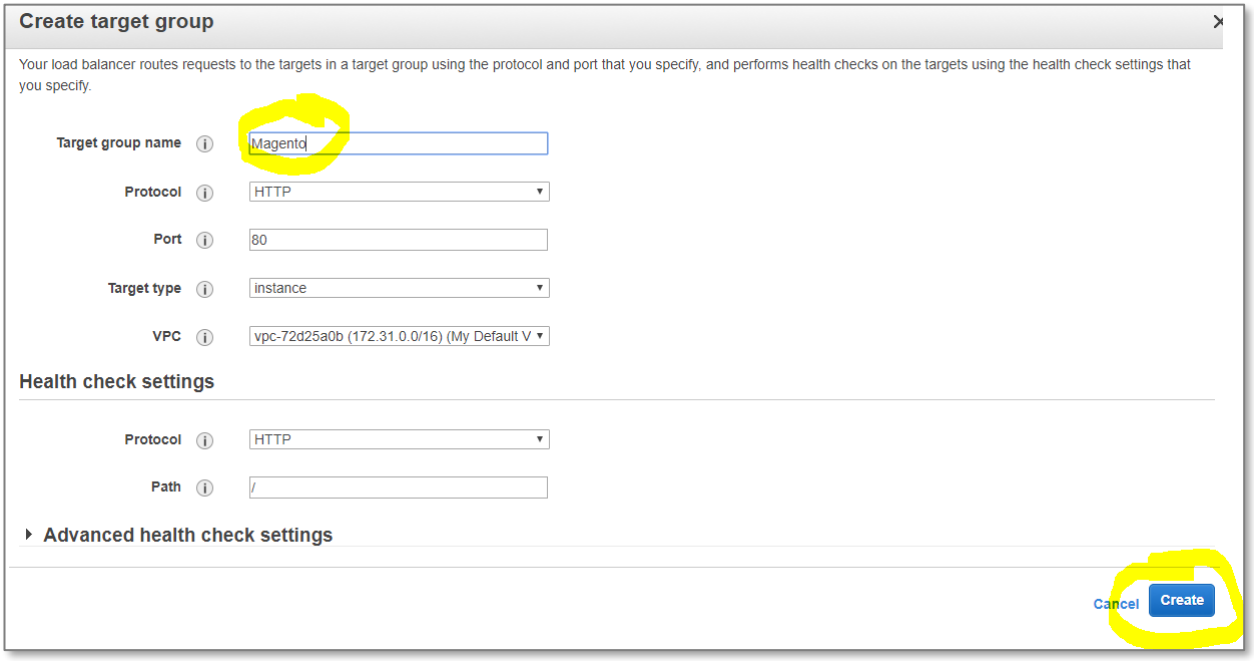
Click “Create Target Group”



Create a target Group called “WordPress”



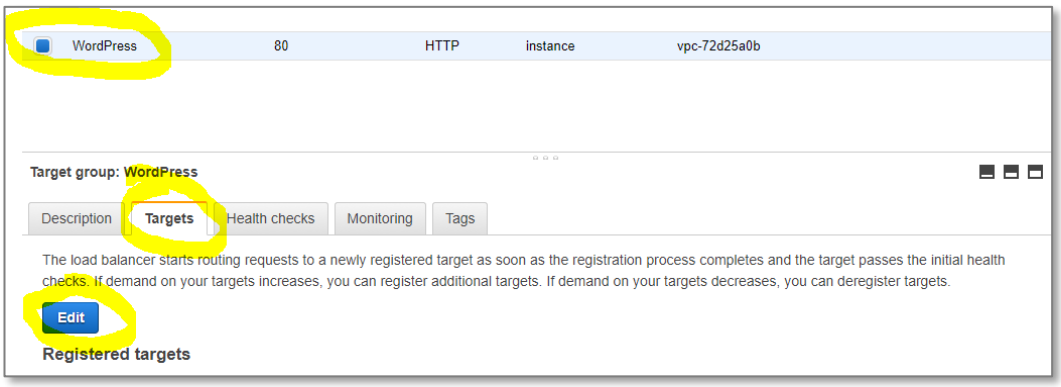
Create another Target Group called “Magento”



Select the WordPress Target Group

Select the “Targets” tab

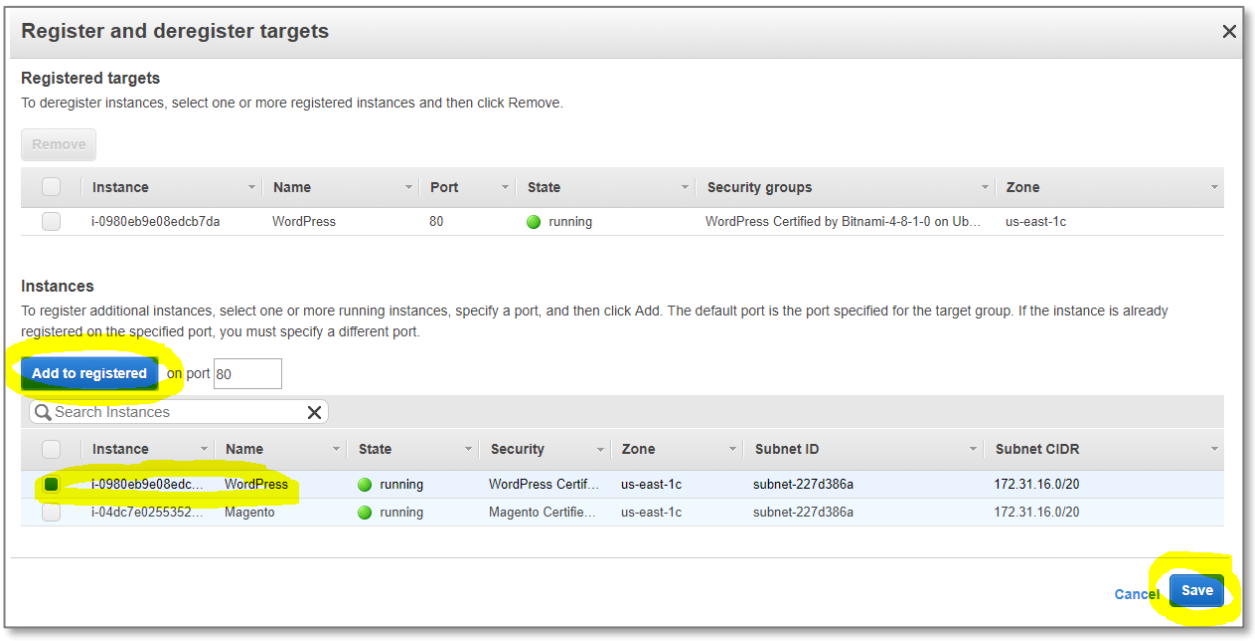
Select “Edit”



Select the WordPress EC2 instance

Click “Add to Registered”

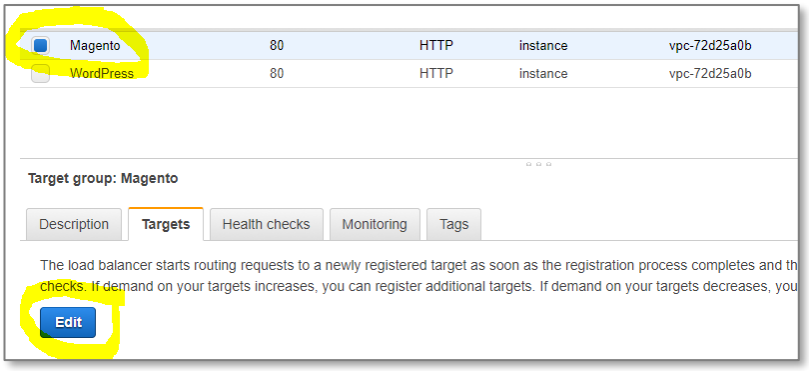
Click “Save”



Select the Magento Target Group

Select the “Targets” tab

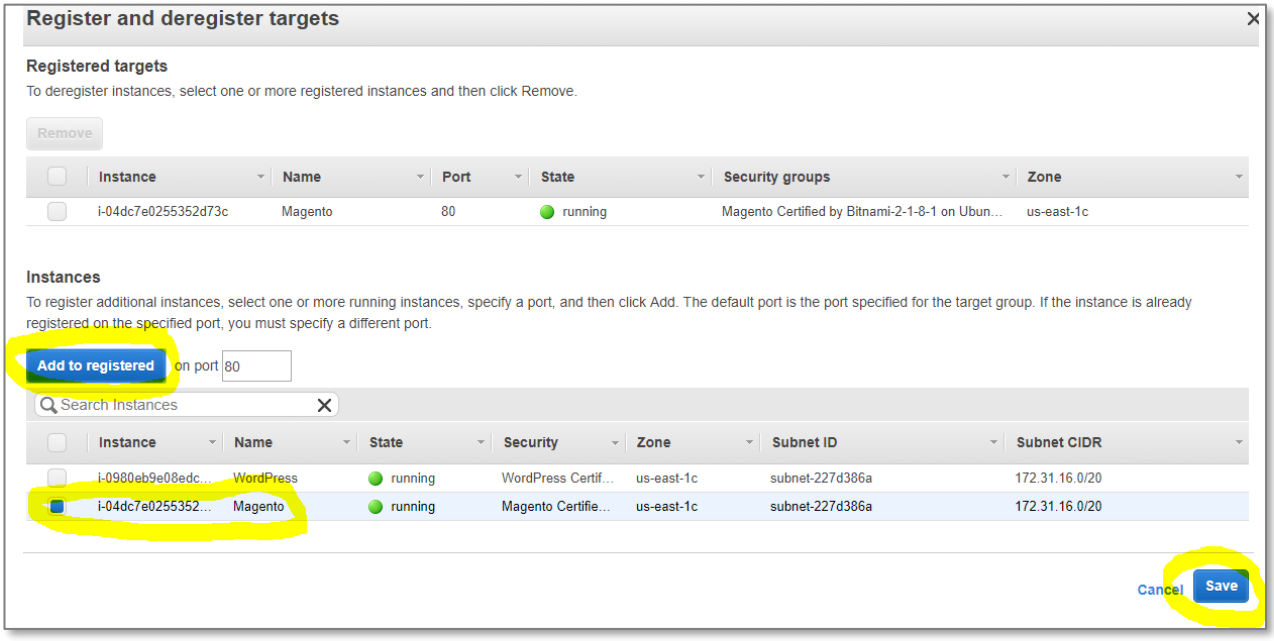
Select “Edit”



Select the Magento EC2 instance

Click “Add to Registered”

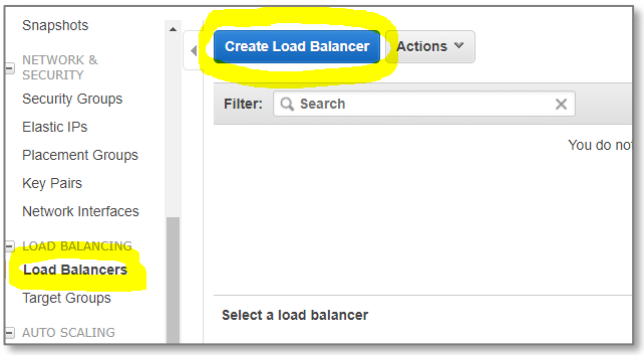
Click “Save”



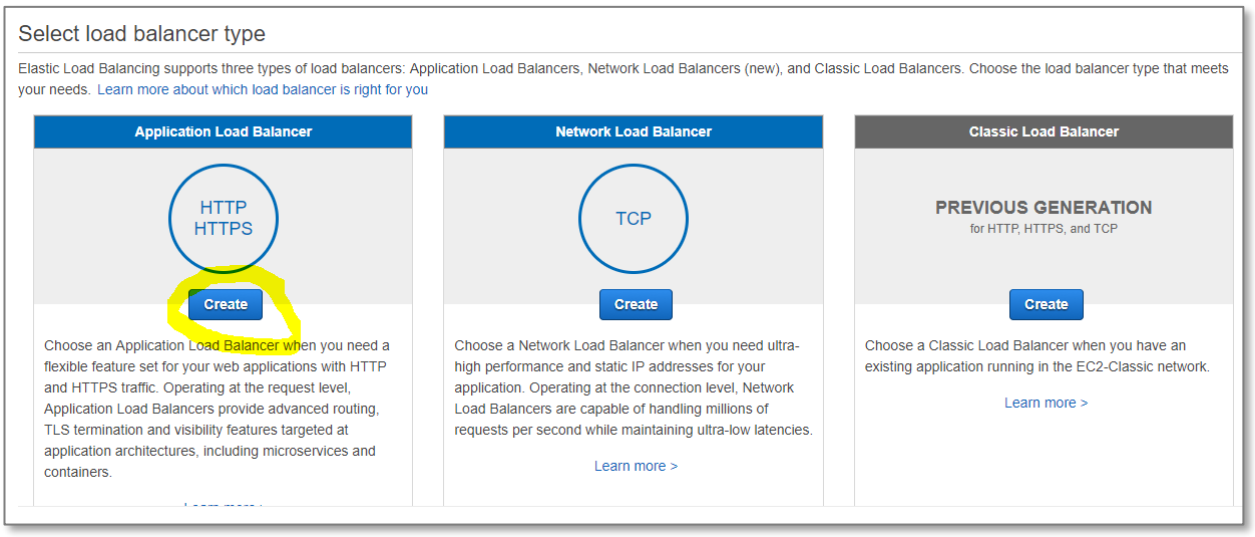
Create an Application Load Balancer

Select “Load Balancers”

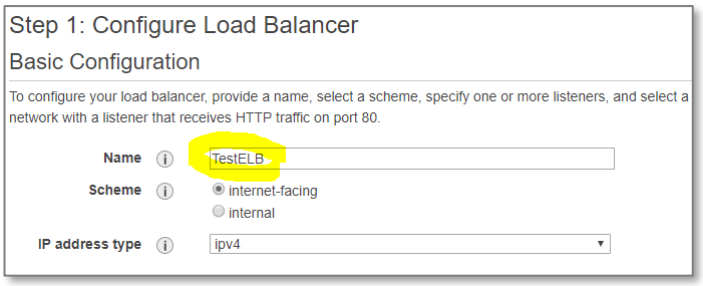
Click on “Create Load Balancer”



Select “Application Load Balancer”

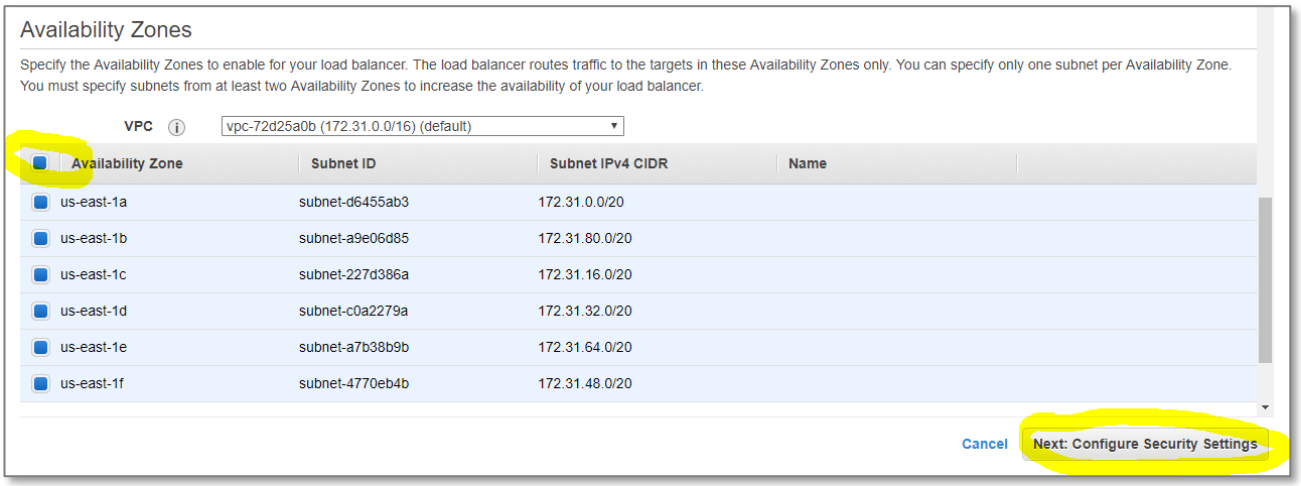


Give the ELB a name



Select all Availability Zones

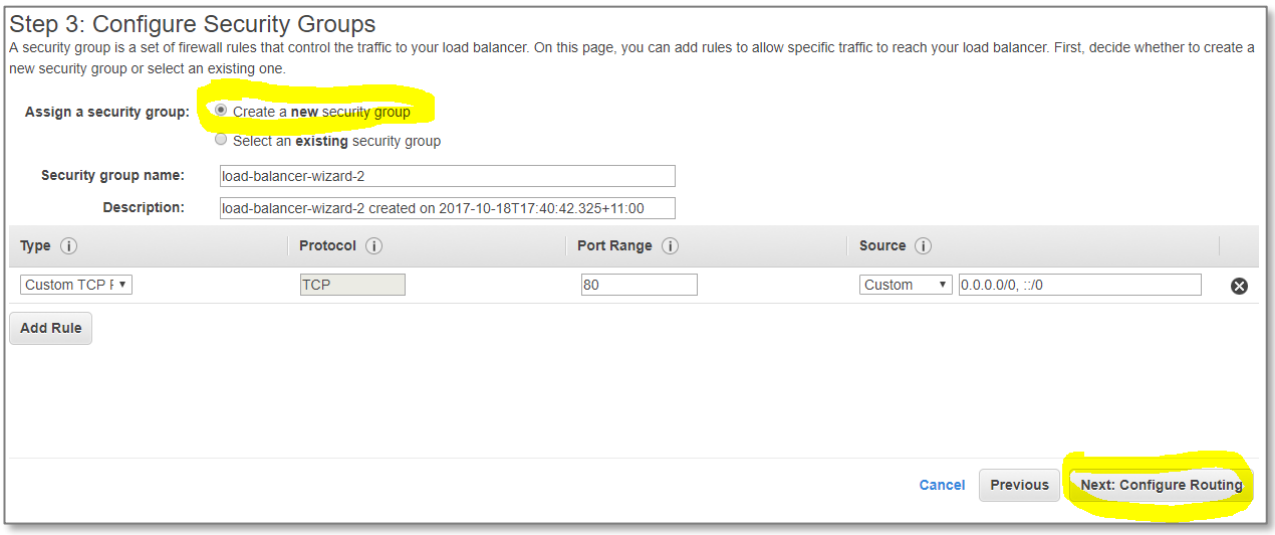
Click “Next: Configure Security Settings”



Click “Next: Configure Security Groups”

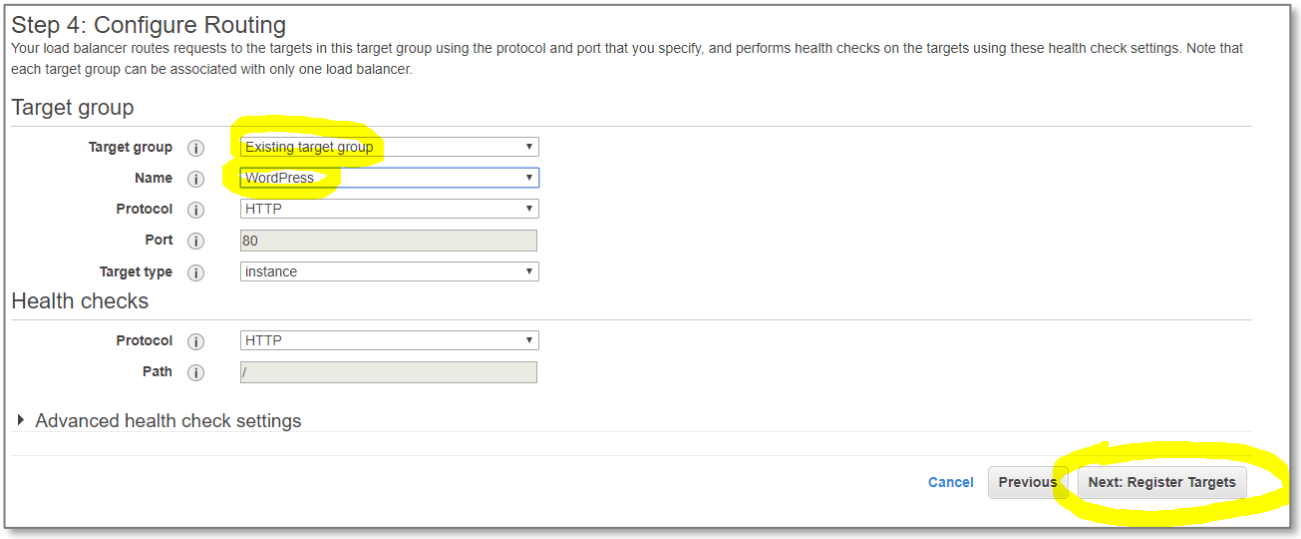
Select “Create a new security group”

Click “Next: Configure Routing”



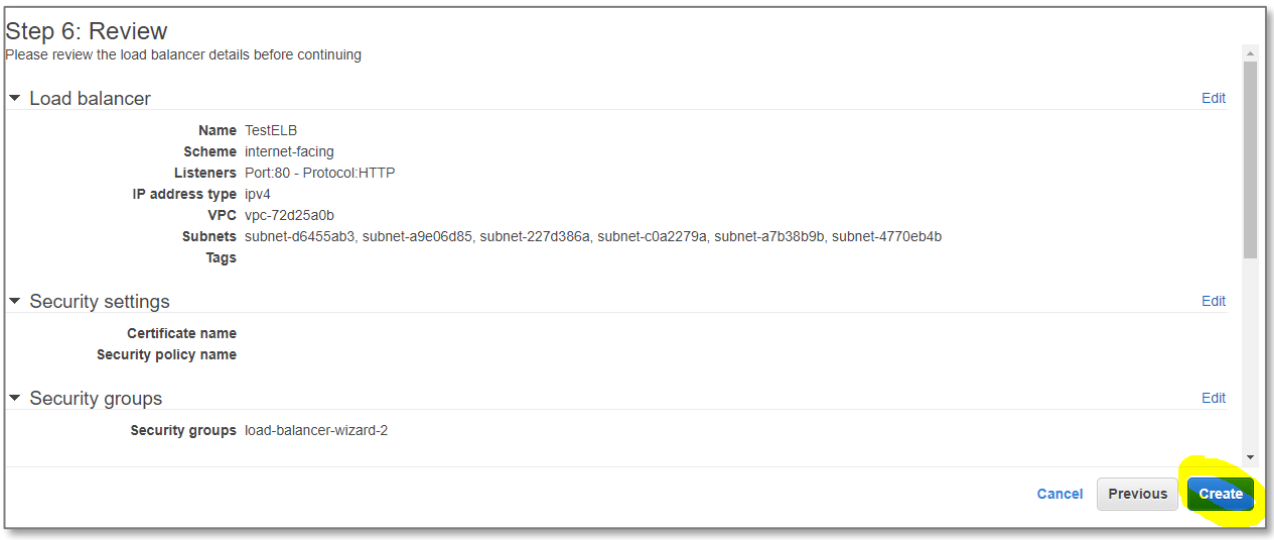
Select the WordPress Target Group as the default routing

Click “Register Targets”



Click “Next Review”

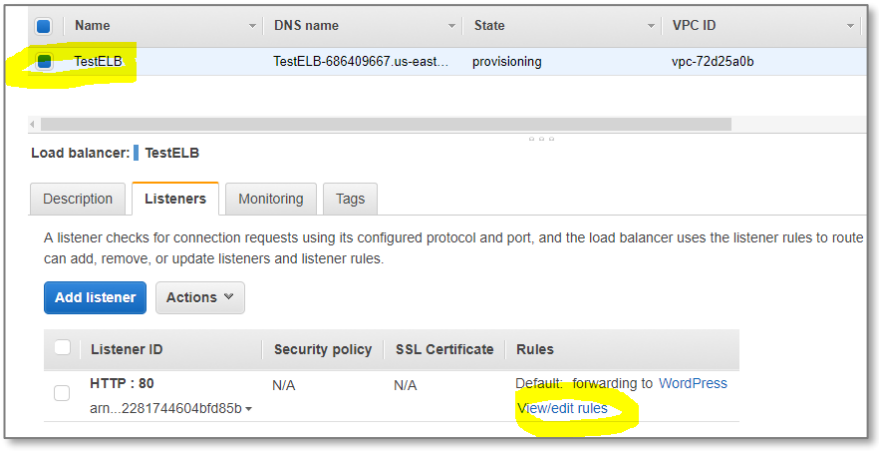
Click “Create”



Create Routing Rules

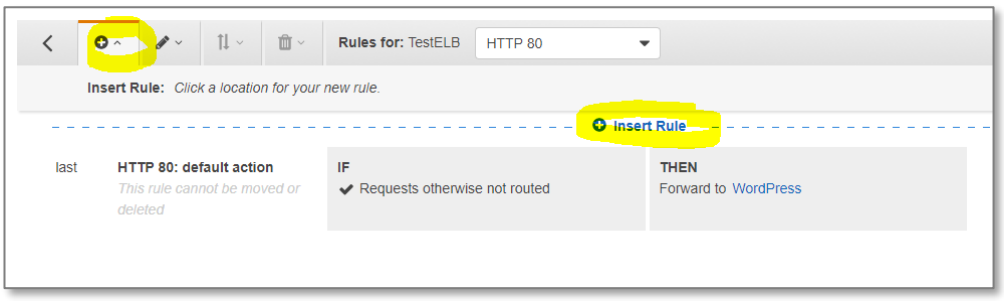
Select the “Listeners” Tab

Click “View/edit rules”



Click add rules icon

Click “Insert Rule”

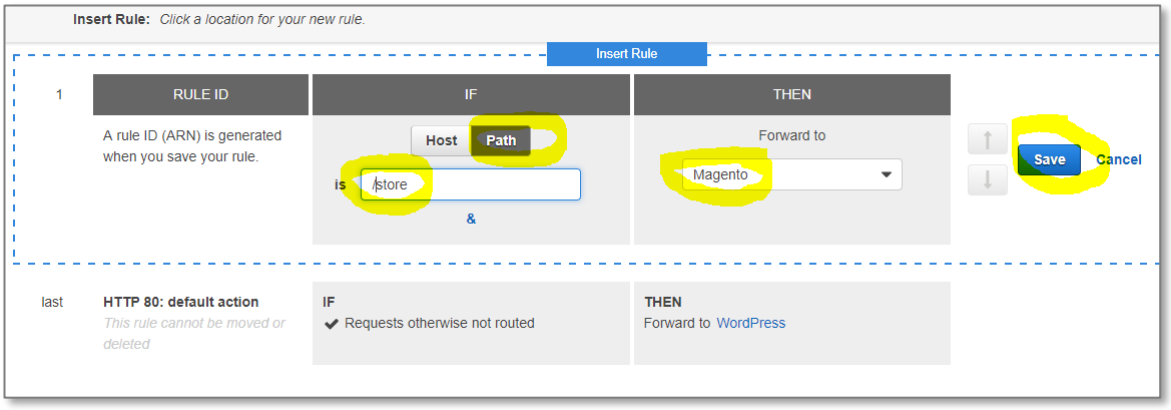


Select “Path”

Enter “/store”

Select “Magento” Target Group

Click “Save”



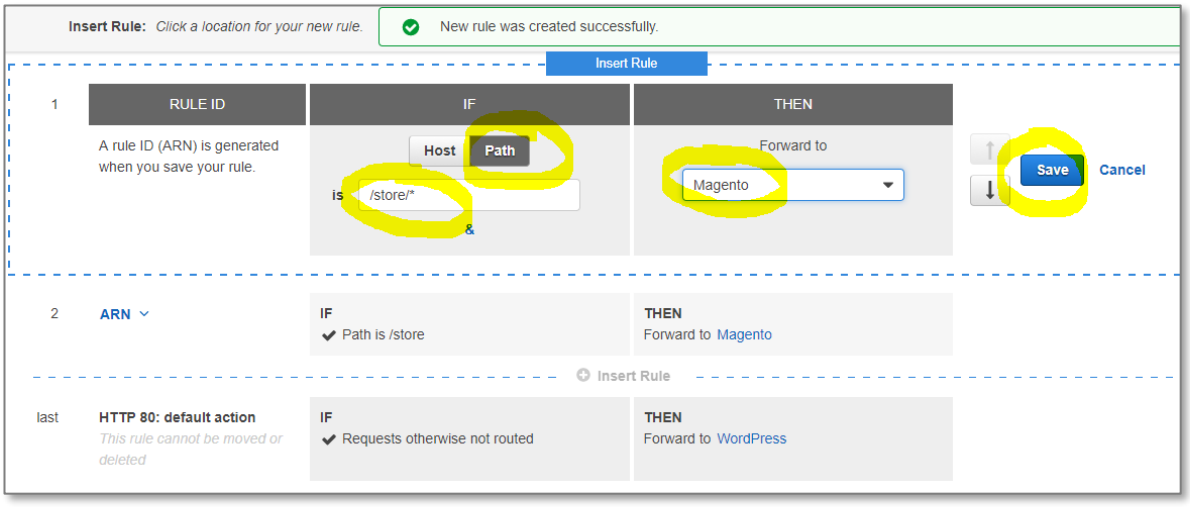
Click add rules icon

Click “Insert Rule”

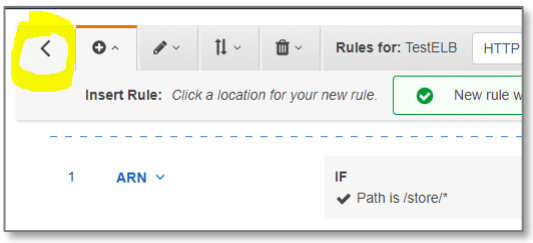
Select “Path”

Enter “/store/*”

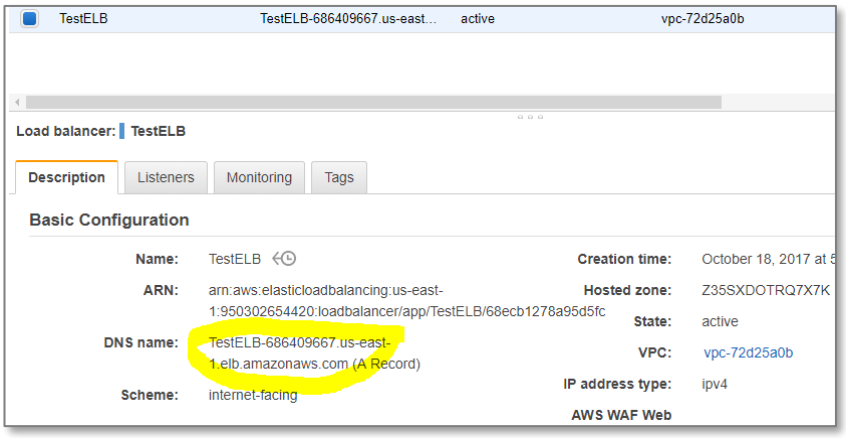
Select “Magento” Target Group



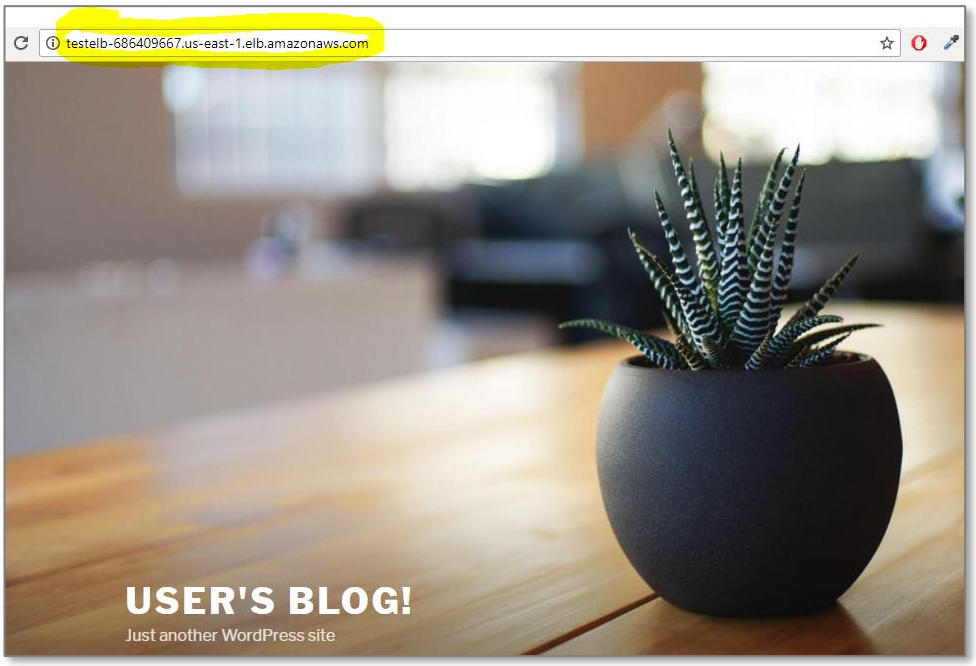
Go Back to Load Balancer



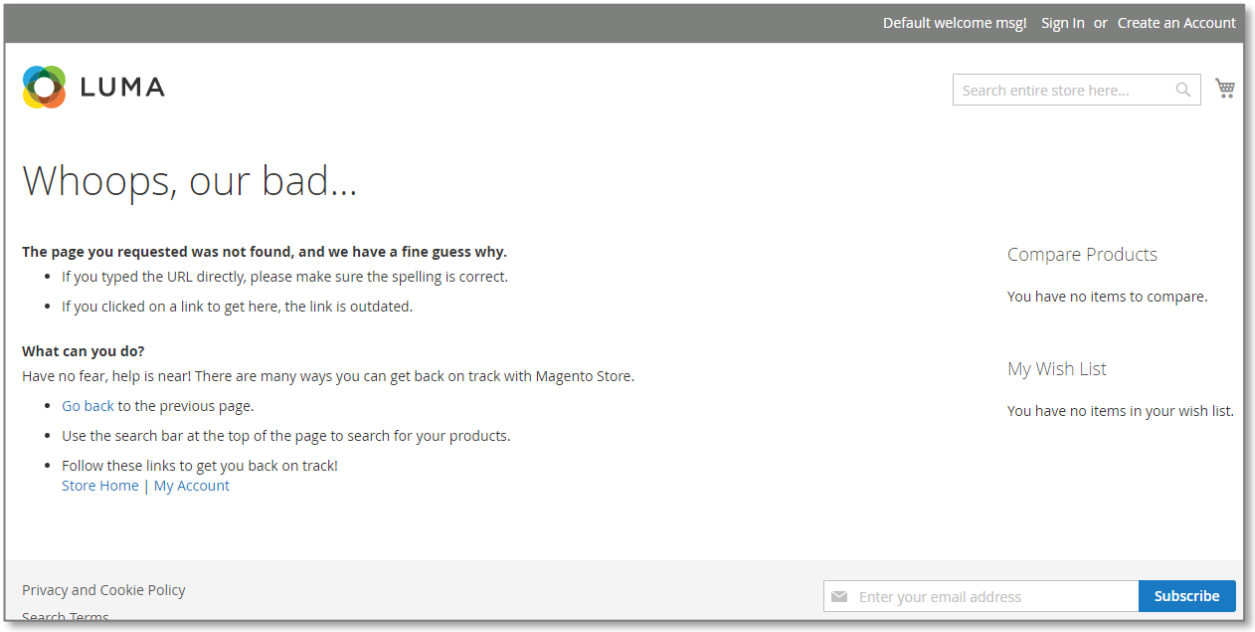
Copy the Load Balancer URL



Open URL in Browser to see the WordPress Blog



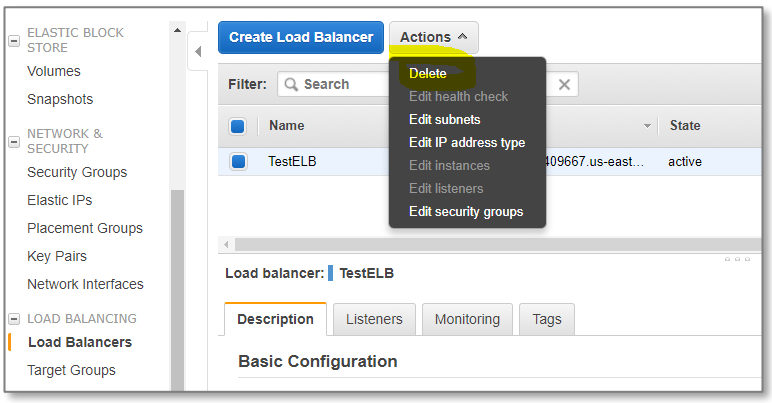
Now add “/store” to the end of the URL to see the Magento Store



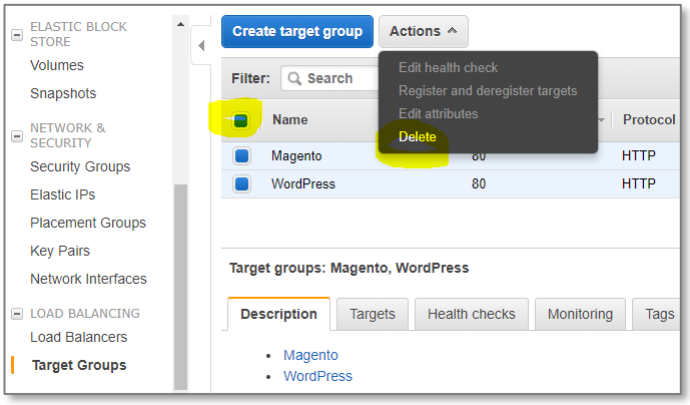
Although it has successfully routed to Magento, the Magento “not found” page will be show as the Magento home page settings will need to be changed to the “/store” path.

Clean Up

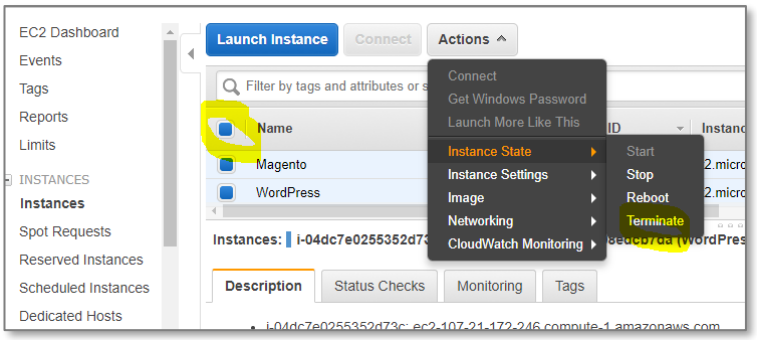
Delete the Load Balancer



Delete the Target Groups



Terminate the instances



Implementing a Network Load Balancer and Auto Scaling

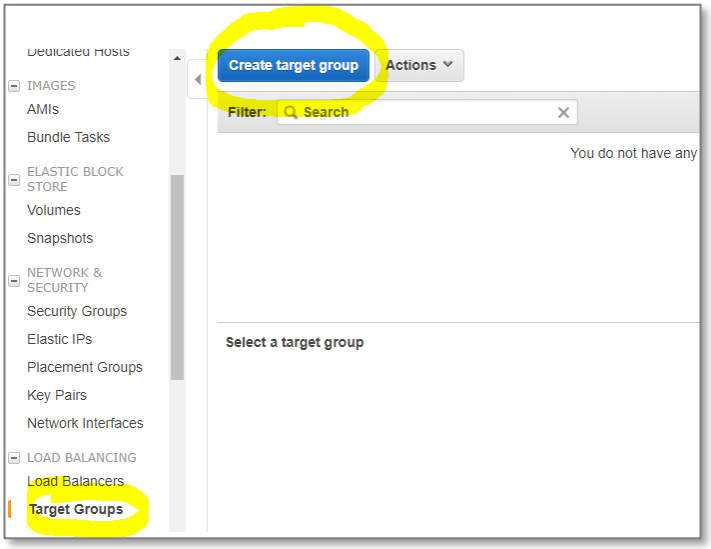
In this section we will create an Auto Scaling group. We will then add this to a target group and create a network load balancer to serve TCP traffic to the Auto Scaling group.

Creating a Target Group

Click on the services menu and select EC2.

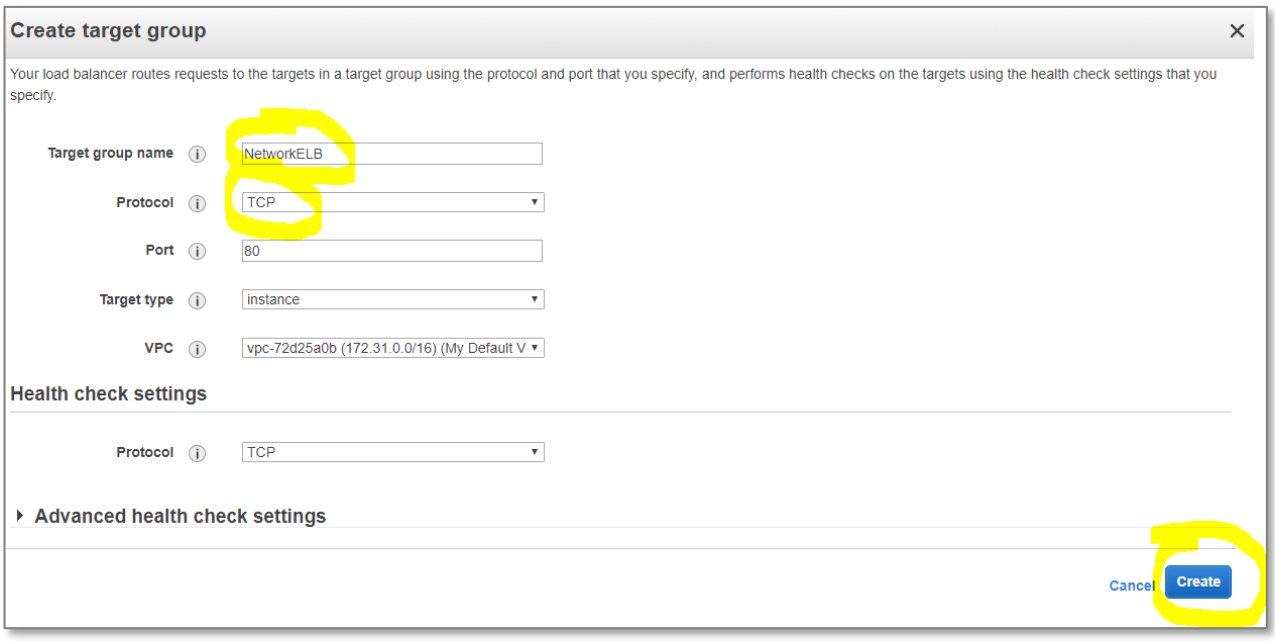
Scroll down and select “Load Balancers”

Click “Create Target Group”

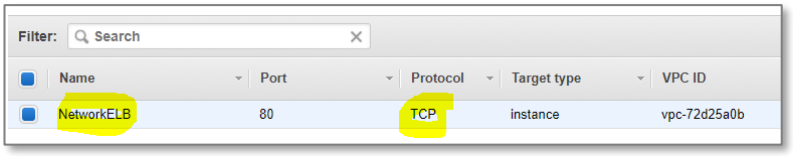


Create a target Group called “NetworkELB”

Select “TCP” for Protocol



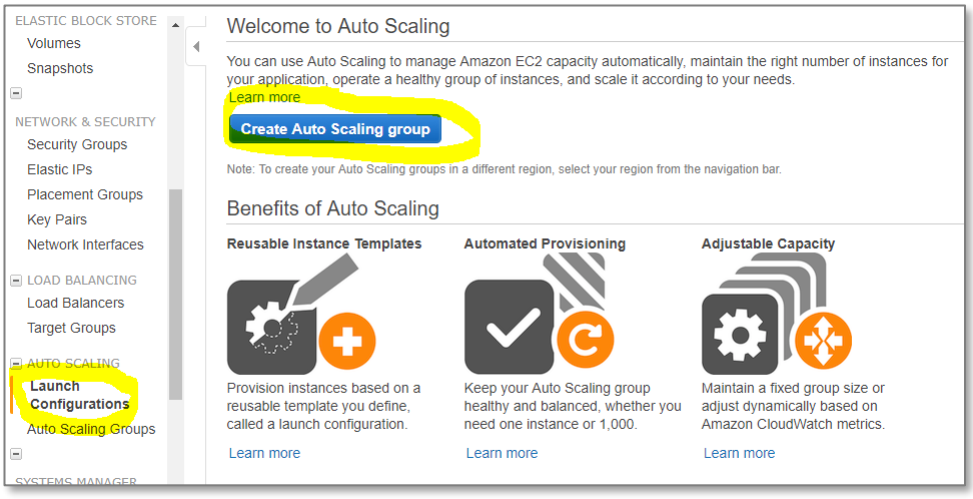
Click “Create”



Creating an Auto Scaling Launch Configuration

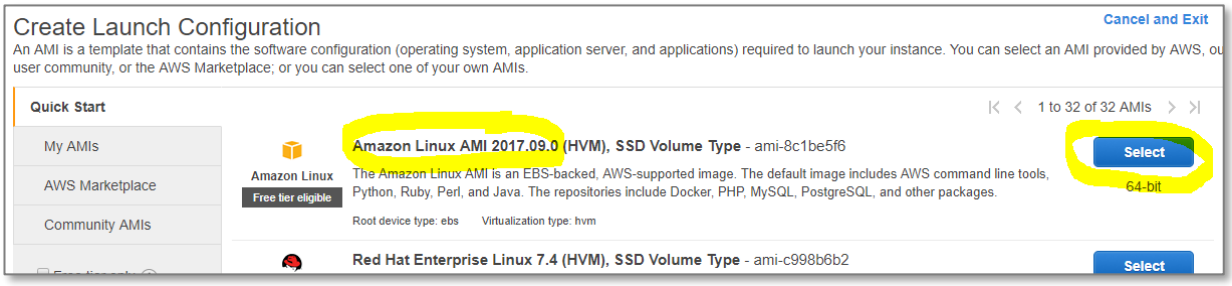
Select “Launch Configurations”

Click “Create Auto Scaling group”



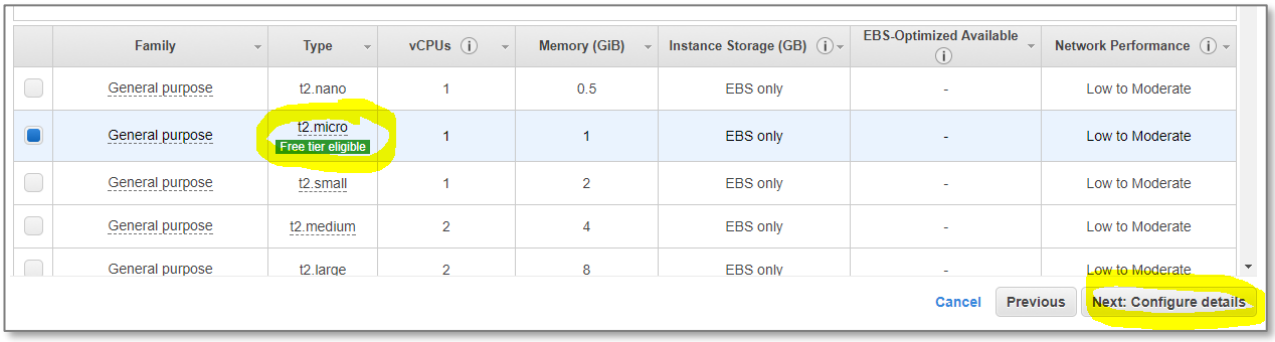
Click “Create launch configuration”

Select the Amazon Linux AMI



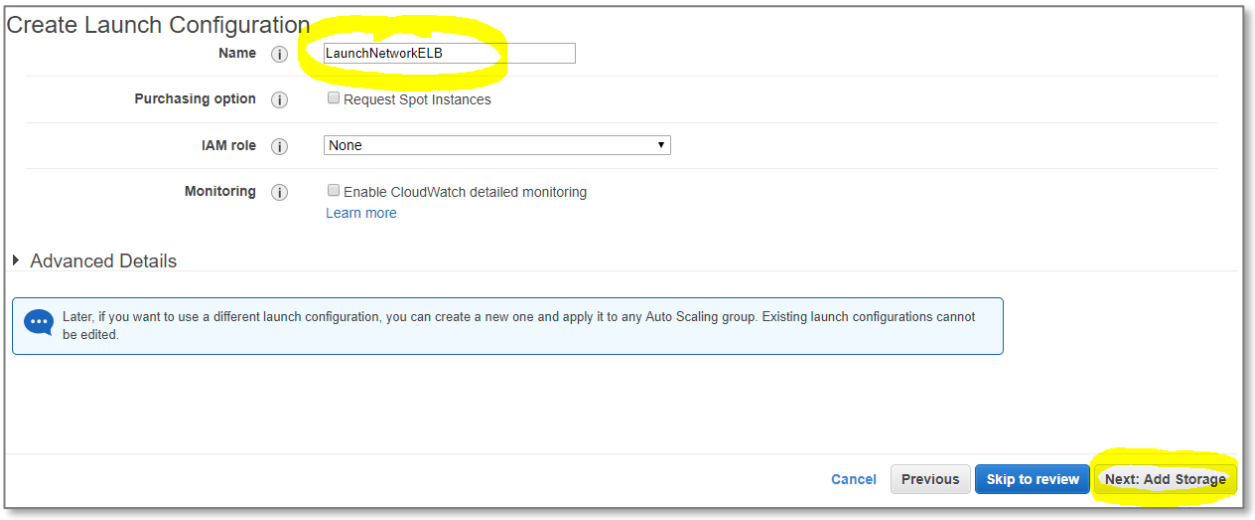
Select t2 micro instance type

Click “Next: Configure details”

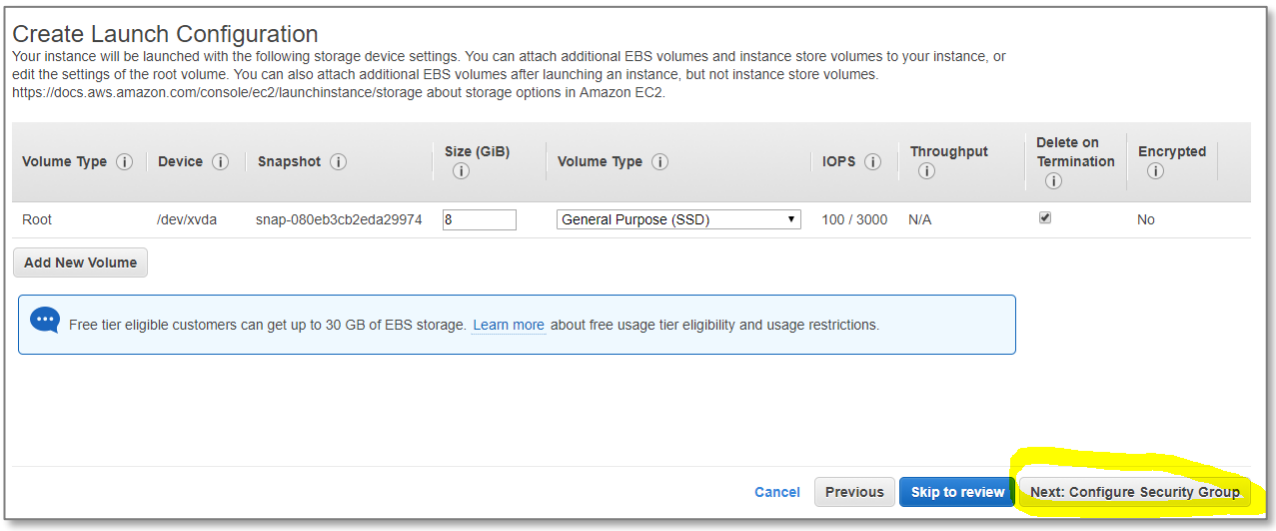


Give the Launch Configuration a name

Click “Next: Add Storage”



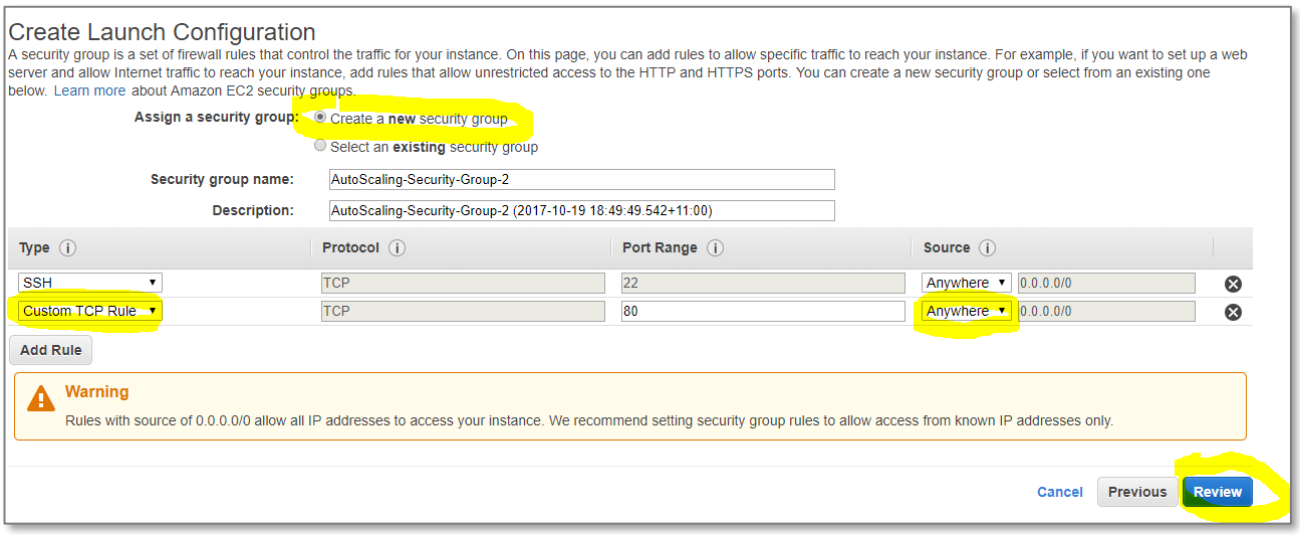
Click “Next: Configure Security Group”



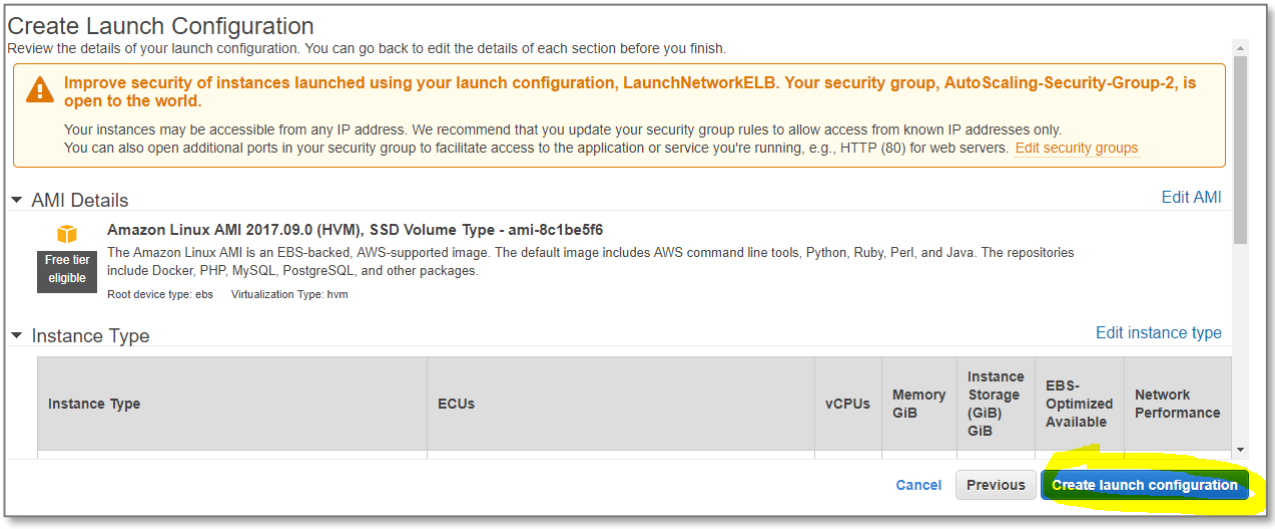
Select “Create a new security group”

Add a rule for TCP on port 80, source anywhere.

Click “Review”



Click “Create Launch Configuration”



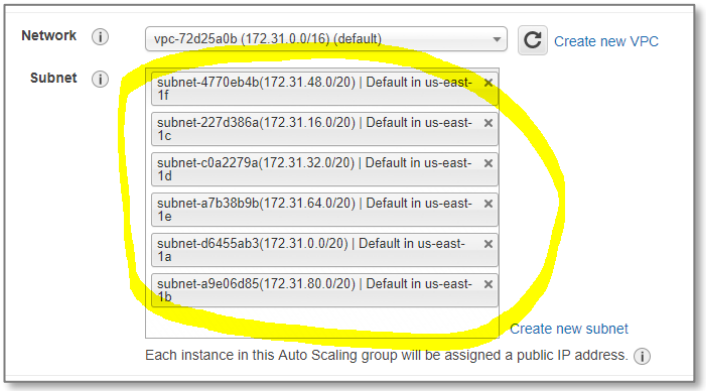
Then select a key pair to finish creating the launch configuration.

Creating an Auto Scaling Group

You will now be presented with the Auto Scaling Group creation page

Give the Group a name

Select the subnets for each availability zone.

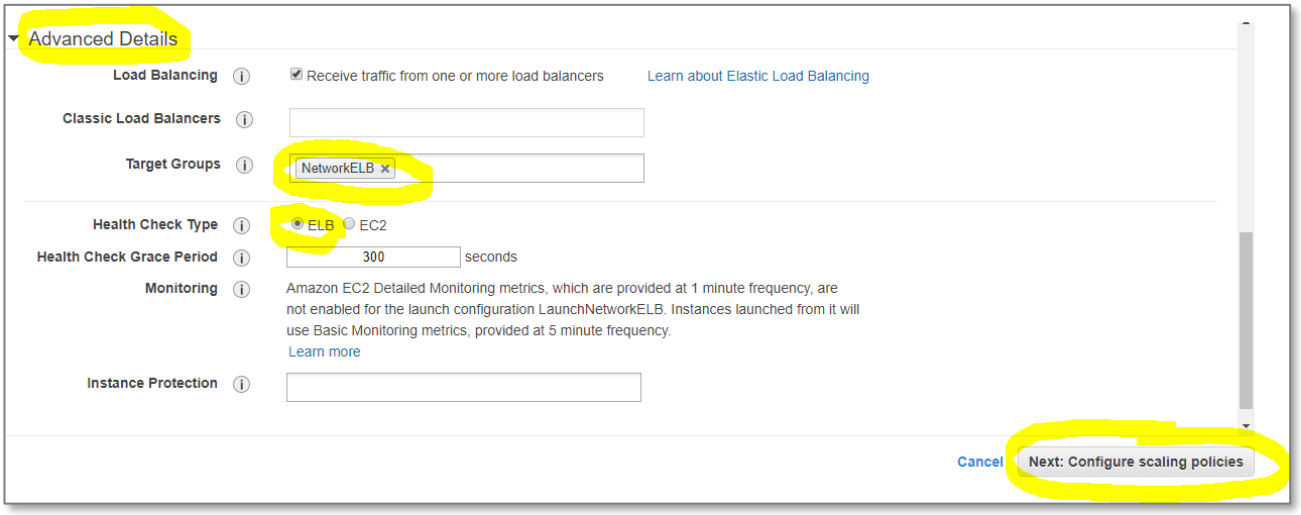


Click to expand “Advanced Details”

Select the Target Group we created

Select “ELB” for Health Check Type

Click “Next: Configure scaling policies”



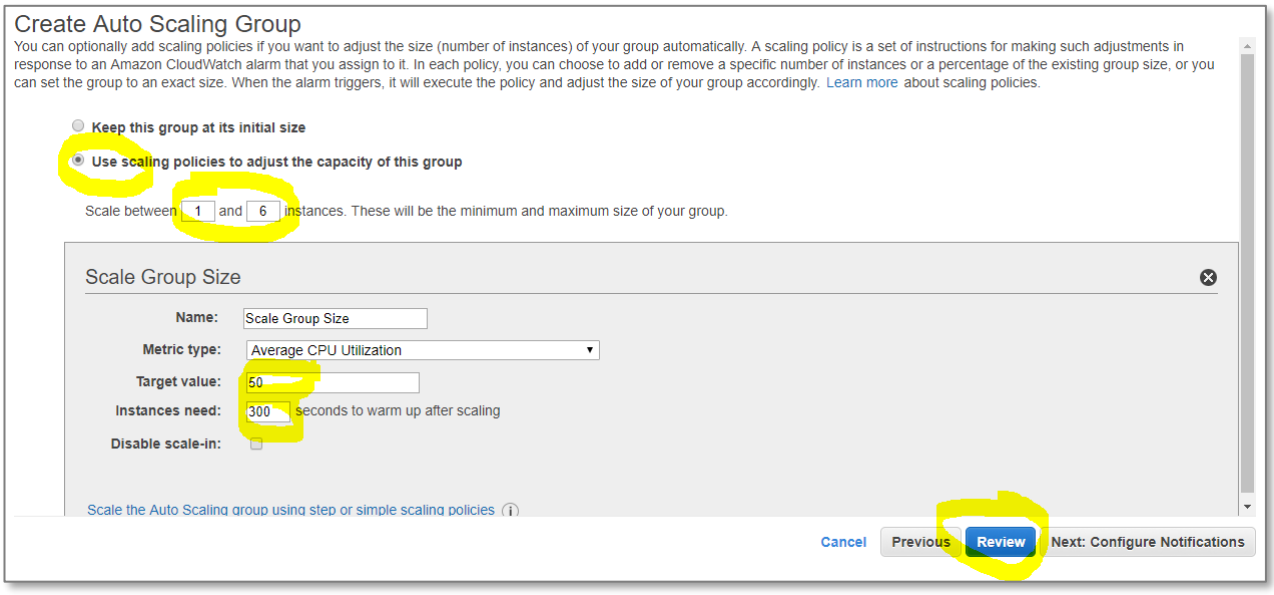
Select “Use scaling policies to adjust the capacity of this group”

Scale between 1 and 6 instances.

Target value: 50

Instances need: 300 seconds to warm up after scaling

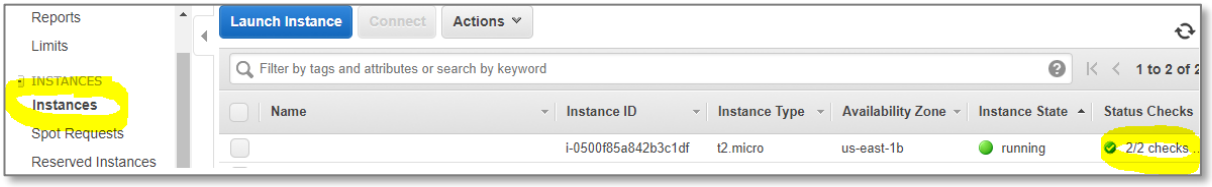
Click “Review”



Click “Create Auto Scaling Group”

Select Instances and you will eventually see an EC2 instance being launched by the Auto Scaling Group.

Wait the until Status Checks have been completed.

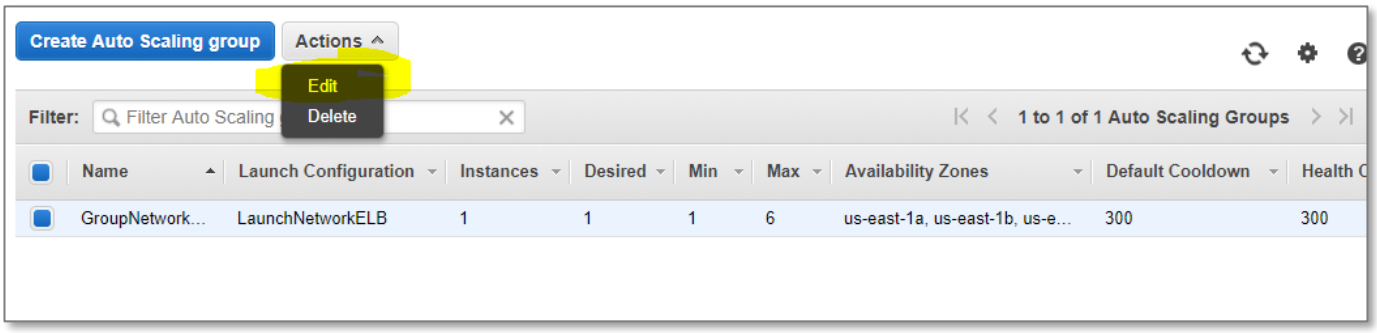


Adding the Auto Scaling Group to an ELB Target Group

Go back to the Auto Scaling page

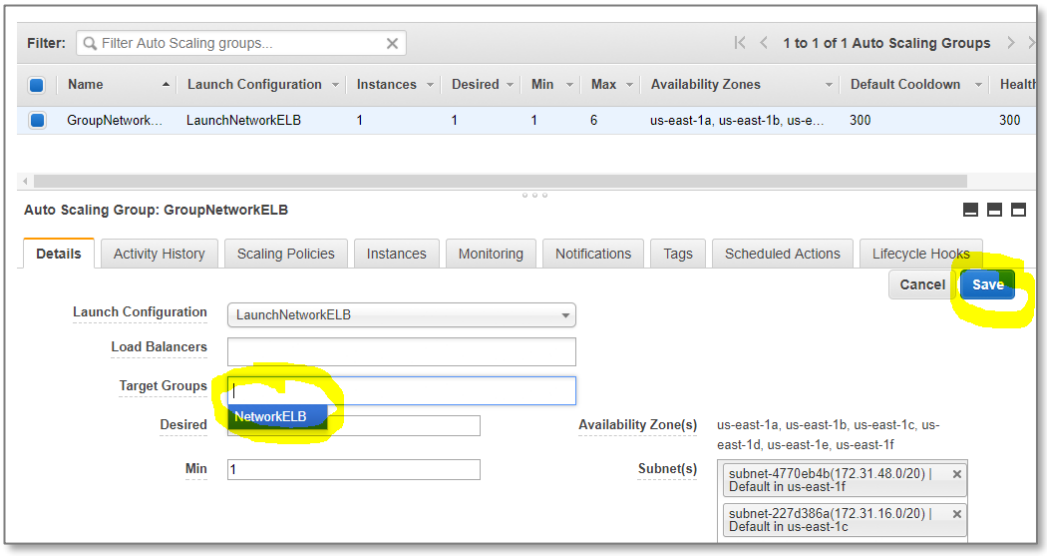
Select the Auto Scaling Group

Select “Actions” “Edit”

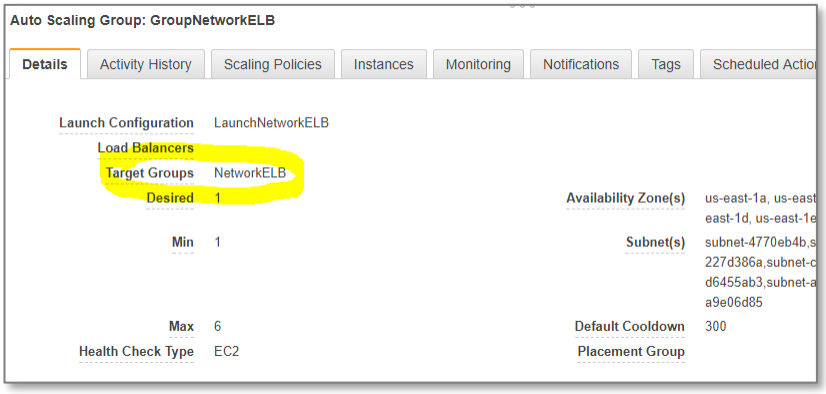


Select the Target Group we created in the “target Groups” field

Click “Save”



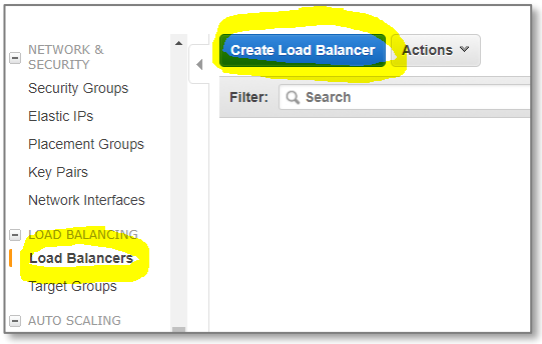
The Target Group will now be added to the Auto Scaling Group



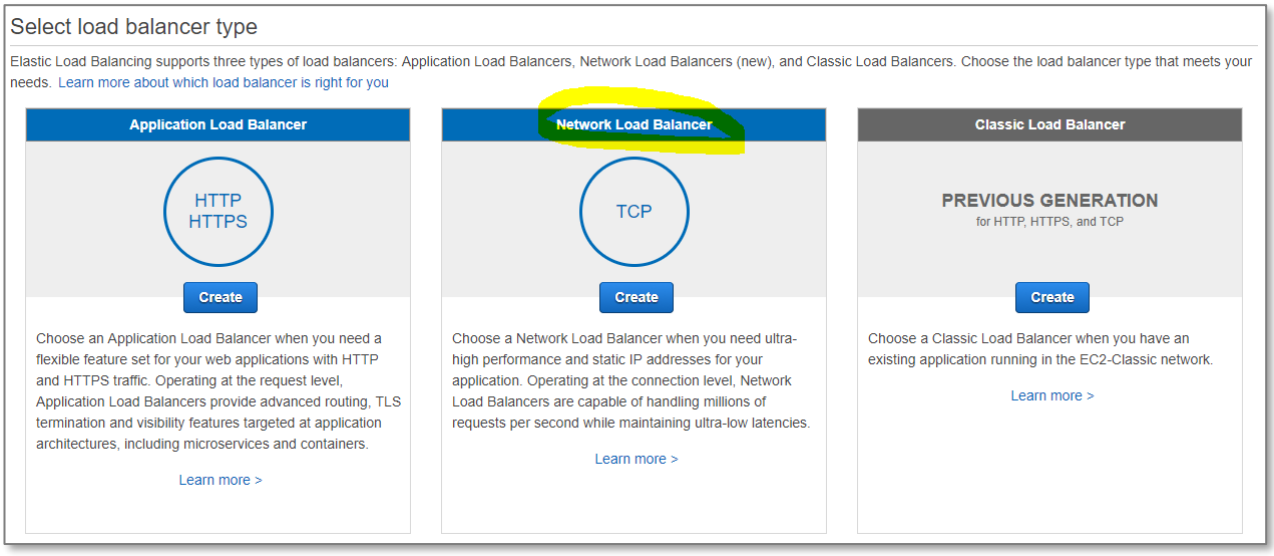
Creating a Network ELB

Select “Load Balancers”

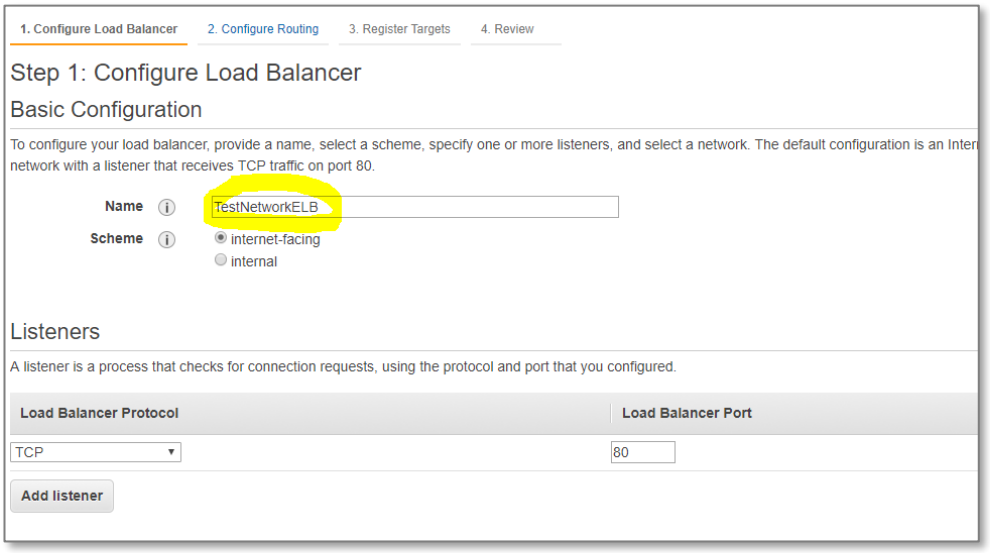
Click “Create Load Balancer”



Select “Network Load Balancer”

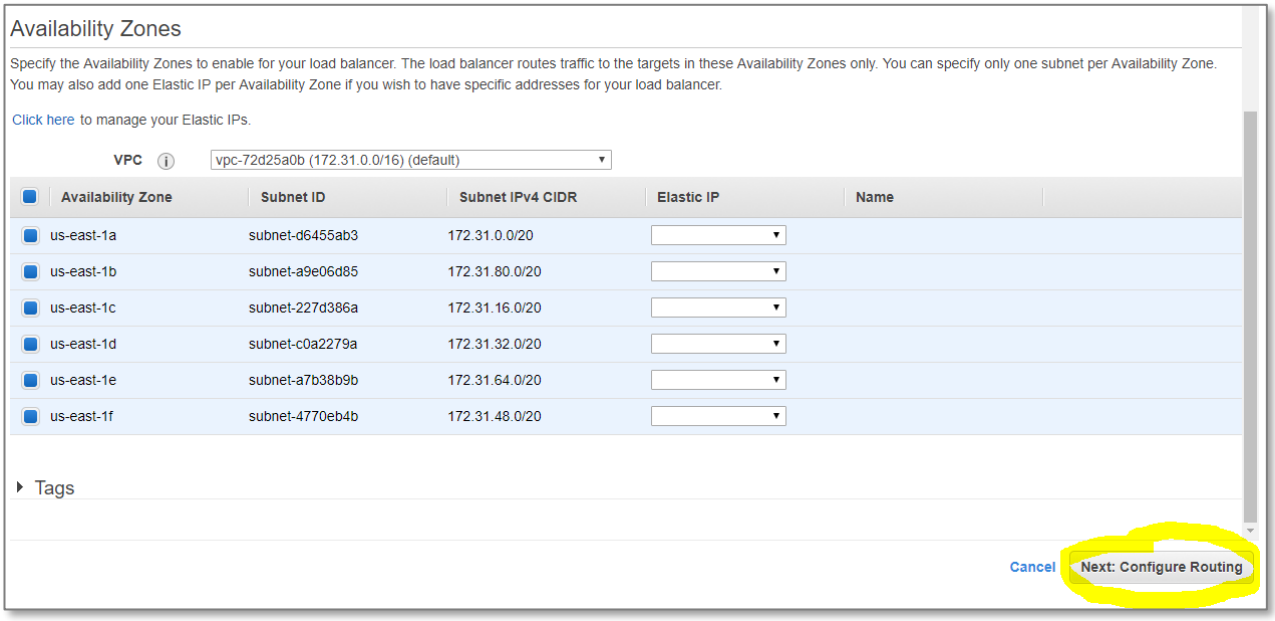


Give the Load Balancer a name



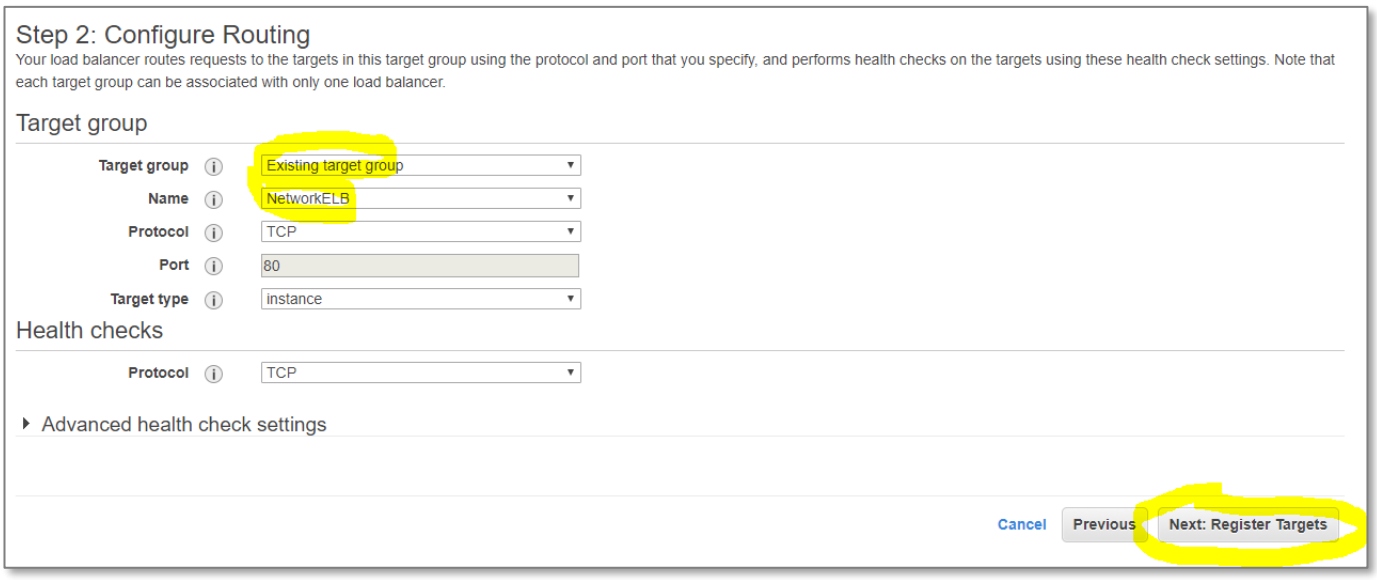
Select all Availability Zones

Click “Next: Configure Routing”



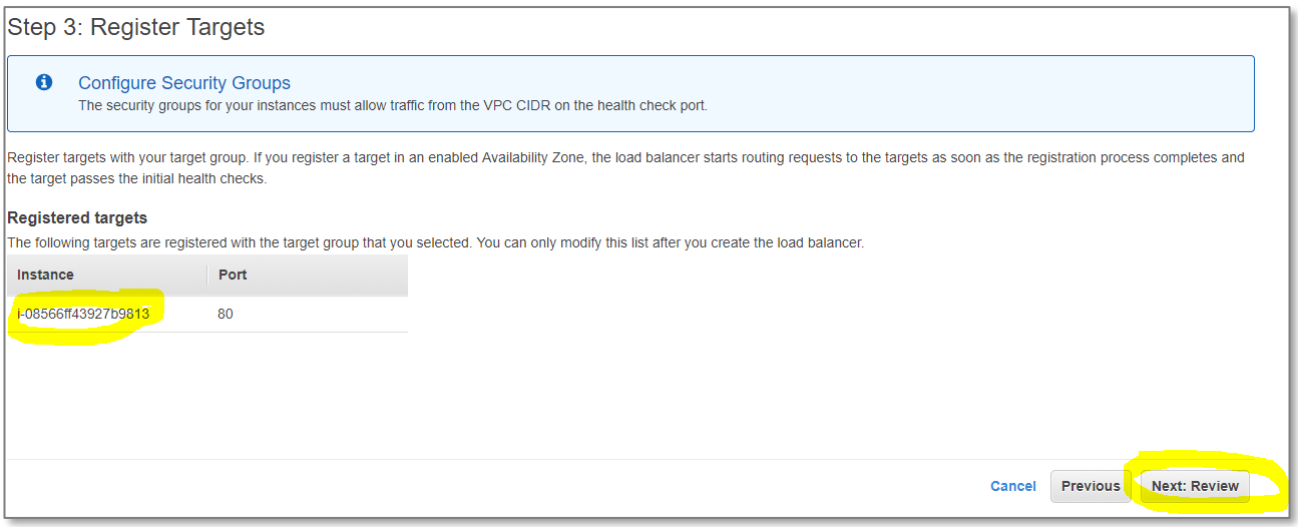
Select the existing “NetworkELB” Target Group we created previously.

Click “Next: Register Targets”

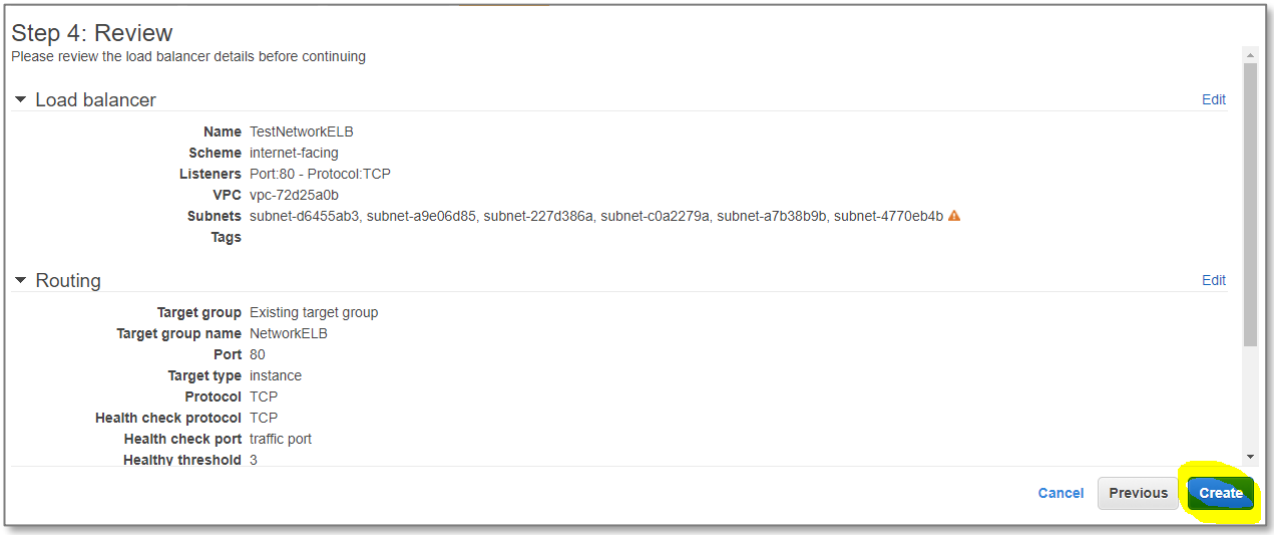


You will see the EC2 instance created by the Auto Scaling has been automatically added to the Target Group.

Click ”Next: Review”



Click “Create”

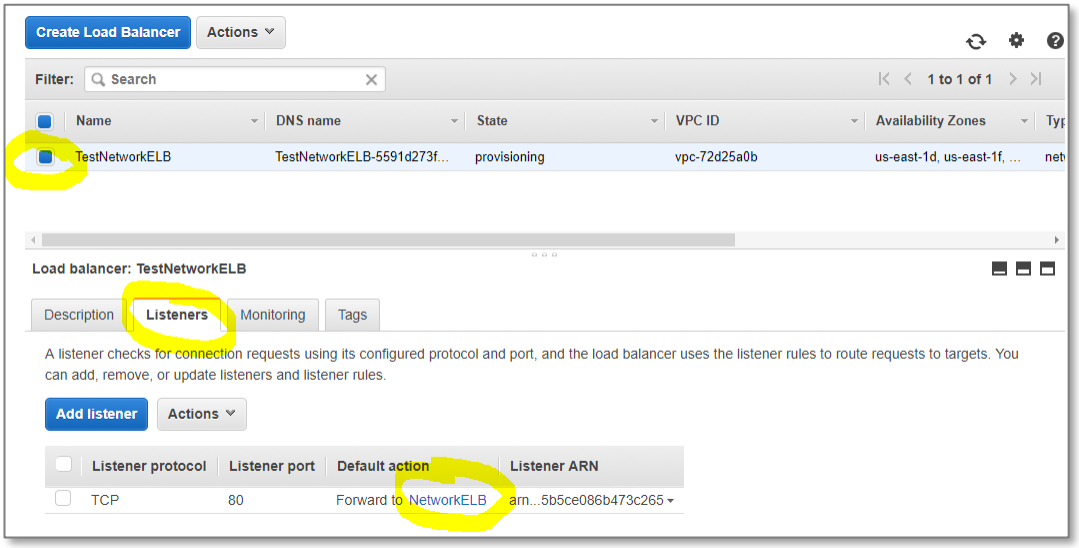


You can now see your Network ELB listed.

Click on the ELB and select the “Listeners” tab.

You can now see the listener is forwarding requests to the Target Group.

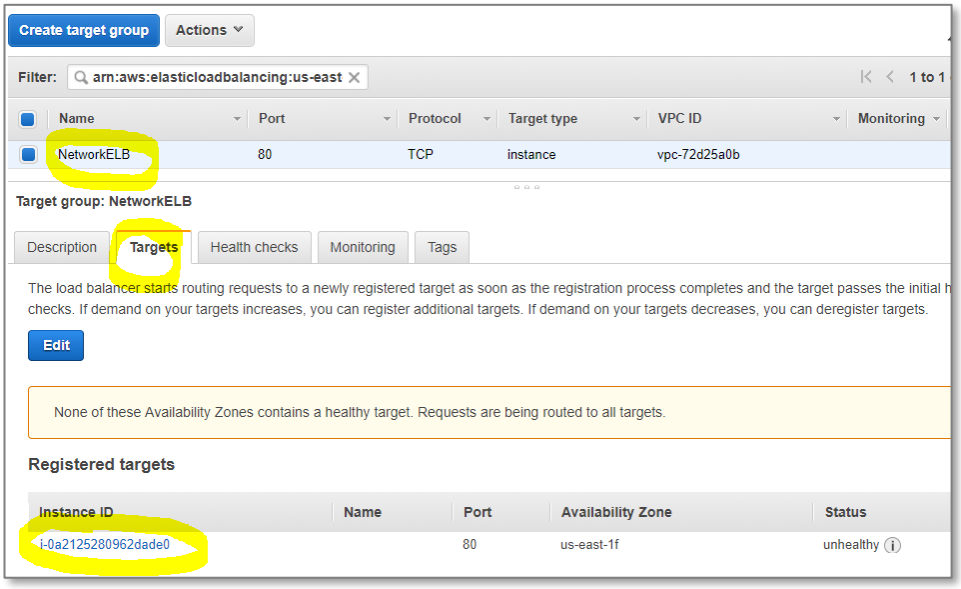
Click on the target group.



Select the “Targets” tab

You will see an instance is registered with the ELB.

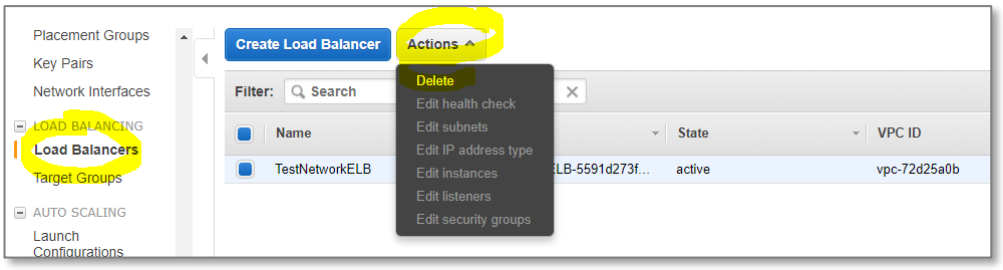
* Please note the instance will be marked as unhealthy because no application is running on the instance to respond to the TCP requests.



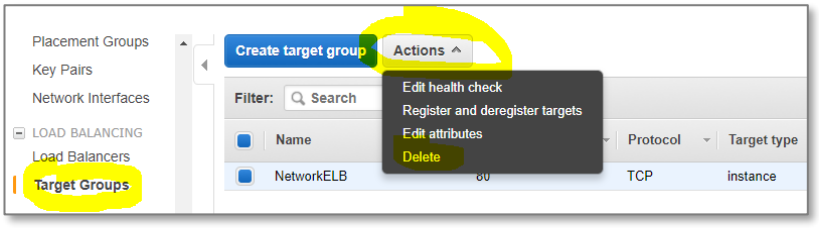
Clean Up

We will now delete the resources.

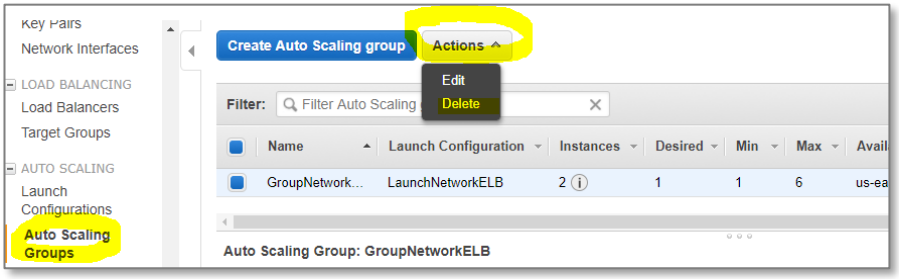
First delete the ELB



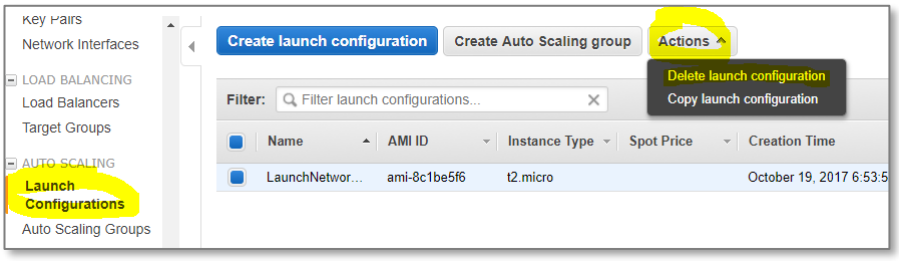
Now delete the Target group



Now delete the Auto Scaling group



Now delete the Launch Configuration



Check that the instances are terminating.