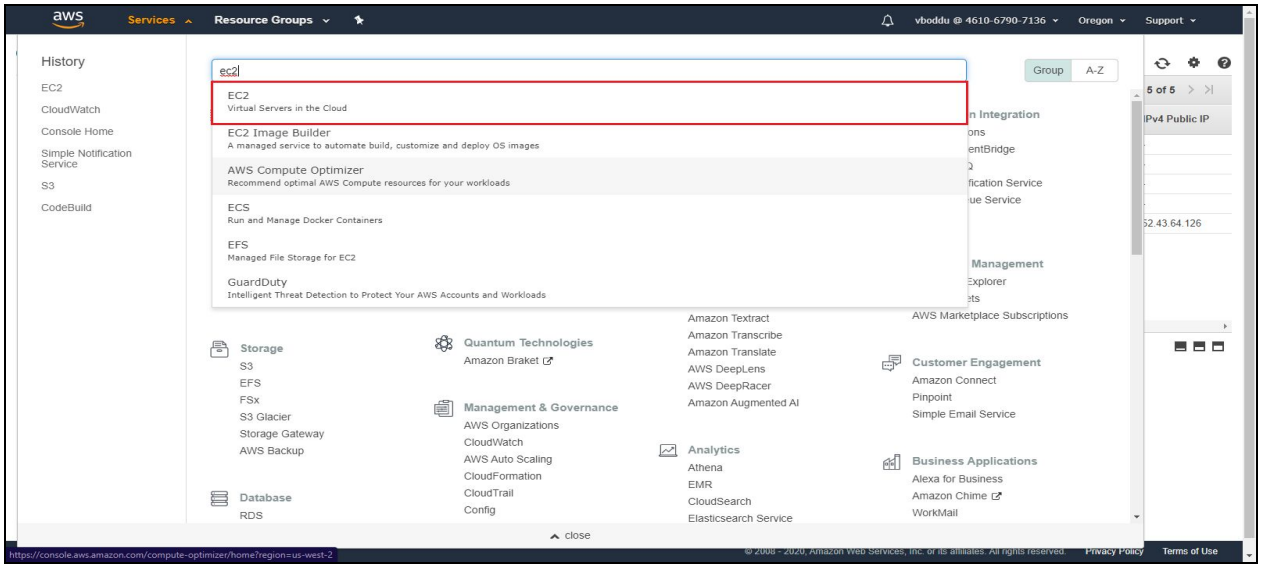
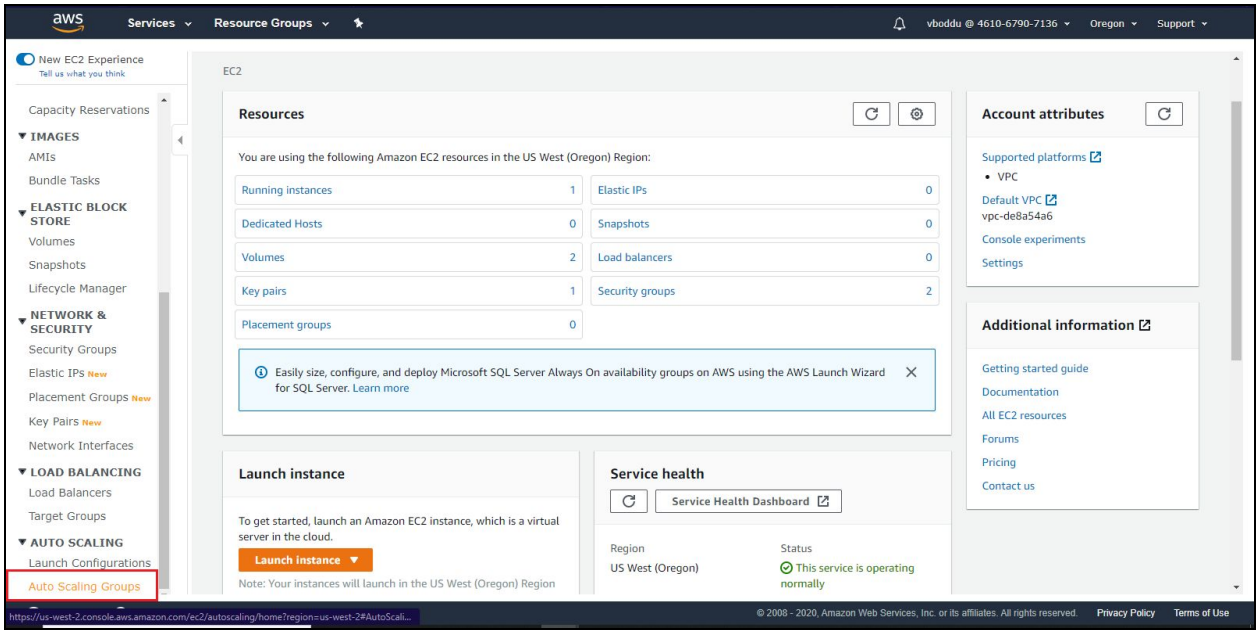
Autoscaling and load balancing

By the end of the document, we will be able to configure Auto scaling and load balancing for the EC2 instance.

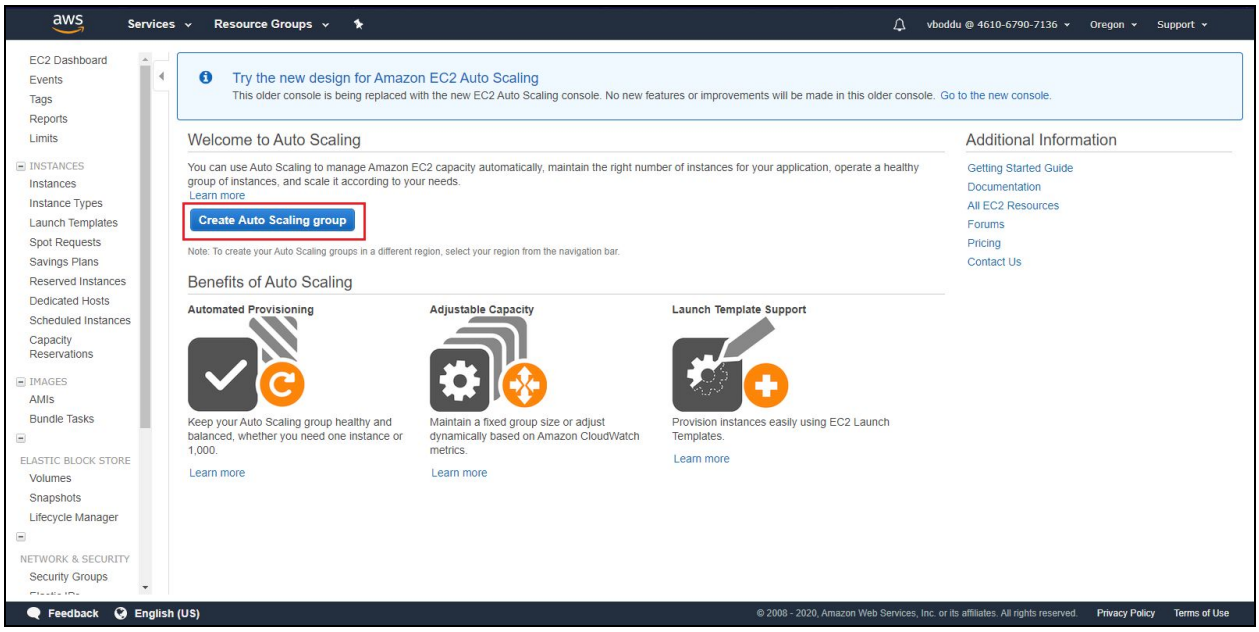
First, we need to login to the AWS console and search for EC2 service.



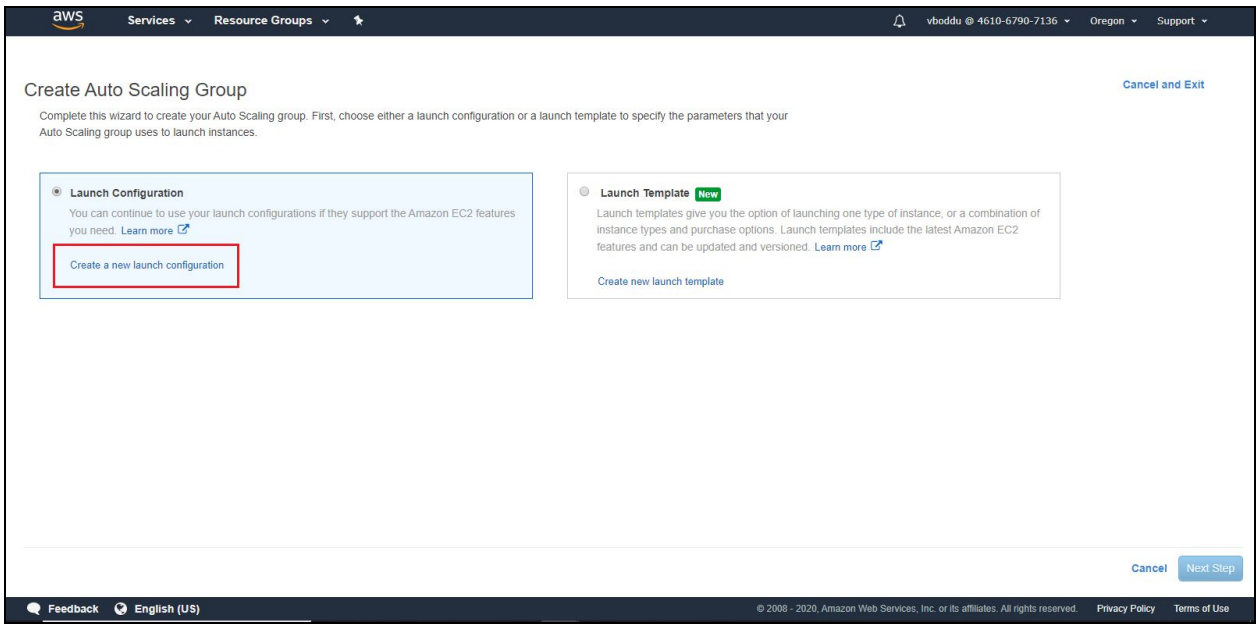
Following is the dashboard for EC2 service and click on Auto Scaling Groups as shown in the dashboard.



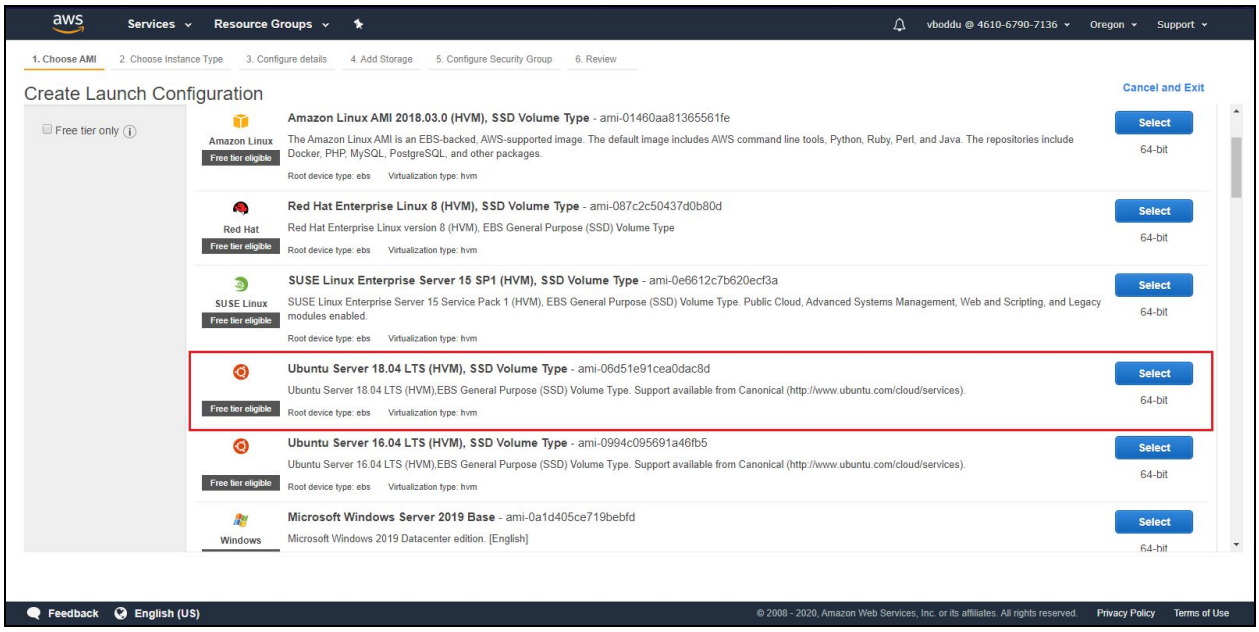
Now click on Create auto-scaling group.



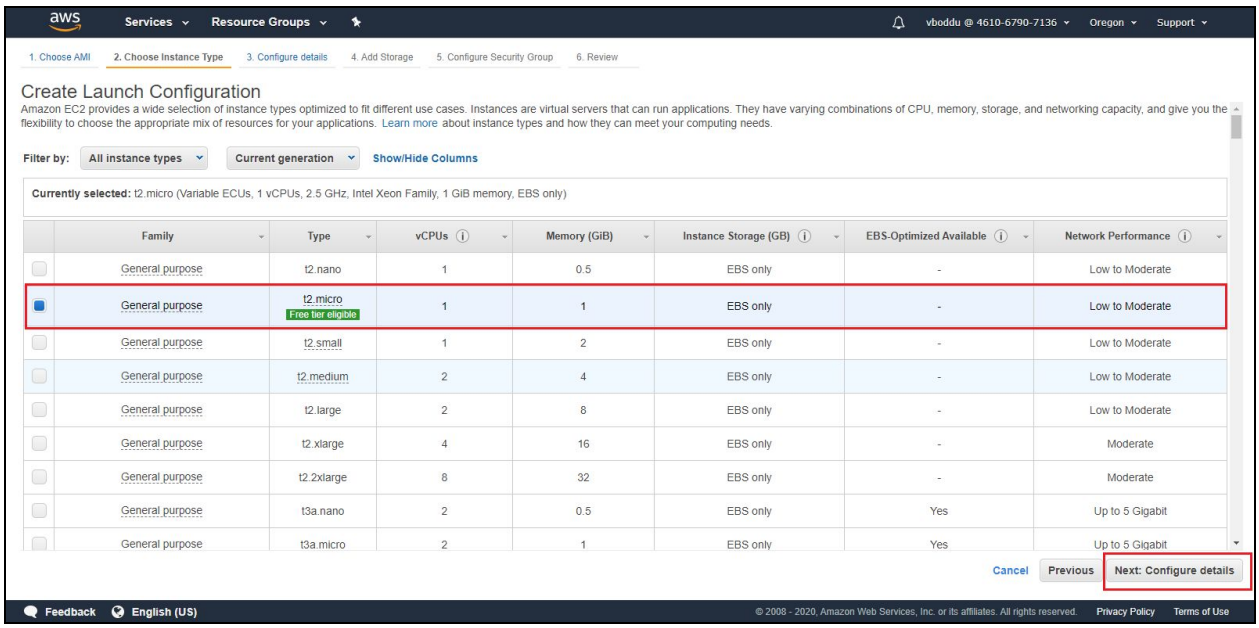
Click on create new launch configuration to configure the Auto Scaling EC2 instance.



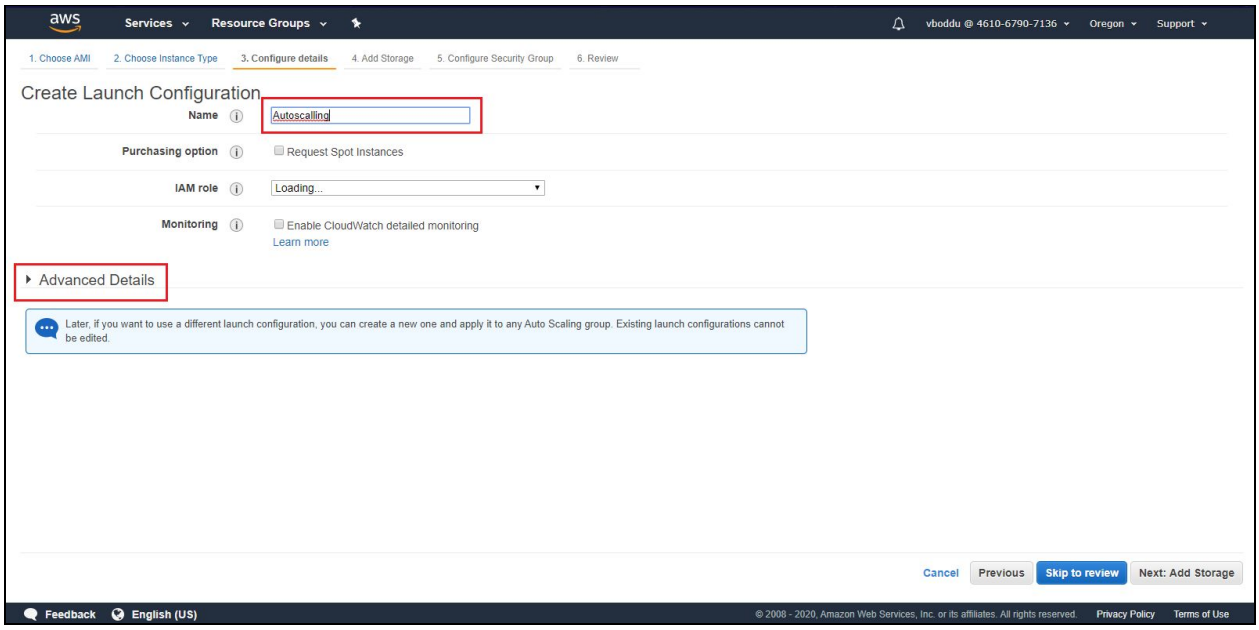
The following are the operating systems that are provided by AWS and Here we need to select one of them as shown below.

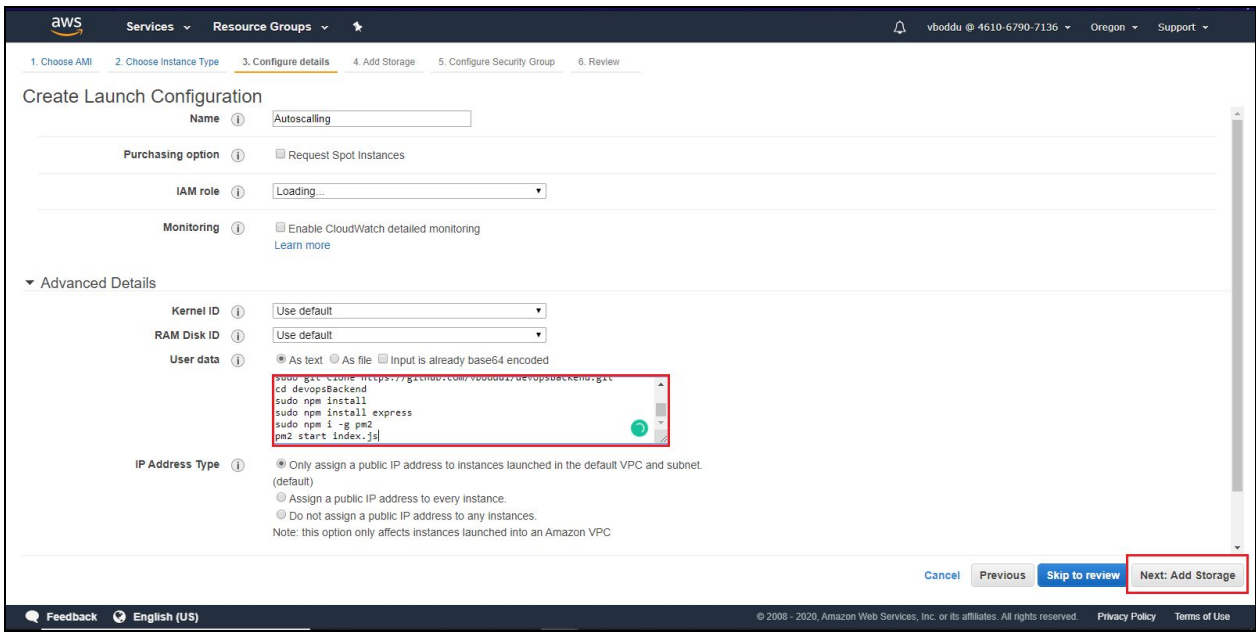


Then Select General purpose t2.micro tier instance for our requirement. It provides 1 vCPU and 1 GB memory for the instance and the performance of the instance is low to moderate. Now click on next to configure the instance.



Once click on configure details we need to provide inputs like the name of the auto scaling, User data and make remaining as default. Click on add storage.

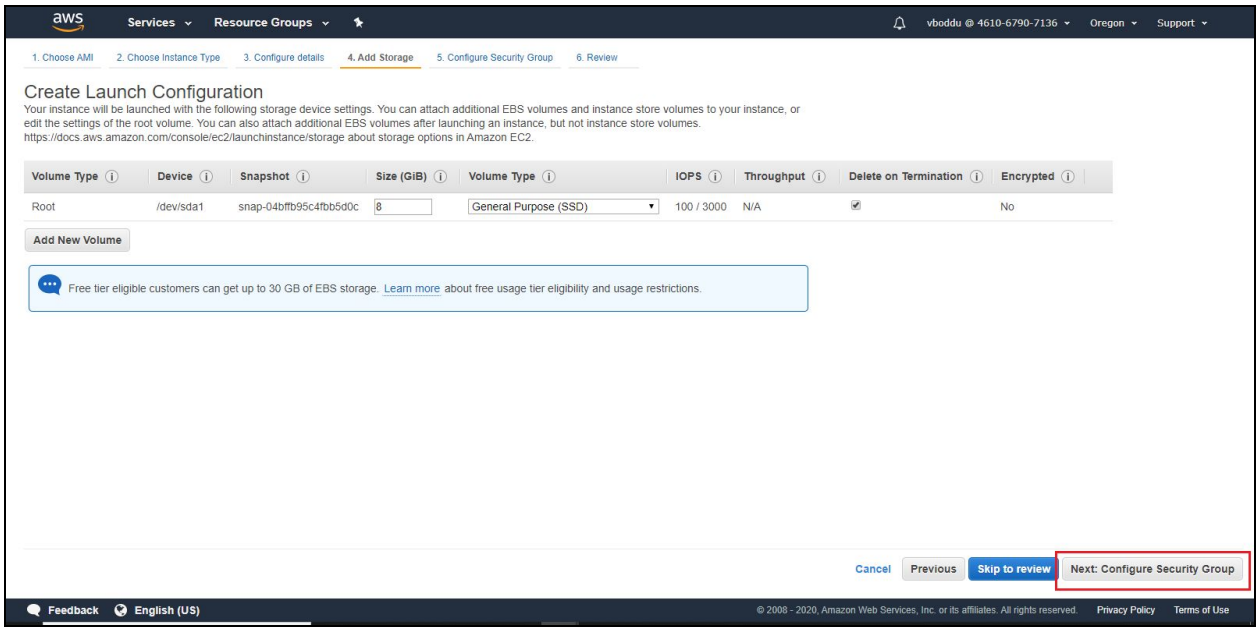




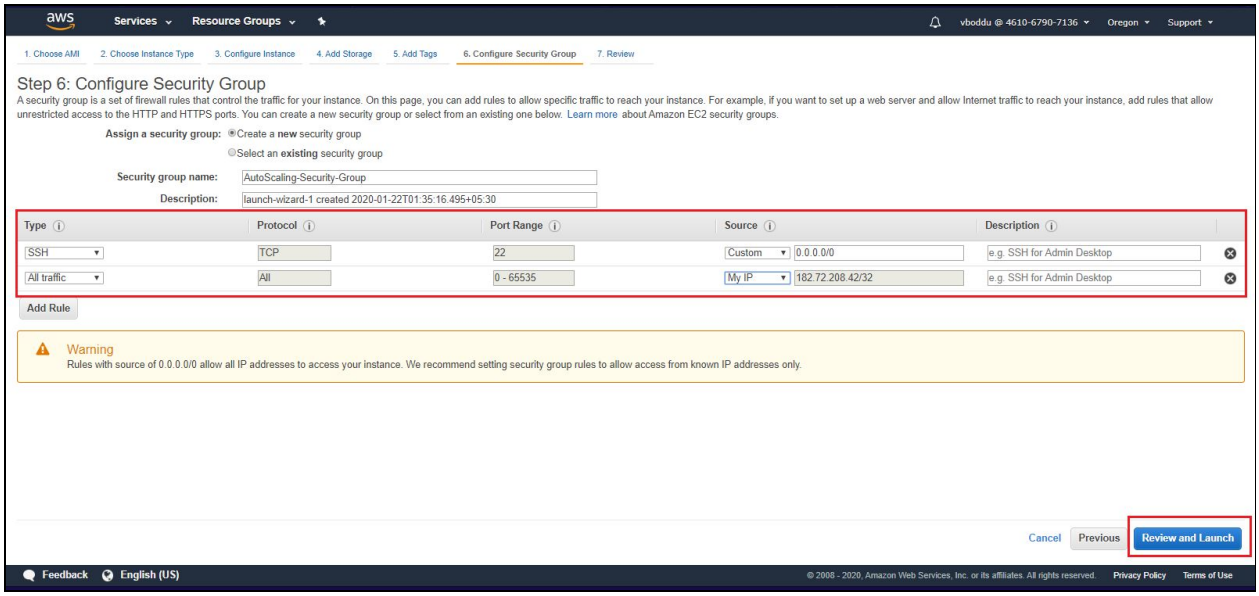
These are the commands that are used in the User data.

```
#!/bin/bash
sudo apt-get update
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install nodejs -y
cd /var/www/
sudo git clone https://github.com/vboddu1/devopsBackend.git
cd devopsBackend
sudo npm install
sudo npm install express
sudo npm i -g pm2
pm2 start index.js
```

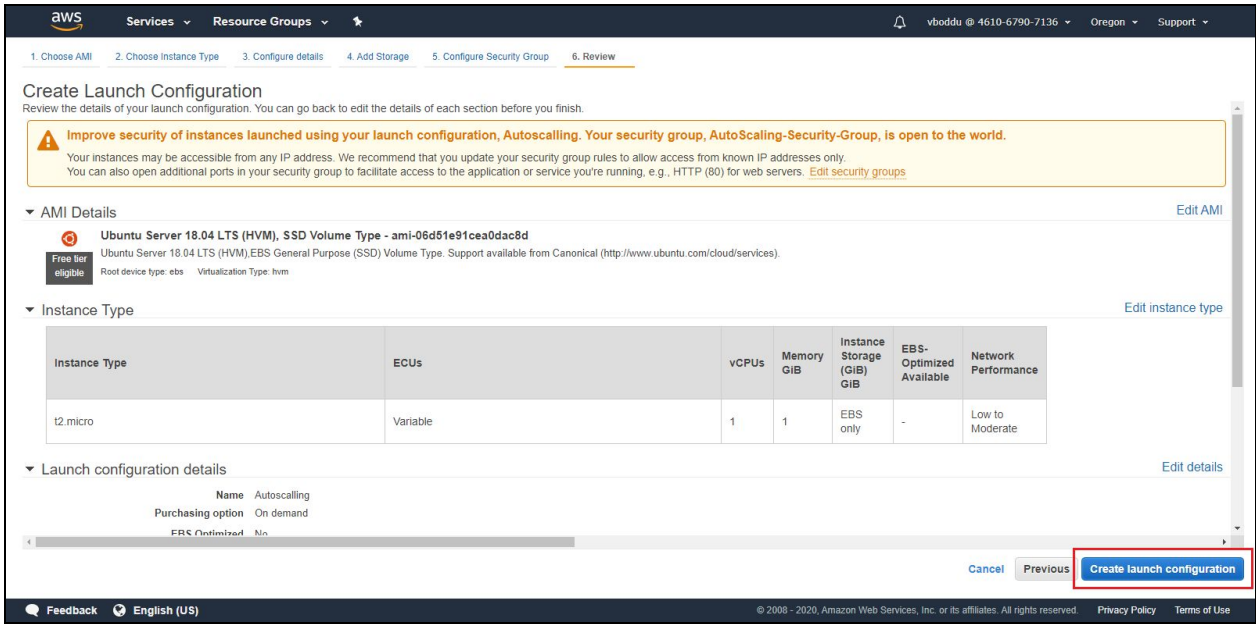
By default, AWS provides 8 GB storage for the t2.micro instance.Now click on next to configure security groups.



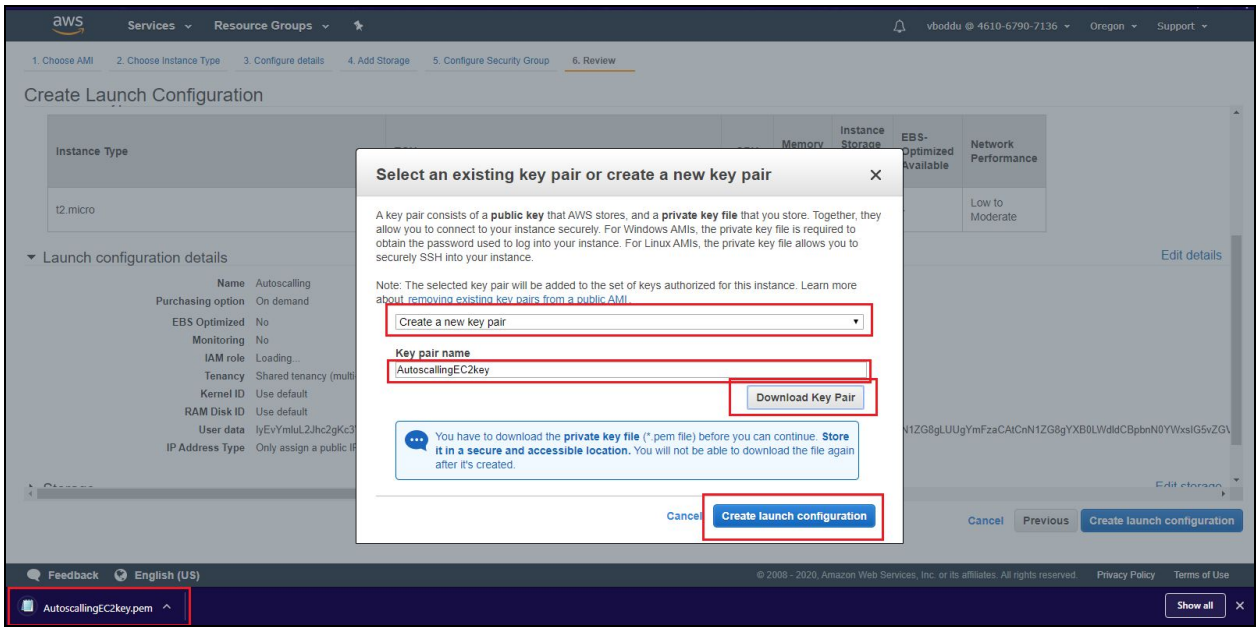
Create a new security group to instance. A security group is a set of firewall rules that control the traffic for our instance. Here we are adding all traffic ports. Now click on review and launch instance.



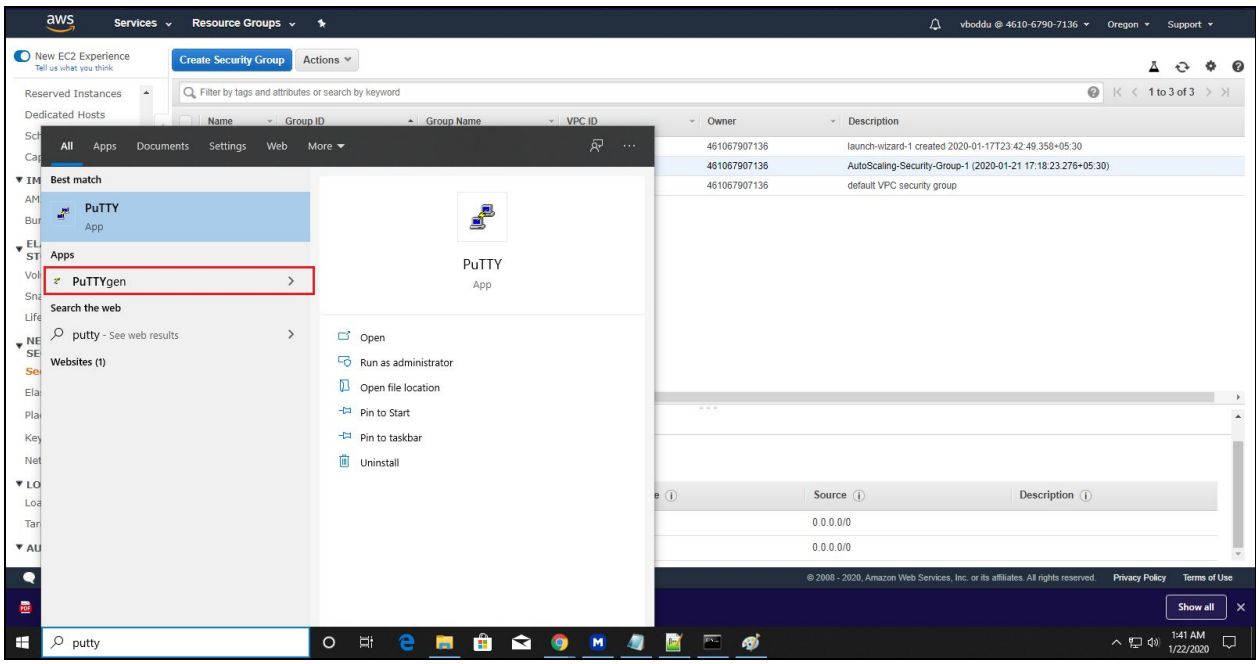
Now review the instance configuration and click on Launch.



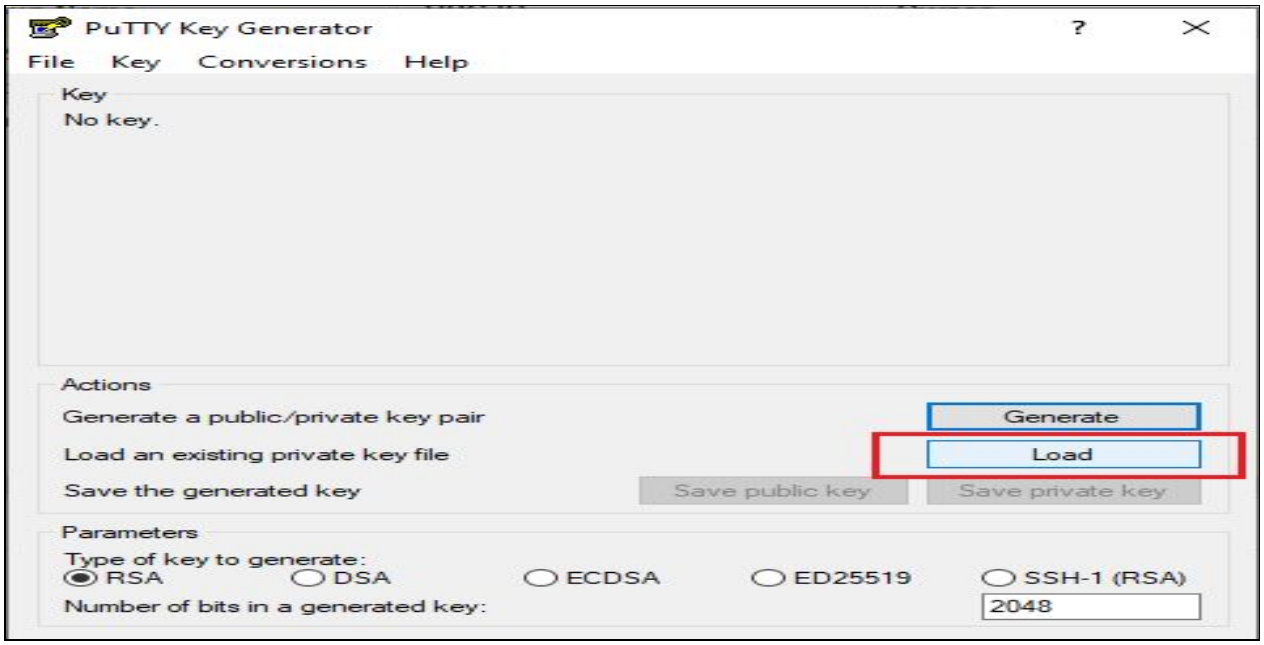
Next, we need to create a key. Using the key we can be able to access the instance. Choose to create a new key and click on the download key pair. A **pem** file will be downloaded.



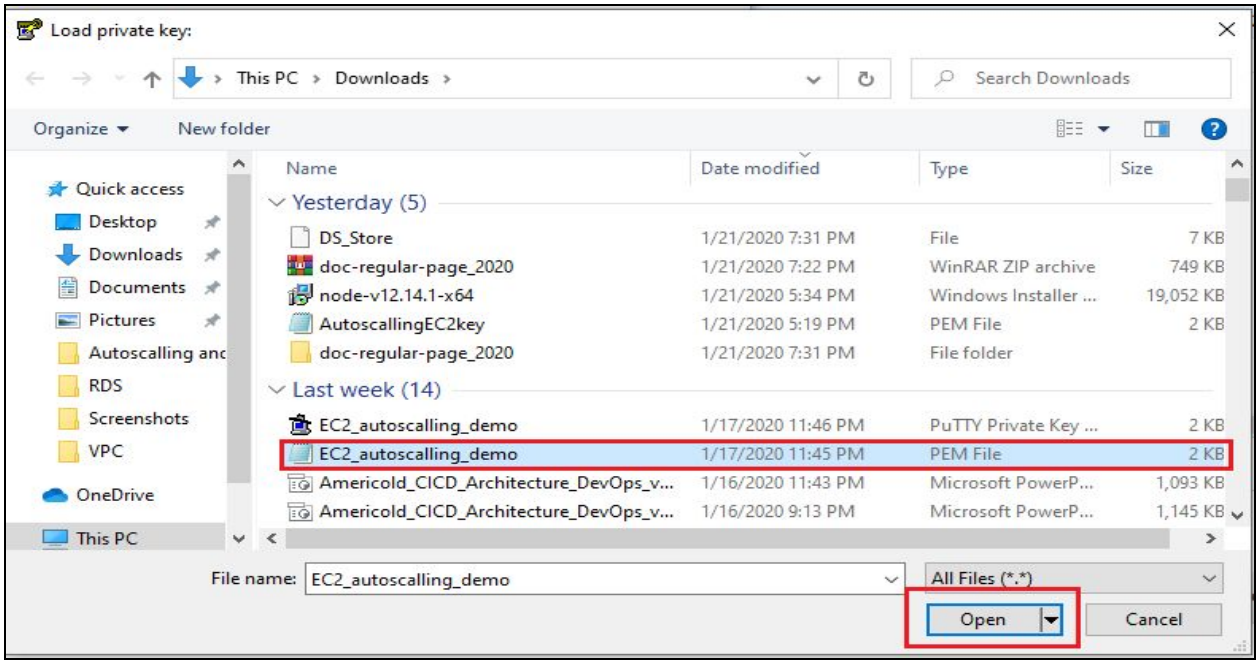
After downloading the key pair we need to generate a public key using the PuTTYgen as follows.



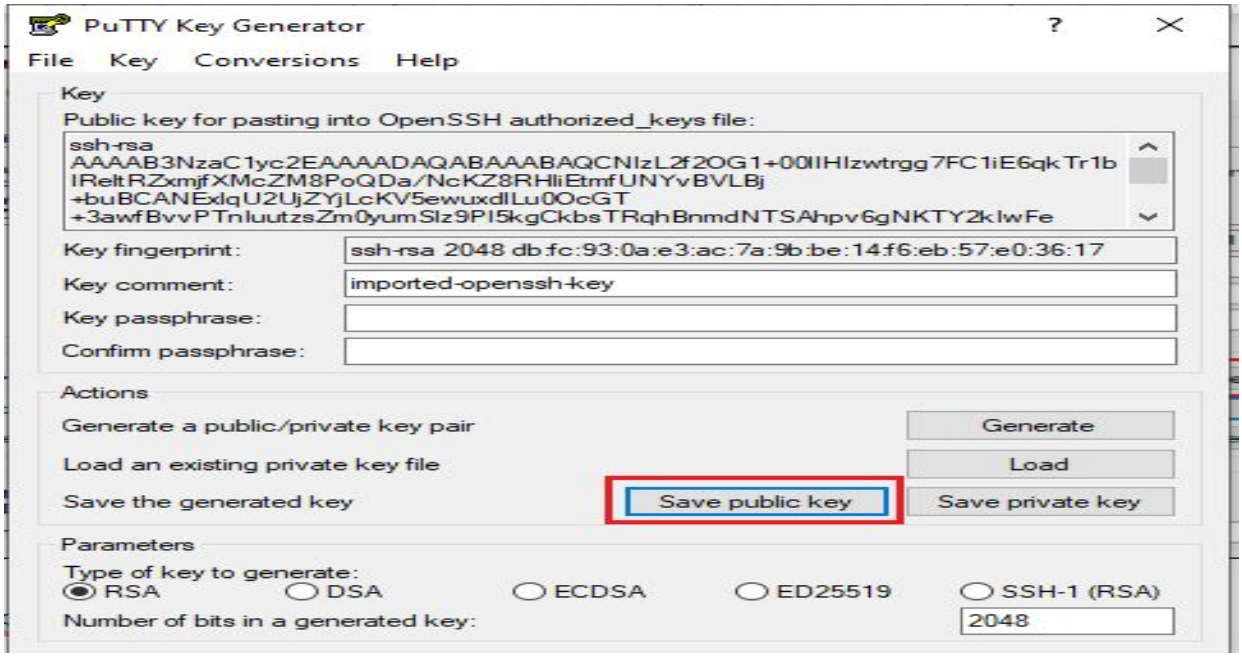
Now, Load the downloaded pem file into the PuTTY Key Generator. Click on load to select the key pair.



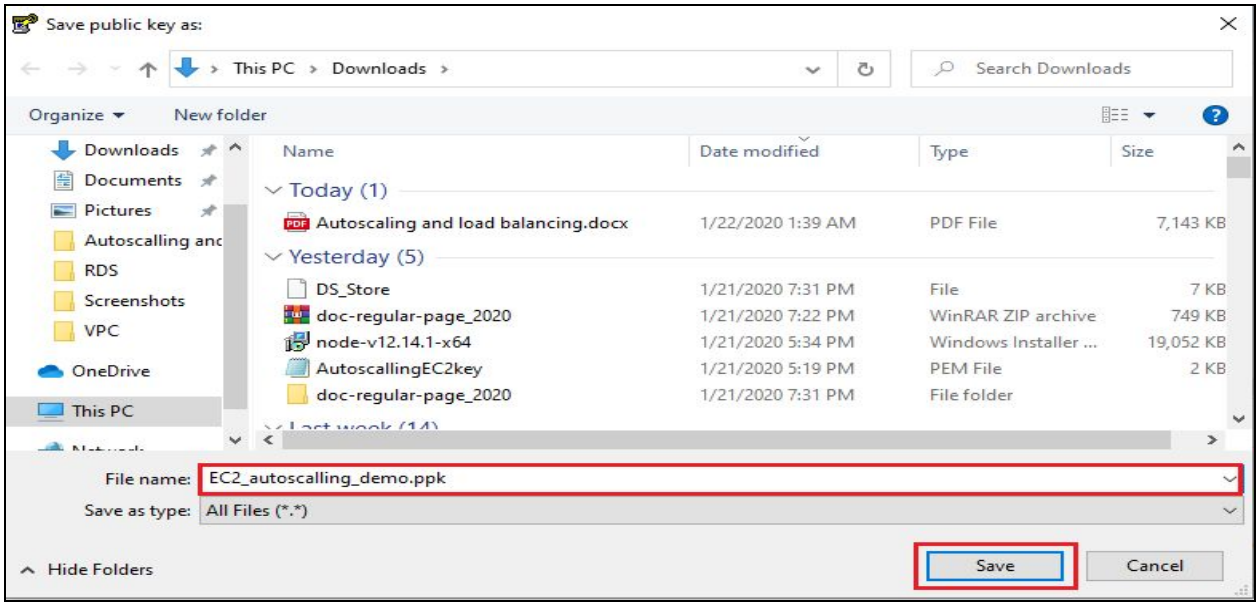
Select the key pair file and click on open



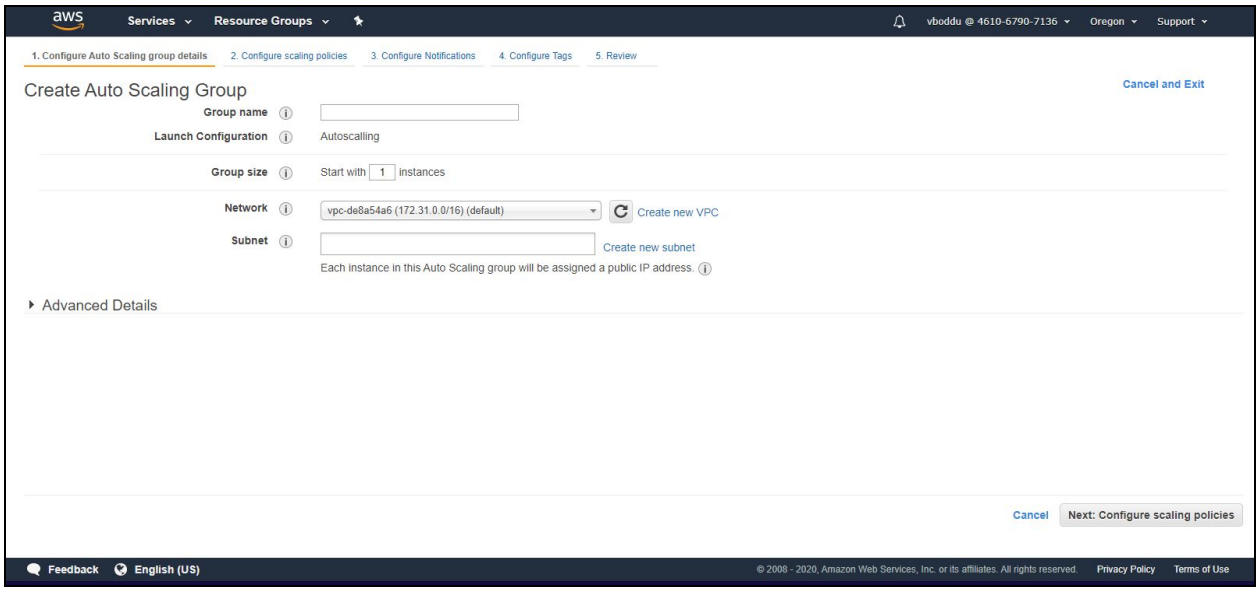
After loading the key pair into Putty key generator click on save public key.



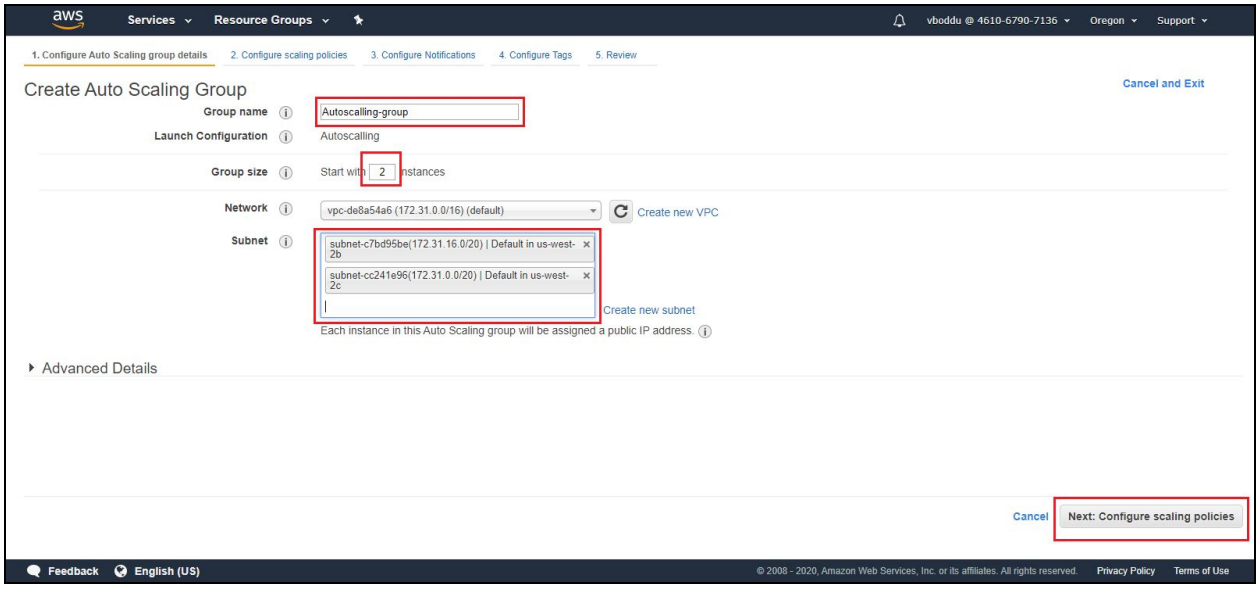
Save the file name with ppk format and click on save.



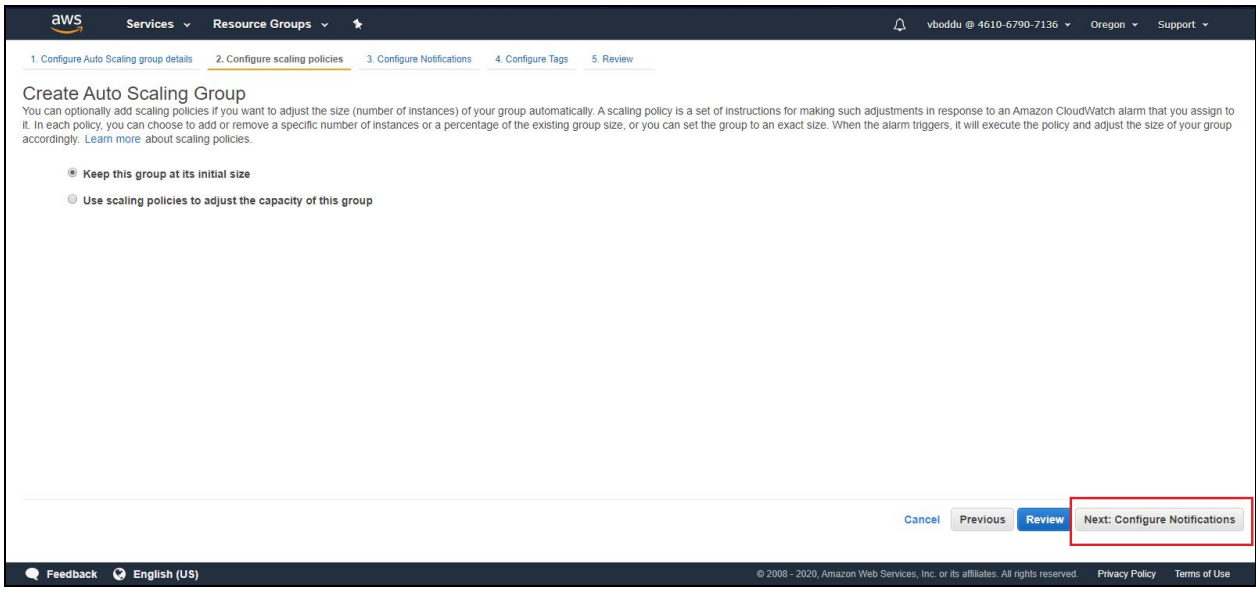
Once we configure the Auto Scaling launch configuration then we need to create the Autoscaling group below is the dashboard for creation of Autoscaling group



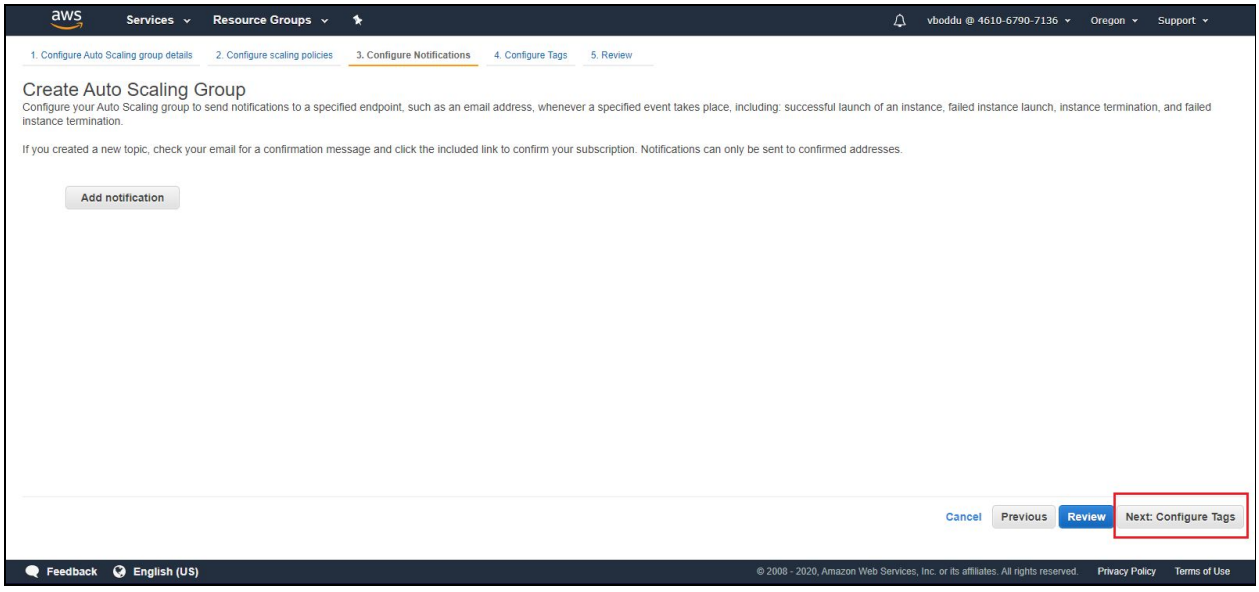
Here We need to provide various inputs like group name, group size and select at least two default subnets and click on create scaling policies.



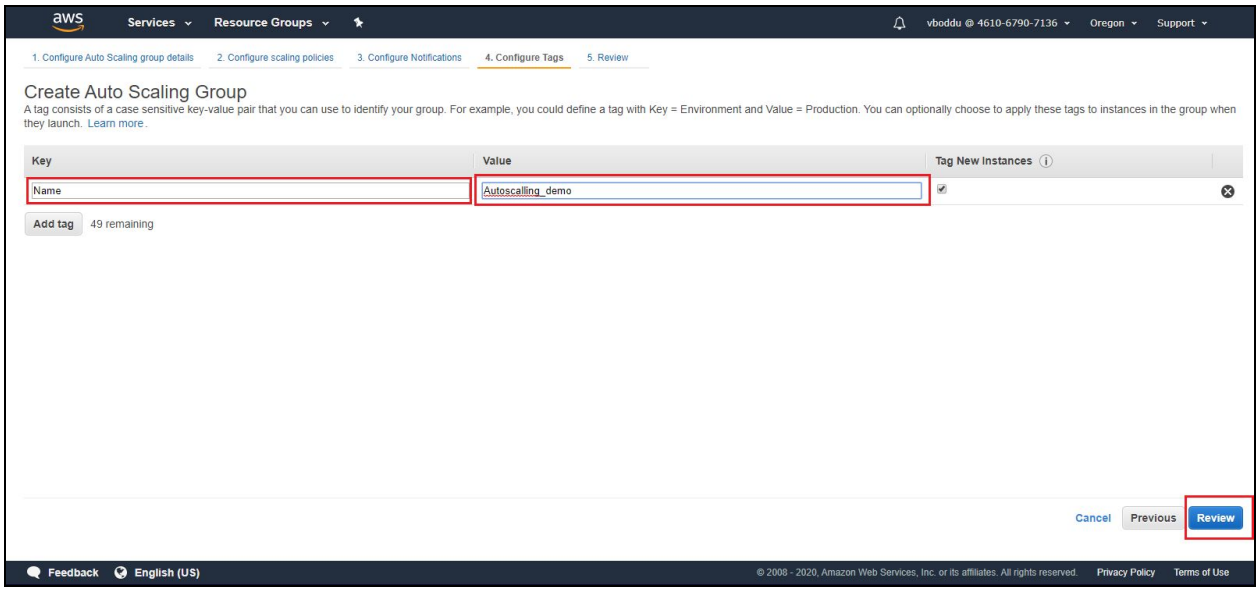
In this section keep everything as default and click on Next.



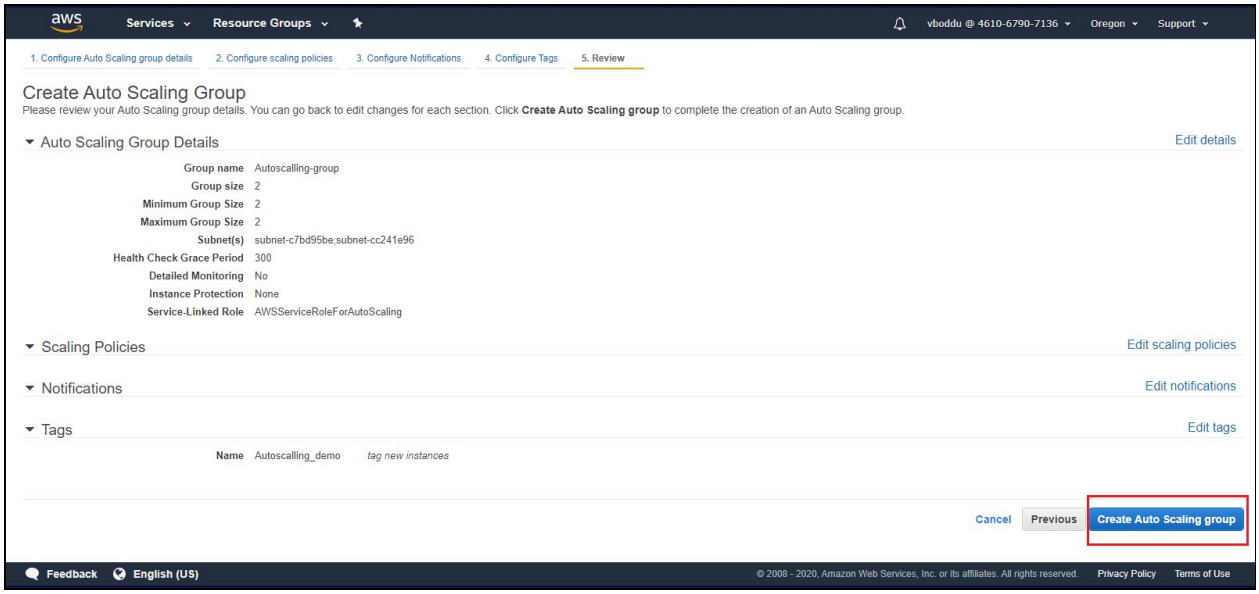
Click on configure tags

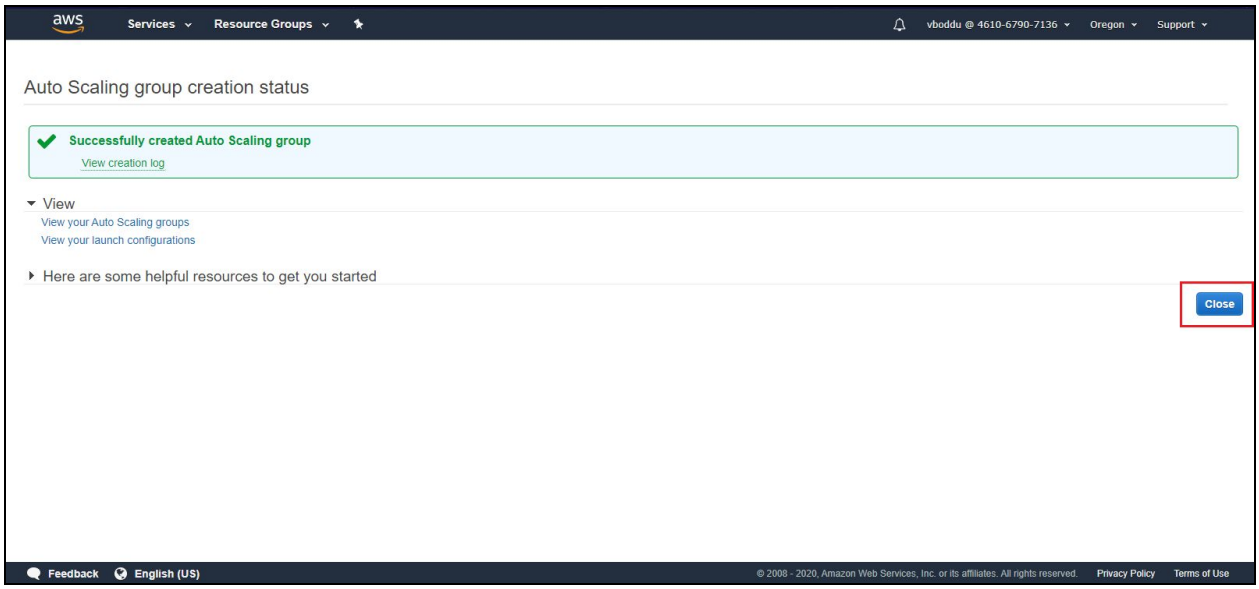


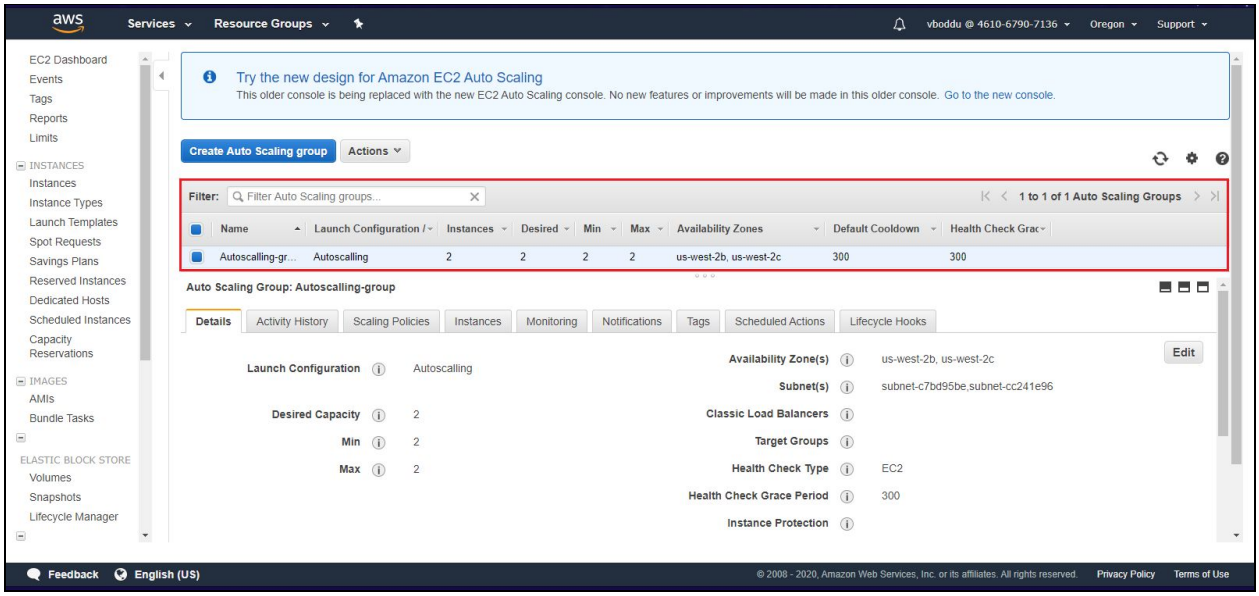
Tags are case-sensitive key-value pairs that define the group. Configure tags and click on the review as shown below.



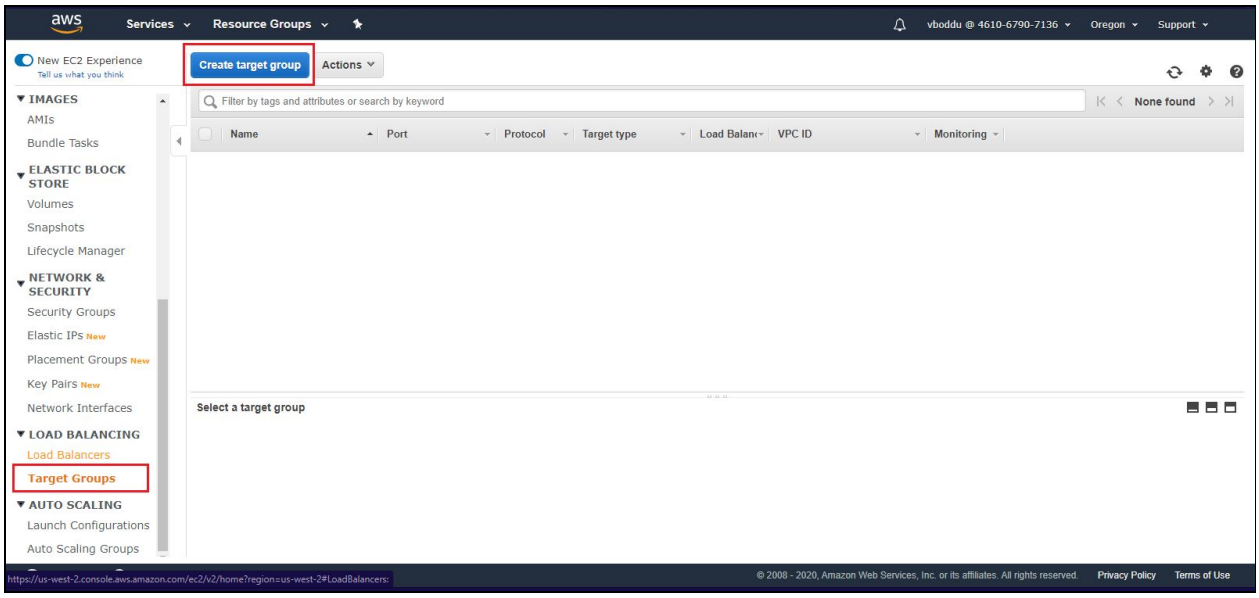
Now review the instance configuration and click on create Autoscaling group.



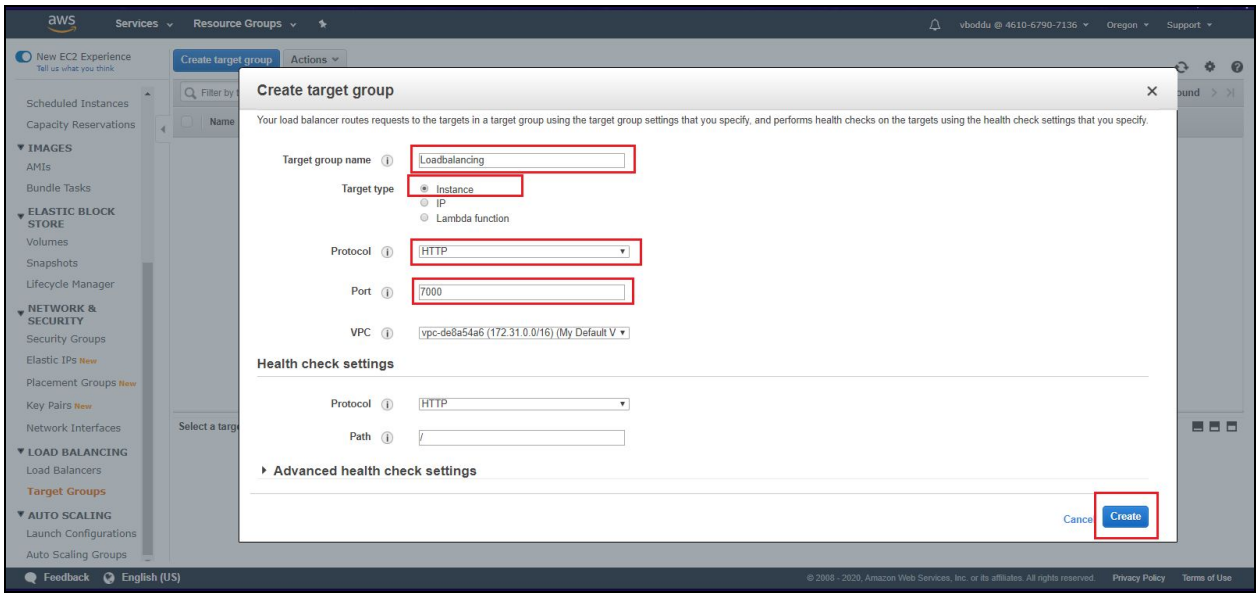


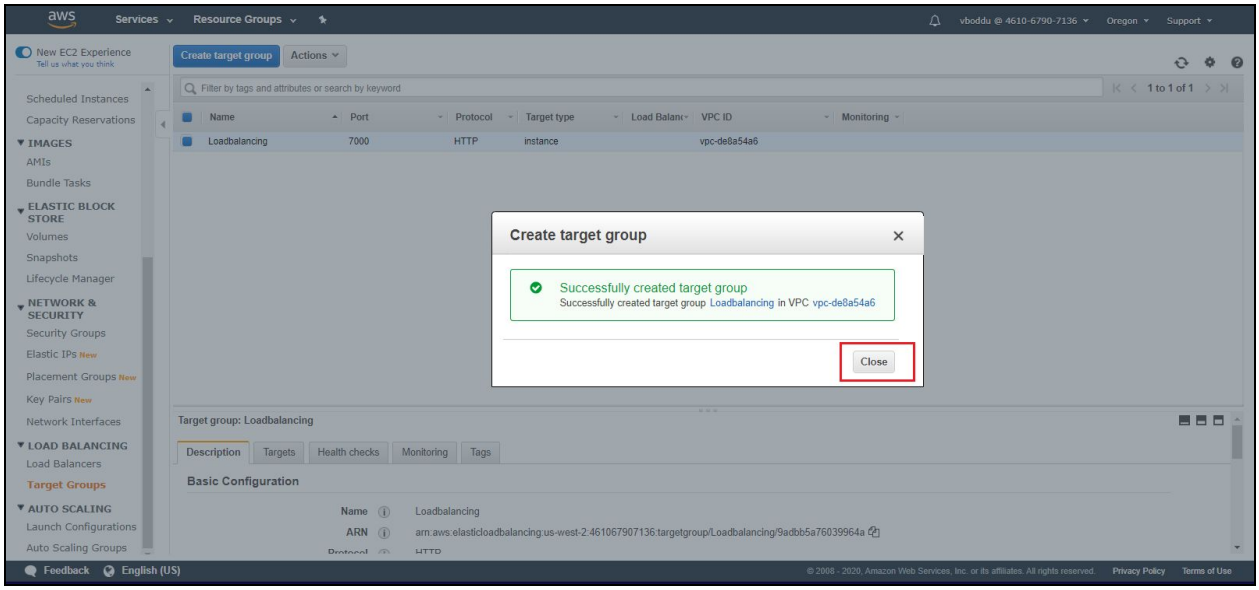


For creation of load balancer first we need to create target groups.

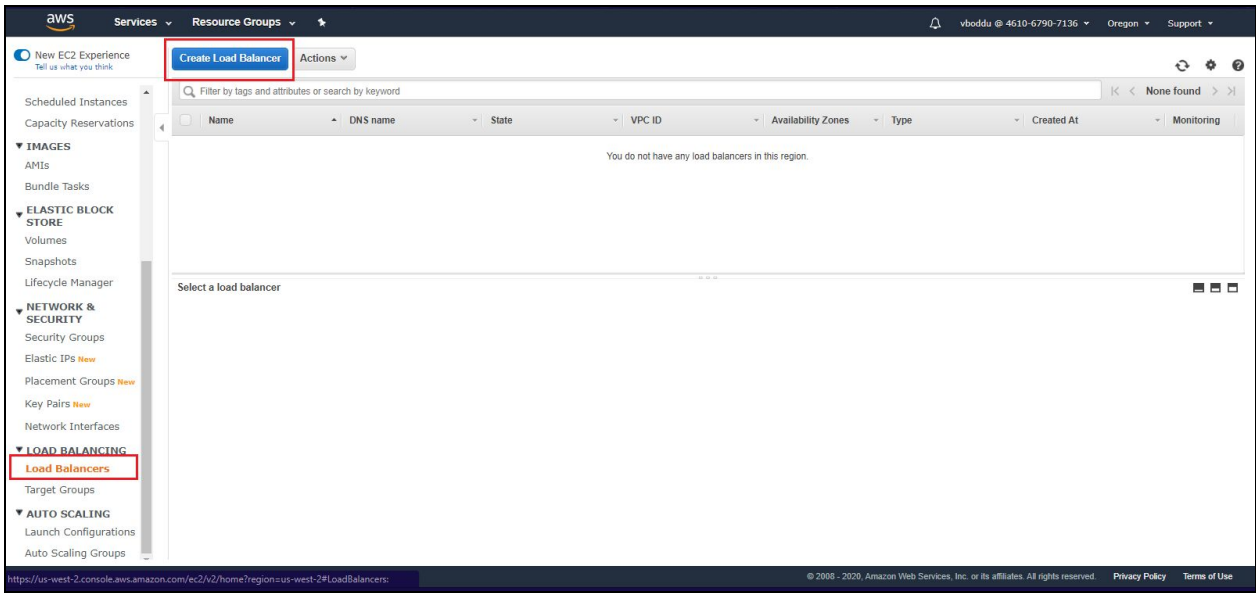


Once we select the create target group, it will list the option where we need to provide various inputs like name of the target group, target type, protocol, port, make remaining as default and create.

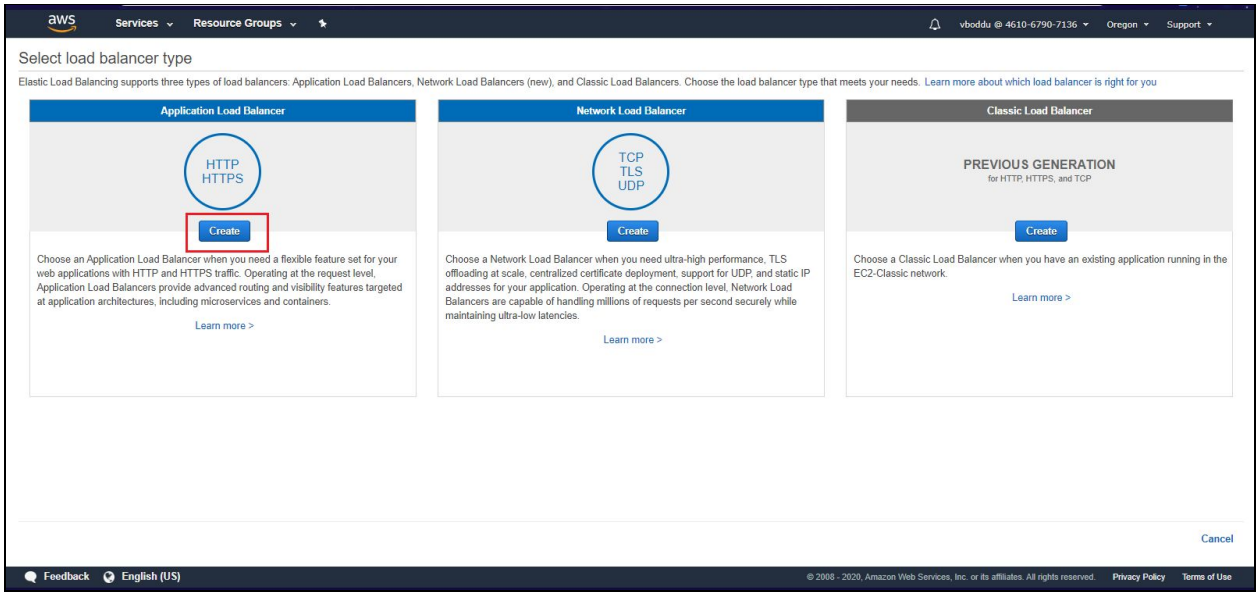




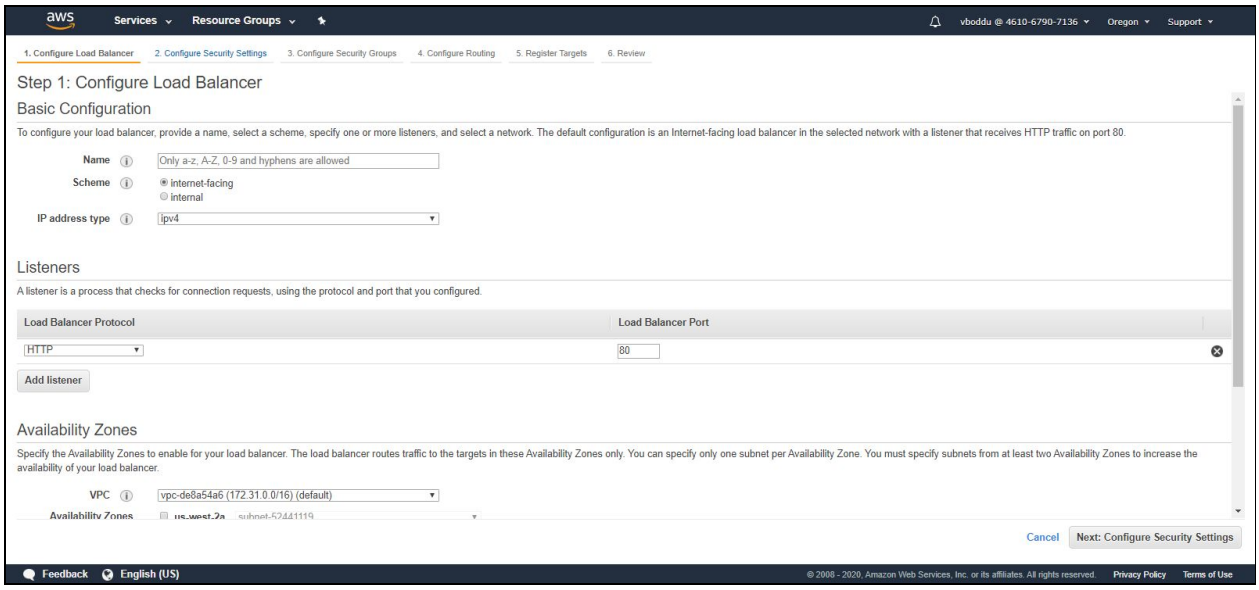
Next we need to create the load balancer, select load balancers and create load balancer.

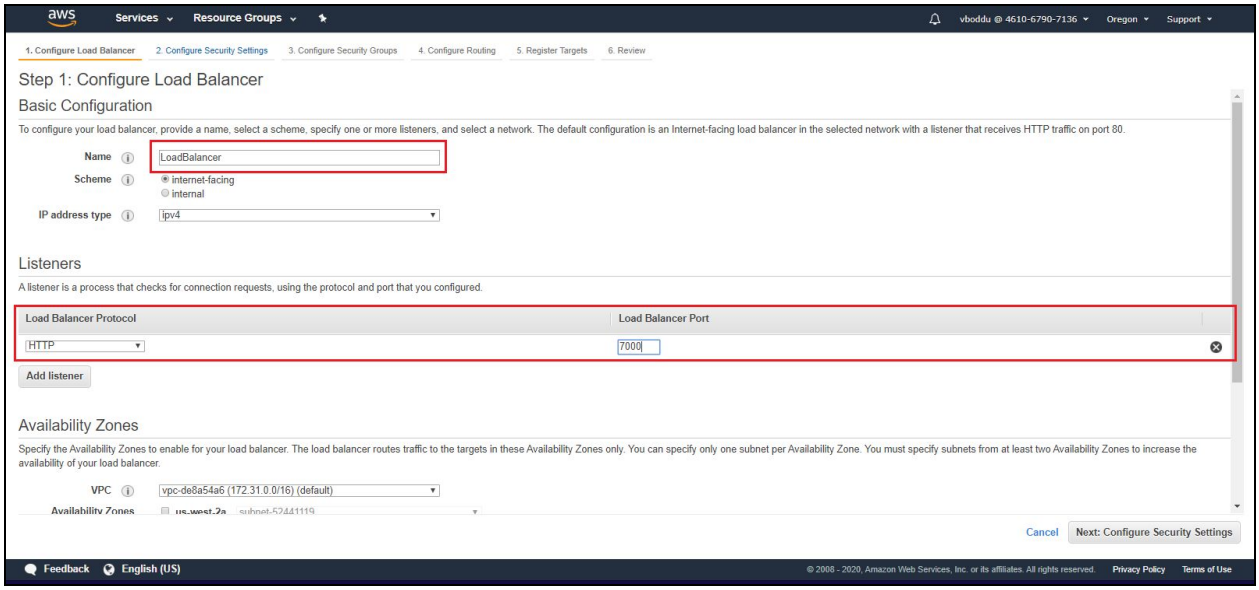


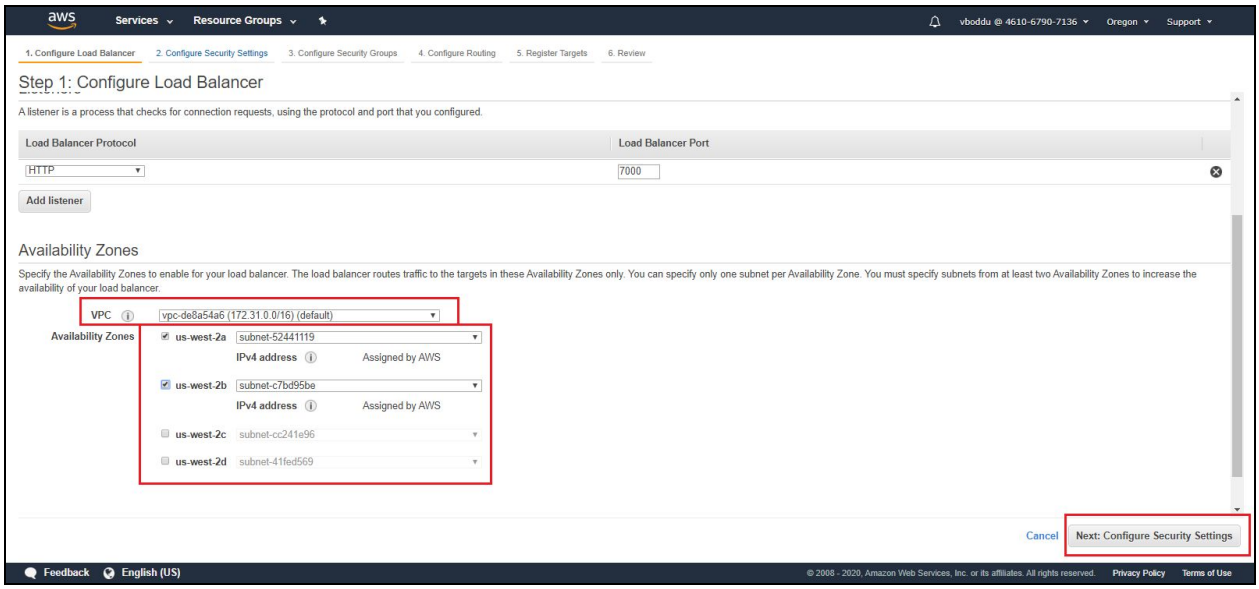
Once we select the create load balancer, it will list the types of load balancer select the application load balancer.



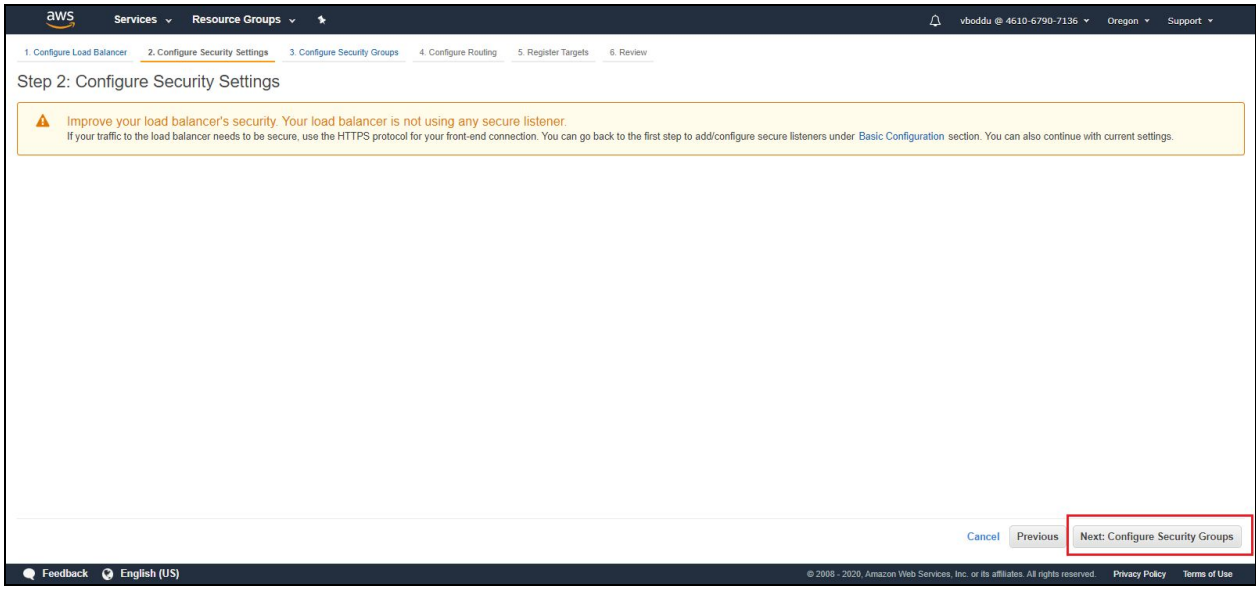
when we select the application load balancer, it will provide various inputs like name of the load balancer, listeners(connection requests) and choose availability zones(subnets). Click on next to configure security settings.



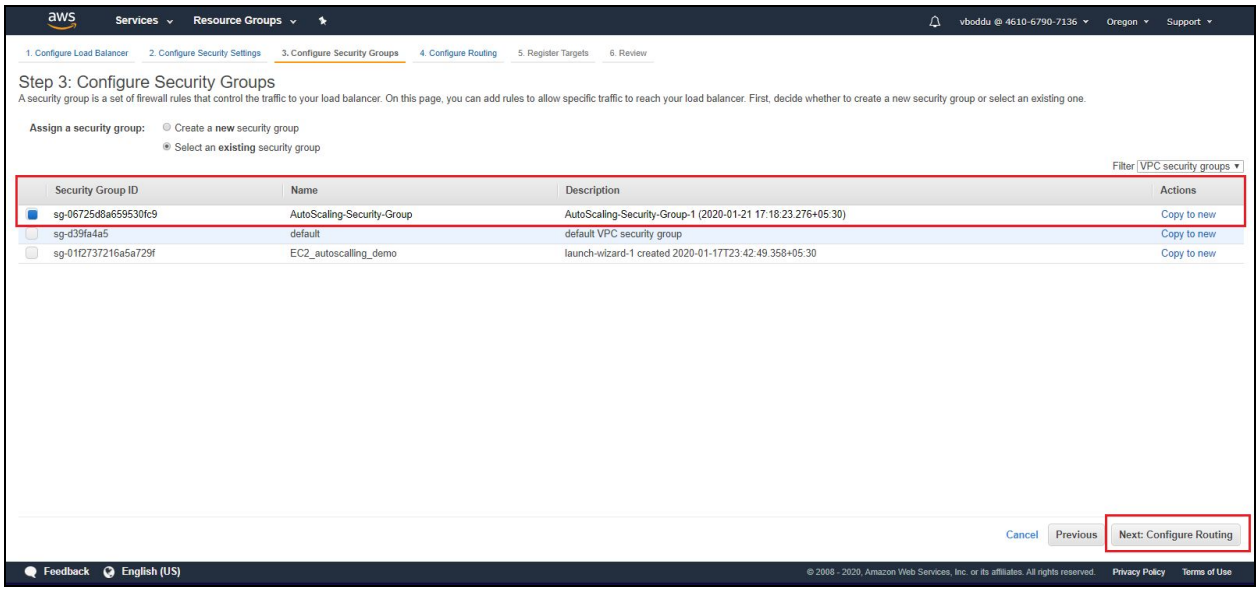




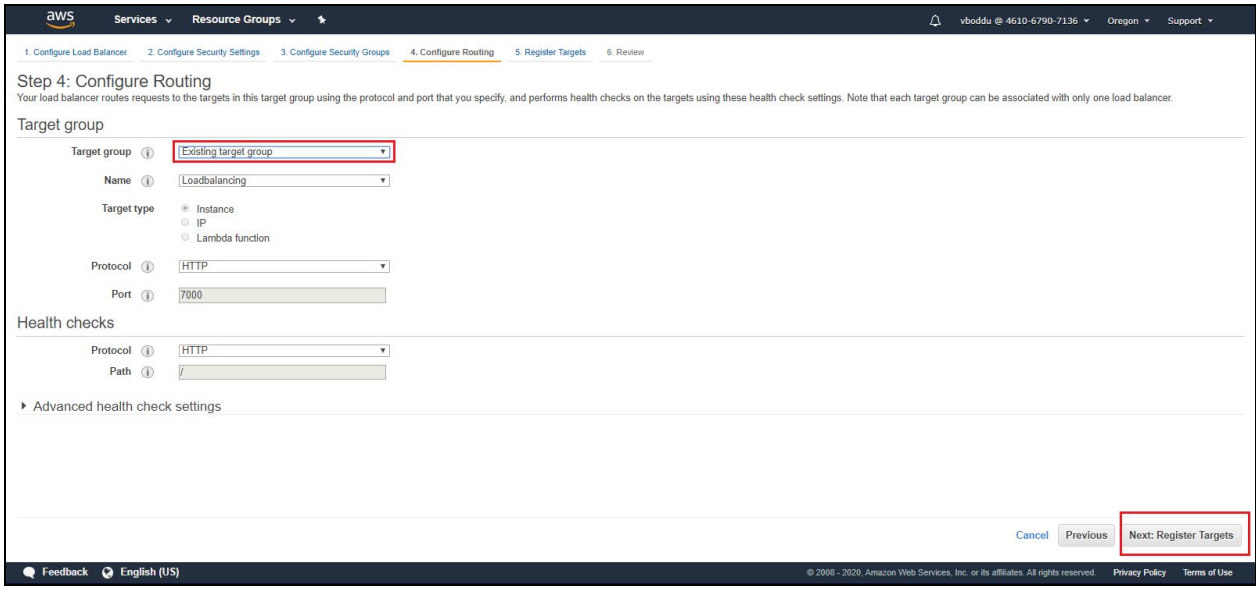
Click on next to configure security groups.



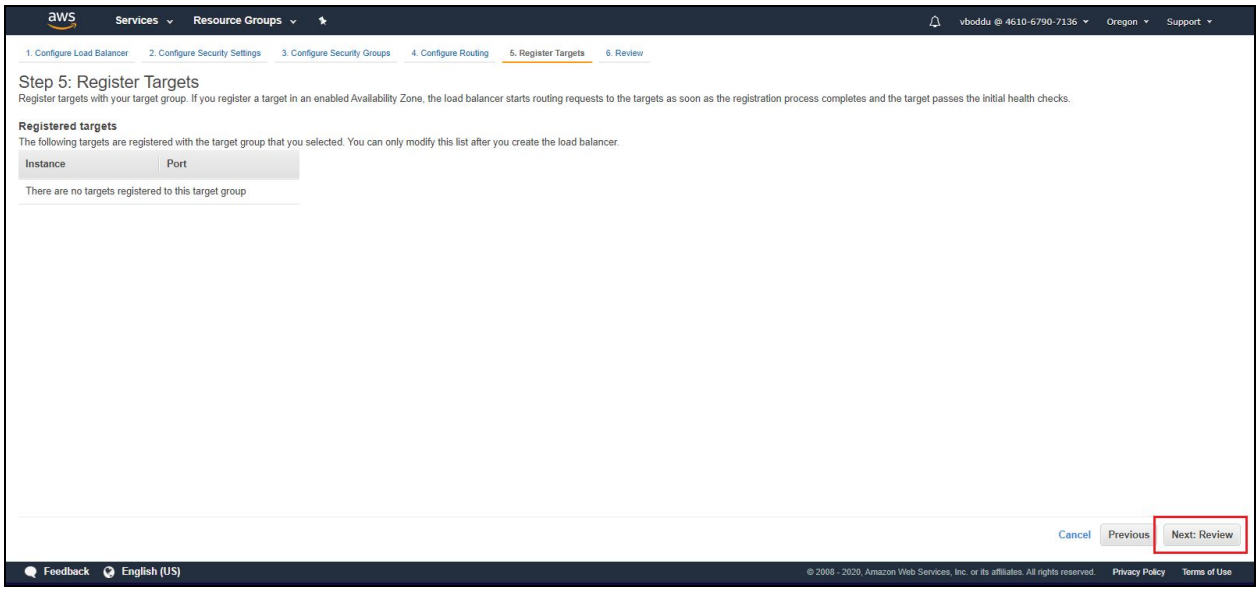
Here choose the security groups and click on configure routing.



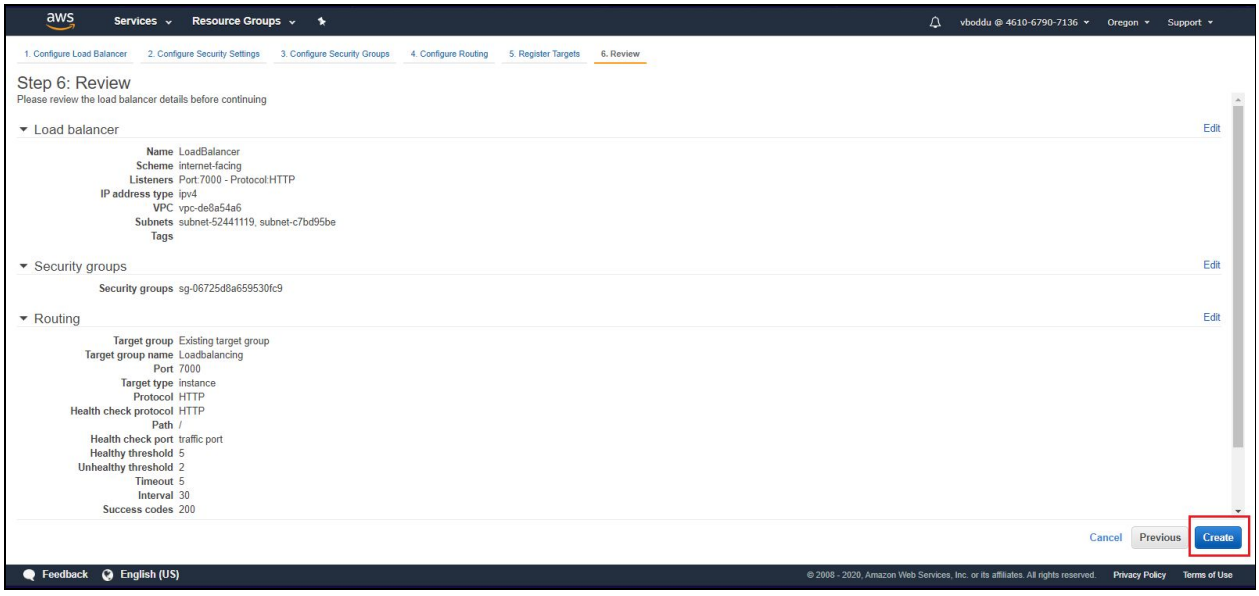
In the configure routing section, select the existing target group that you created before, and click on register targets

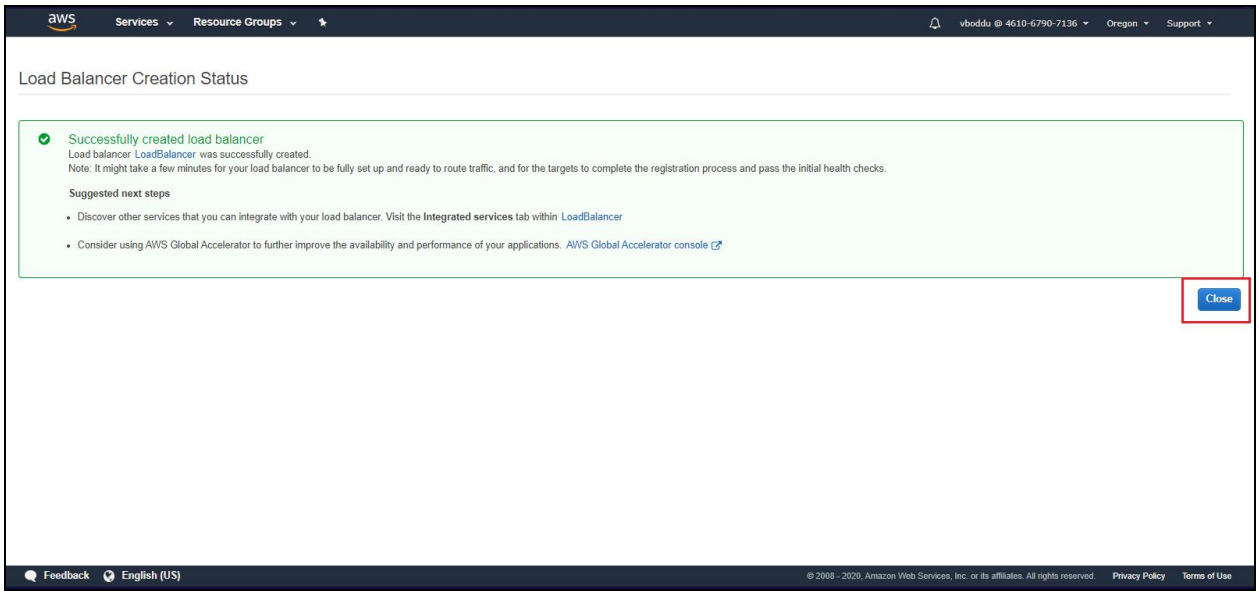


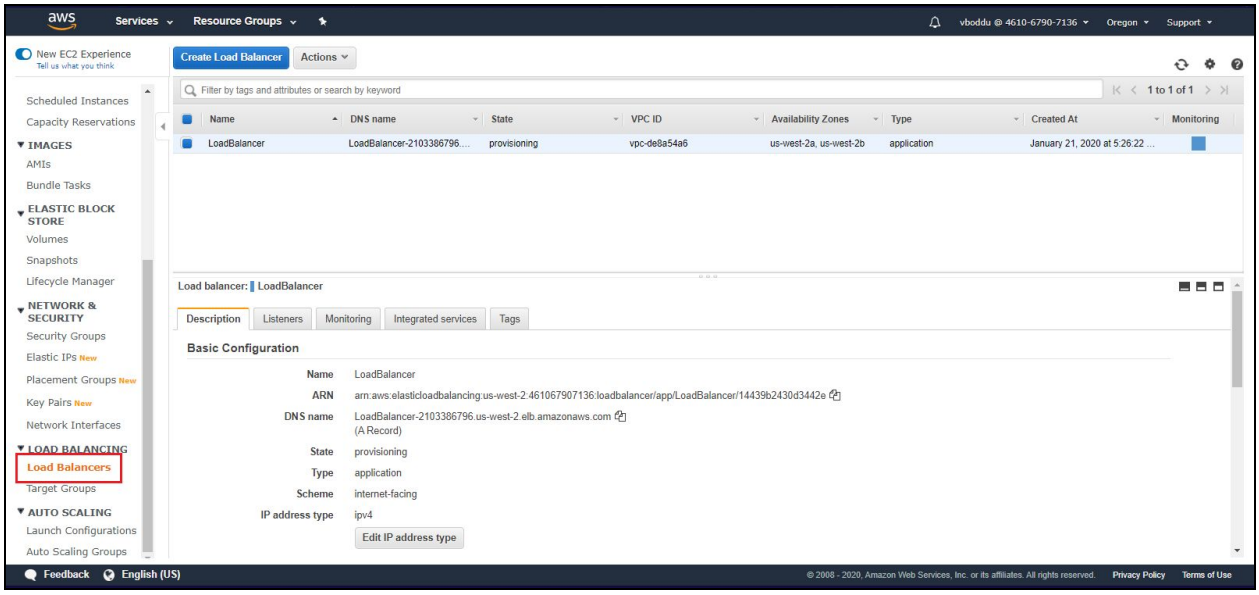
Here click on next



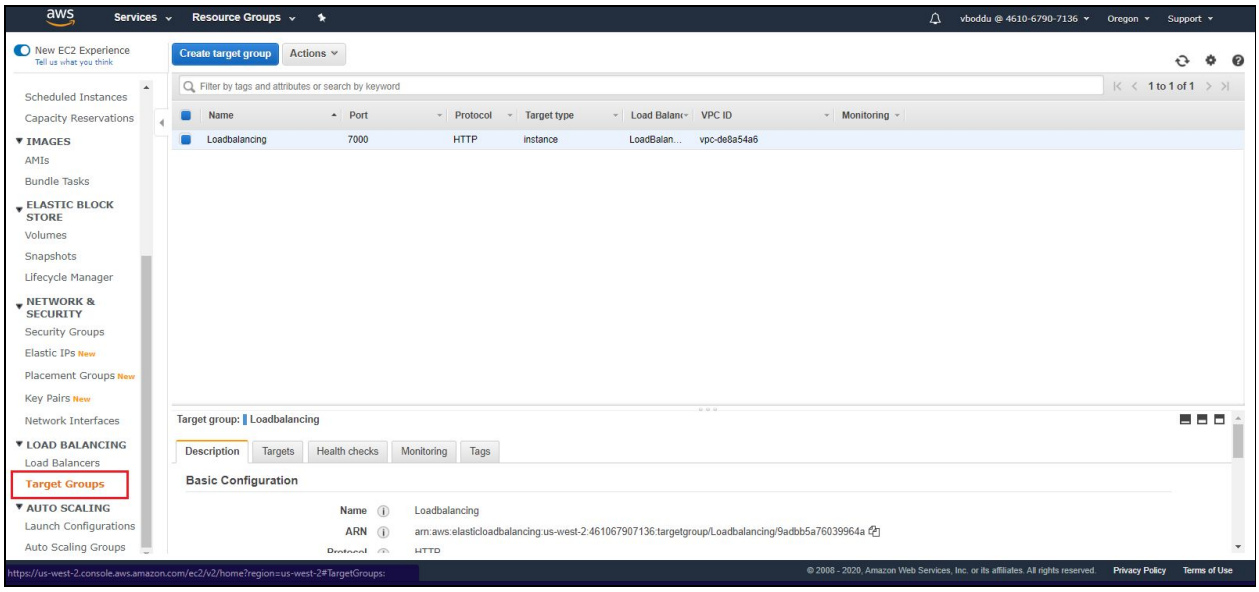
Now review the configuration and click on Create.



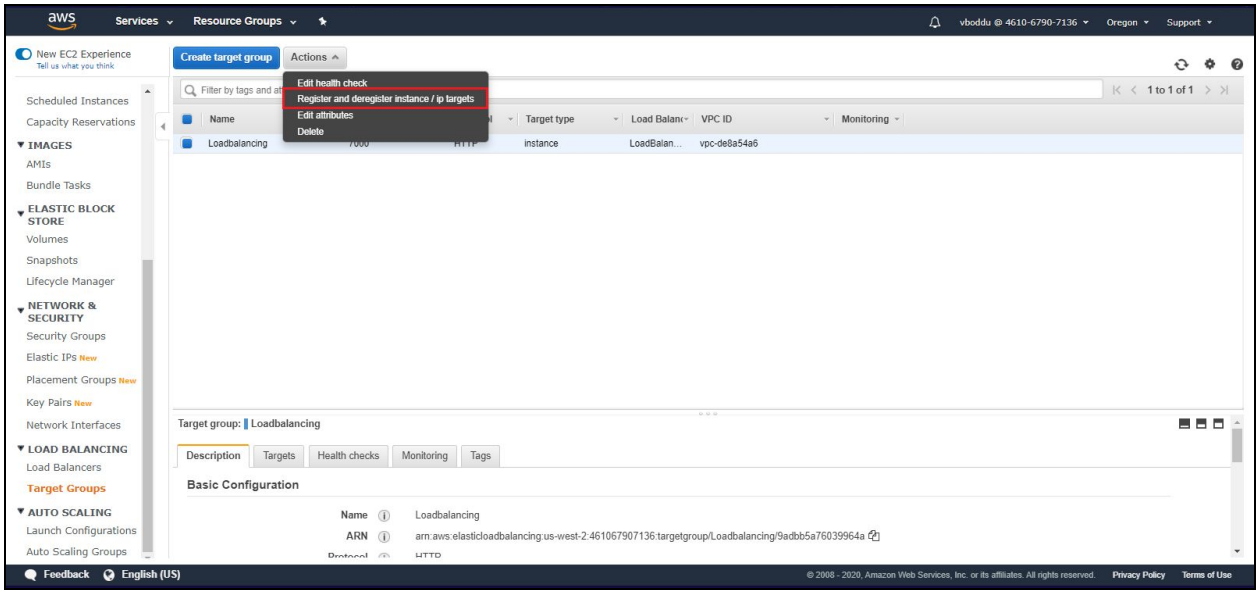

aws
Services
Resource Groups
vboddu @ 4610-6790-7136
Oregon
Support
Load Balancer Creation Status
Successfully created load balancer
Load balancer LoadBalancer was successfully created.
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.
Suggested next steps
Discover other services that you can integrate with your load balancer. Visit the Integrated services tab within LoadBalancer
Consider using AWS Global Accelerator to further improve the availability and performance of your applications. AWS Global Accelerator console
Close
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
vboddu @ 4610-6790-7136
Oregon
Support
New EC2 Experience
Tell us what you think
Create Load Balancer
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
DNS name
State
VPC ID
Availability Zones
Type
Created At
Monitoring
LoadBalancer
LoadBalancer-2103386796....
provisioning
vpc-de8a54a6
us-west-2a, us-west-2b
application
January 21, 2020 at 5:26:22 ...
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Load balancer: LoadBalancer
Description
Listeners
Monitoring
Integrated services
Tags
Basic Configuration
Name LoadBalancer
ARN arn:aws:elasticloadbalancing:us-west-2:461067907136:loadbalancer/app/LoadBalancer/14439b2430d3442e
DNS name LoadBalancer-2103386796.us-west-2.elb.amazonaws.com (A Record)
State provisioning
Type application
Scheme internet-facing
IP address type ipv4
Edit IP address type
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

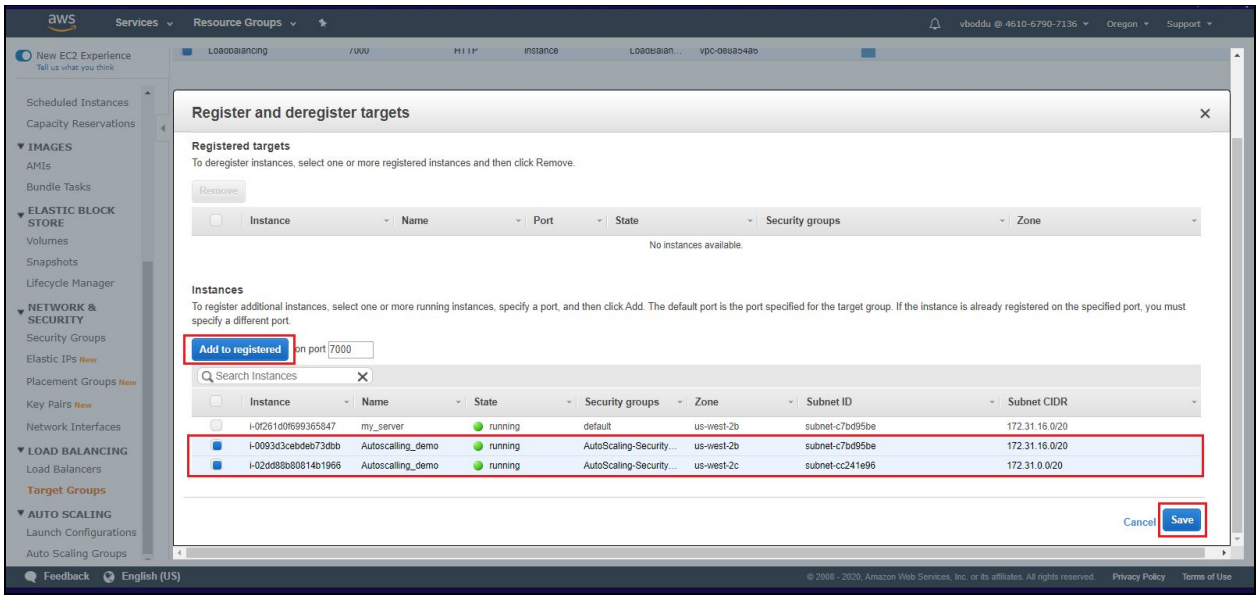
Click on target groups to register the instances as shown below.



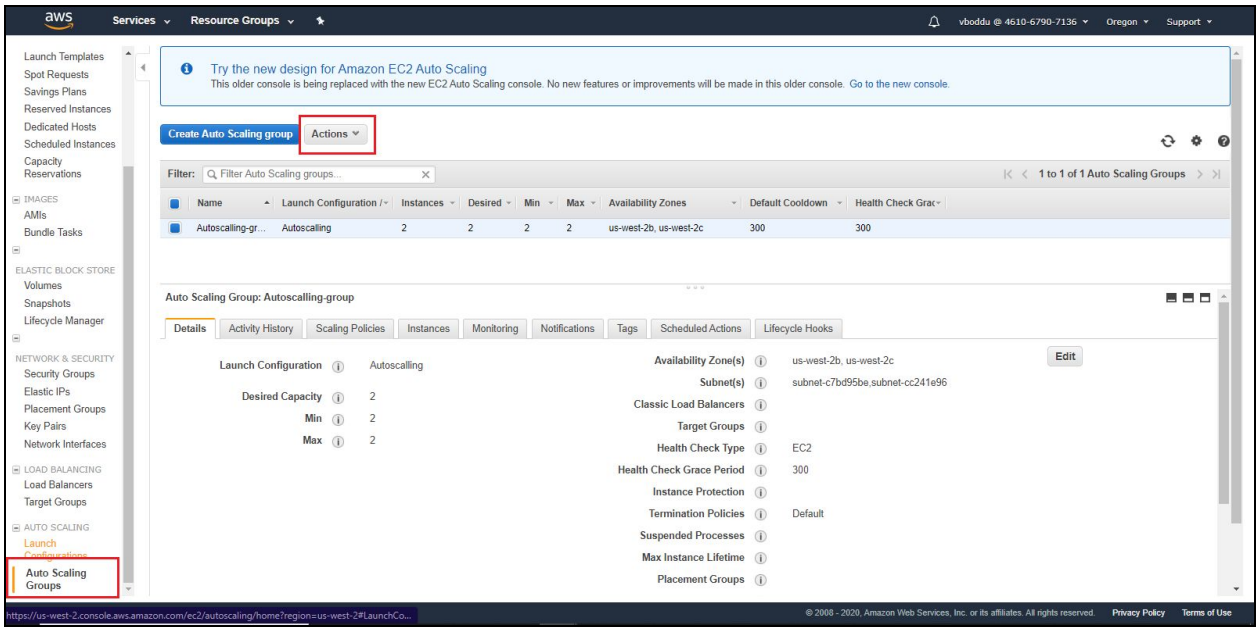
Choose actions and select Register ip targets



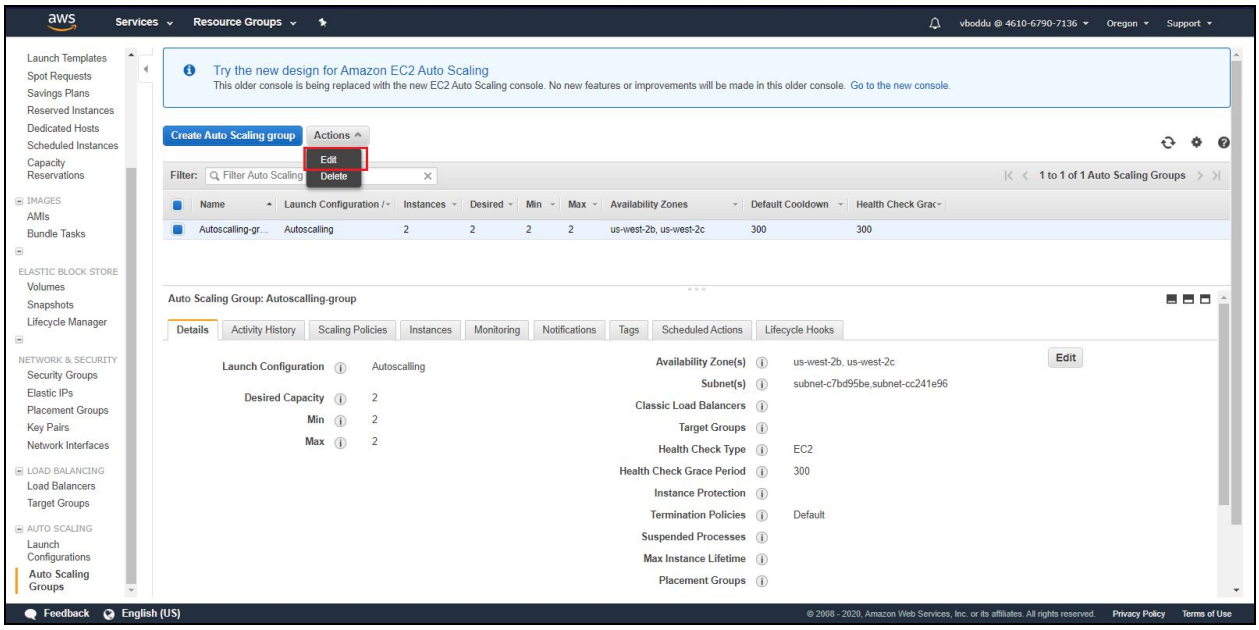
Here select the Autoscaling instances and click on save.



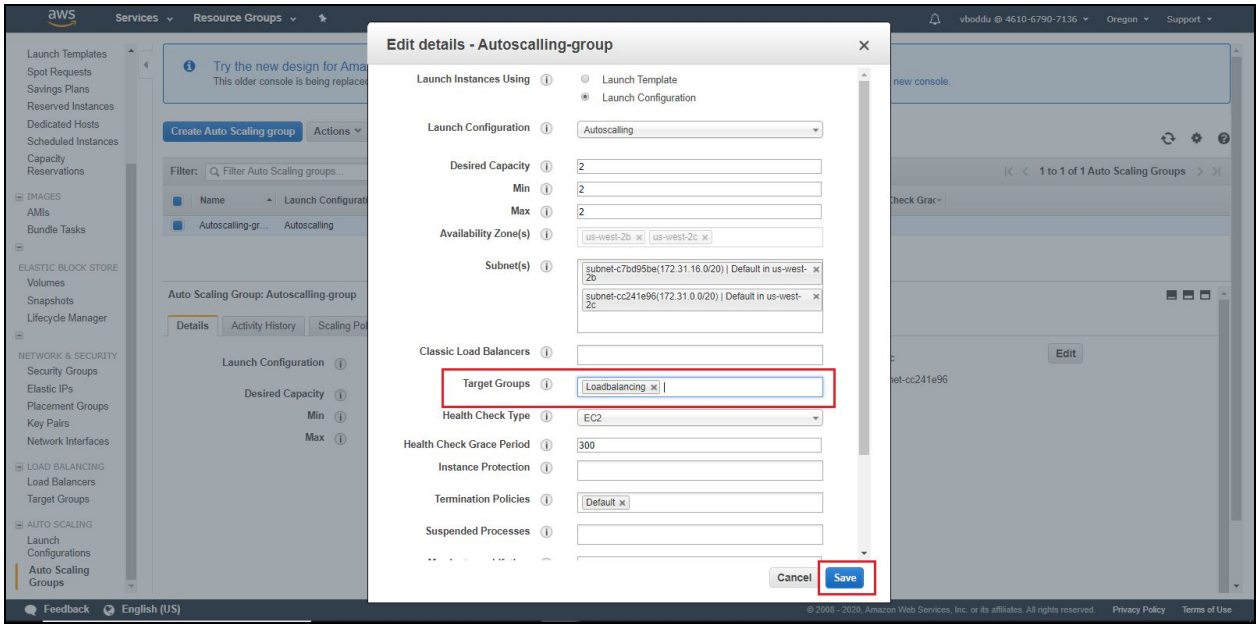
To attach load balancer to auto scaling group click on auto scaling groups and then select actions dropdown



Click on edit.

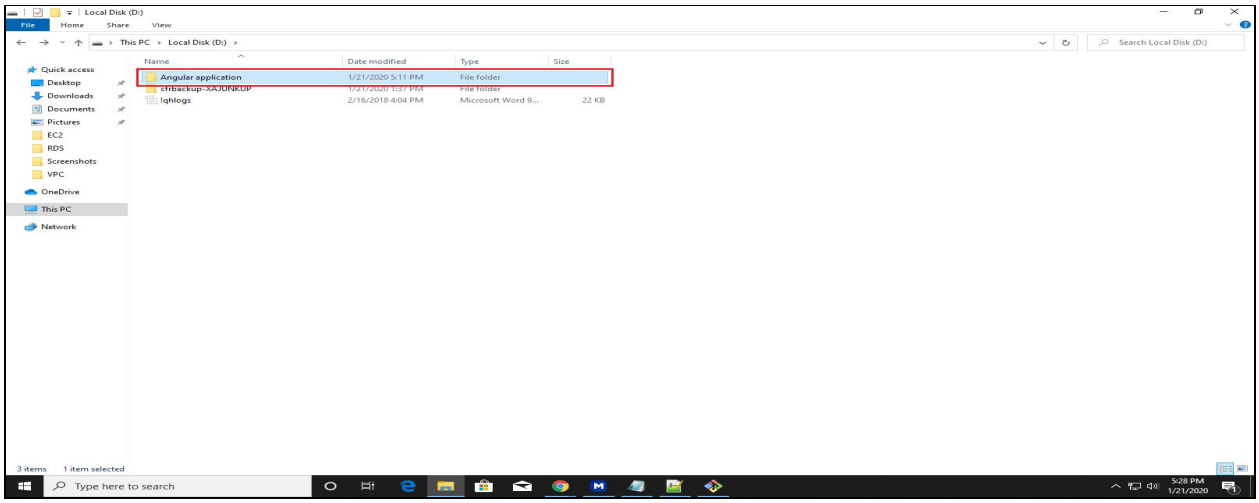


Here choose the load balancer target group and click on the save as shown below.

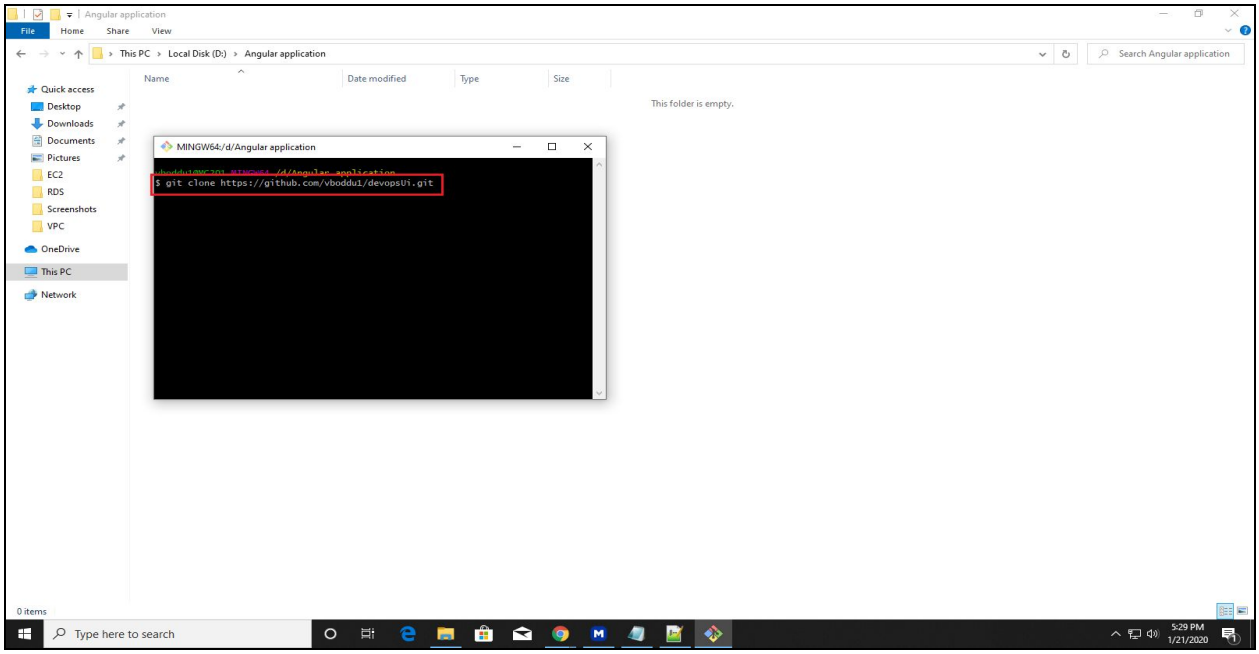


Static Web Hosting on S3

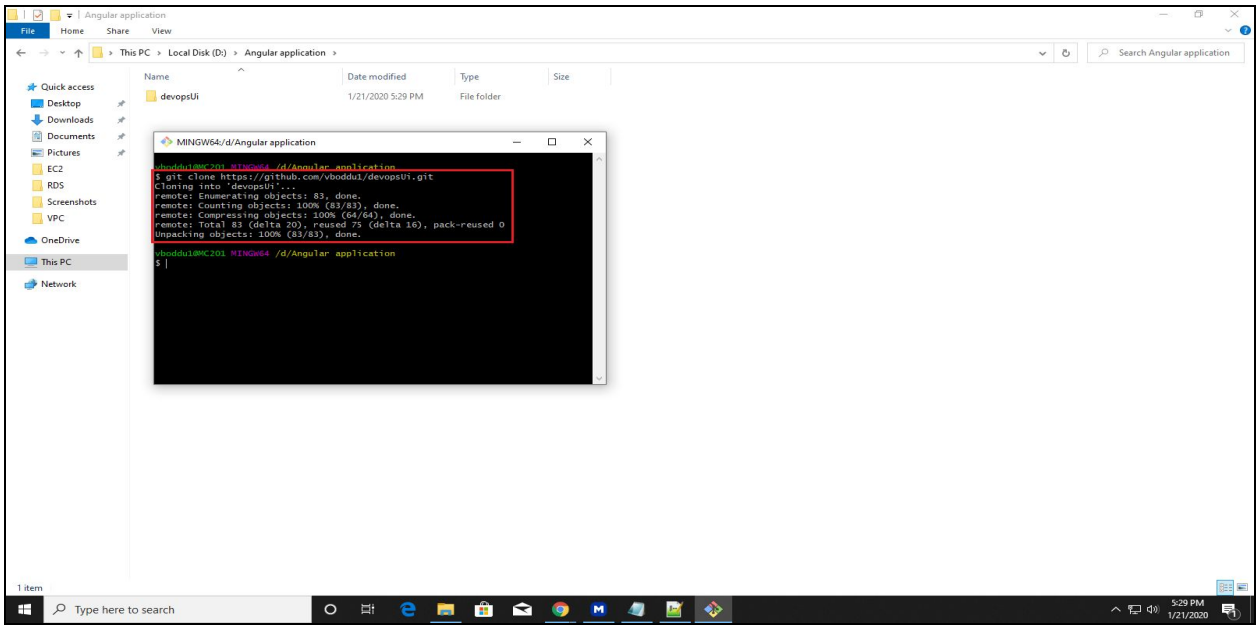
Create a folder

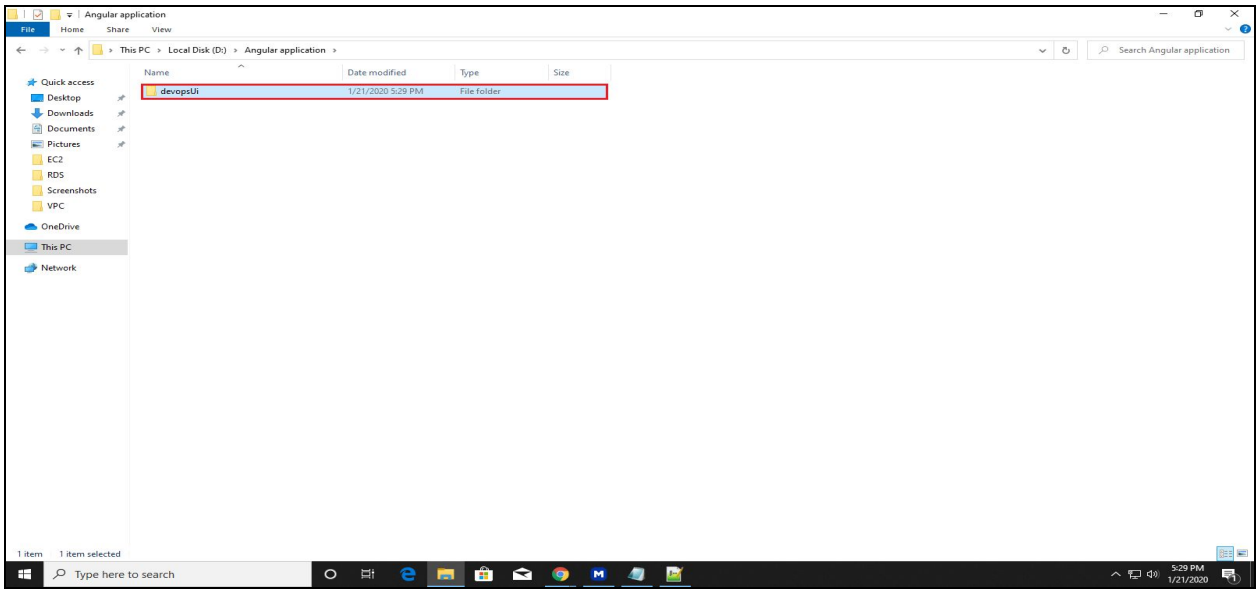


Clone your application using below git URL https://github.com/vboddu1/devopsUi.git

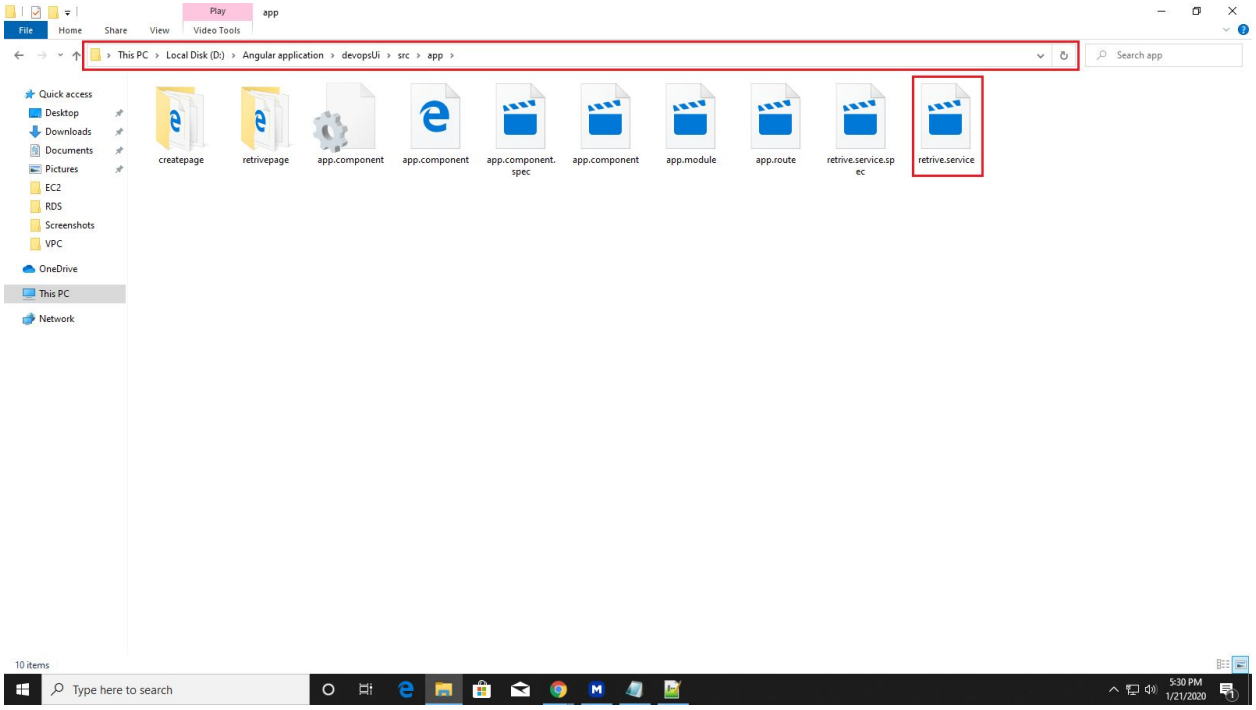


Your application was cloned successfully

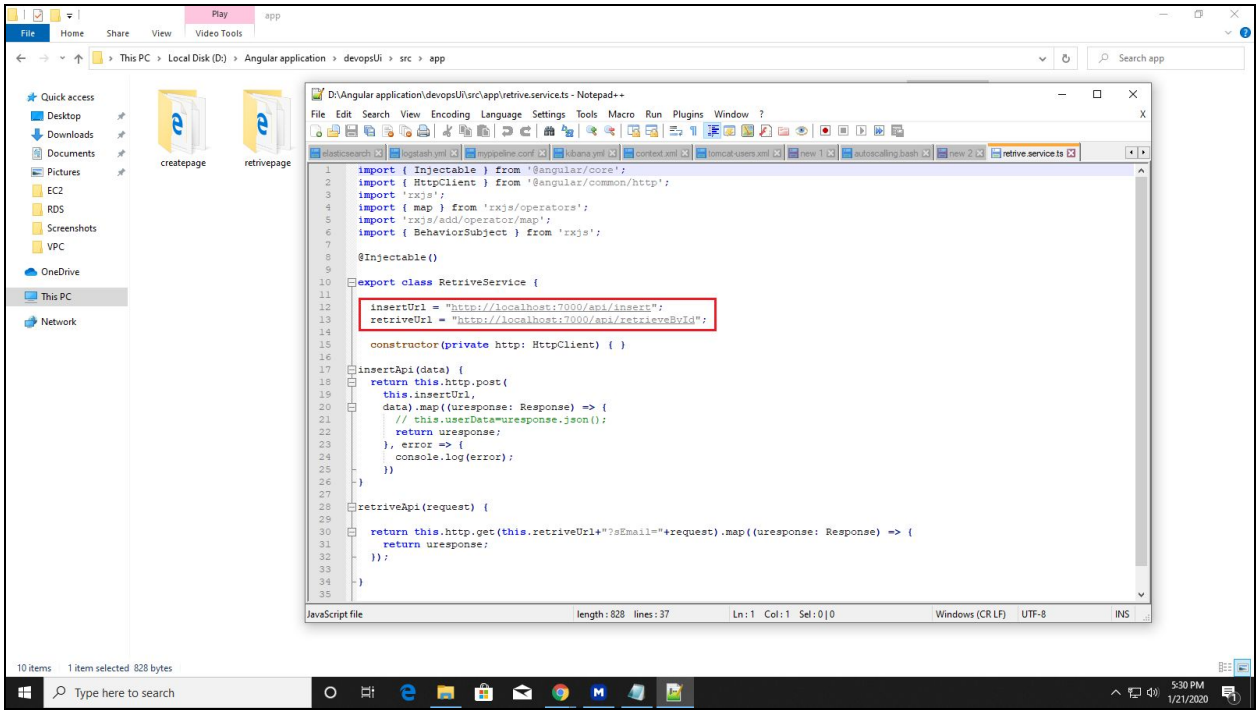


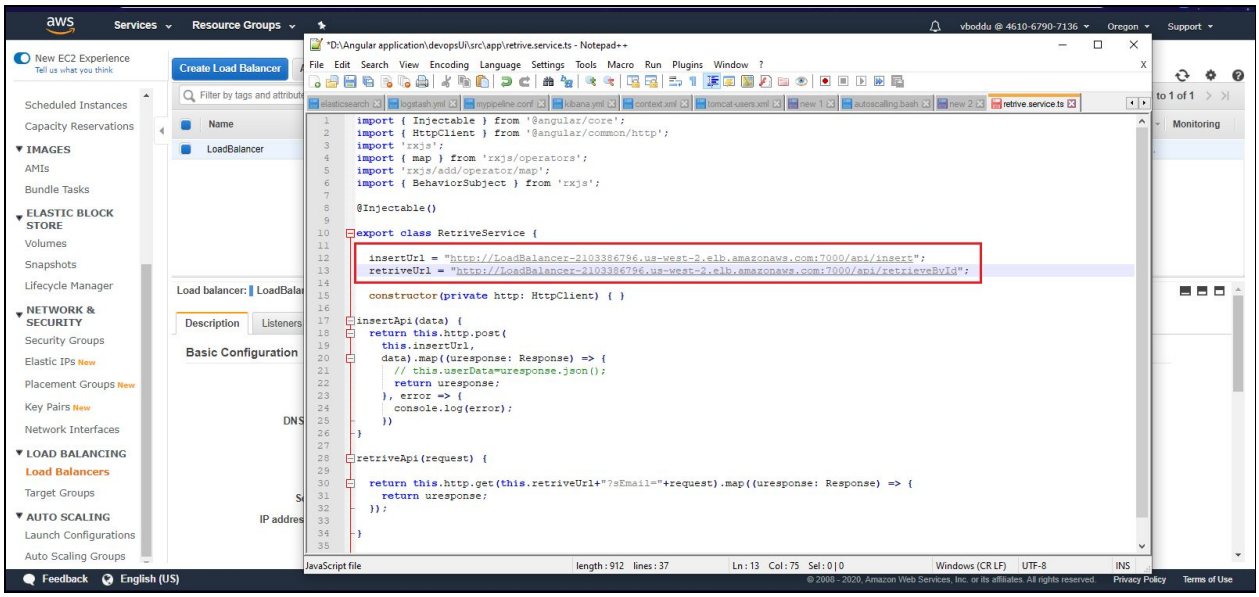


Go to the path and open retrive.service

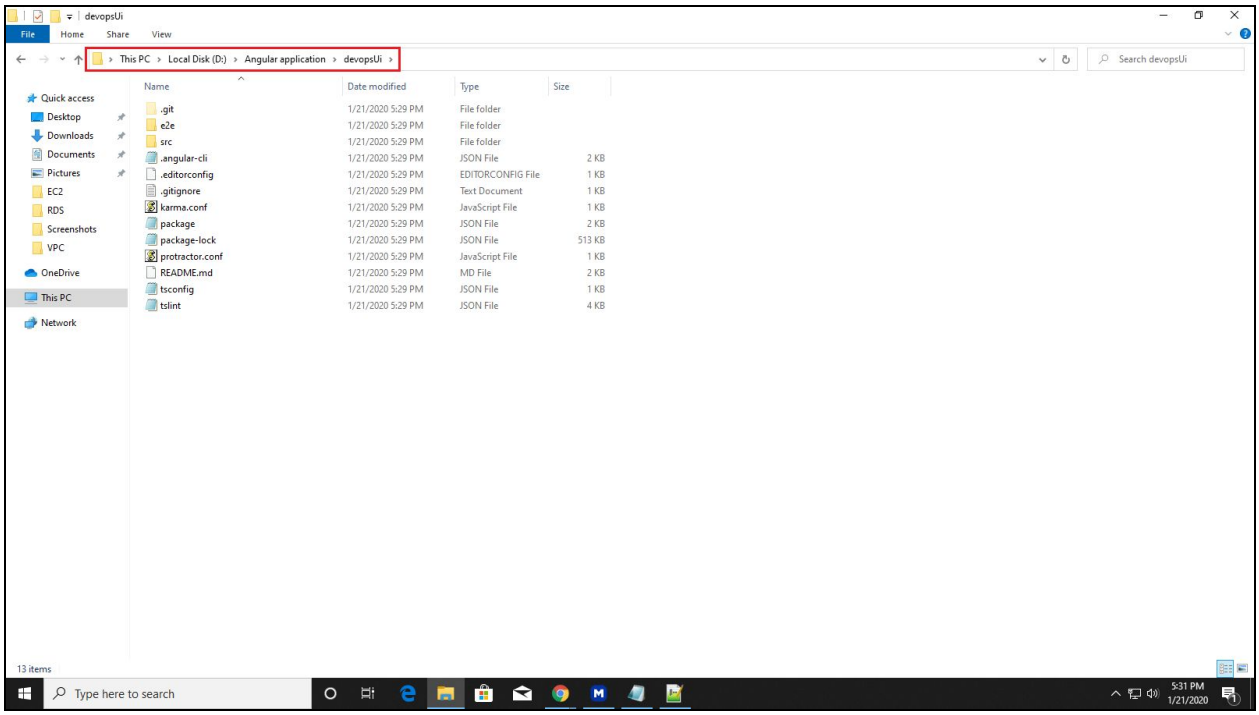


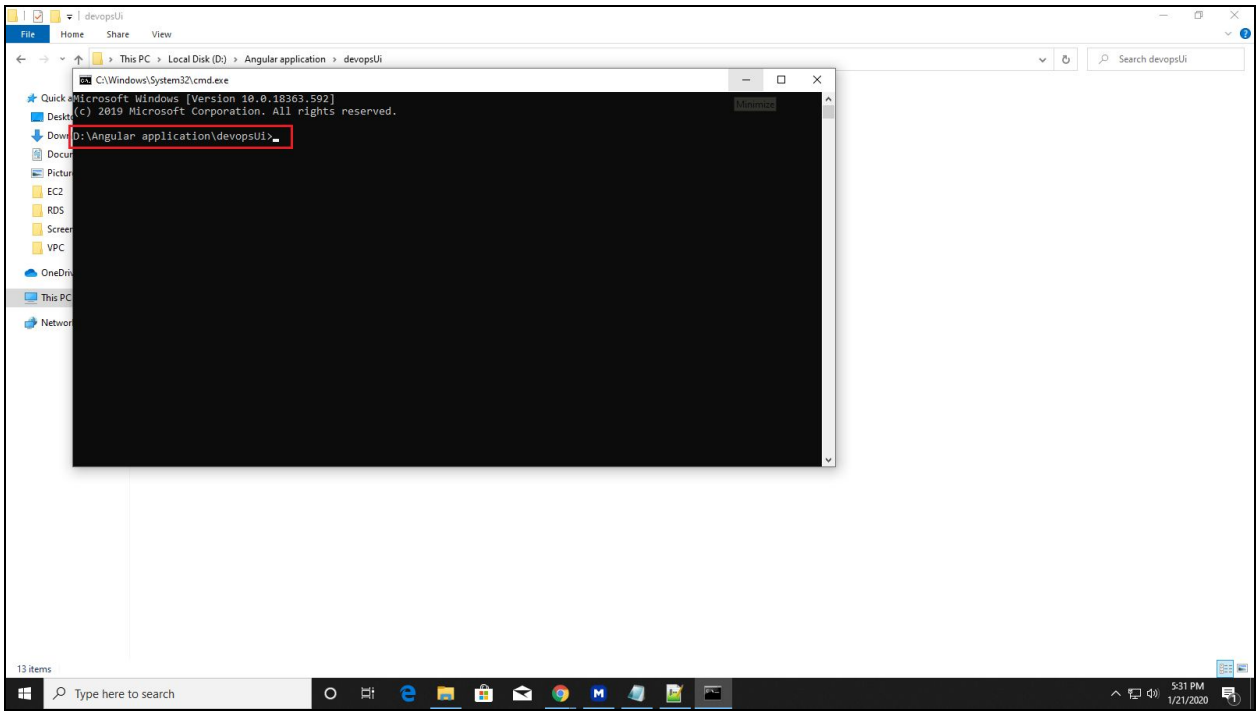
Replace localhost with load balancing URL



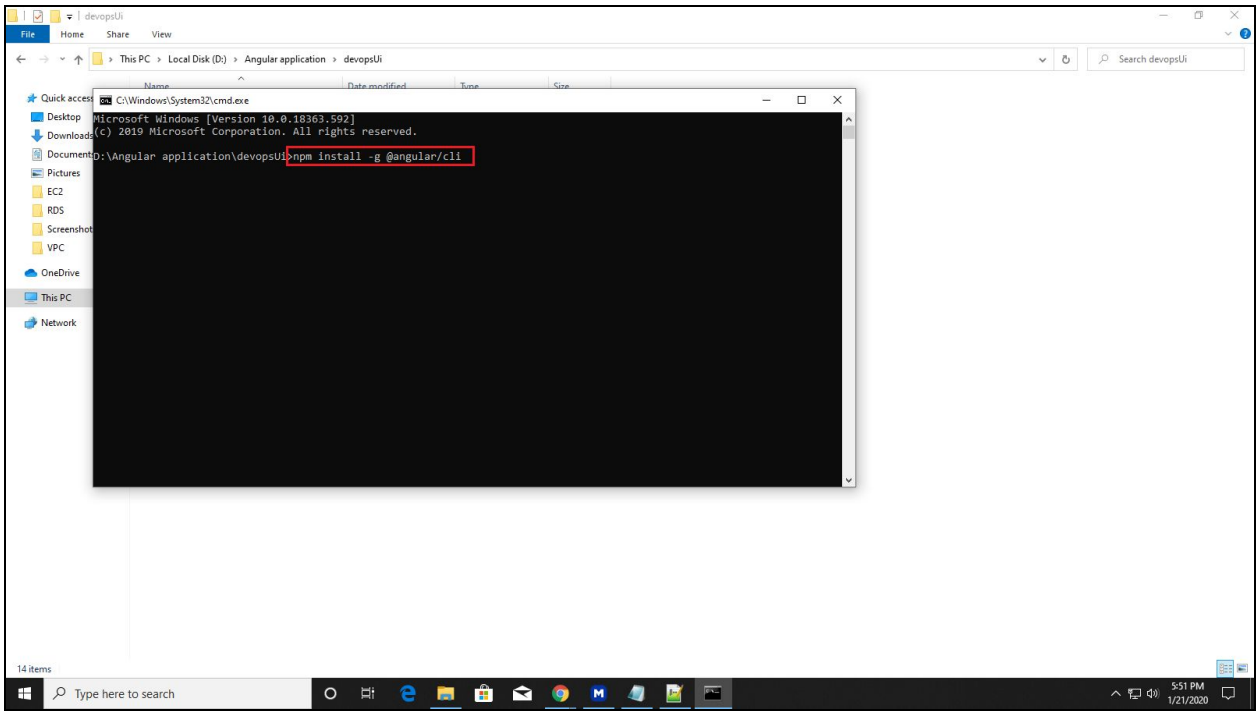


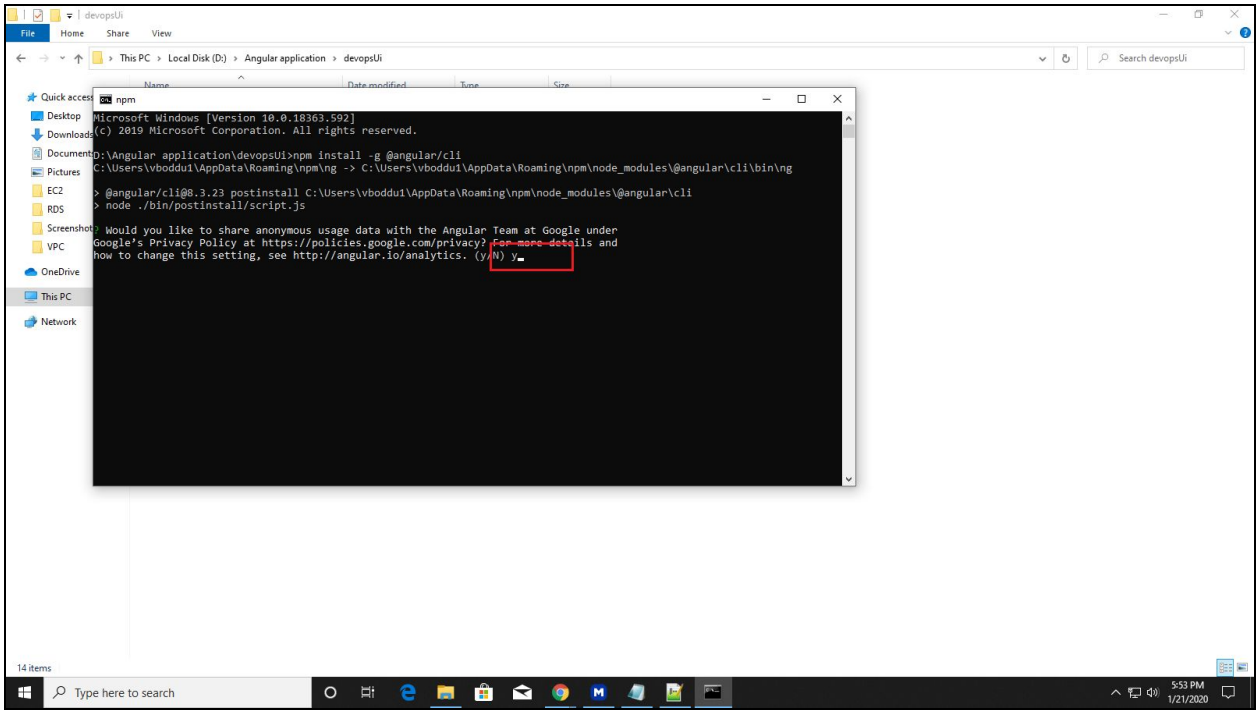
Once you are done with your modifications open cmd



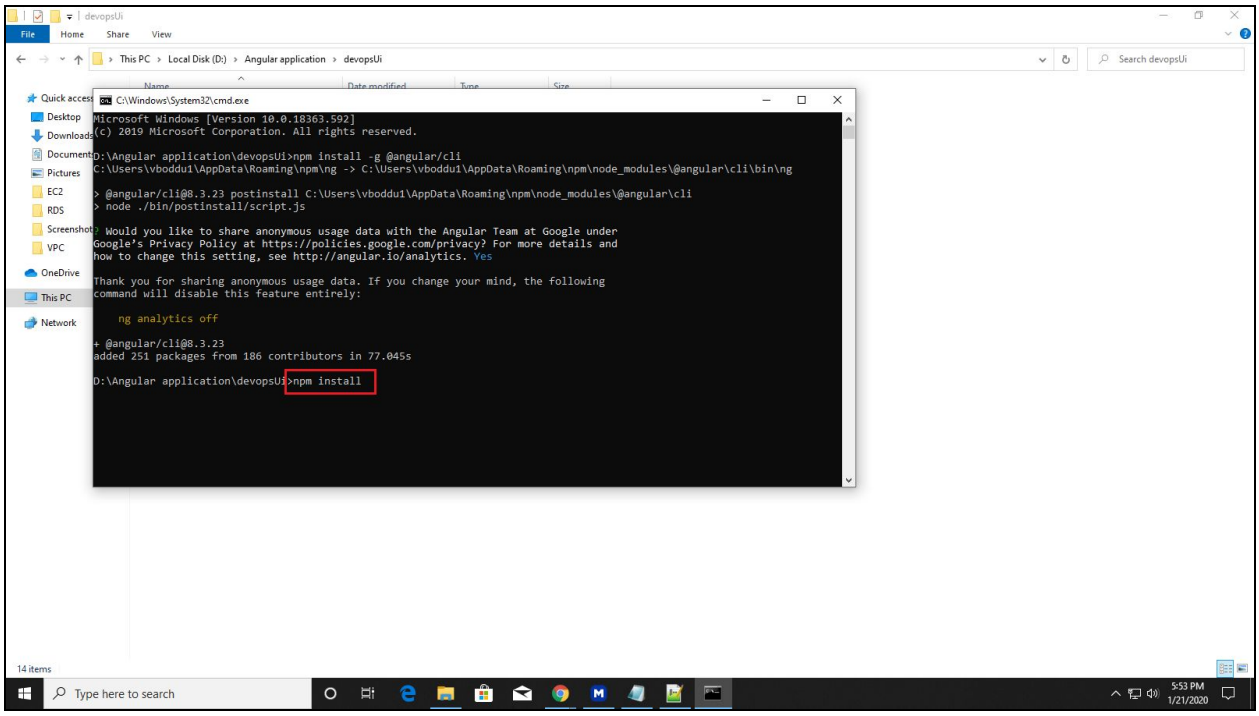


Run the below command to install angular/cli

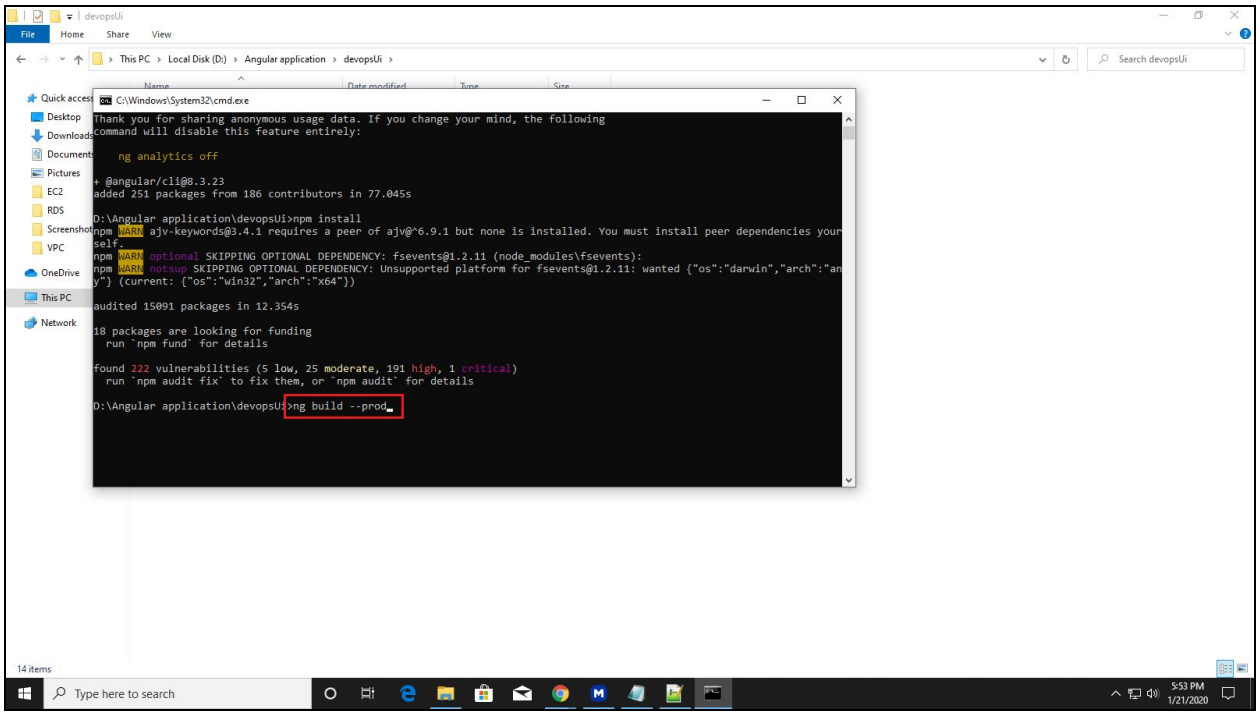


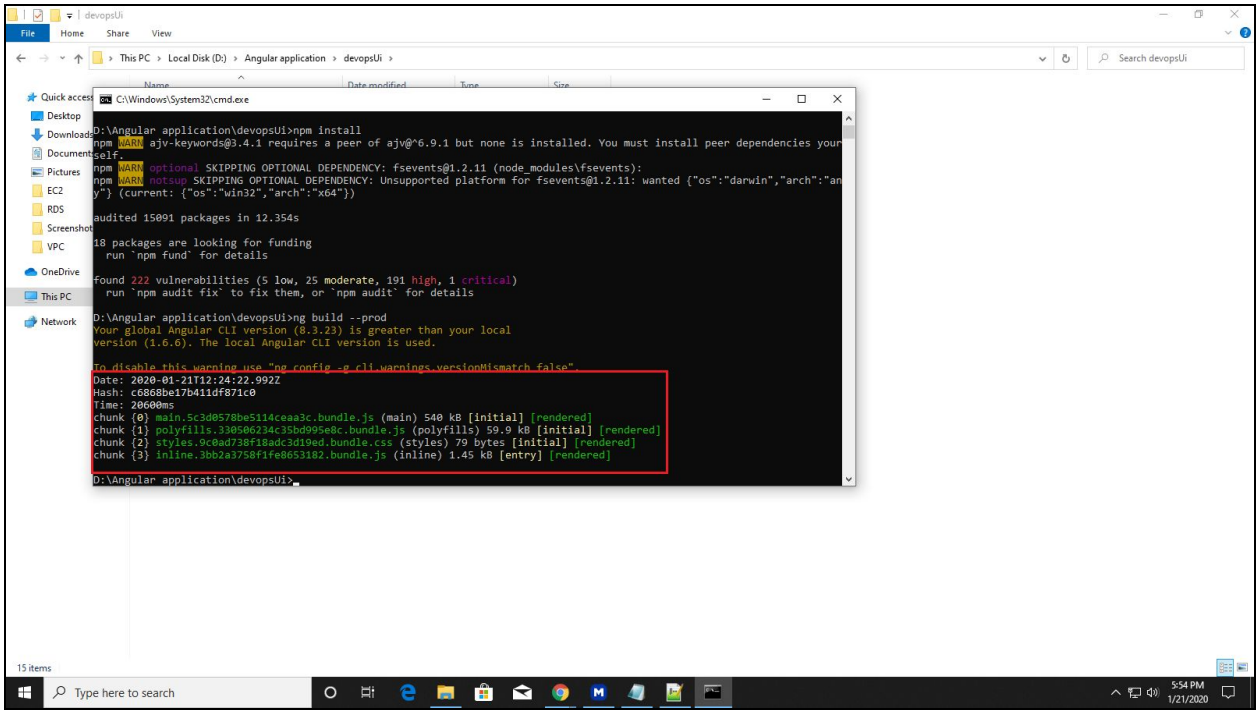


Run the below command to install node modules

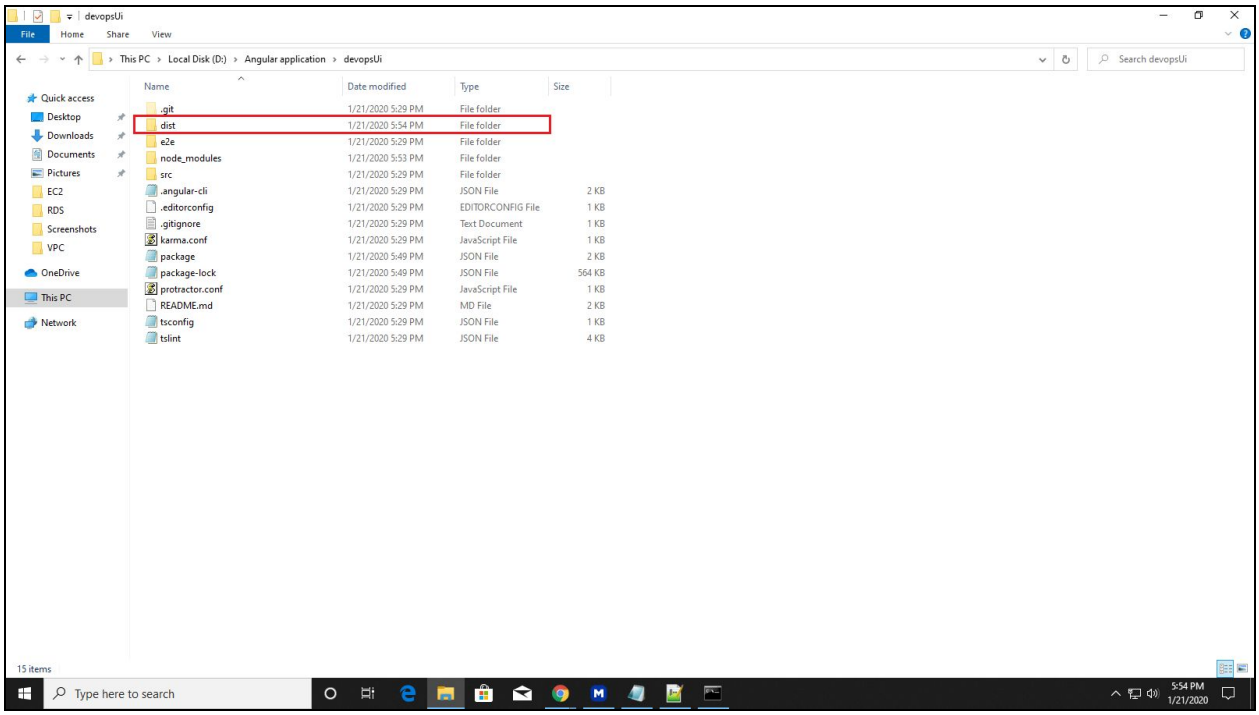


Run the below command to build your application and to create dist folder

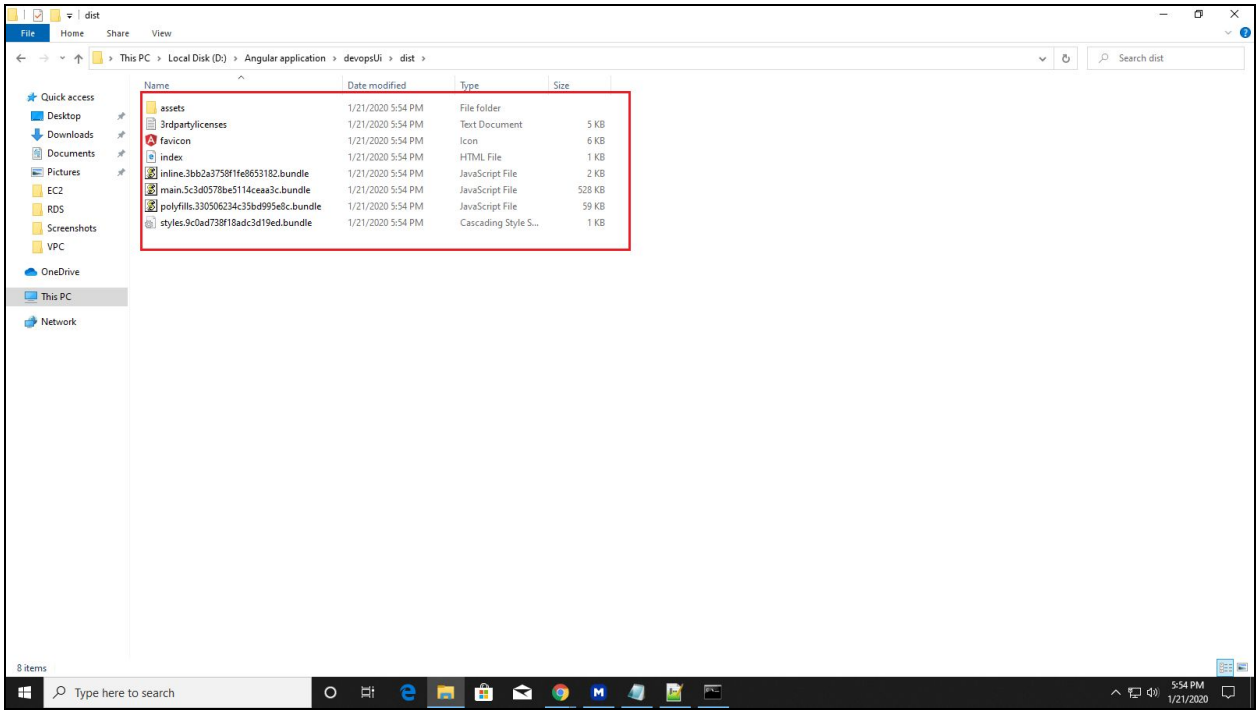




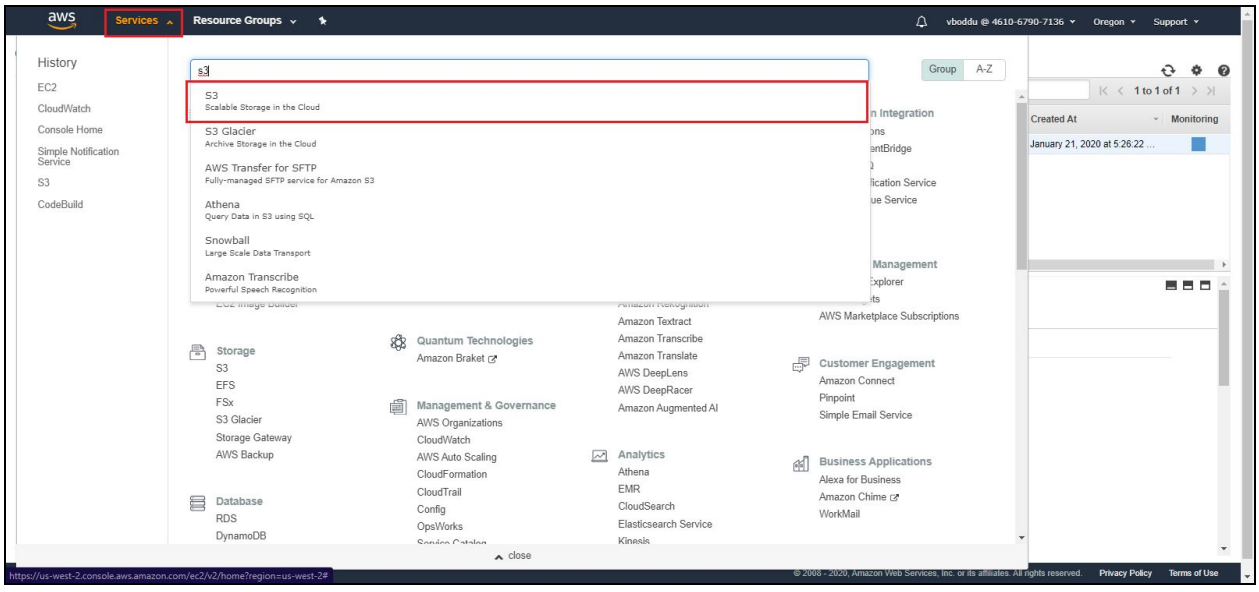
Once your application successfully built the dist folder will be created as shown below



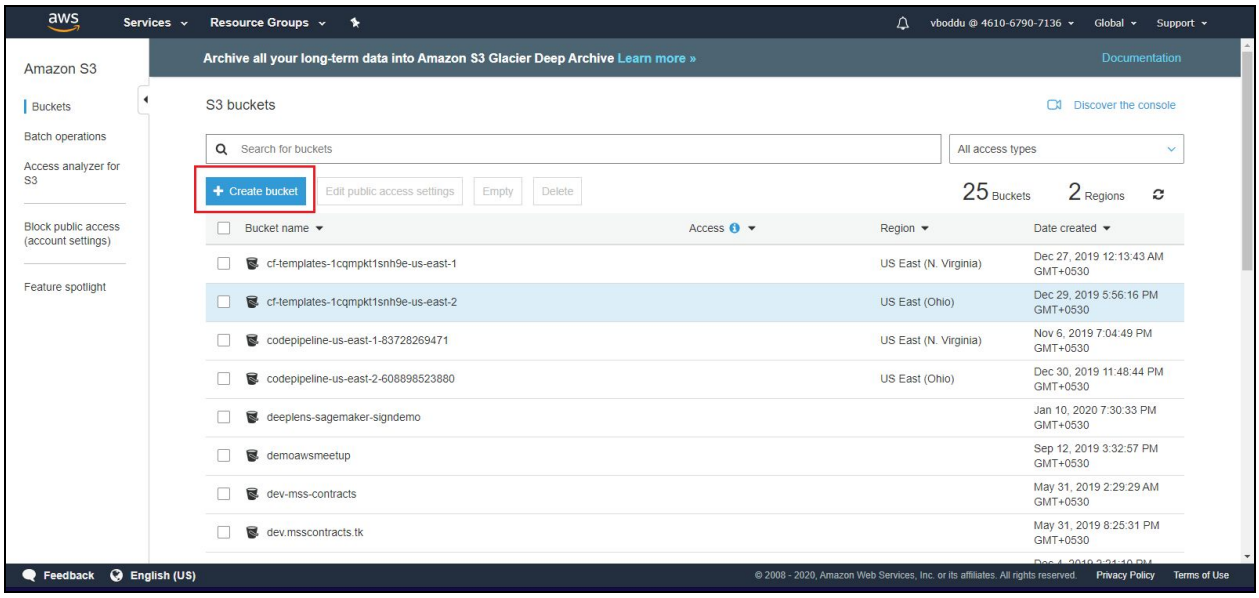
Copy all the files that are present in your dist folder



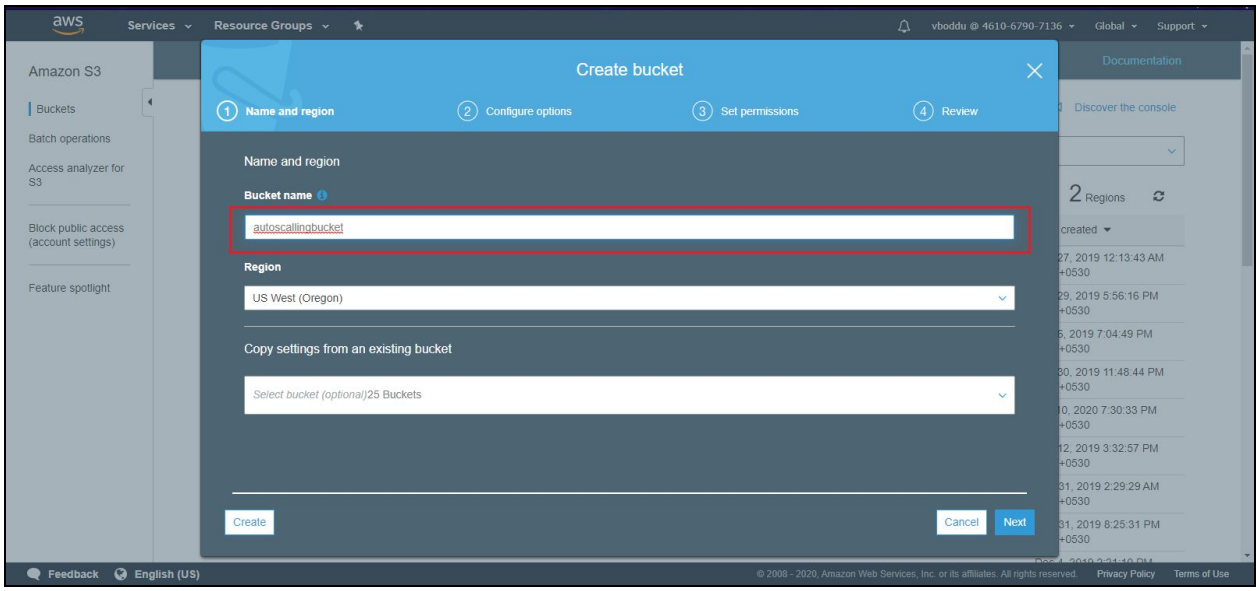
Go to your AWS console and search for S3



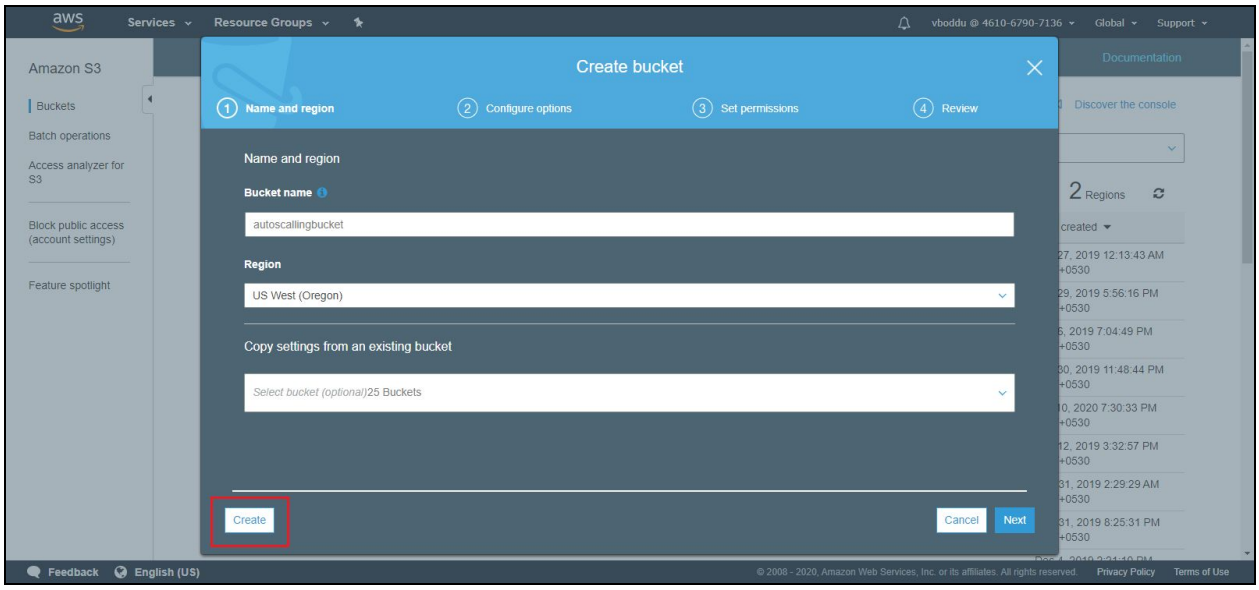
Click on create bucket



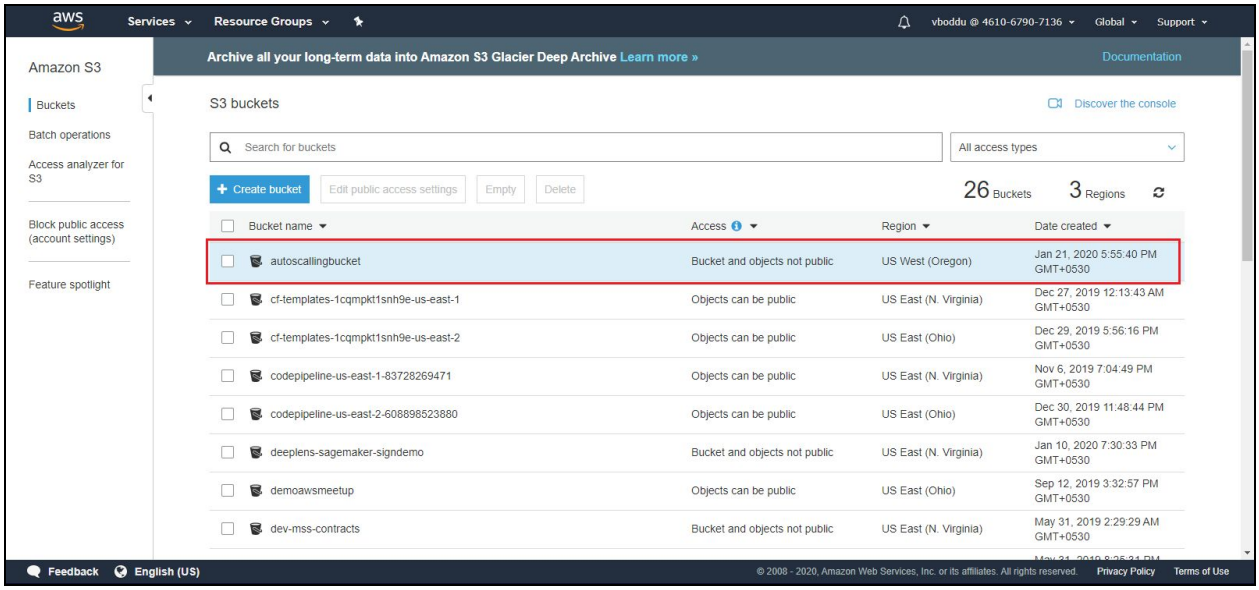
Provide S3 bucket name and region



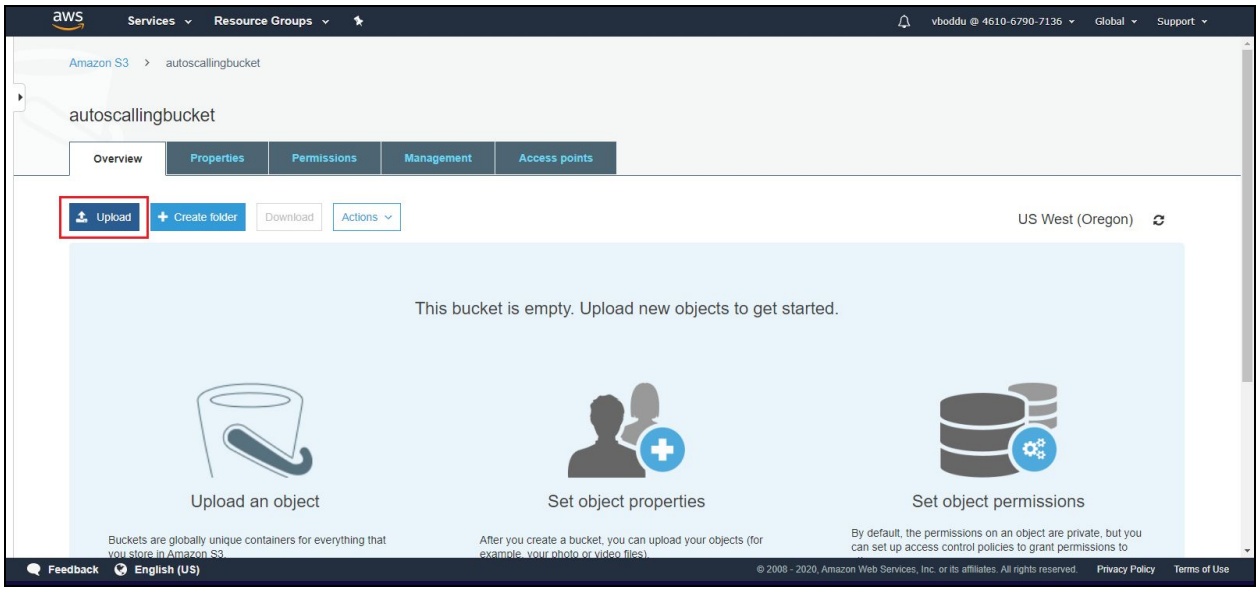
Click on create



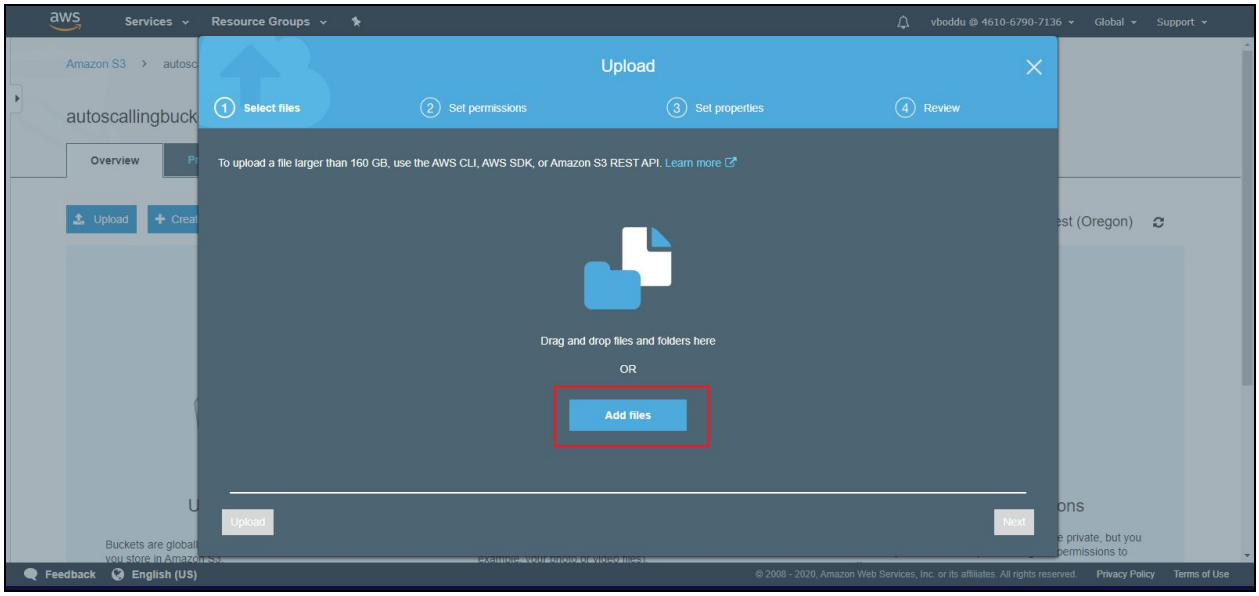
Your bucket will be created as shown below, click on your bucket



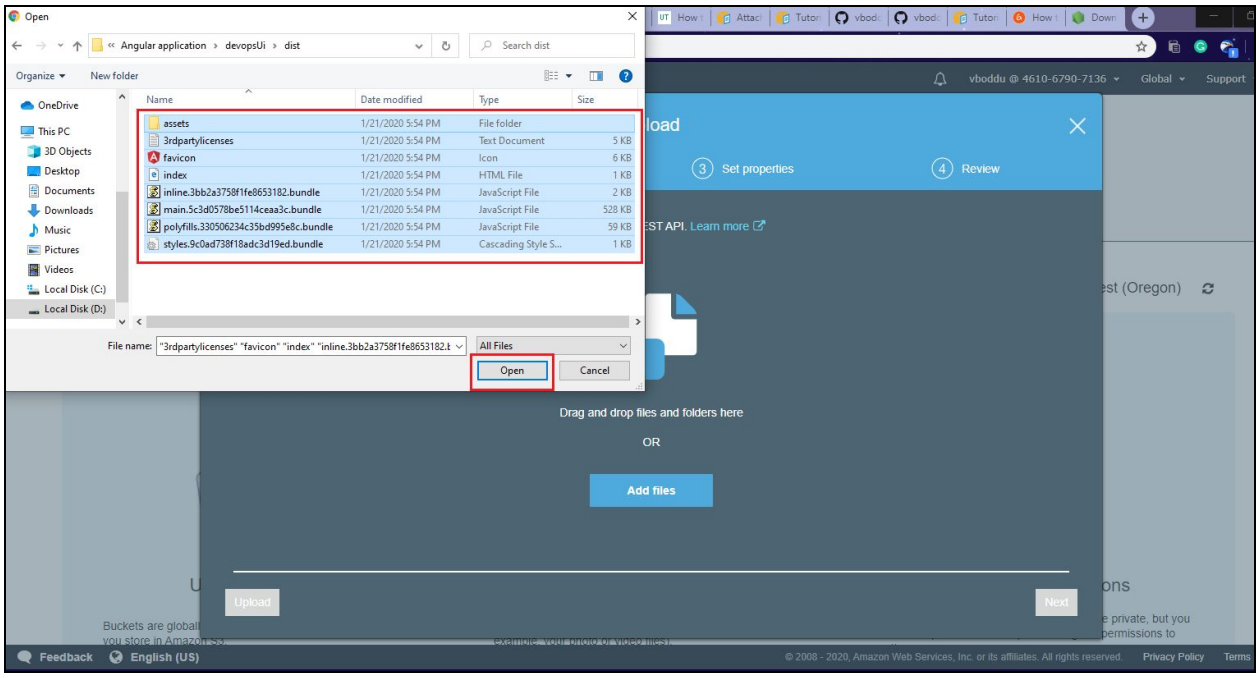
Click on upload



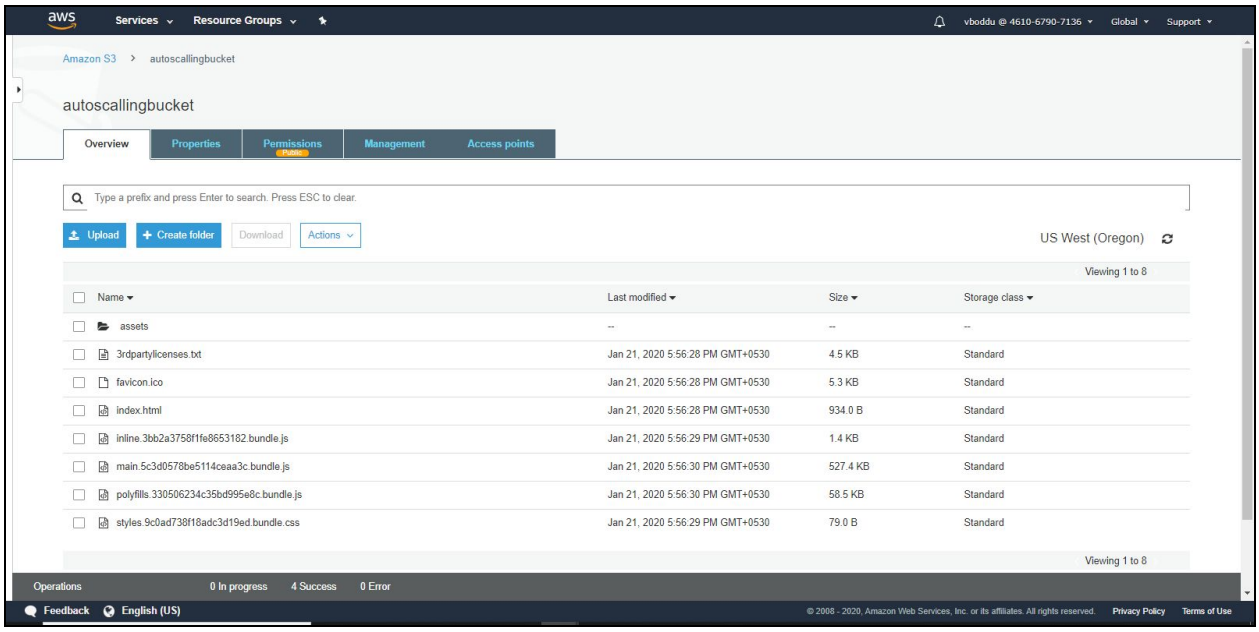
Click on the Add files



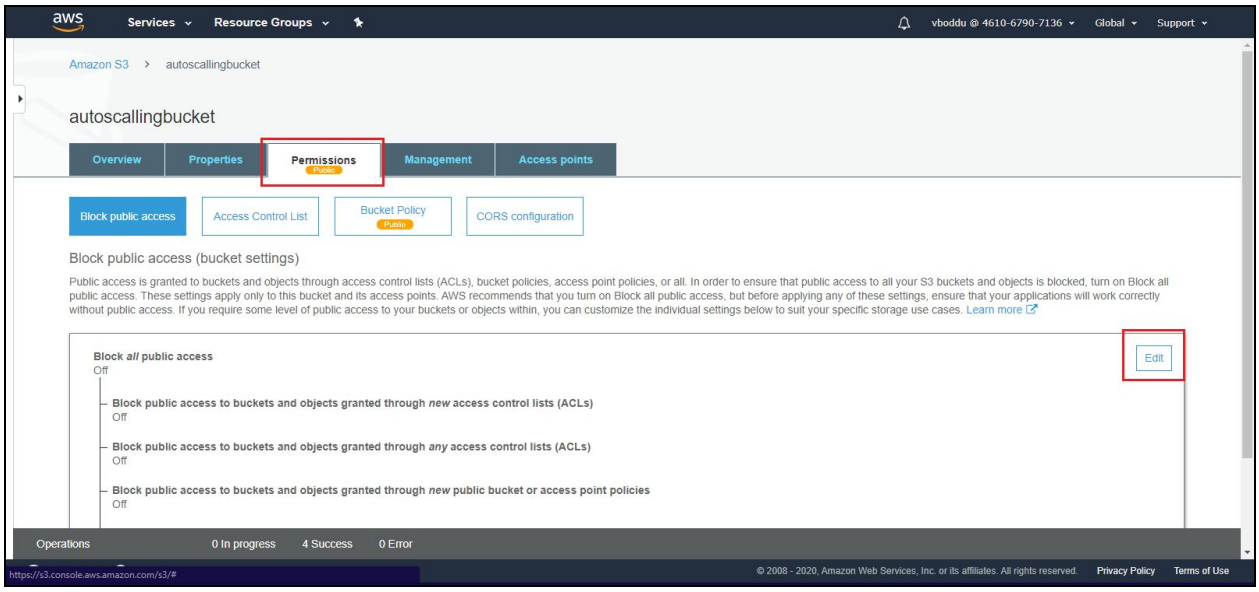
Go to your Angular application dist folder and then drag and drop all the files onto your S3 bucket



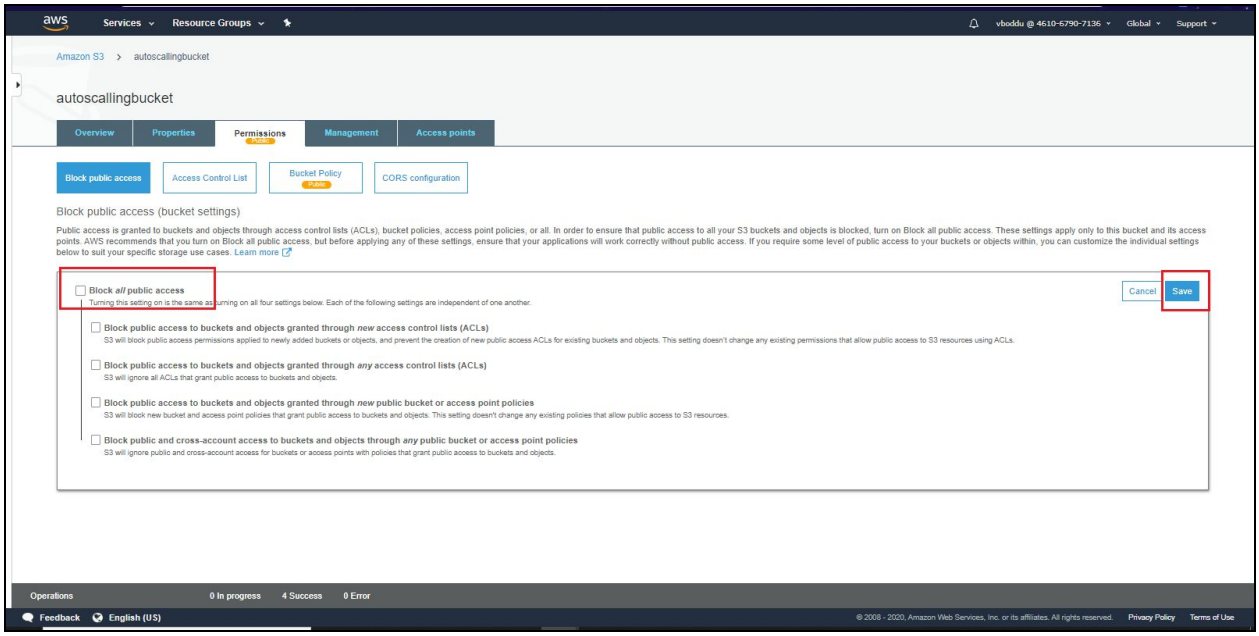
All the files in your dist folder will get uploaded onto your S3 bucket as shown below



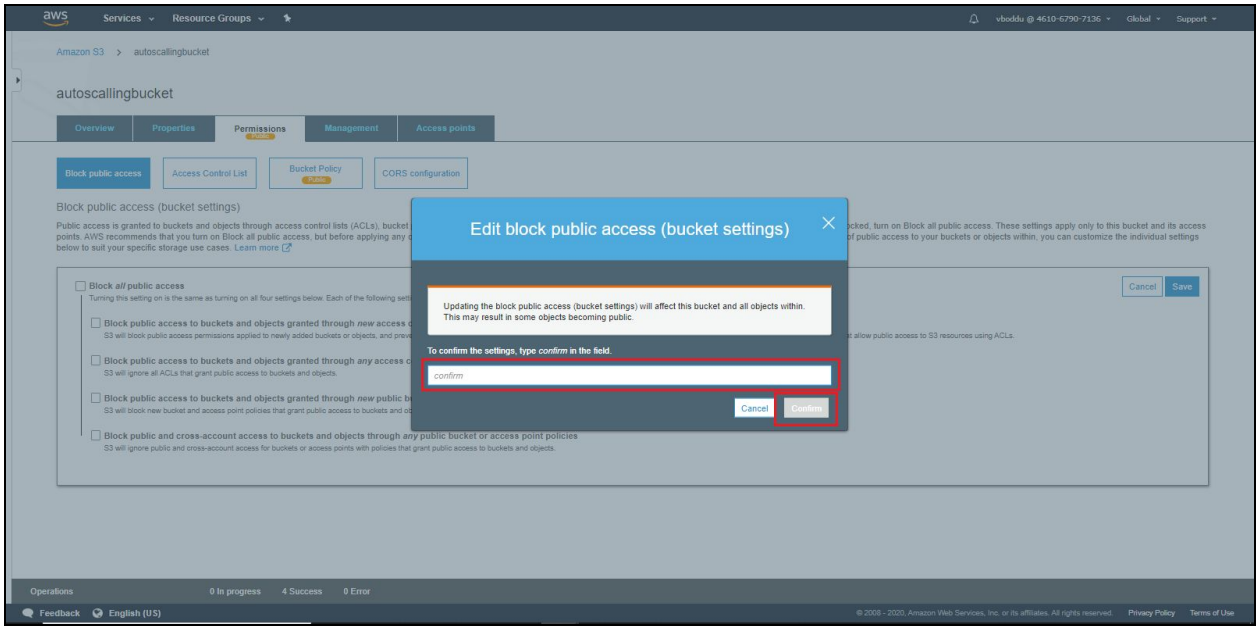
Click on Permissions, select block public access and then click on edit



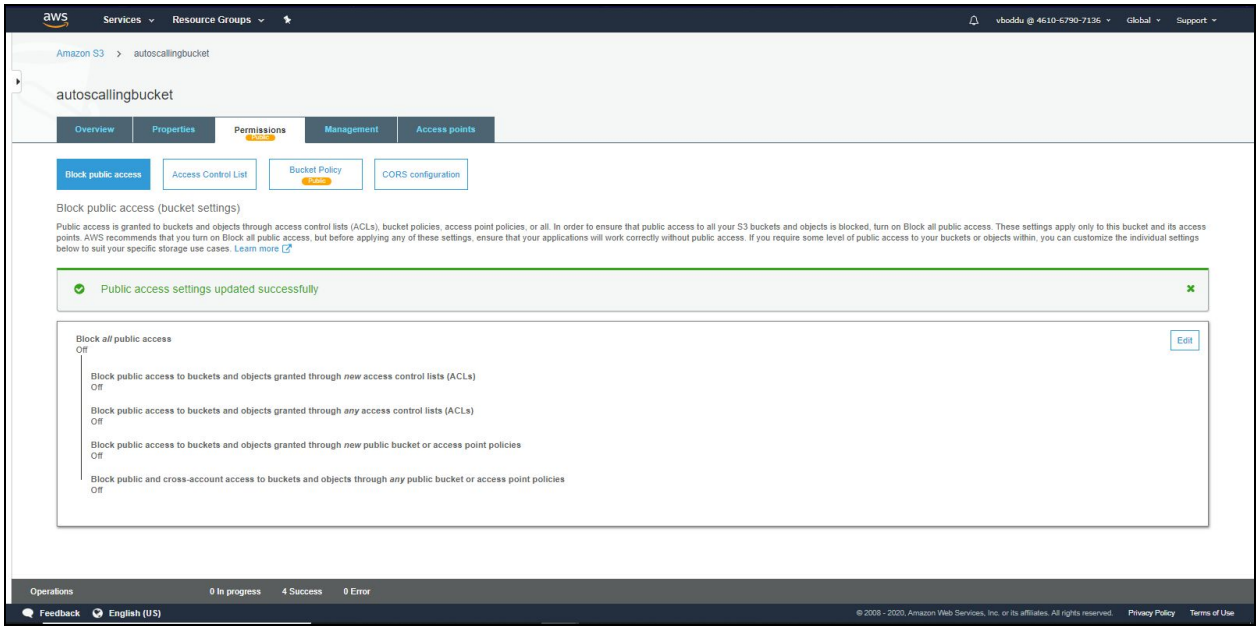
Disable block all public access and click on save



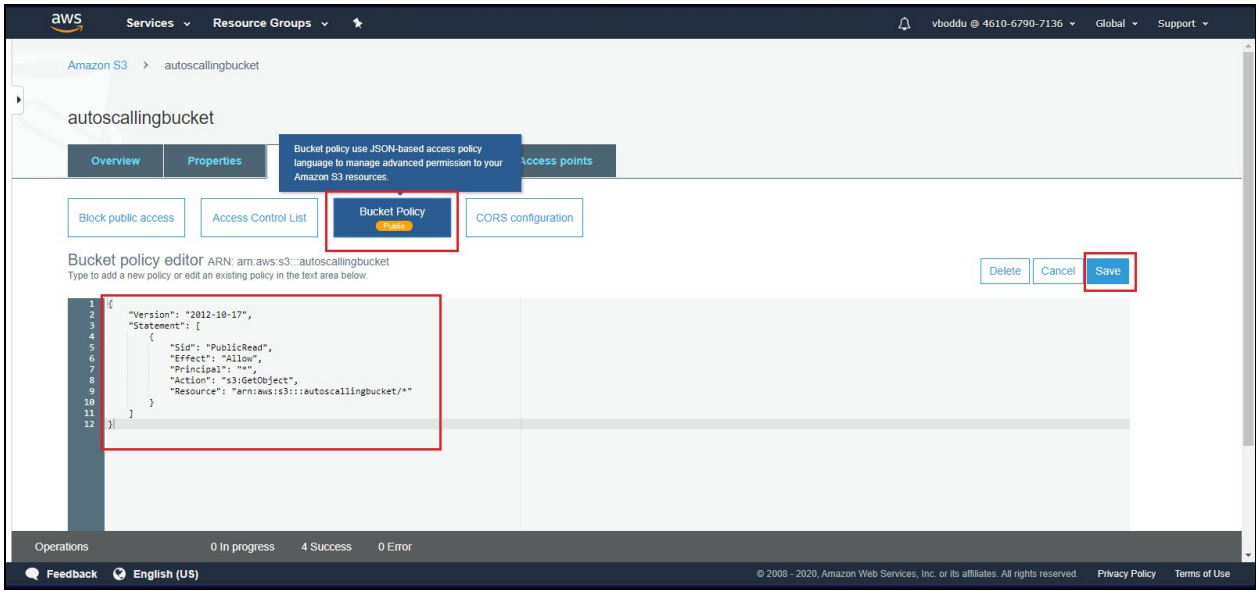
It will ask for confirmation so type confirm and click on confirm



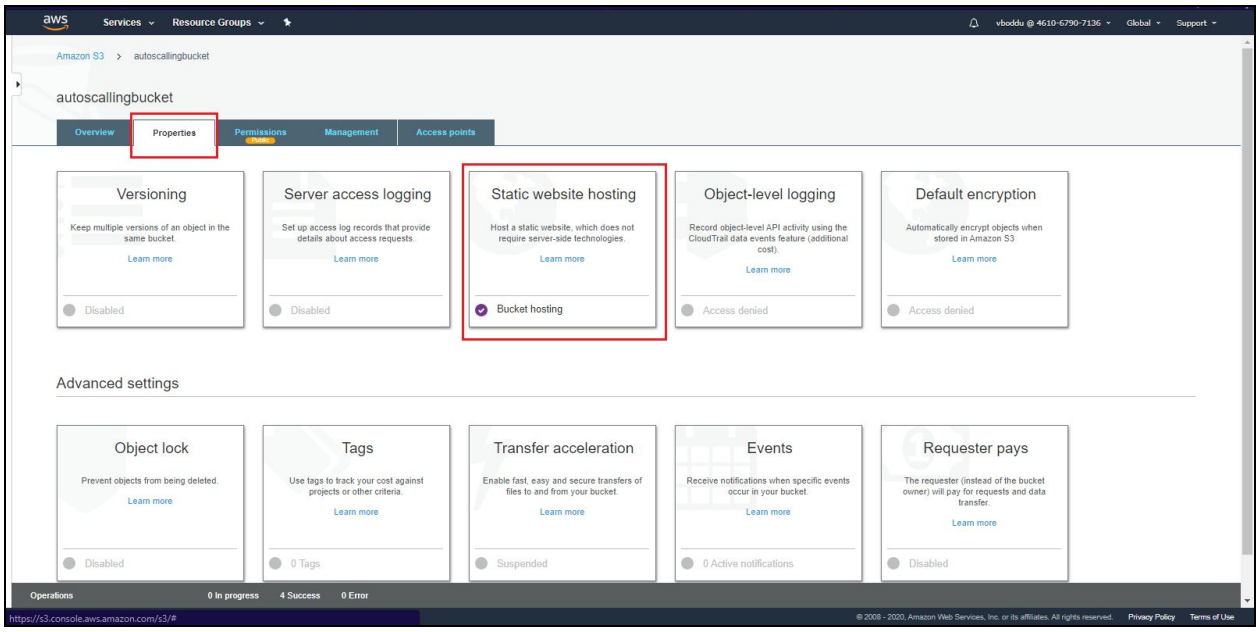
Public access setting were updated successfully



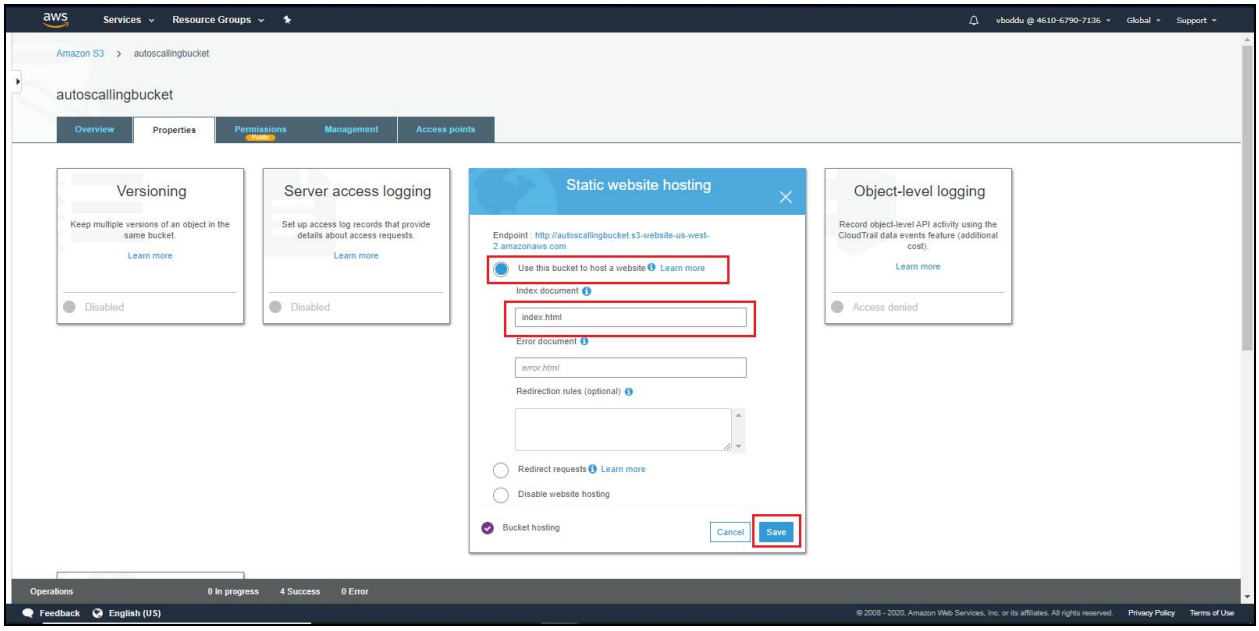
Click on bucket policy and provide the below script and then click on save



Click on properties and select static web hosting



Select use the bucket to host a website and provide index.html and then click on save



Now click on the end point to access the application