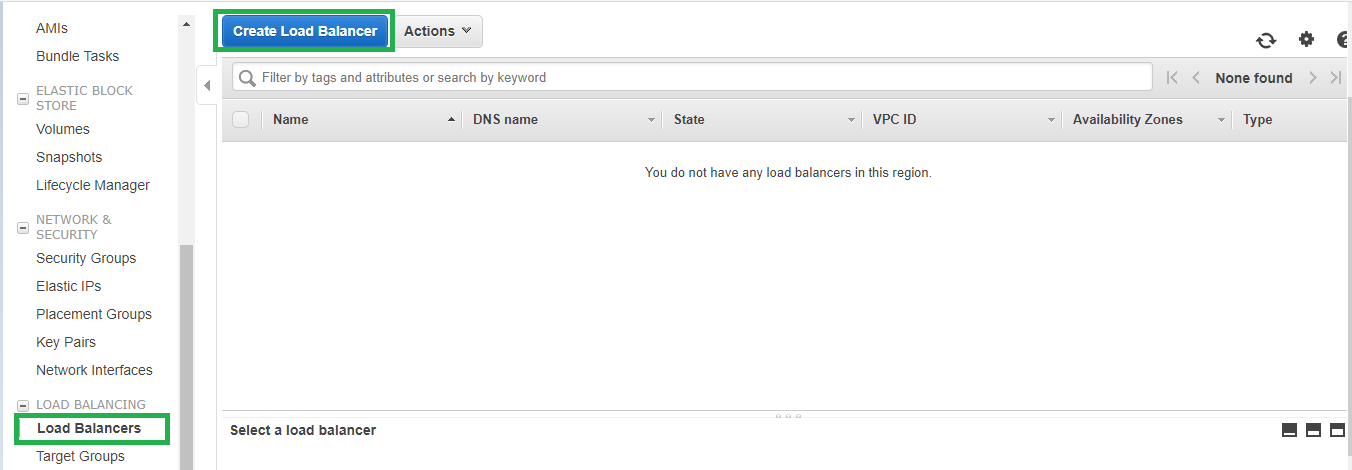
**Lab: Configuring ELB with ASG**

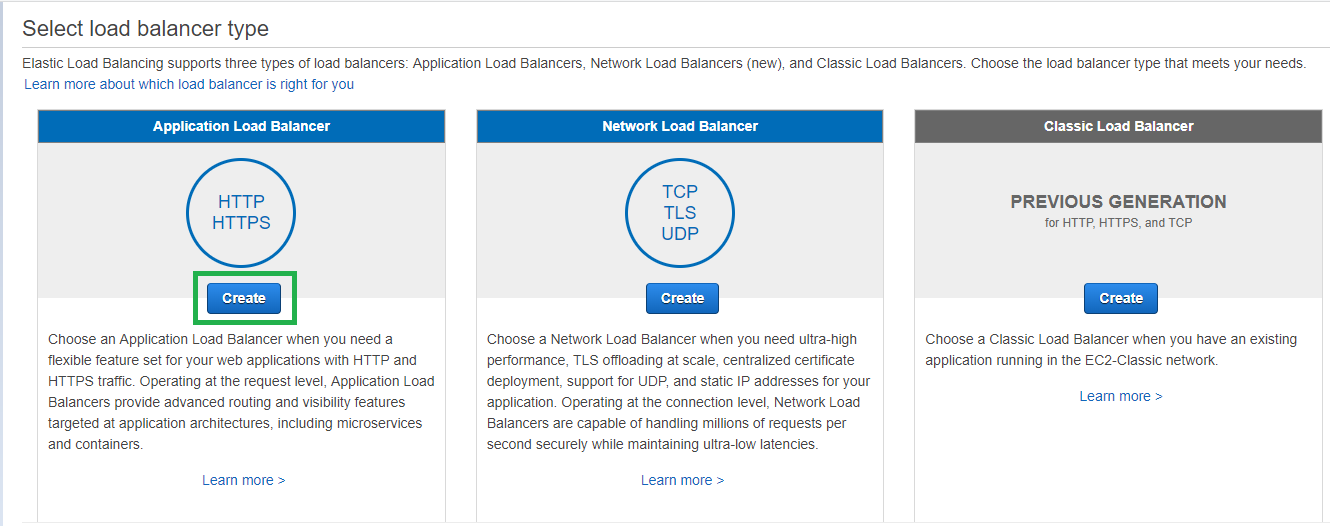
**Services Used: ELB & ASG**

**ELB configuration:**

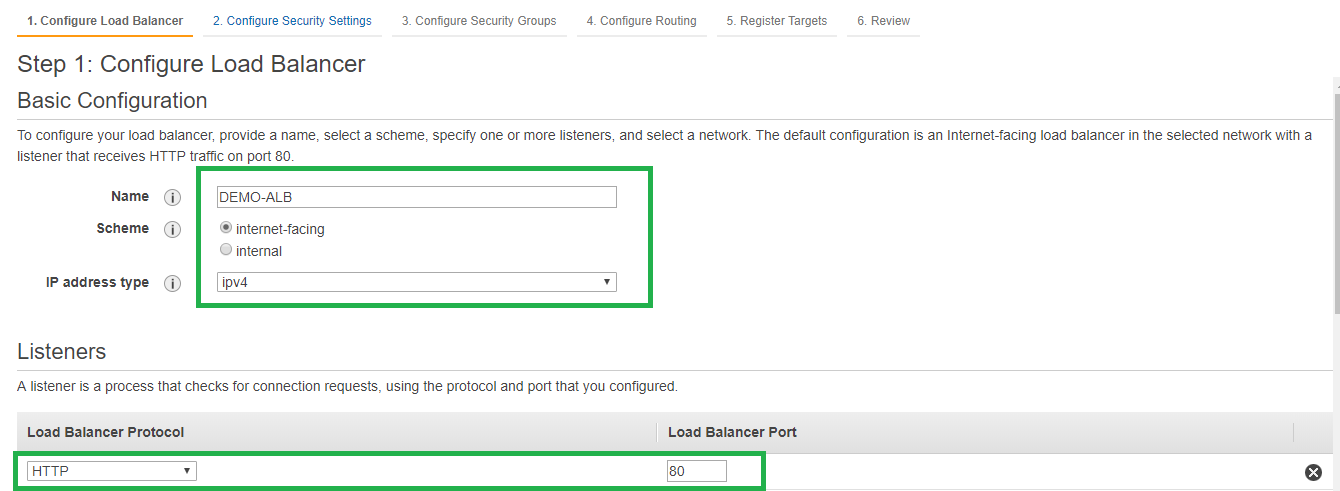
1. Go to Ec2 Service --> Load Balancers 🡪 Create Load Balancer



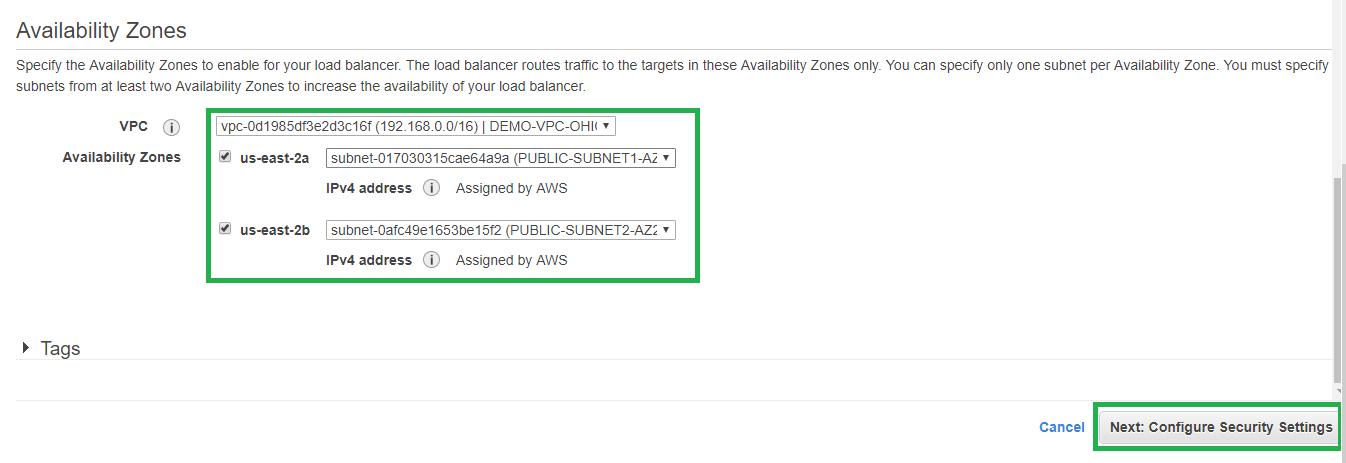
1. Select Application Load Balancer and Create



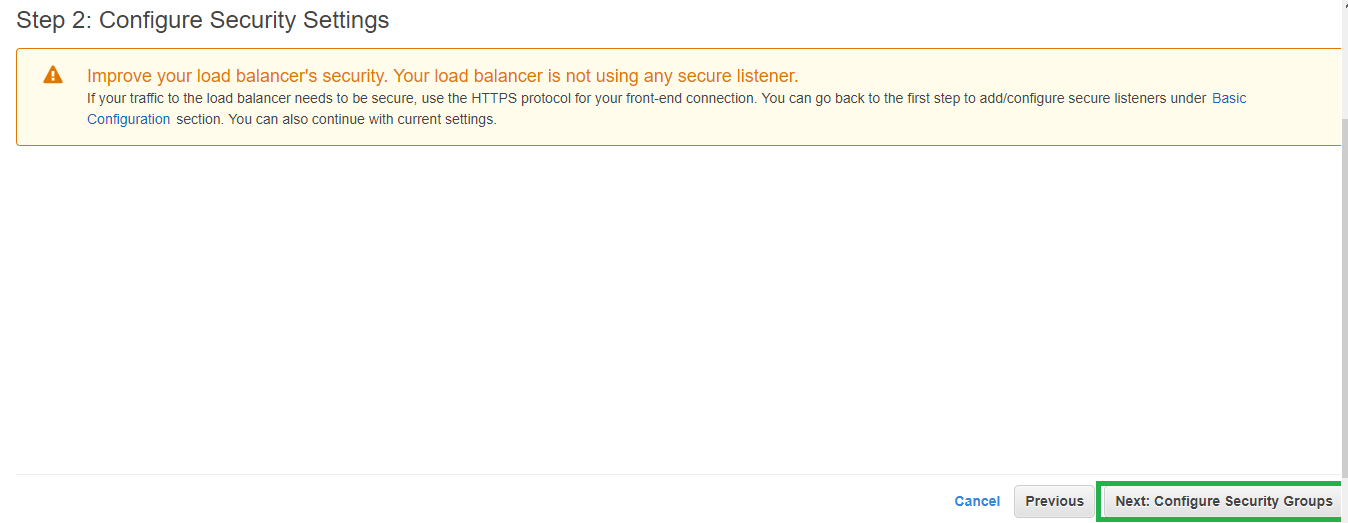
1. Provide ELB Name, Scheme as Internet Facing, IP Address type as IPv4, Listener as HTTP.



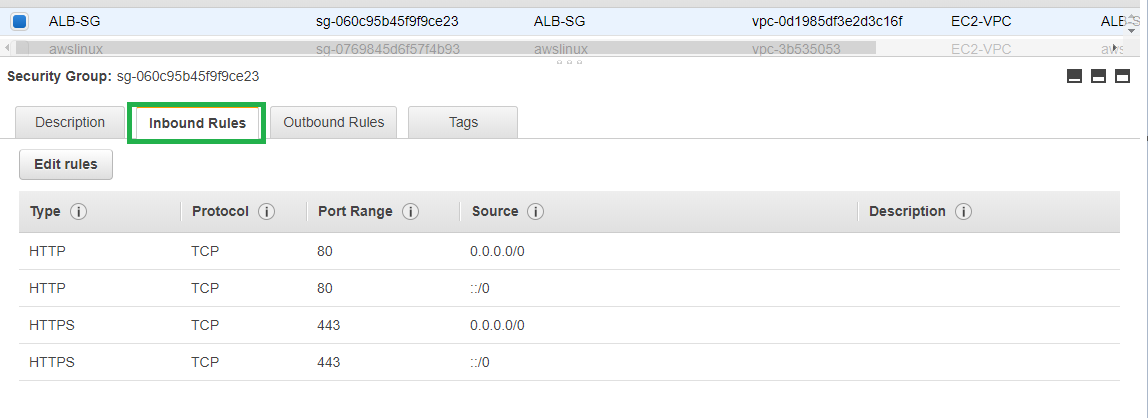
1. Select all availability zones that you may want to use for ELB and select public subnets in those AZs.

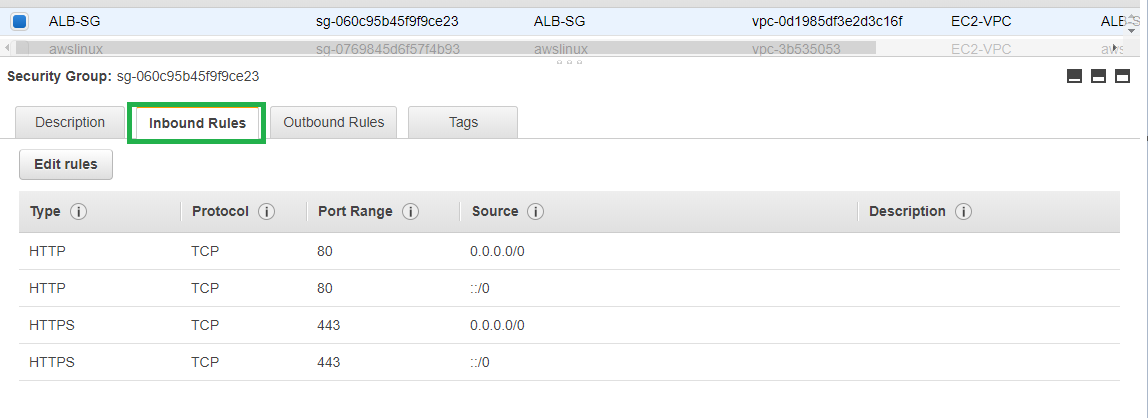


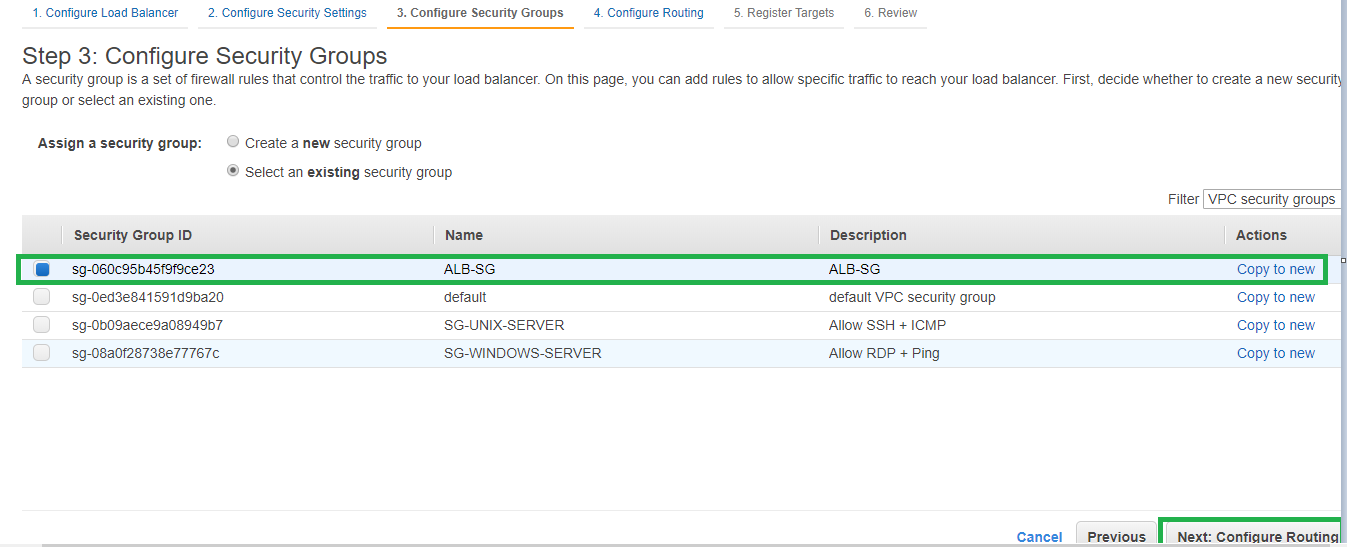
1. Configure Security Groups.



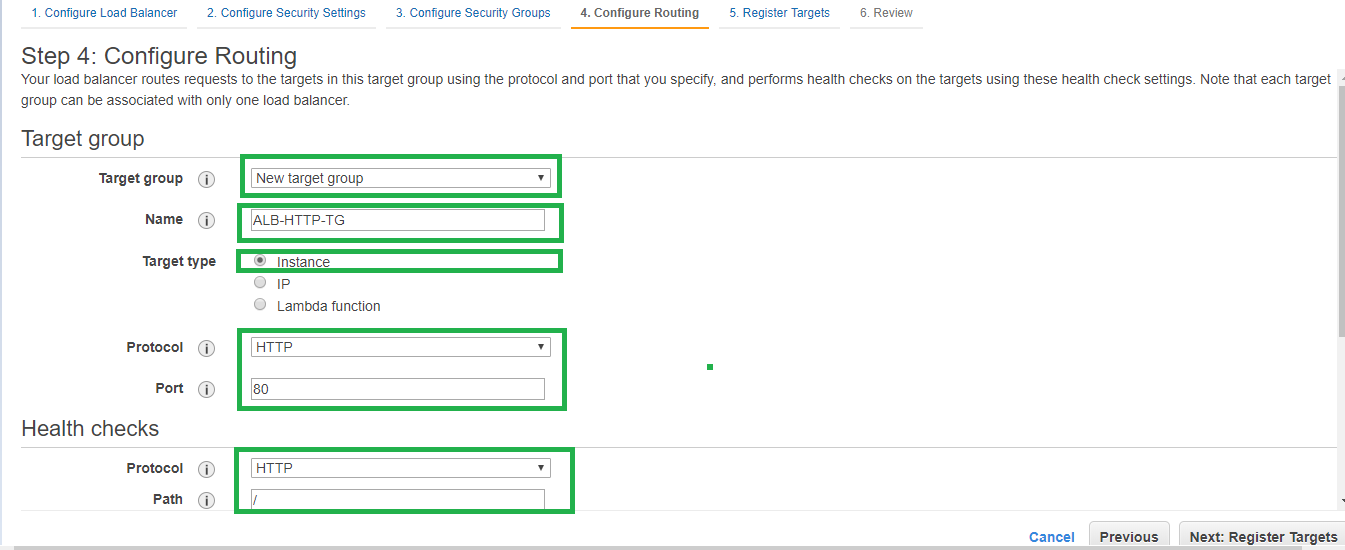
Ensure Security Groups allow HTTP/HTTPS from Internet.

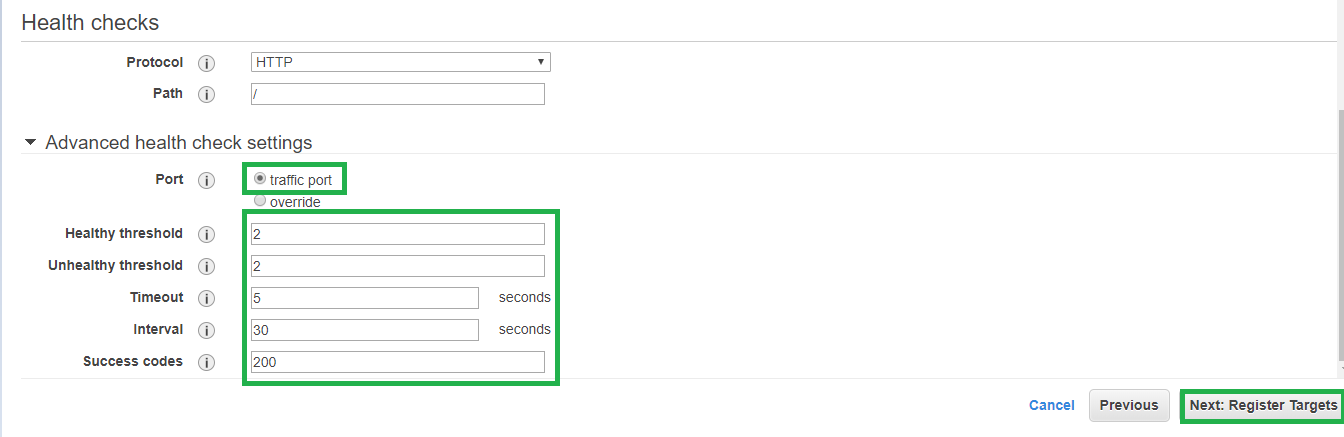




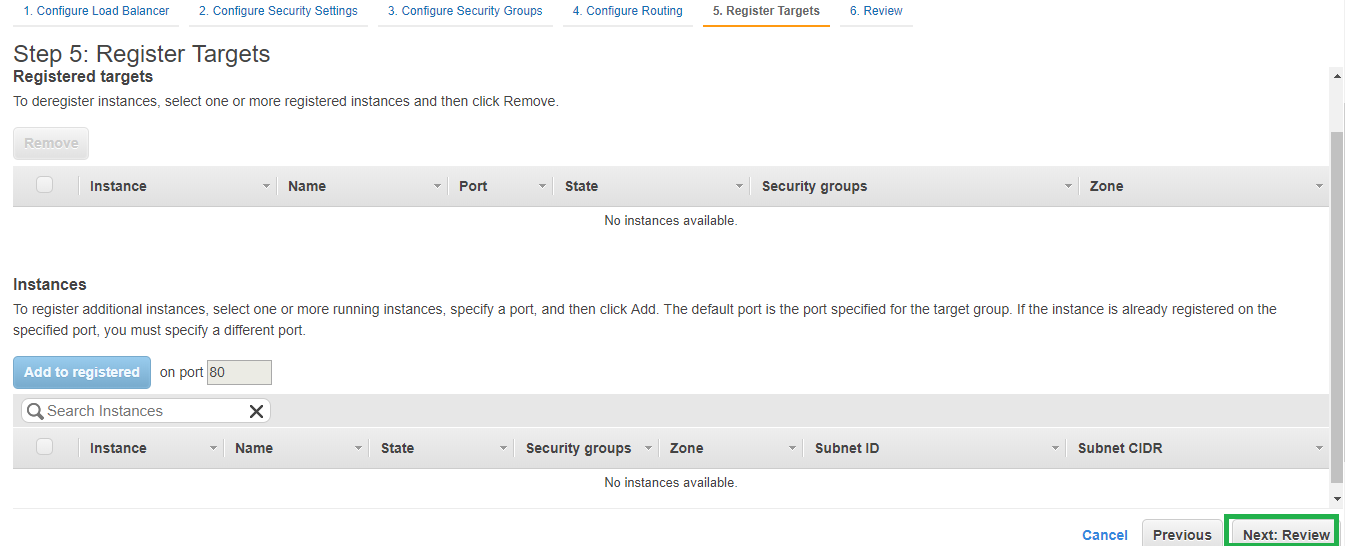


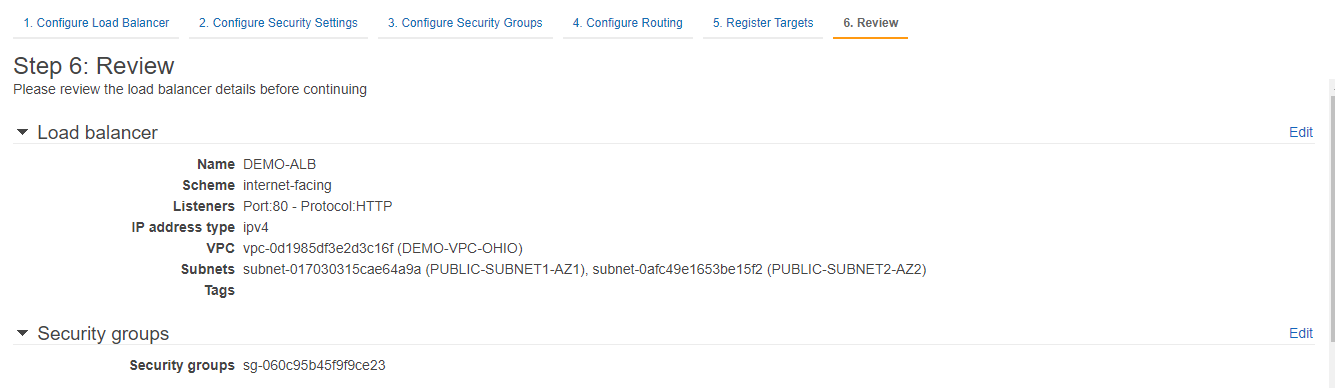
1. Configure Routing with Target Type as Instance, Protocol as HTTP, and Port as 80.

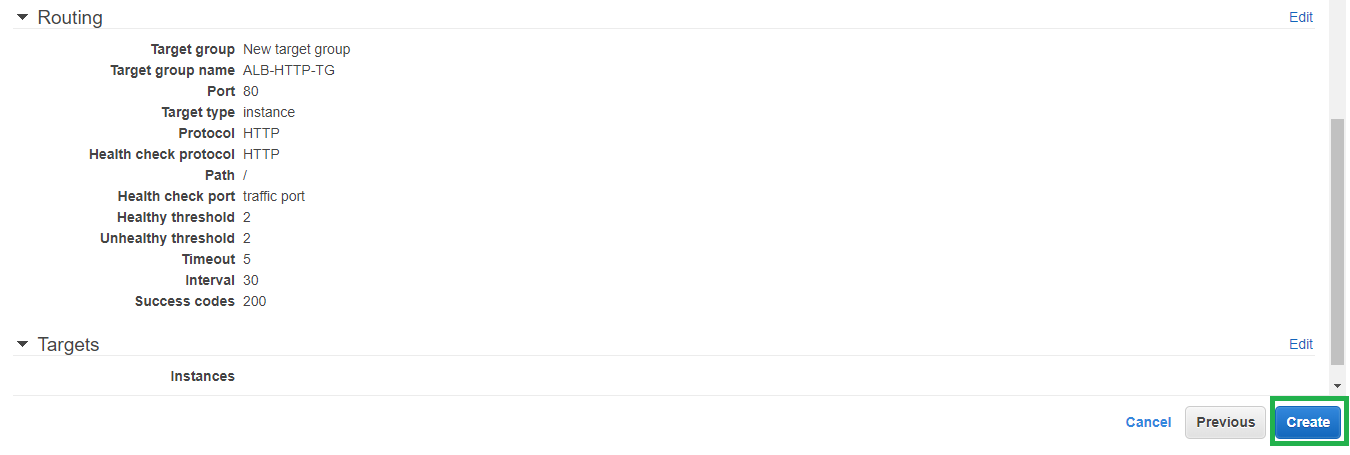


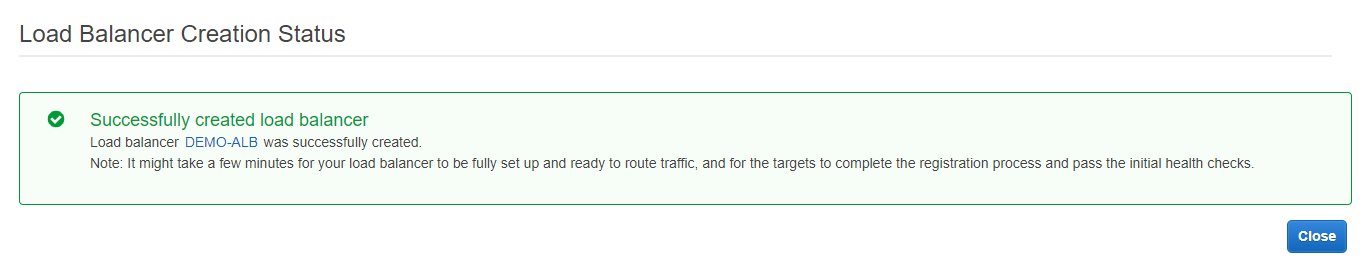


1. Register Targets: If you are manually configuring Instances you can add instances here. If through ASG leave it blank and proceed.

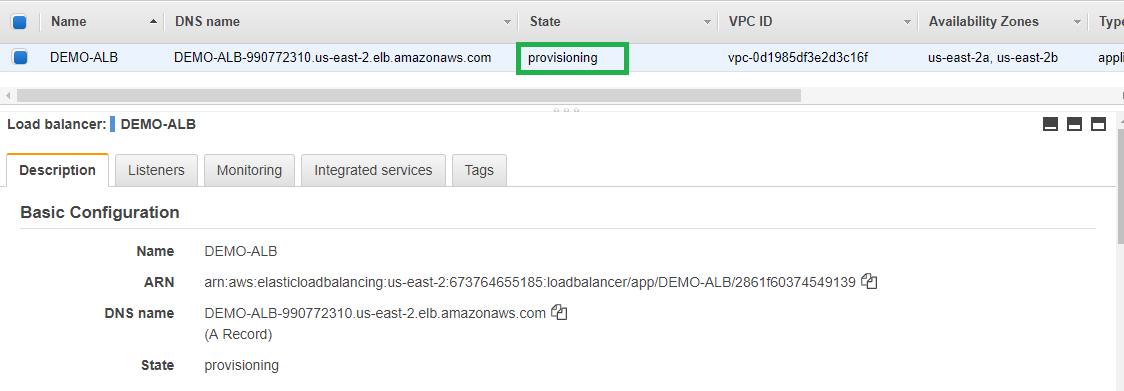


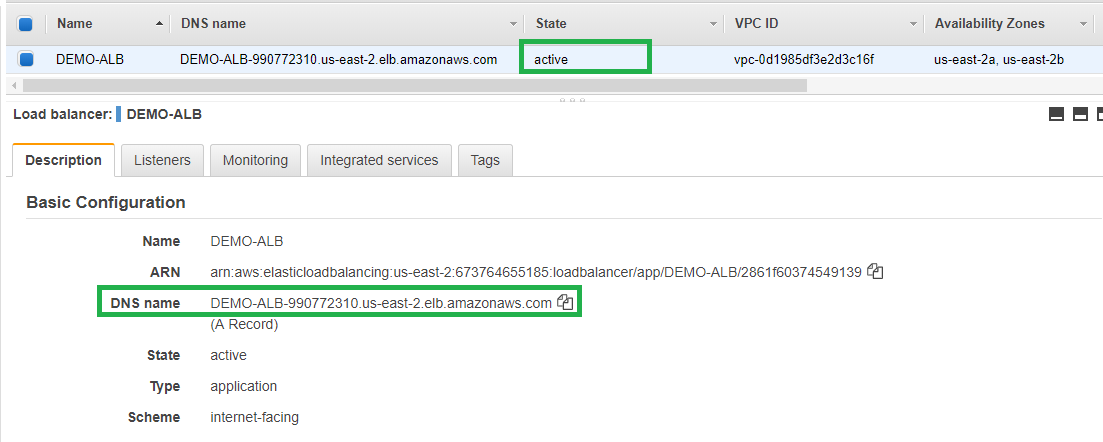






ELB is now being provisioned wait for it to be active.





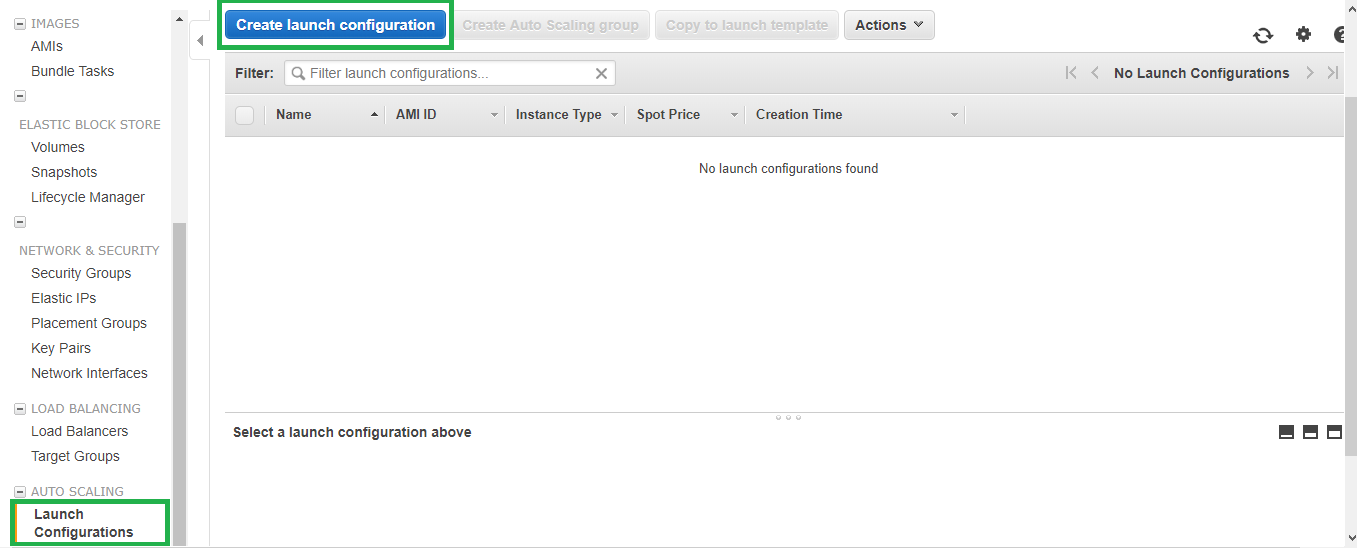
Once the ELB is active verify the accessibility and it should return service unavailable as no instances configured so far. Which we will be doing with ASG in next steps.



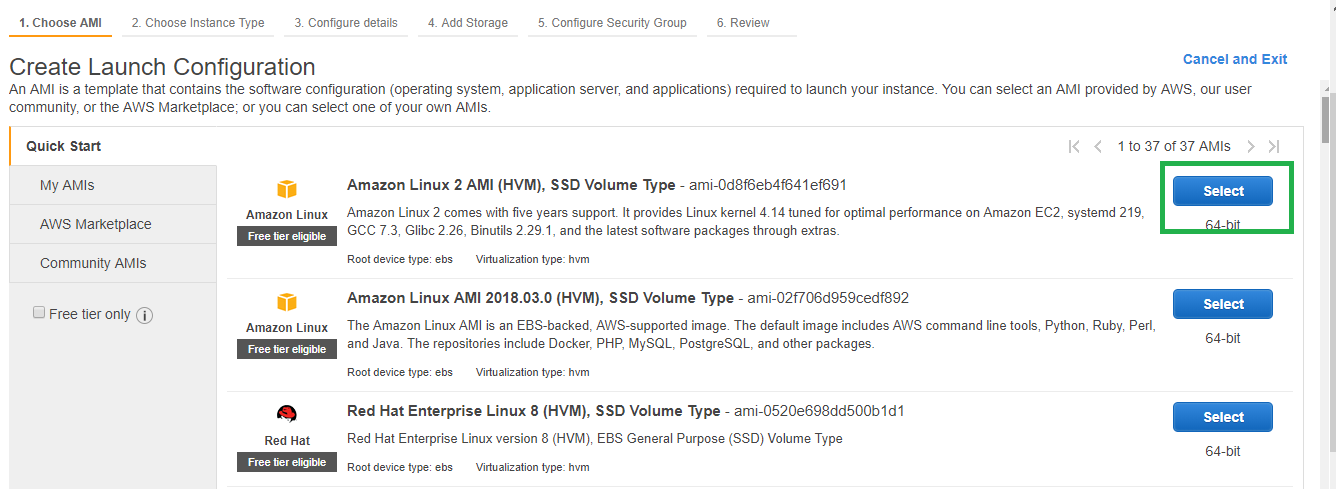
**ASG CONFIGURATION:** ASG configuration involves creating launch configuration and Auto scaling group.

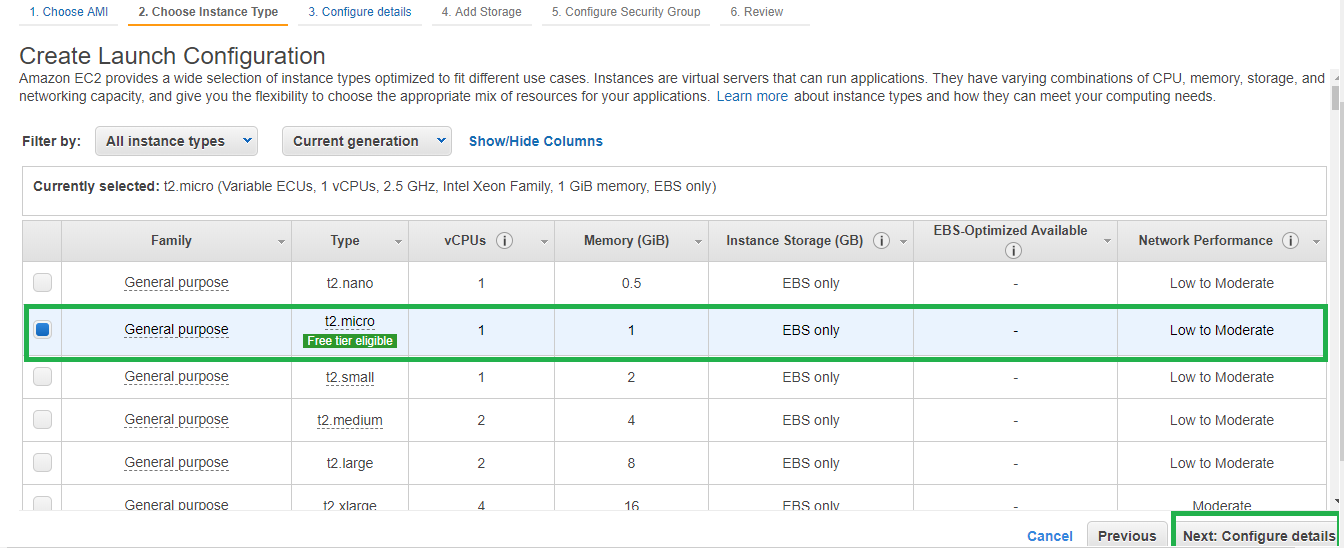
**Launch Configuration Creation:**

1. Create Launch configuration under EC2 Dashboard 🡪 Launch Configurations

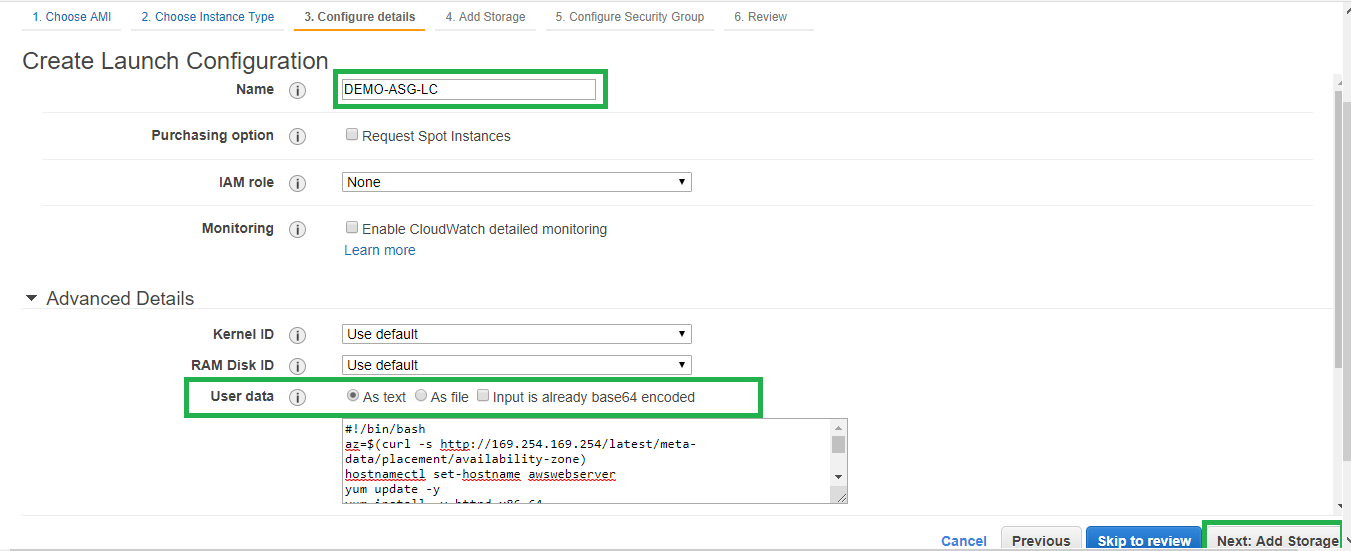


1. Select AWS Linux and t2.micro free version.

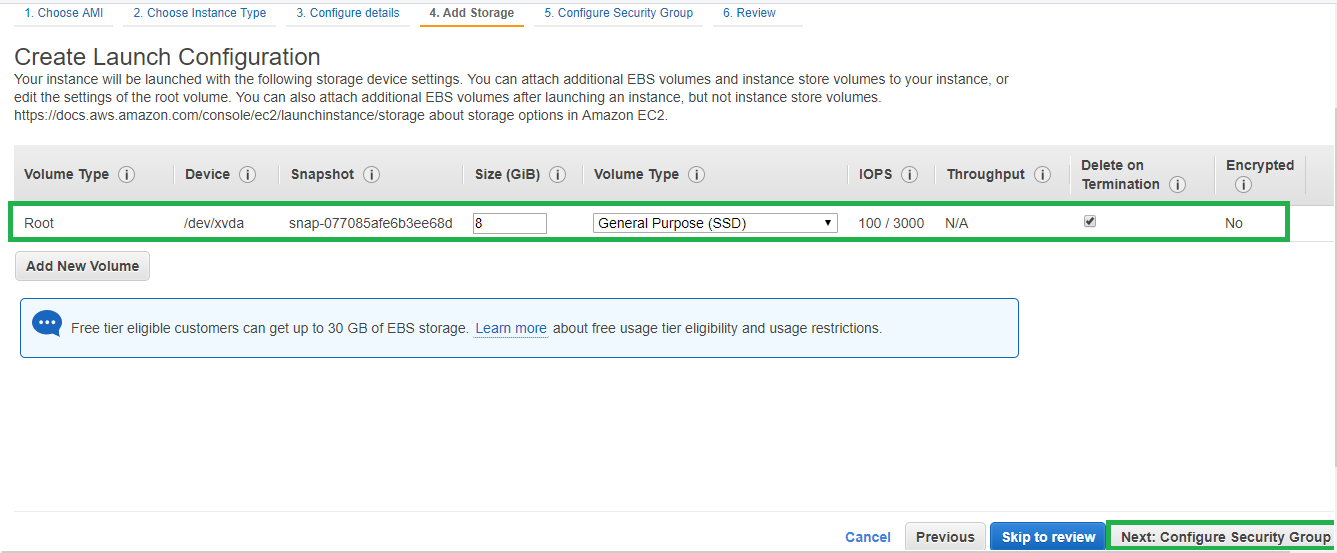




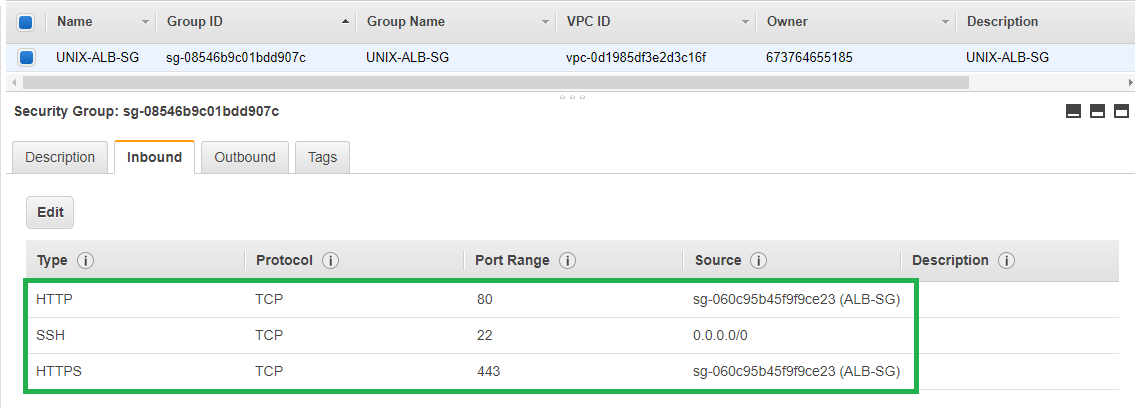
1. Provide name for the launch configuration. In the user data section provide the web server configuration.

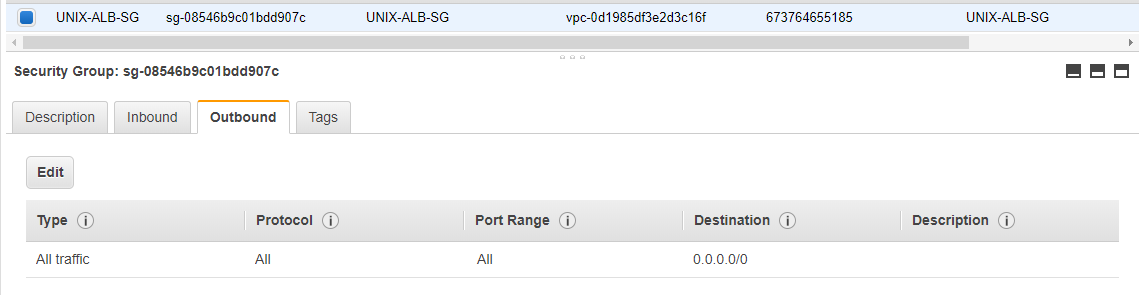


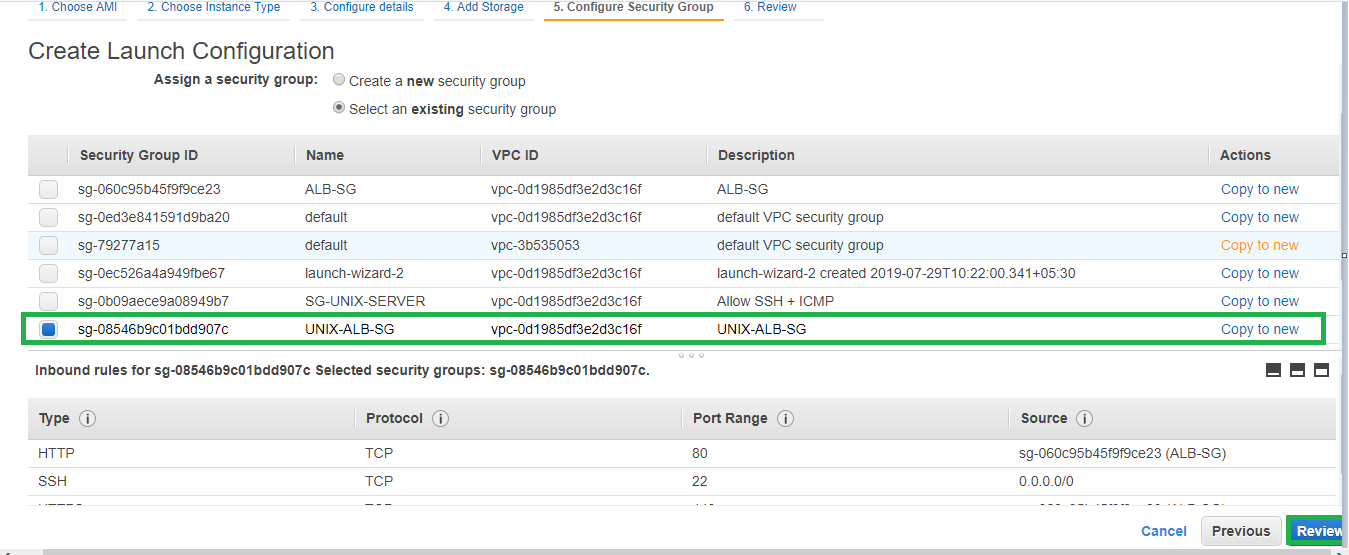
1. Provide storage configuration.

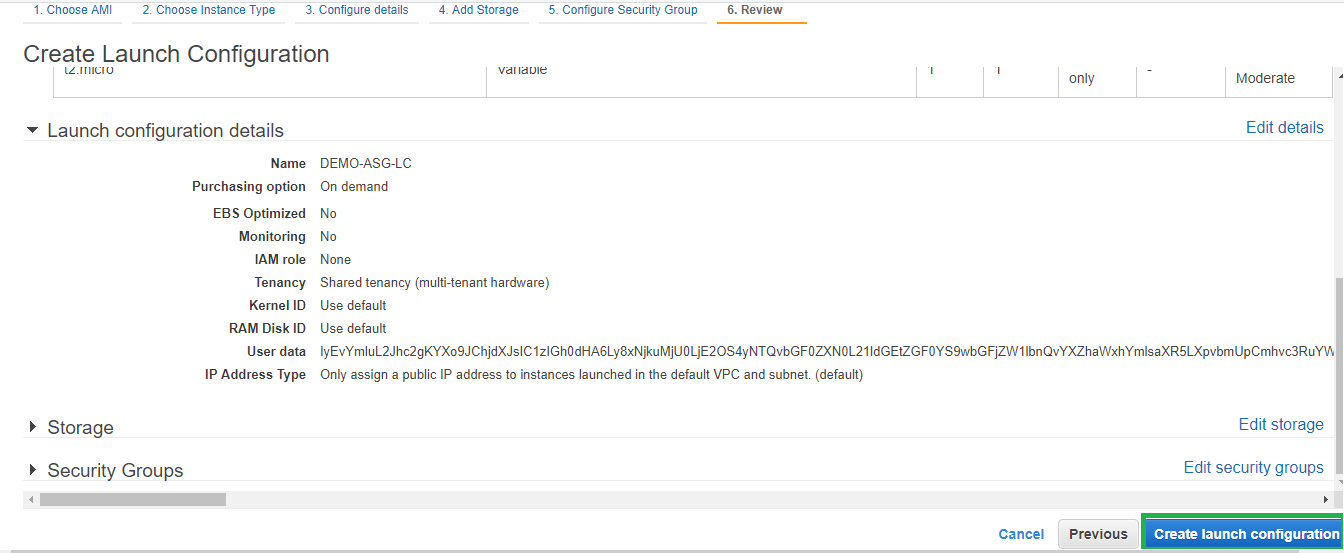


1. Define security group and ensure inbound port allows HTTP/HTTPS/ from ELB Security group and ssh from Internet or from your configured bastion host.

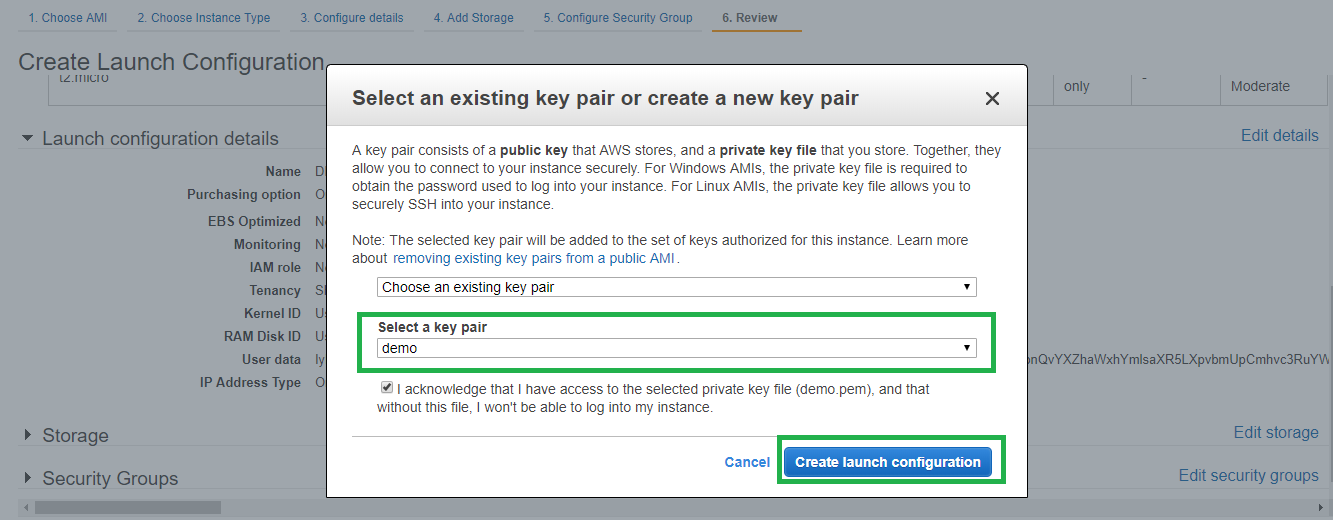


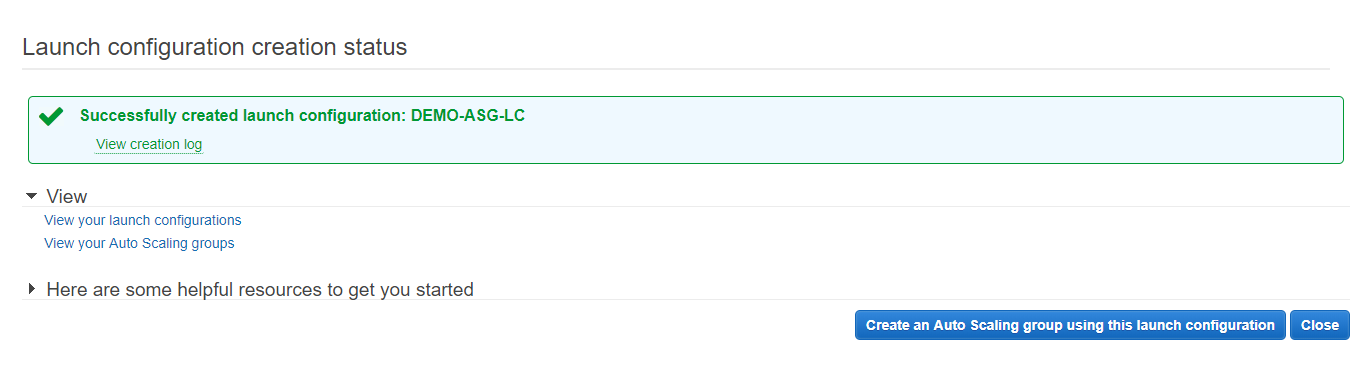






Review and save the key pair for later logging into the instances.

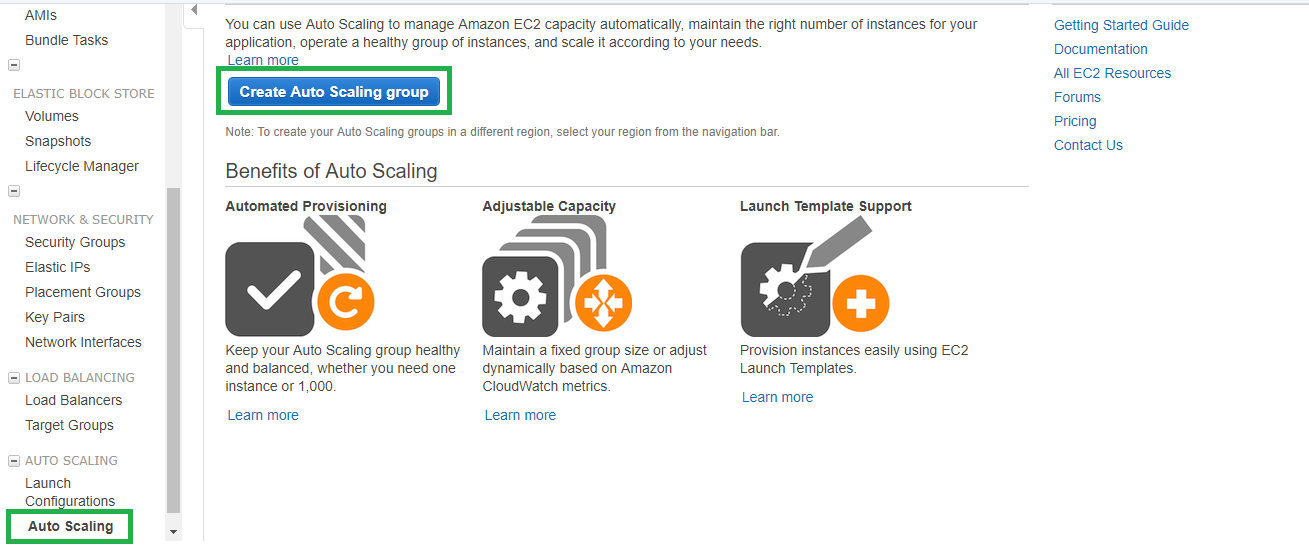




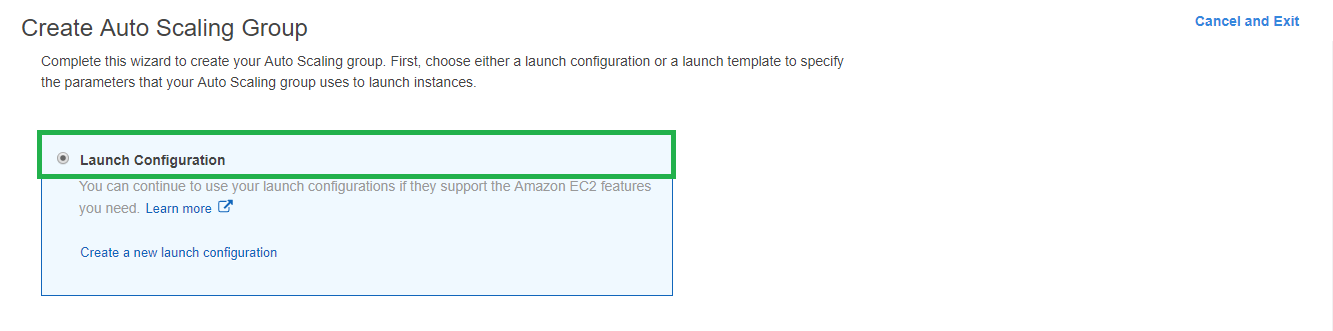
Launch configuration is completed, proceed with ASG creation.

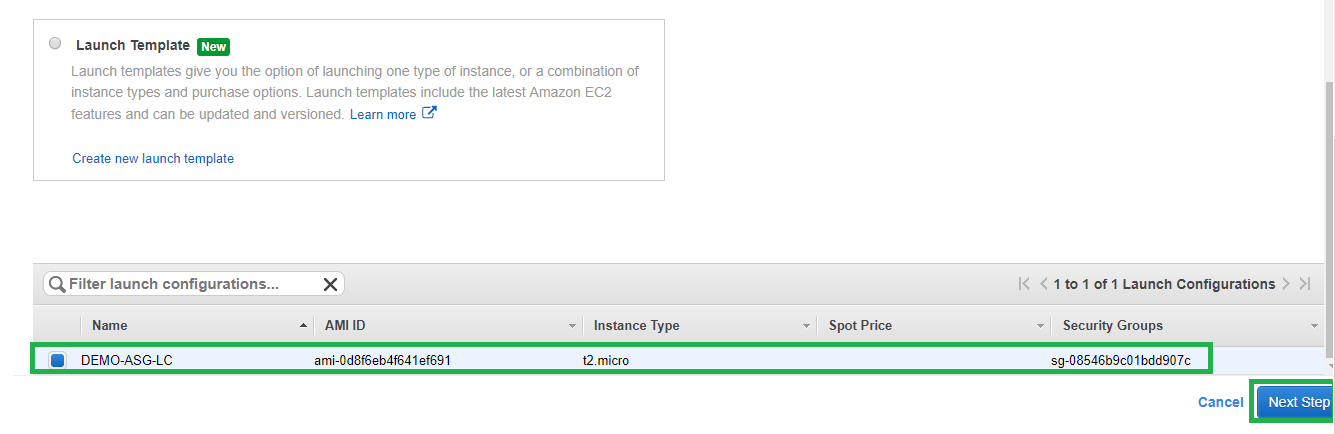
**ASG Creation:**

1. EC2 Dashboard 🡪 Auto Scaling 🡪 Create Auto Scaling Group

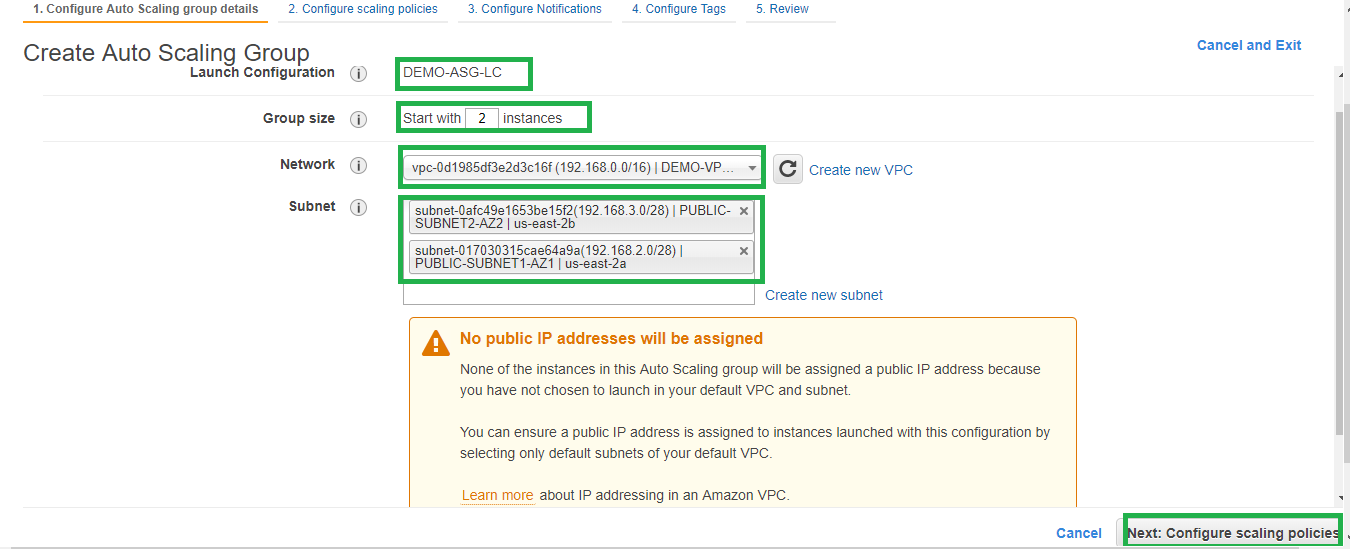


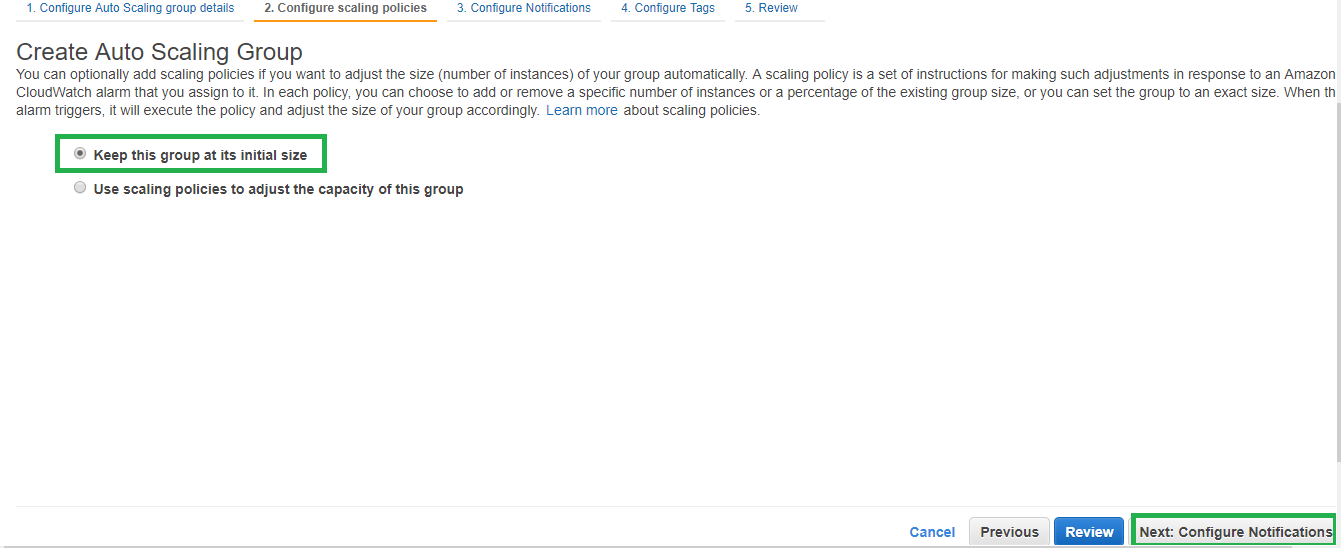
1. Select the Launch configuration that you created previously.

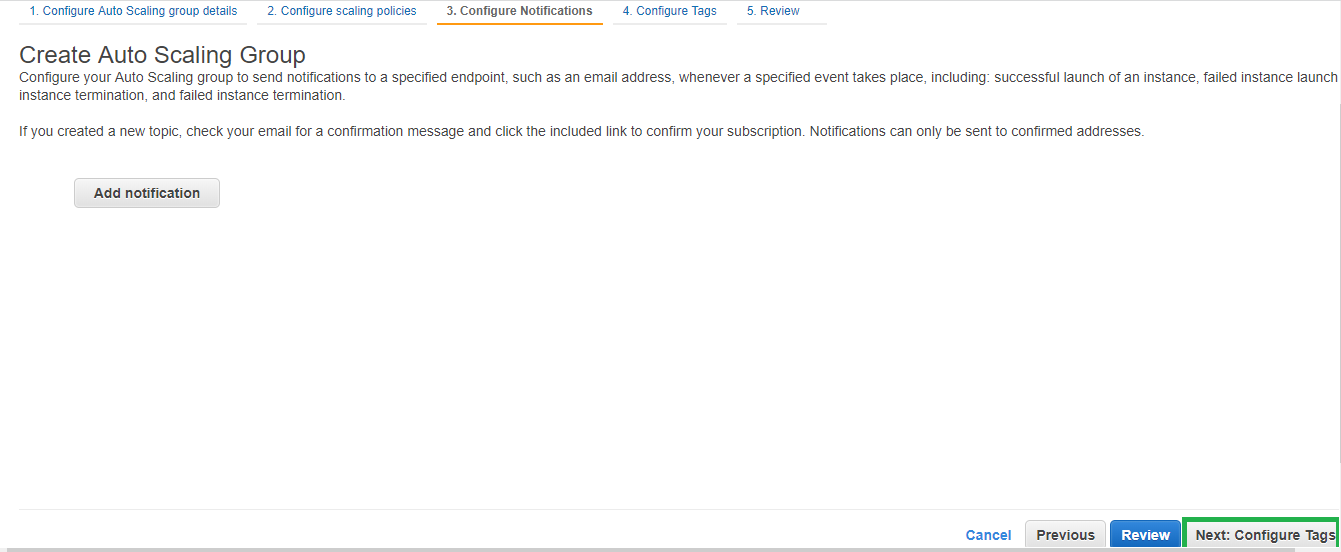


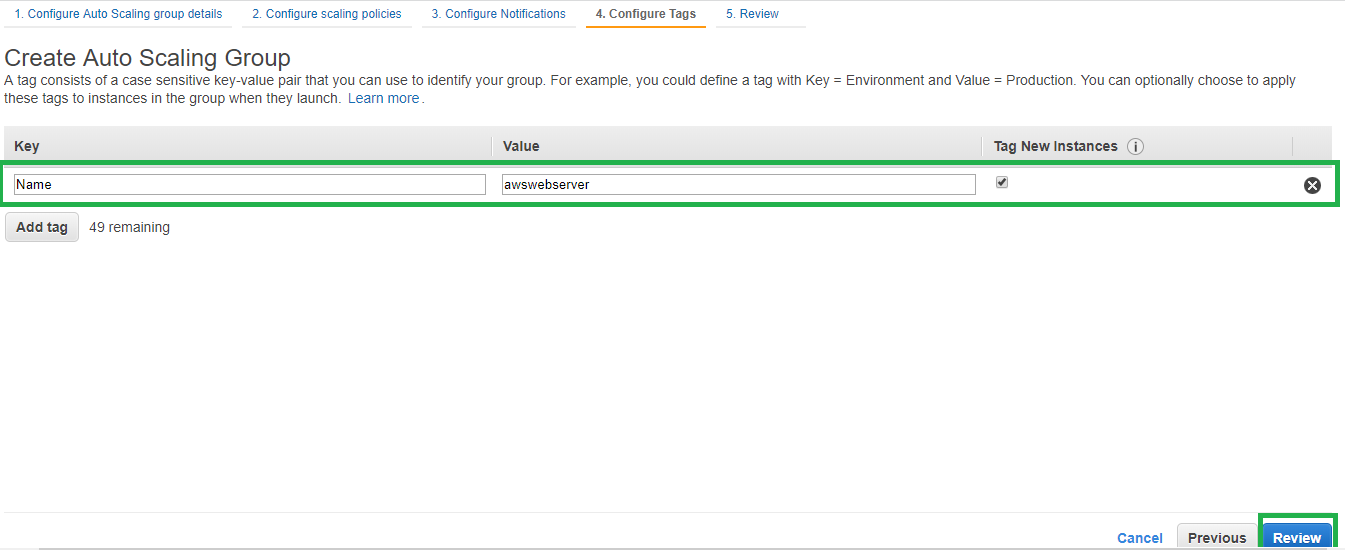


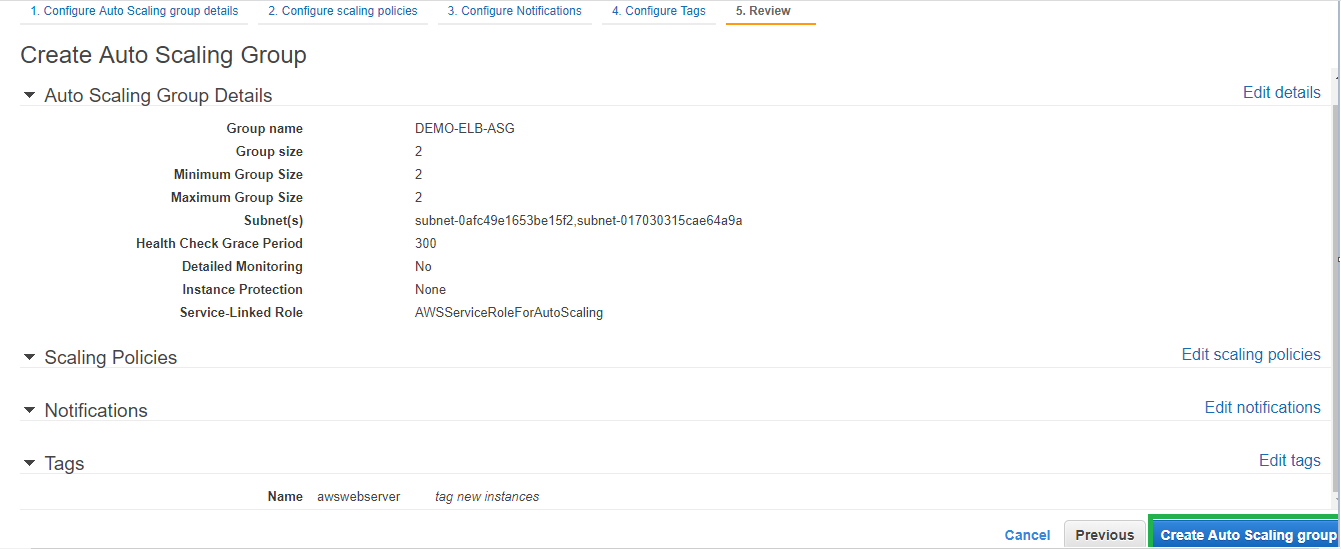
1. Define the group size, VPC, Subnets.

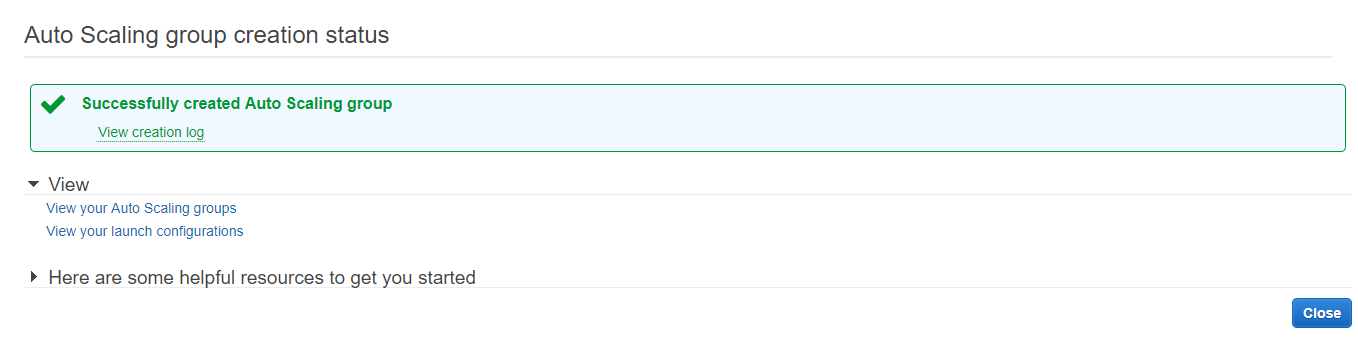




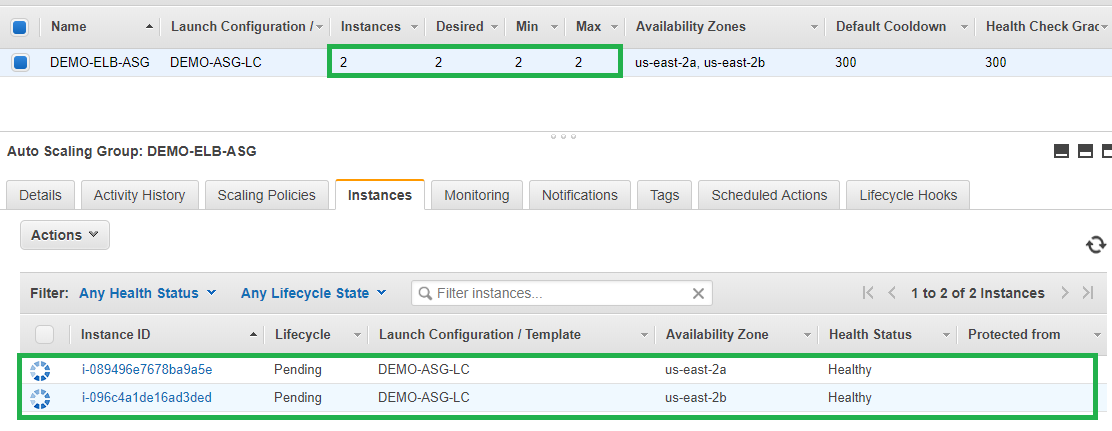


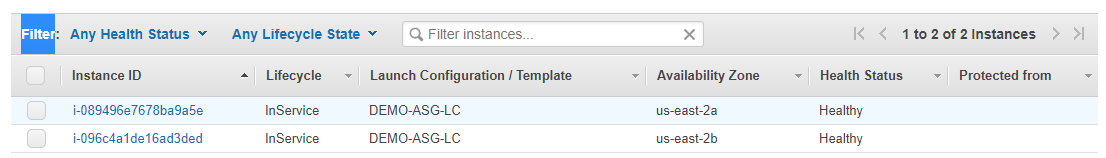


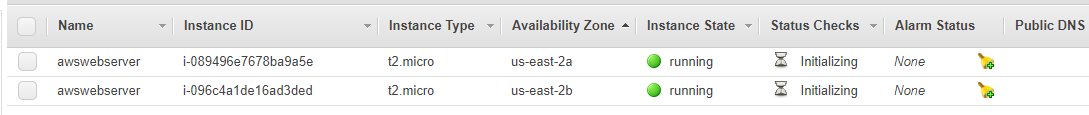


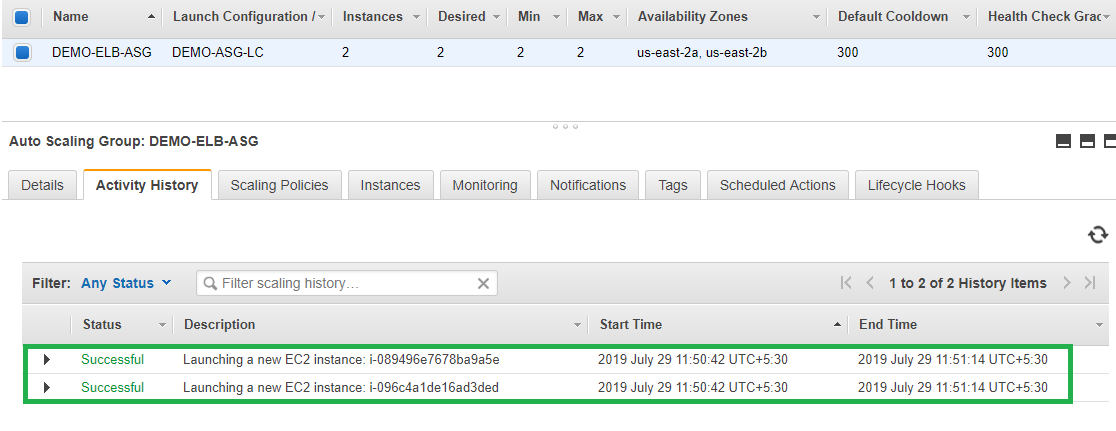


Once ASG is created modify the instances desired, minimum, and maximum as per requirement and as soon as the ASG is created we should see new instances provisioned.

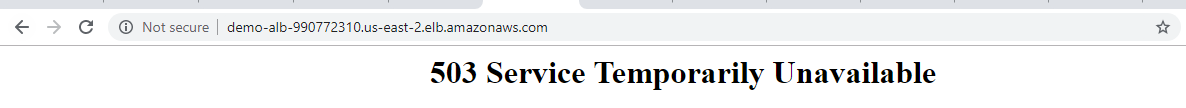


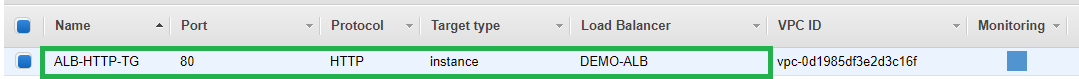




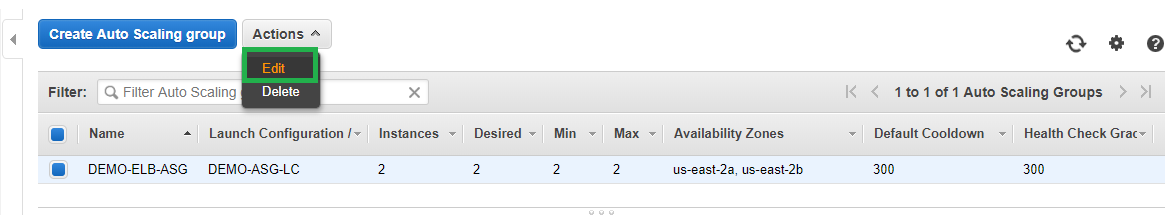


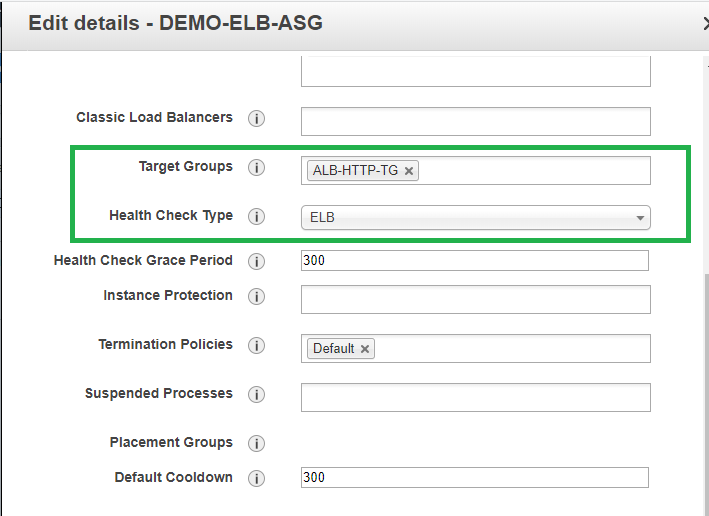
However you can still see ELB is not servicing.



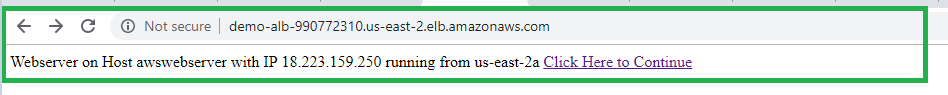


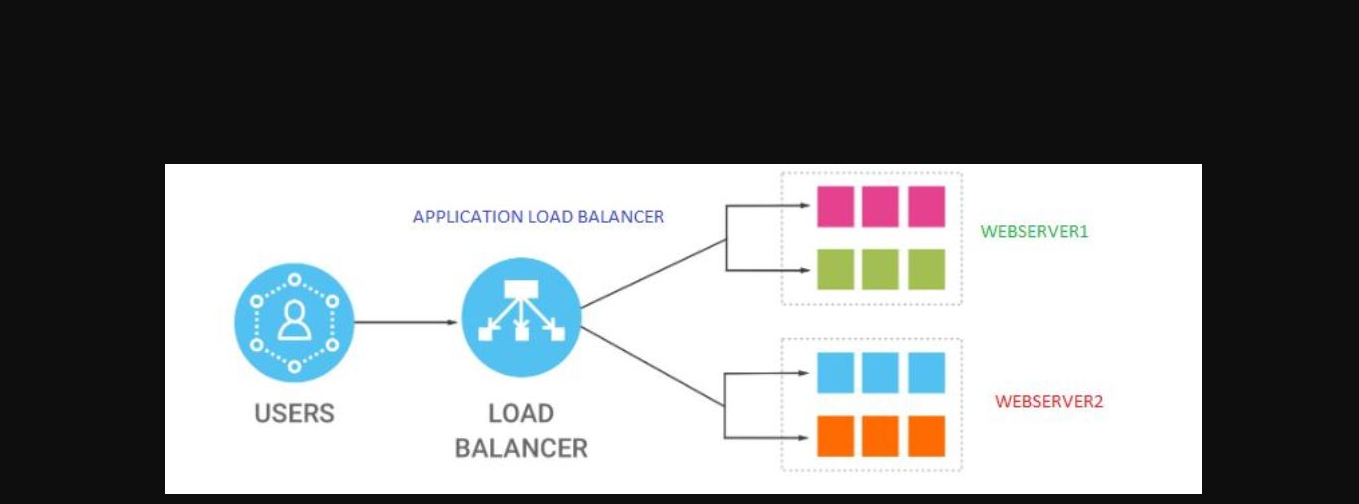
Edit the ELB to select the Target Group to make ELB in service and select Health check type as ELB

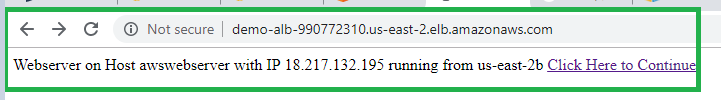


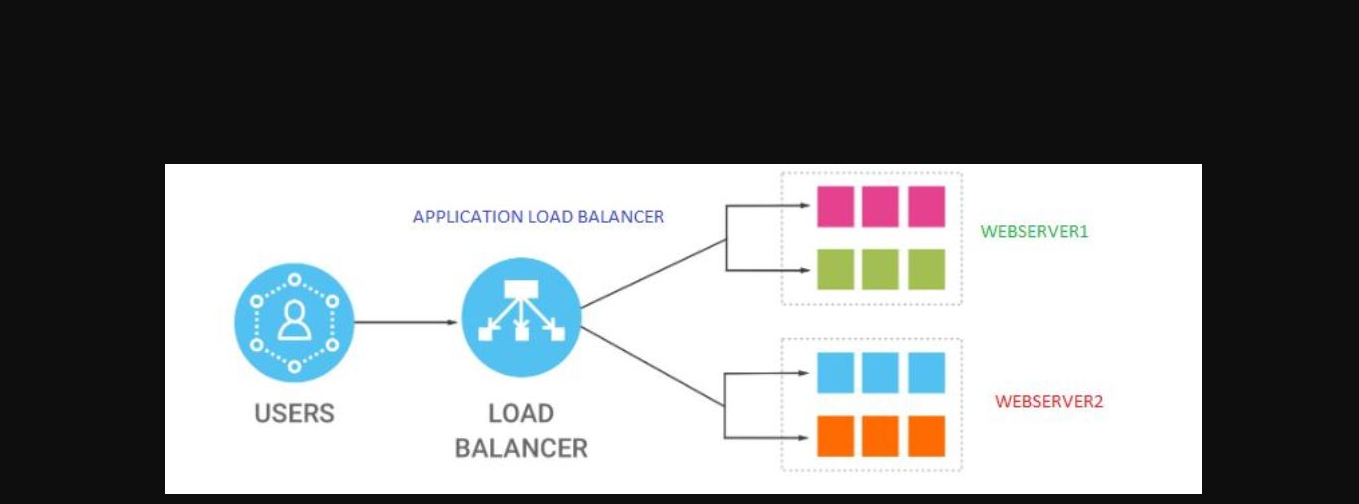


Verify ELB status and should see the configured instances load balancing.

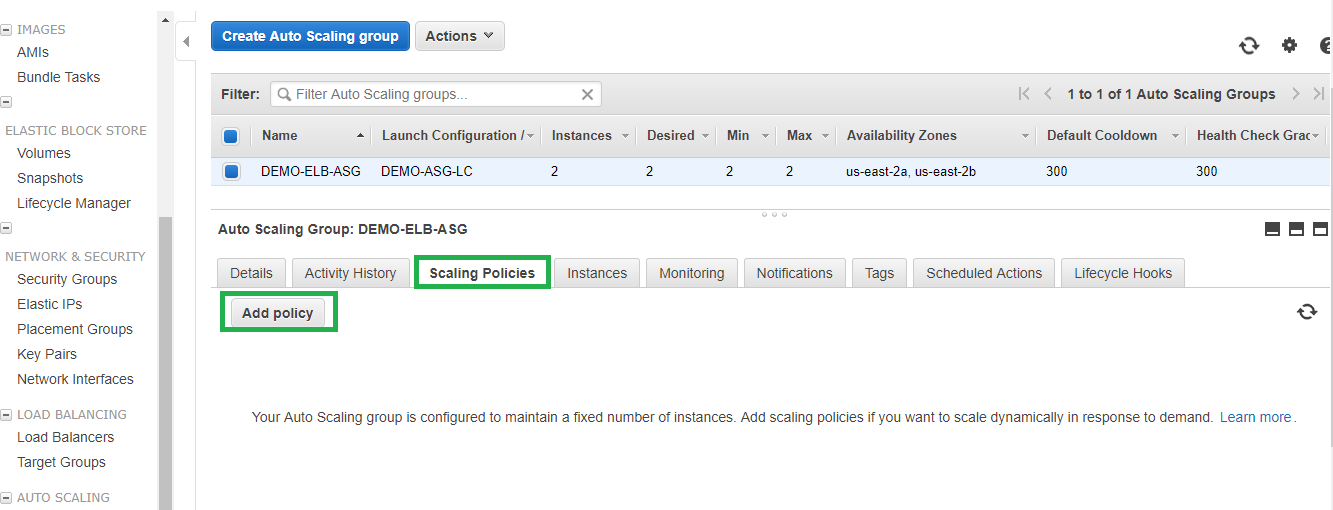


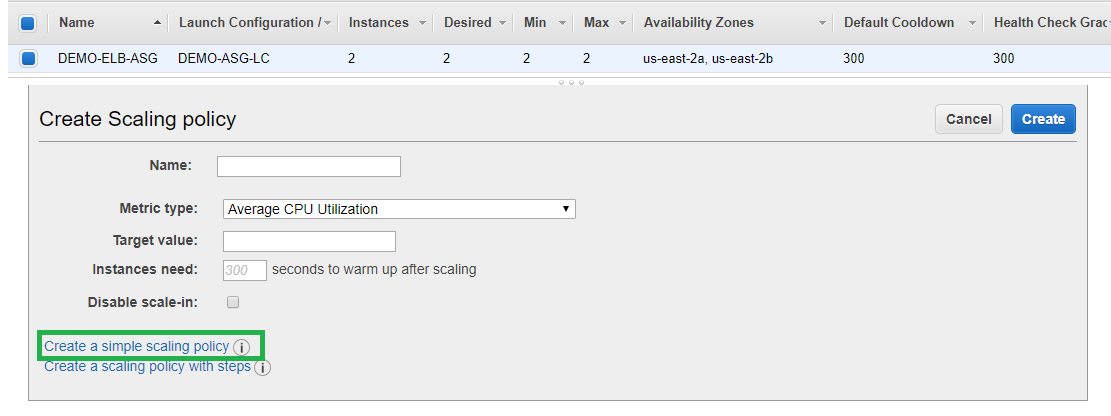


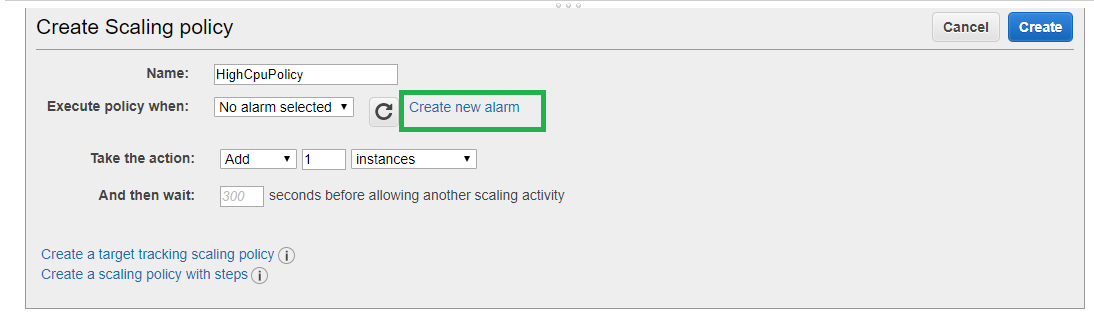


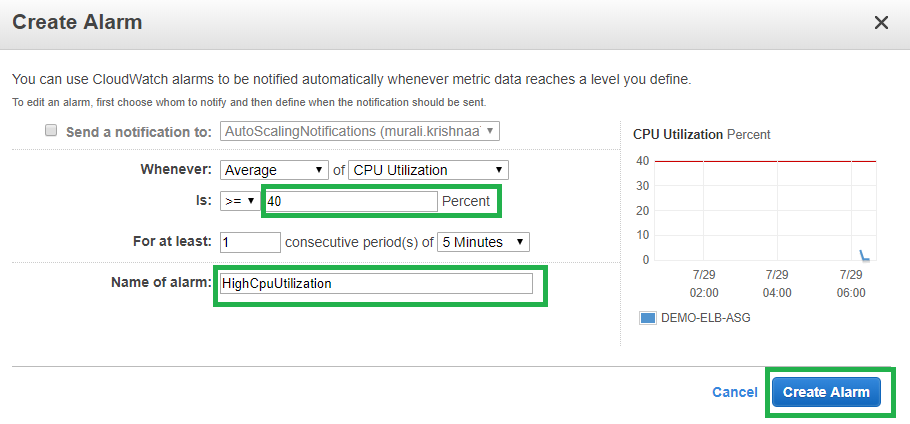


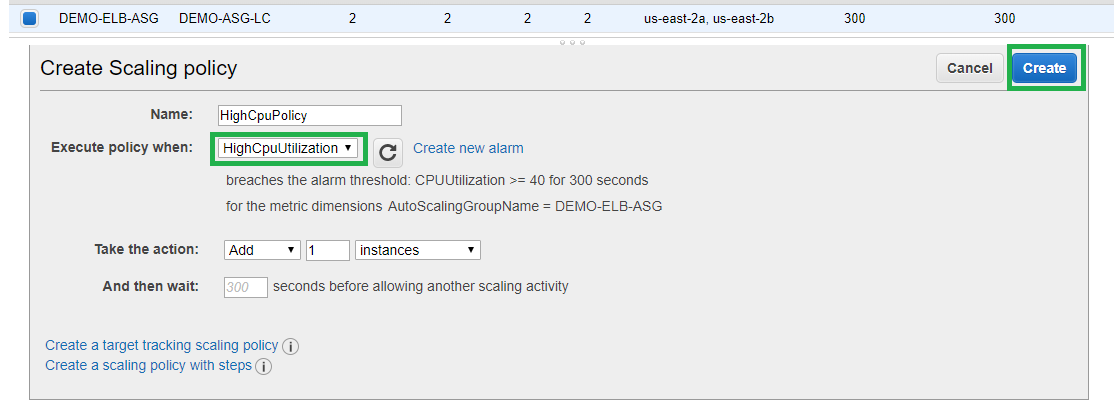
**Scaling Policies for ASG:** Create two scaling policies one for scale out when high cpu utilization and one for scale in when low cpu utilization.

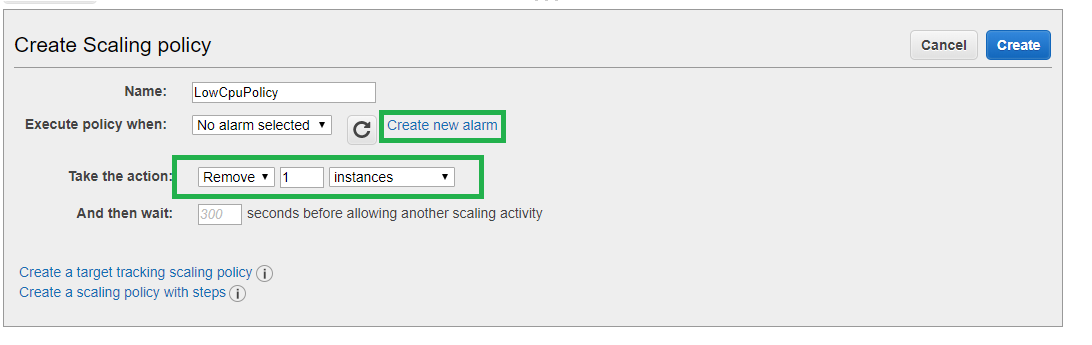


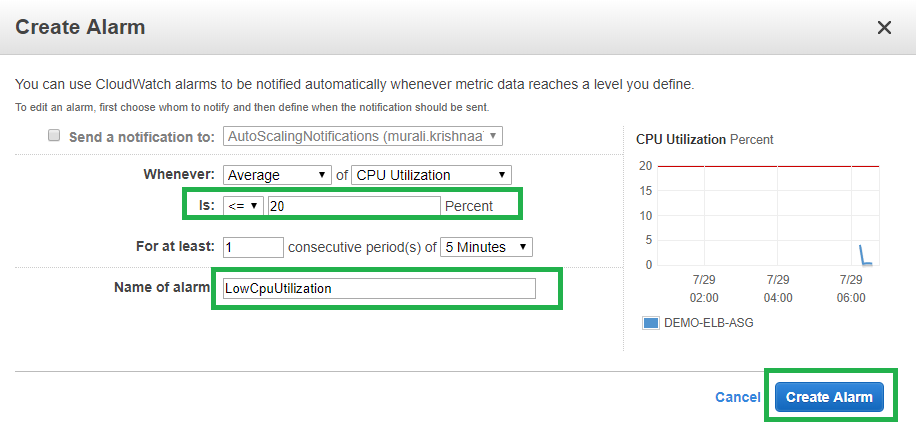


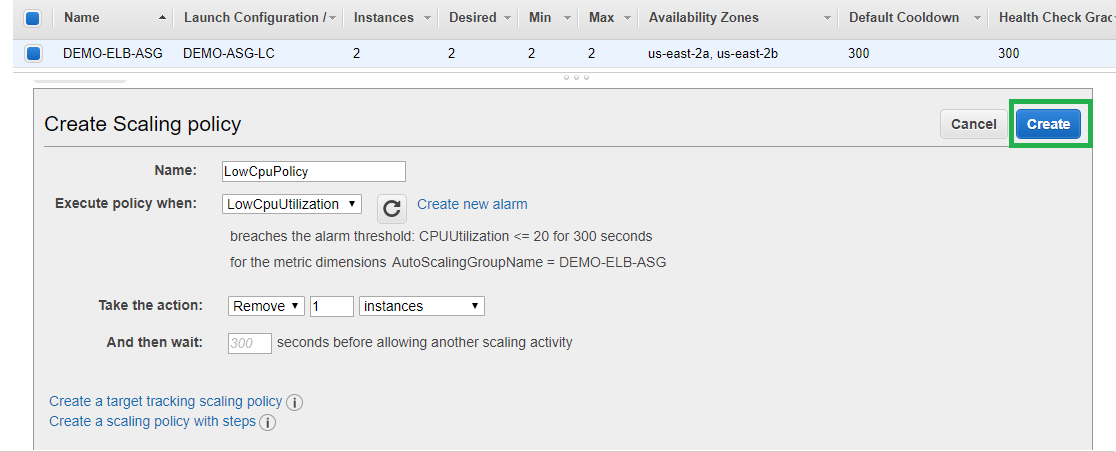


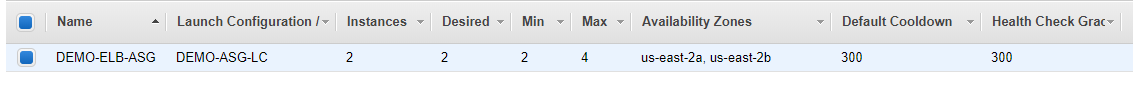




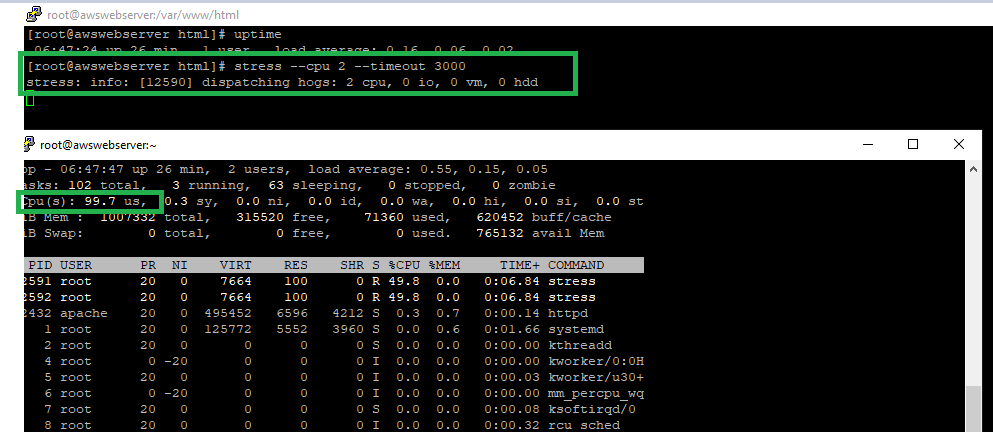


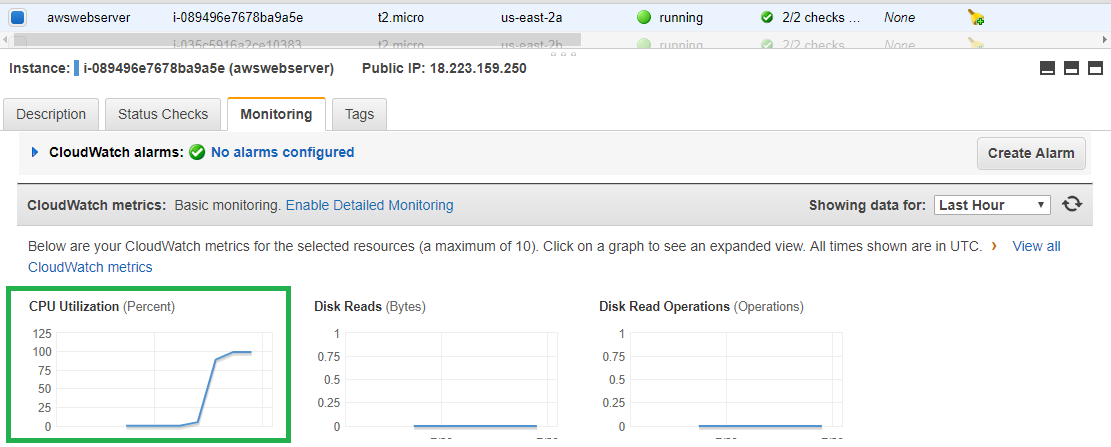




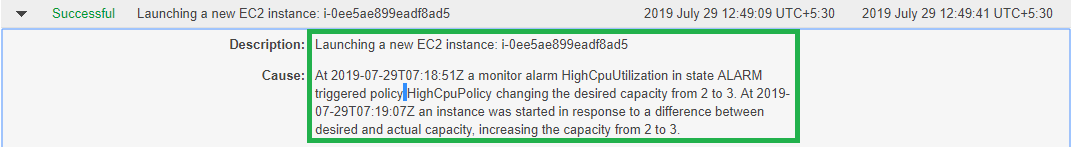


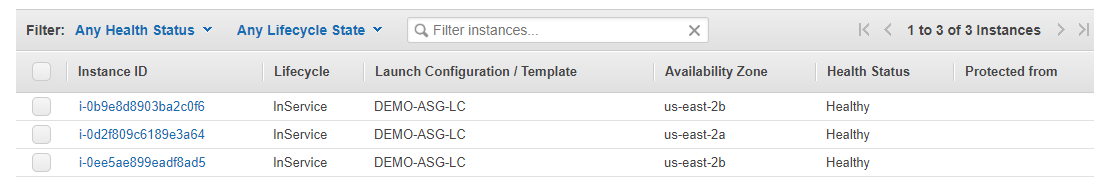
As Scaling policies are created now we can simulate them by increasing the load on the server using stress command.

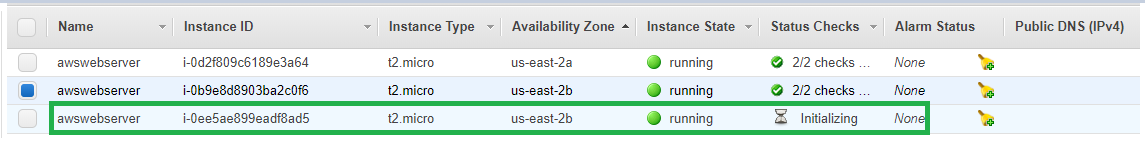


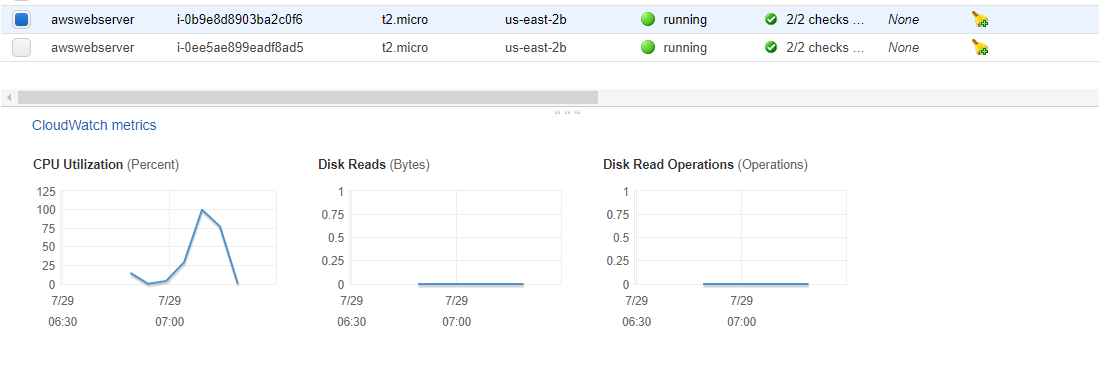


After CPU usage cross our scaling policy alarm we can see new instance being launched.



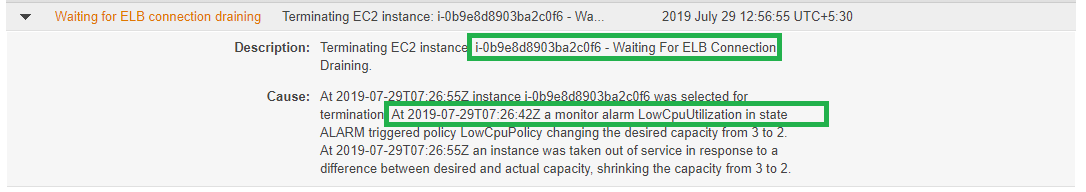


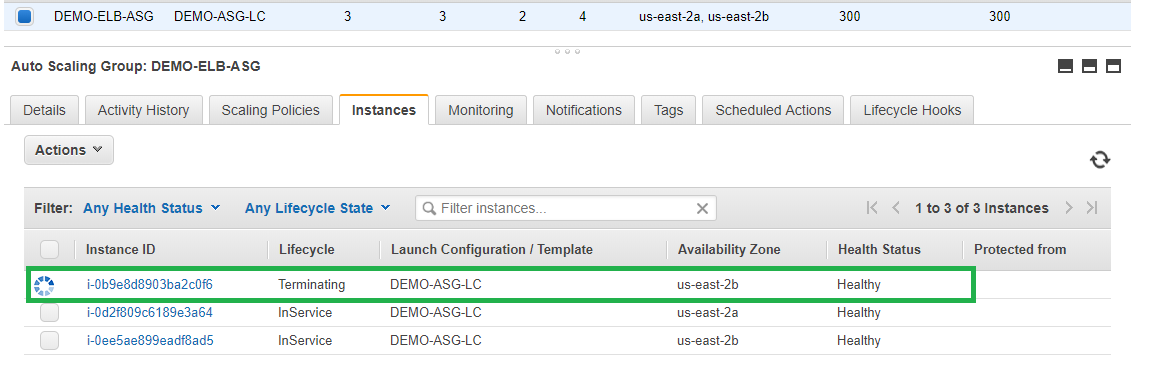


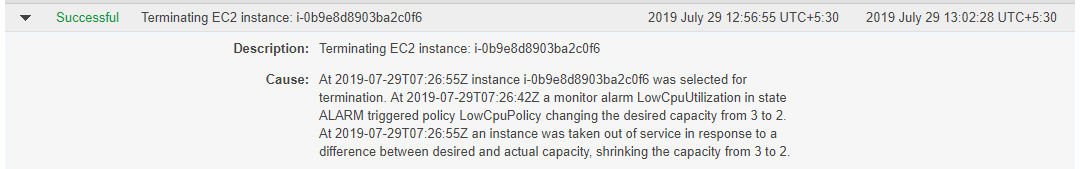


Now stop the stress command to reduce the cpu utilization which triggers the low cpu usage alert and ASG terminates the instance based on the termination policy.









**Note: After completing the lab make sure to delete all the resources.**