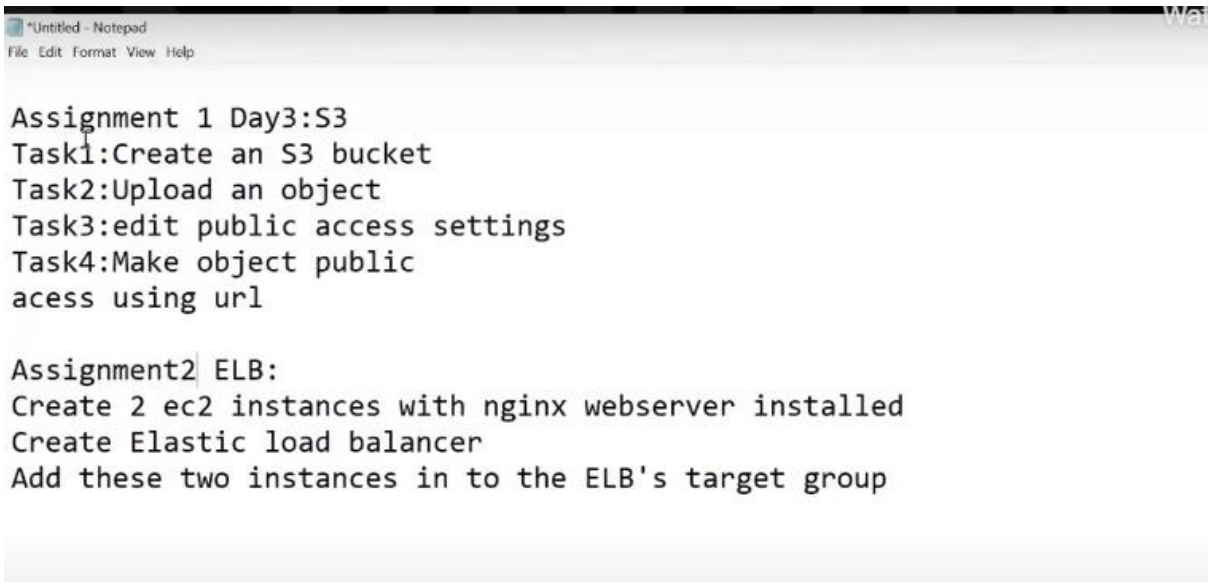
Assignments

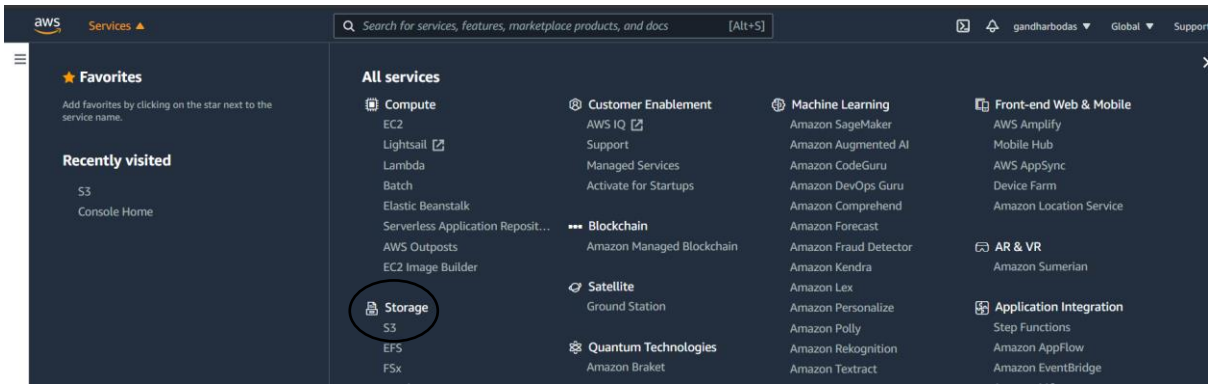
Day 3: Assignments: Part a) & b) S3:



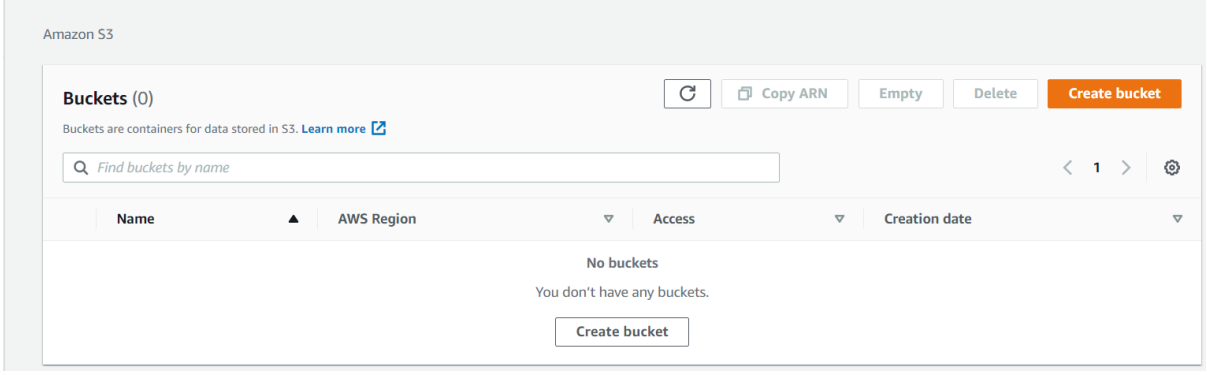
Answer:

Part a):

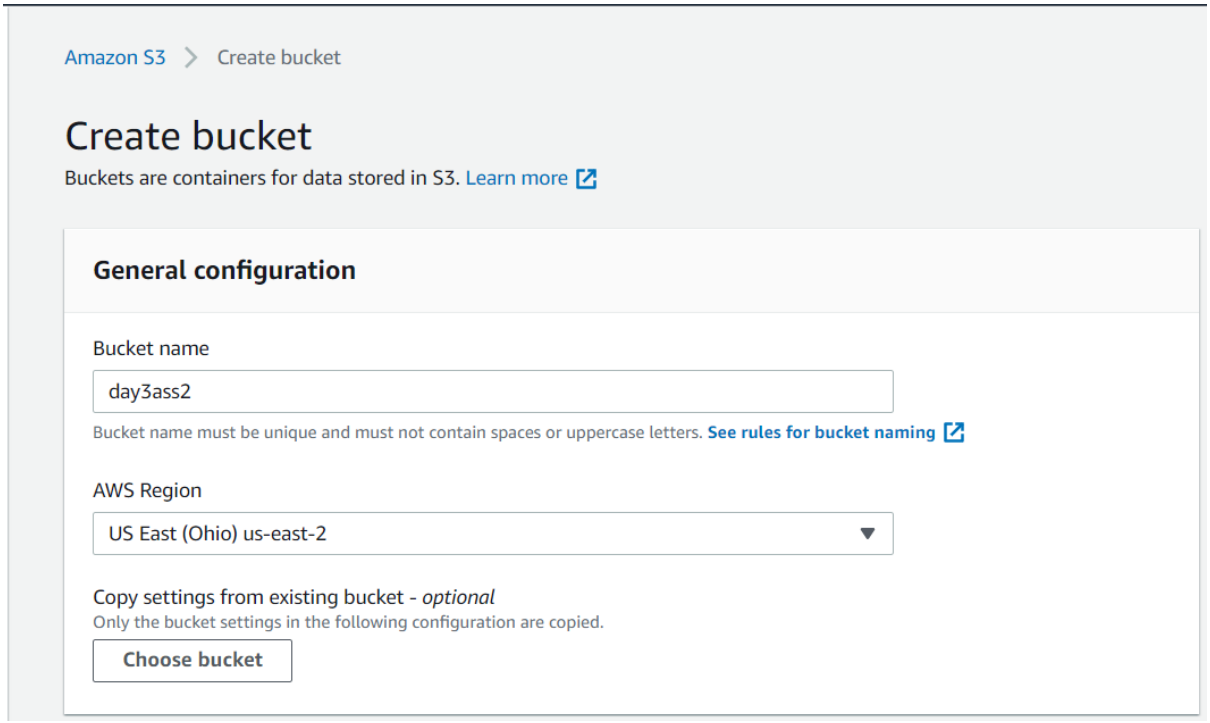
Step 1: Go to the Service-→Storage-→S3

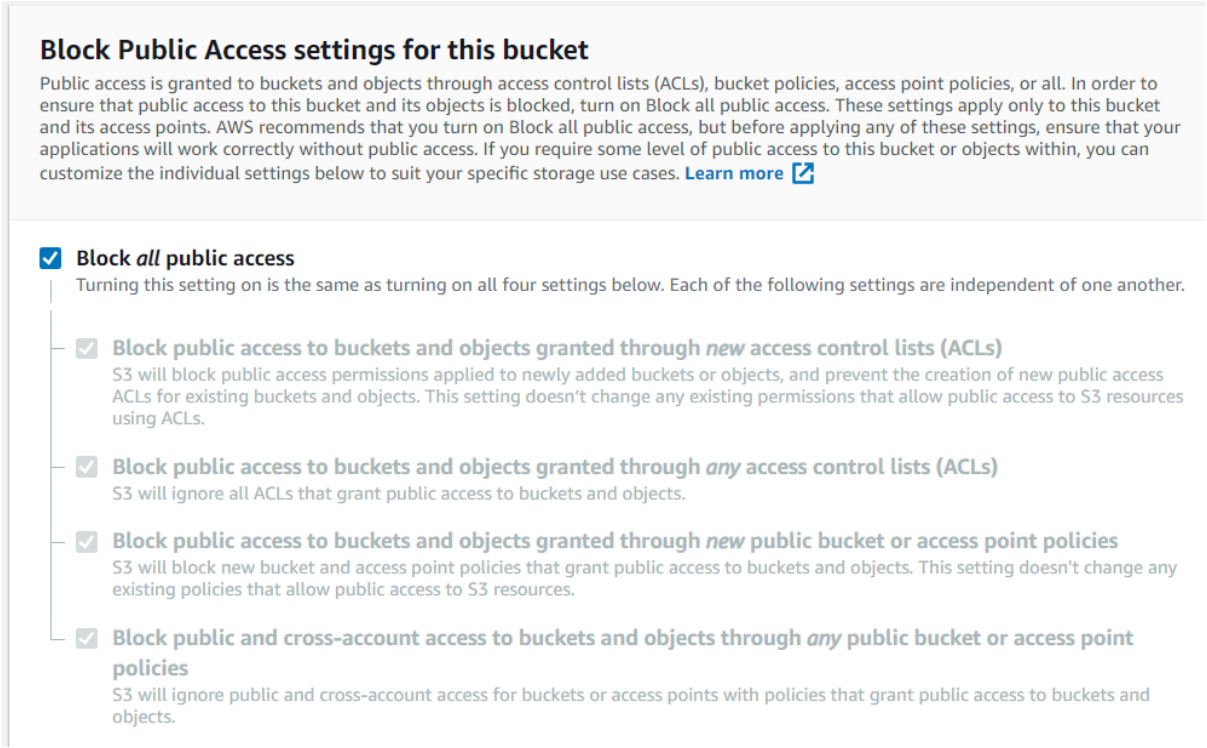


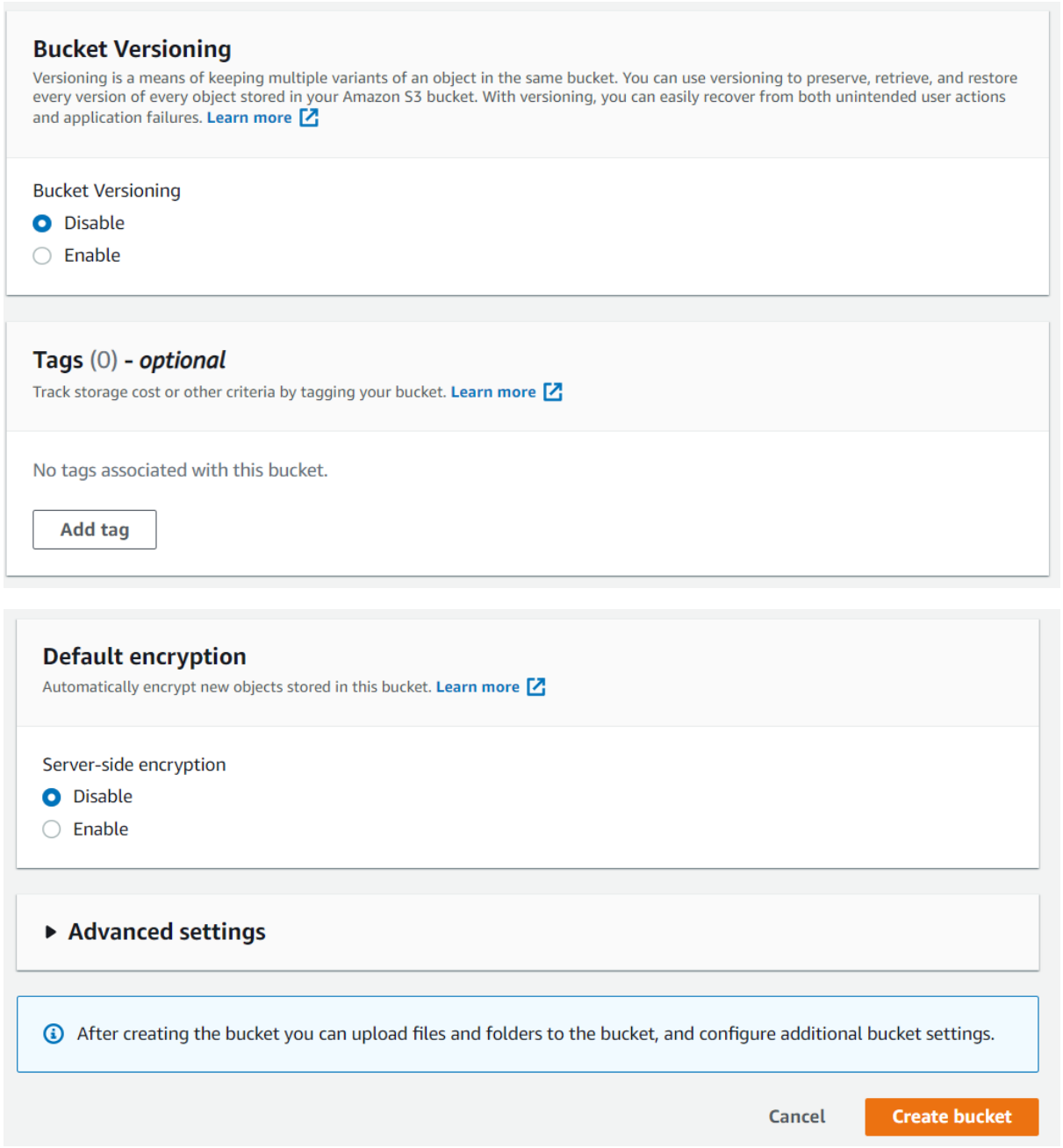
Step 2: *Create an S3 bucket*



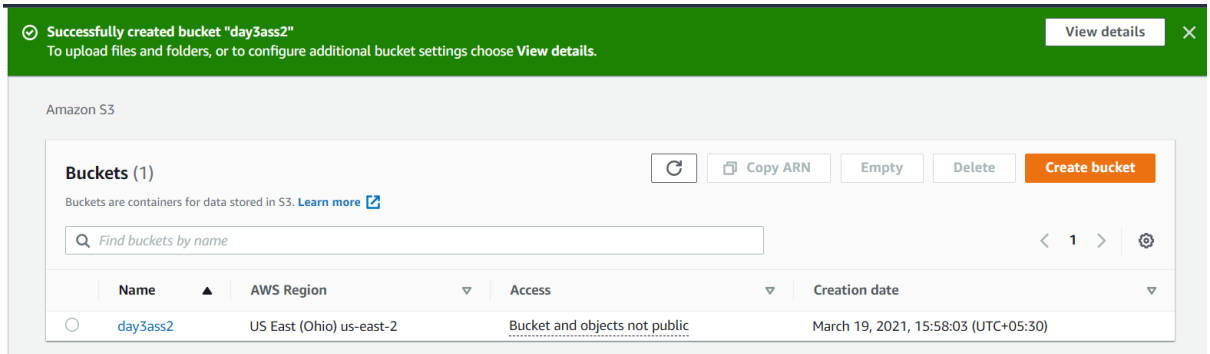
Click on “Create Bucket” button and specify the “Bucket name: day3ass2”, keep “AWS Region as it is”, “Block Public Access settings for this bucket : keep as it is “, “Bucket Versioning: Disable” and “Default encryption: Disable”.





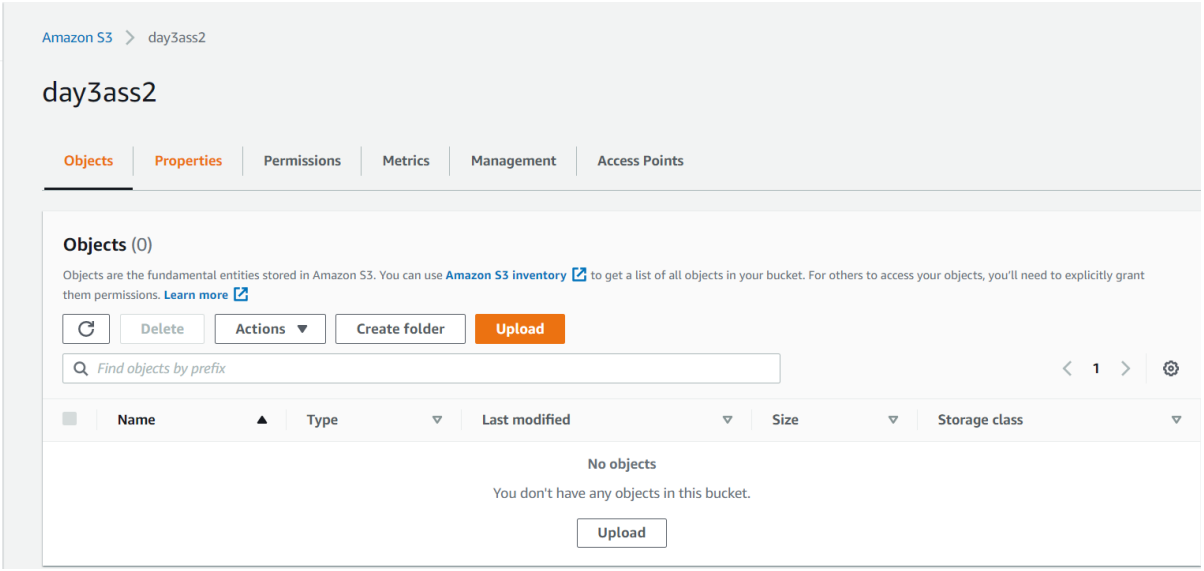


Now click on “Create bucket” button.

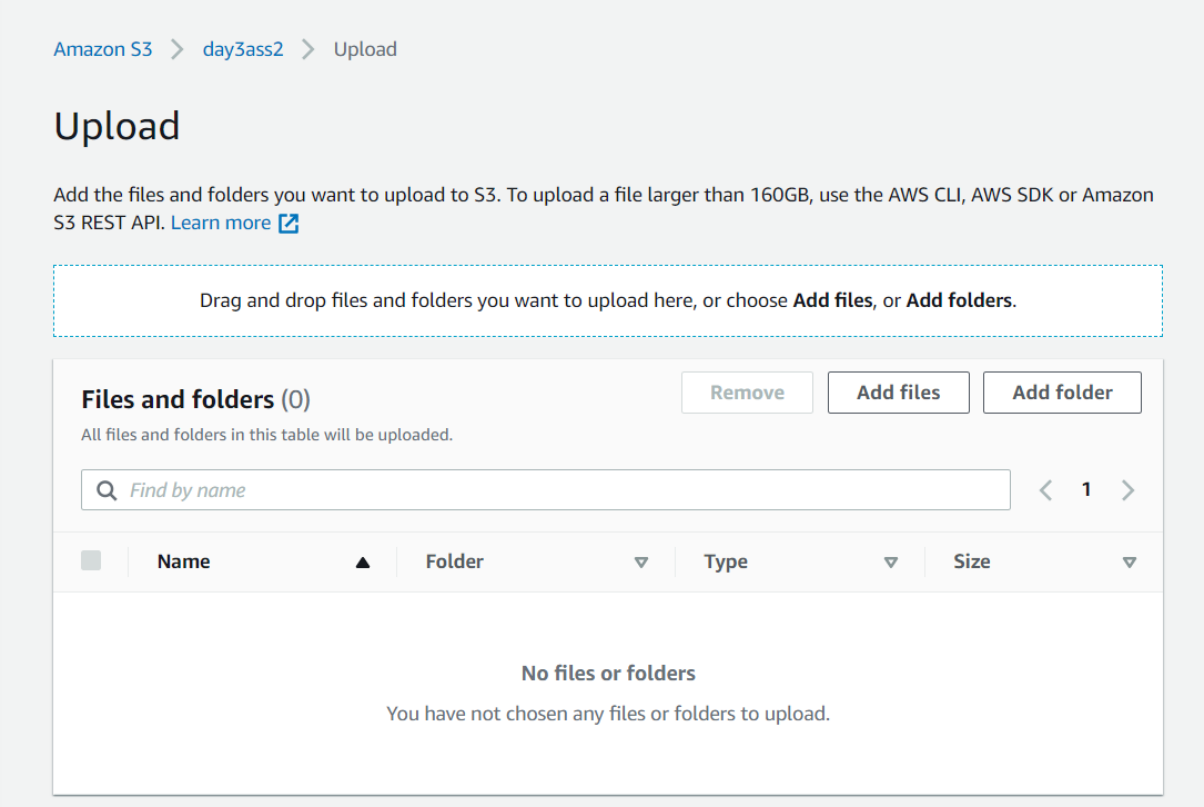


Step 3: *Upload an object*

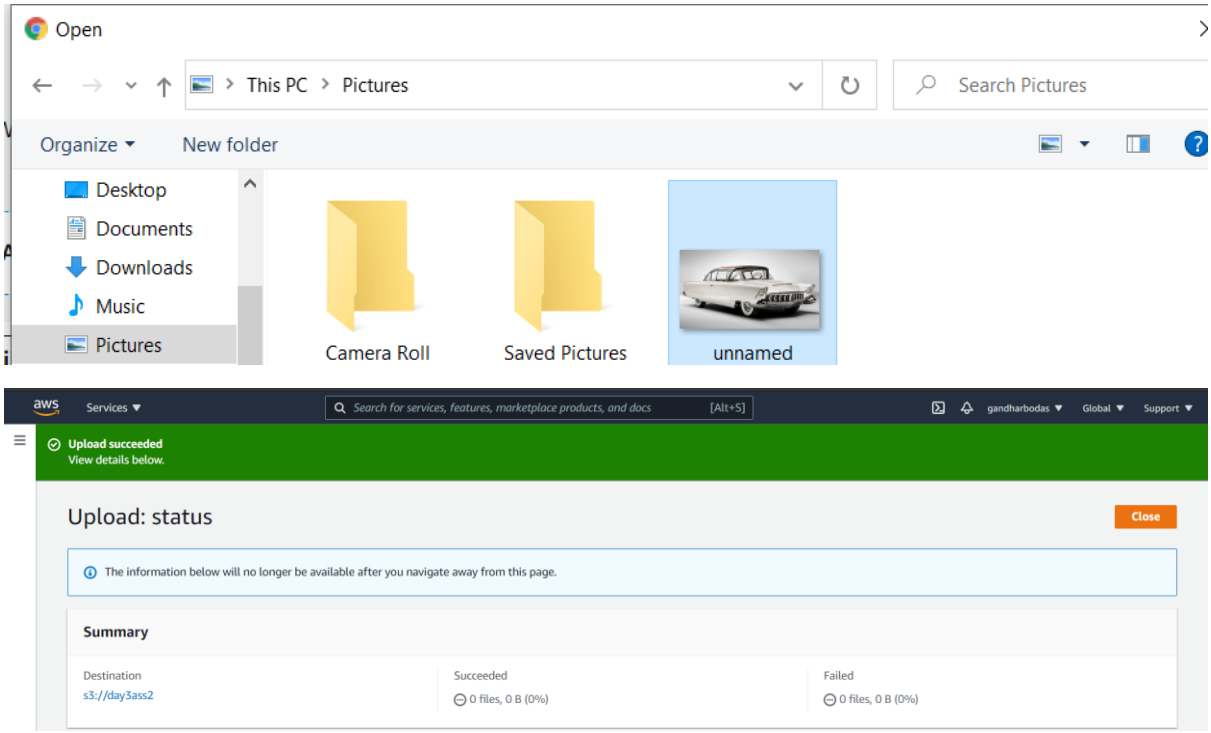
Click on bucket name: day3ass2



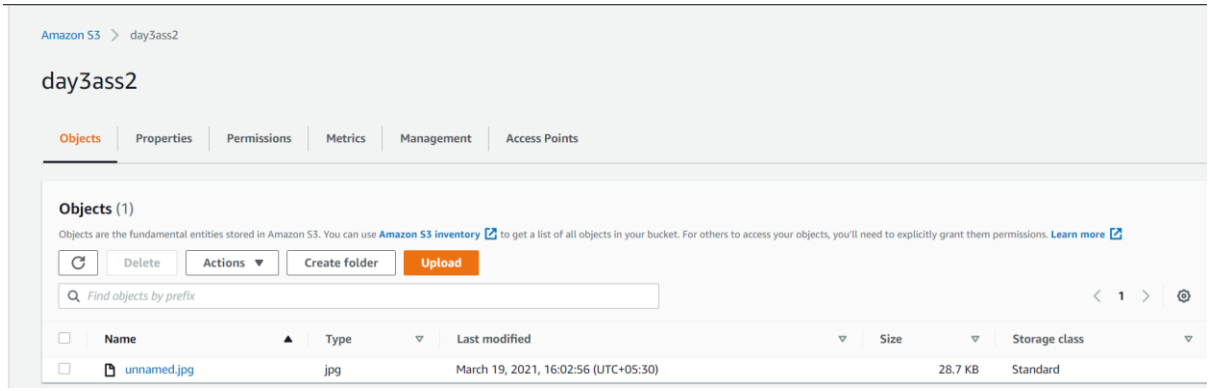
Here, uploads an object so for that click on “upload button” and then



Click on “Add files” and then again click on “Upload button”



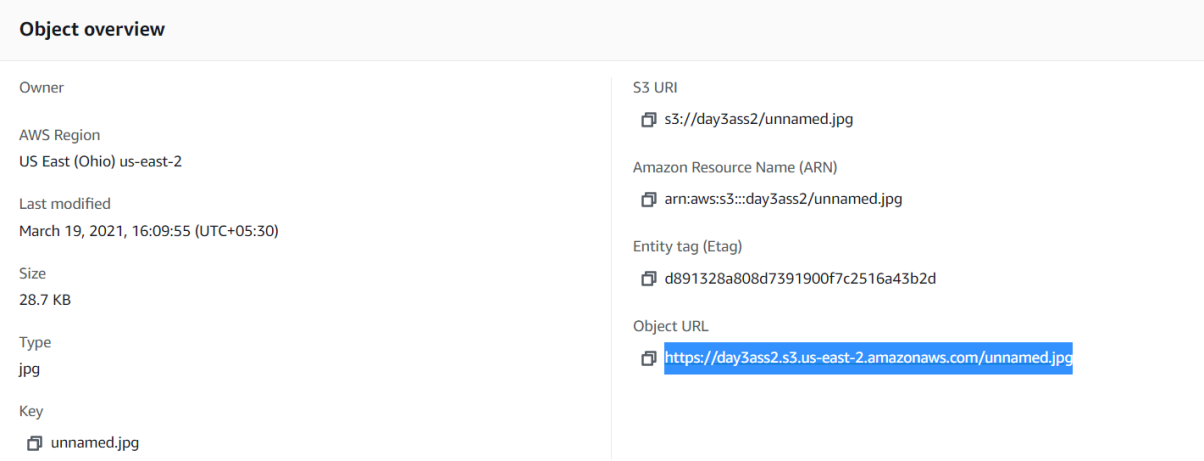
Uploaded successfully now click on Close button.



Step 4: *Edit public access settings*

Select object and click on it and you will be able to see the below page.

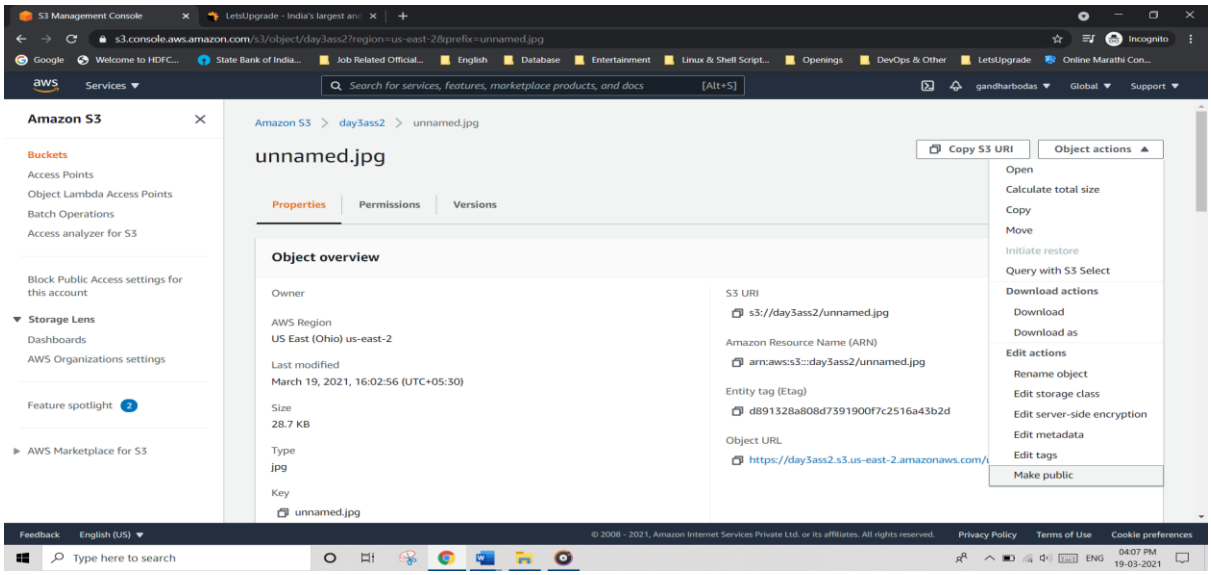
Now copy Object URL: https://day3ass2.s3.us-east-2.amazonaws.com/unnamed.jpg and paste it on a new tab you will be able to see the XML file page where **Access Denied.**

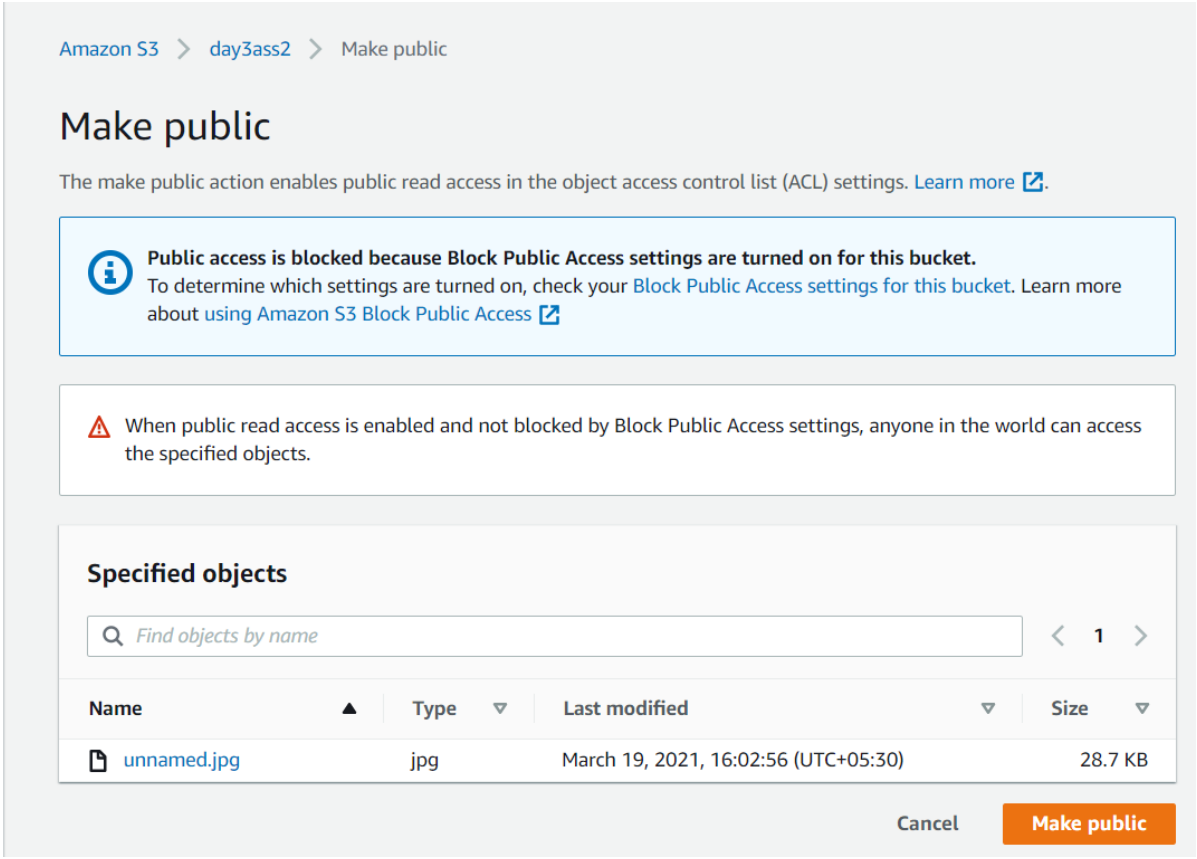


This XML file does not appear to have any style information associated with it. The document tree is shown below.

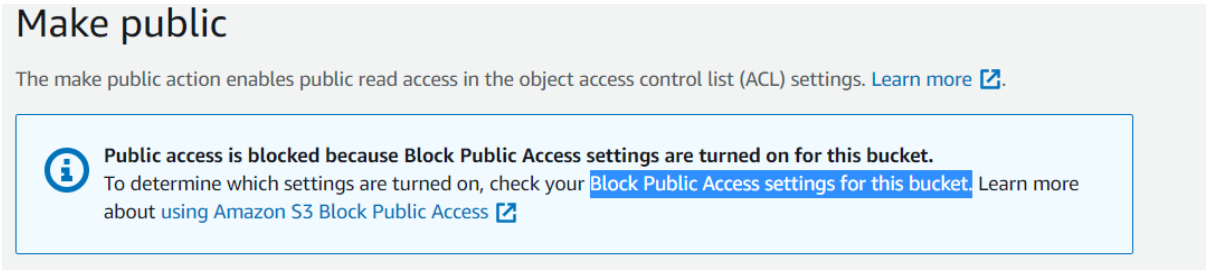
```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>BJ3RJ32H01GEV2K3</RequestId>
   <HostId>kGXUbGa5AQoljM02xWyxzYF93+6oE25tQLSGbbQiGeJZJbD29DPY3t48Az8xNsJVzD+c6YVuEdE=</HostId>
 </Error>
```

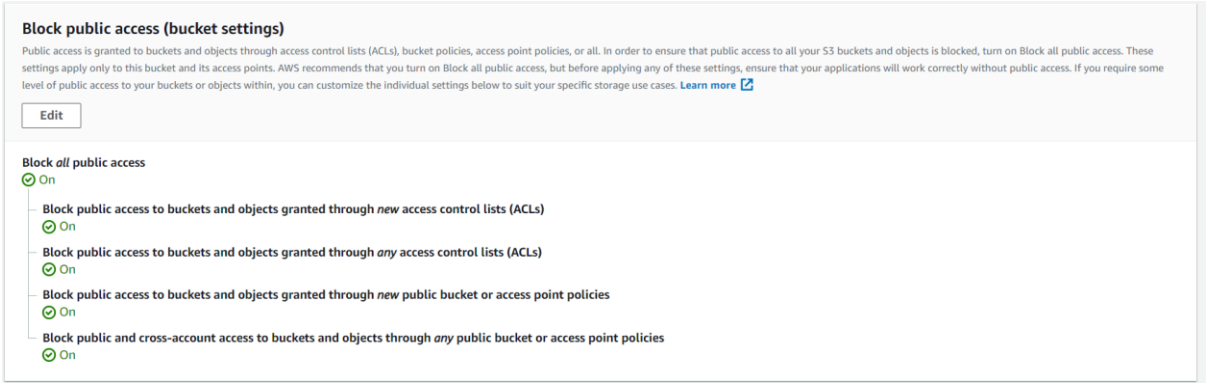
On this page to towards right hand side--→Object Section-→Click on Make public option.



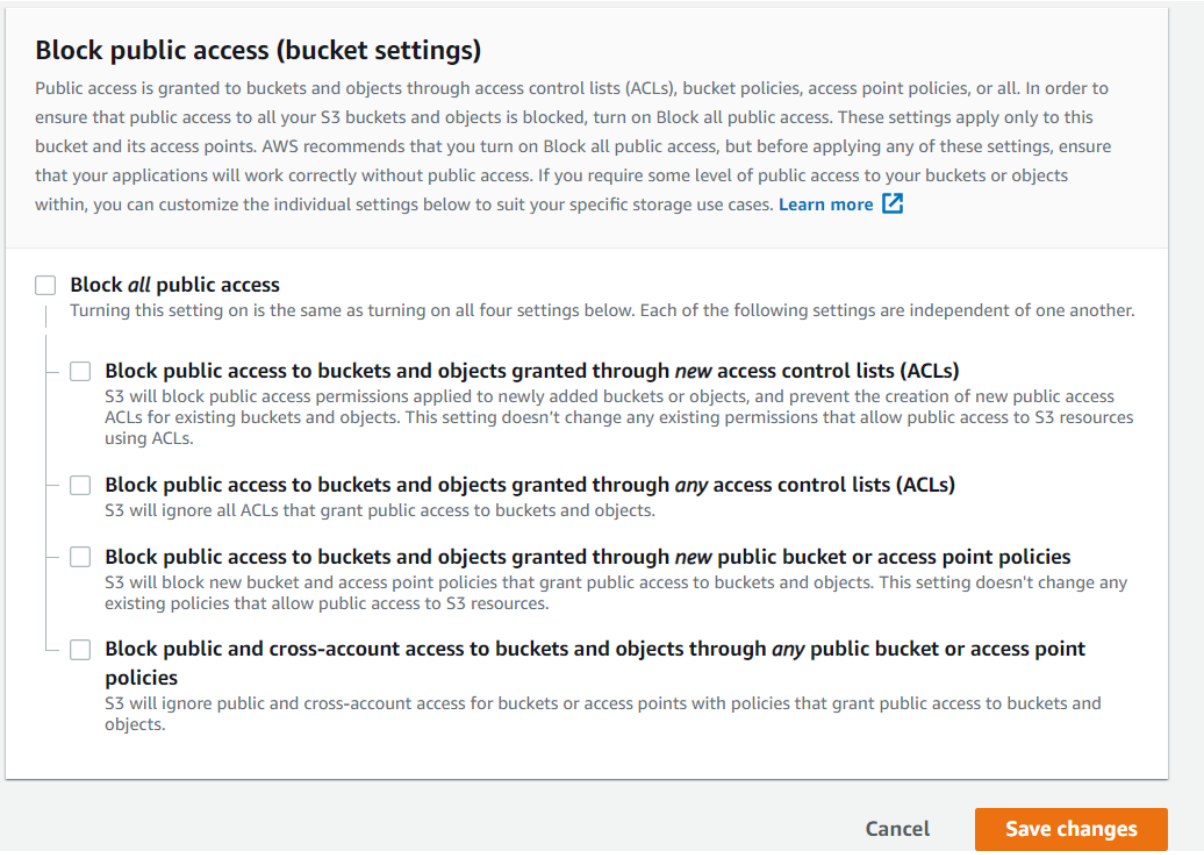


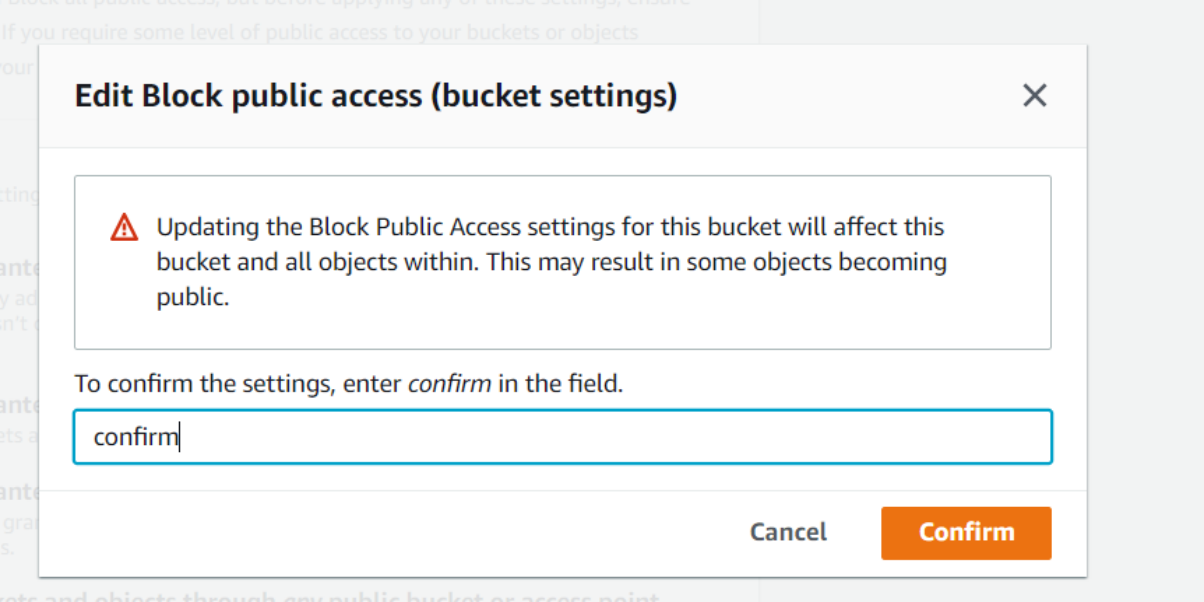
Now click on this " Block Public Access settings for this bucket." You will get below page



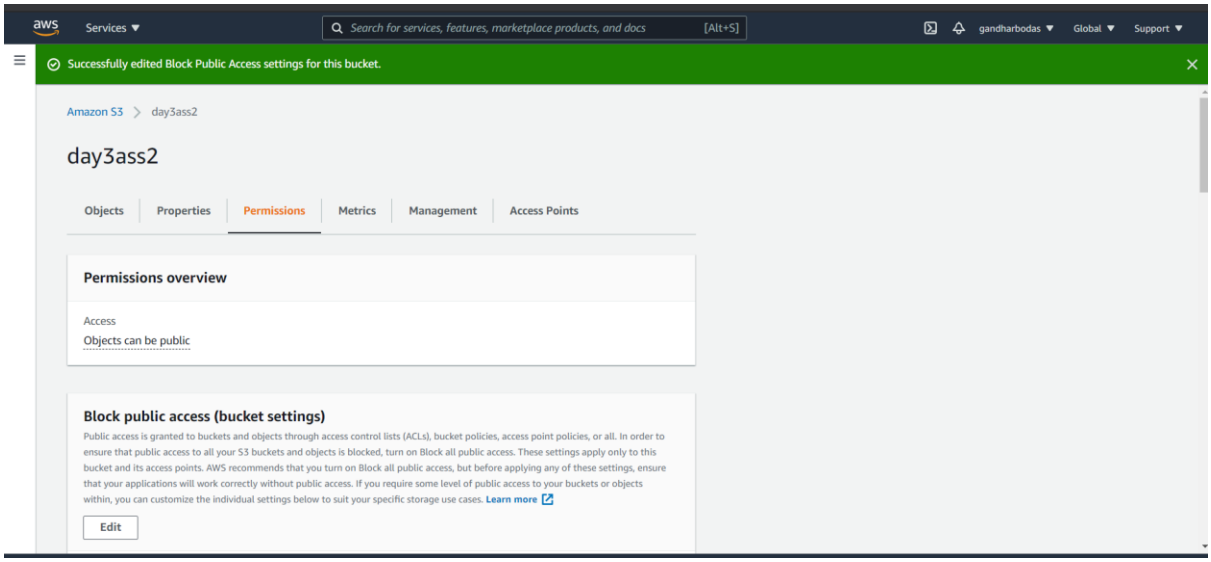


Now click on “Edit button” and uncheck “Block all public access” and then save changes and confirmed.

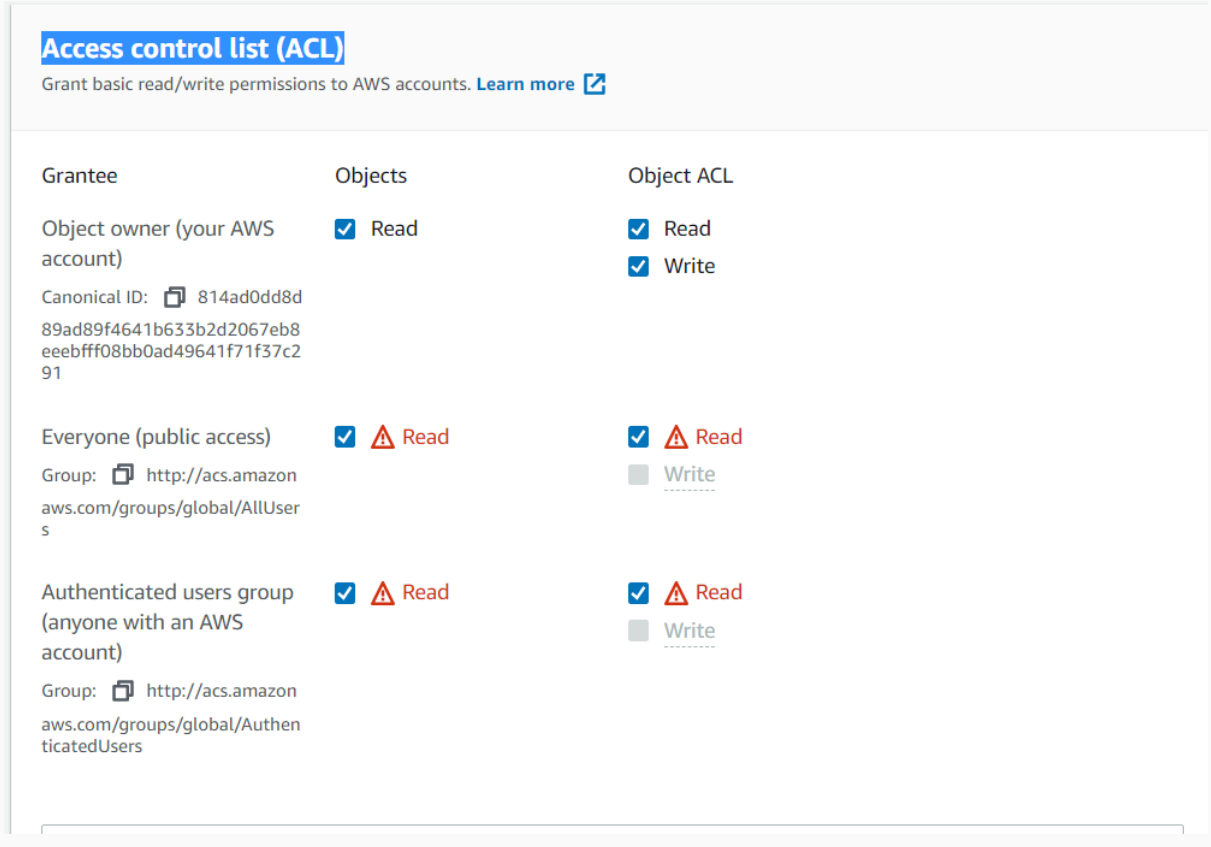




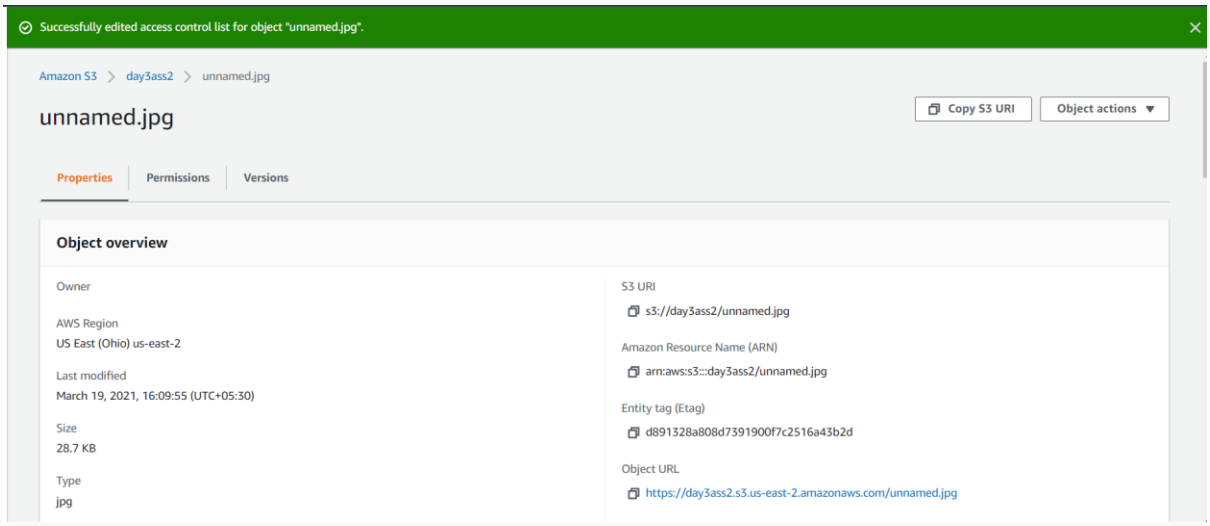
Successfully edited Block Public Access settings.



Step 5: *Now click on “Permission” and later on click on “Edit button” and under “Access control list (ACL)” give read and write permissions to an object as such*:

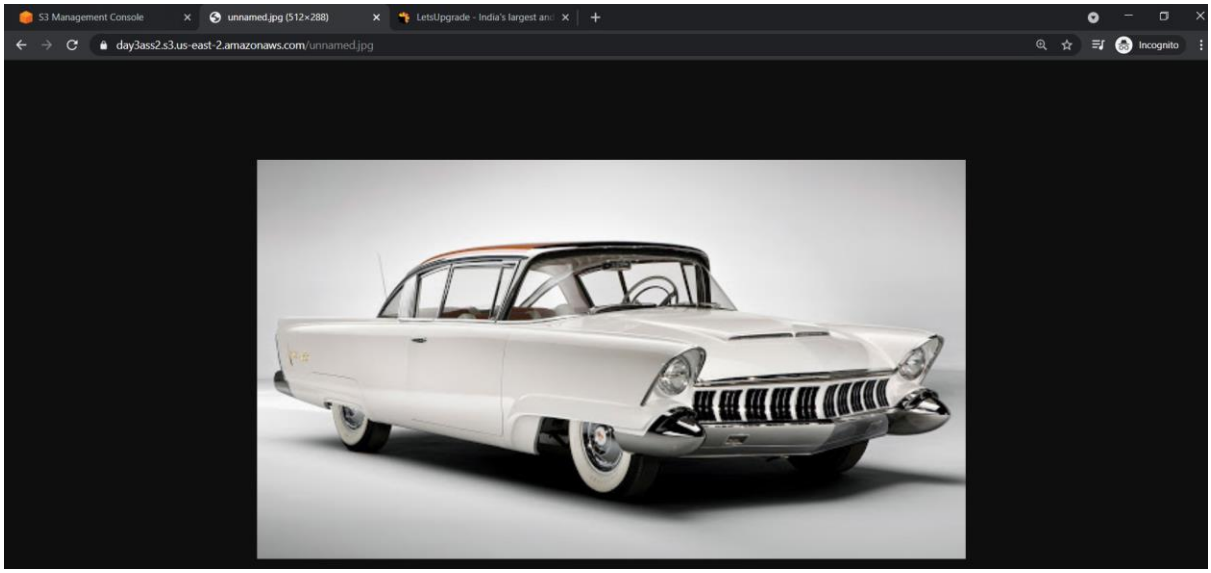


Then save changes.



Step 6: Again, copy the same Object URL: https://day3ass2.s3.us-east-2.amazonaws.com/unnamed.jpg and open on a new tab then output is as below:

OUTPUT:



END

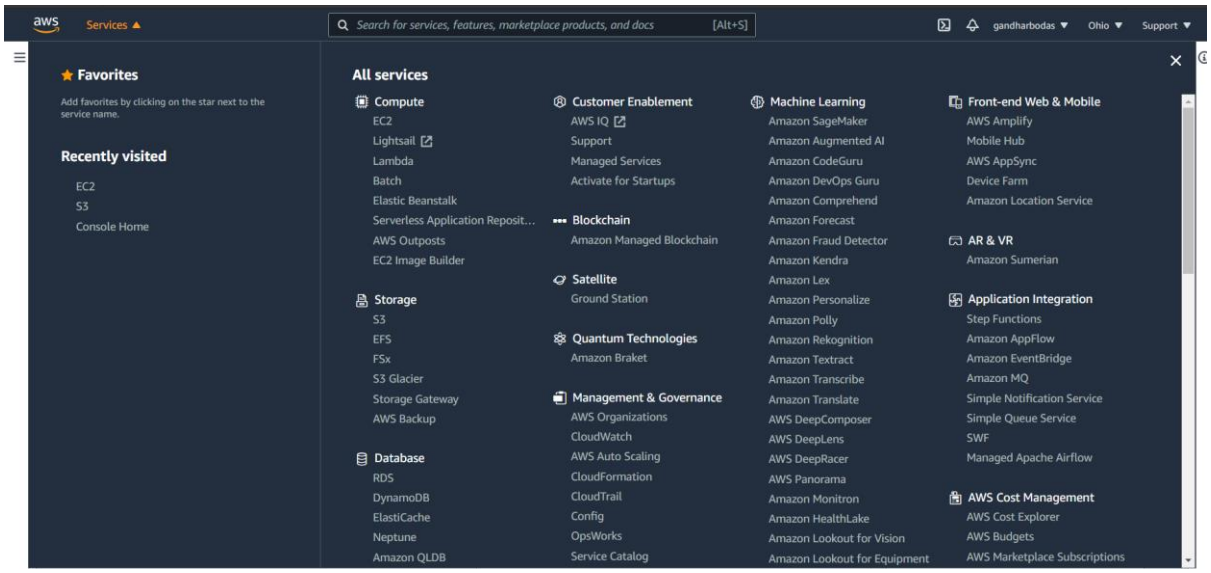
Answer:

Part b):

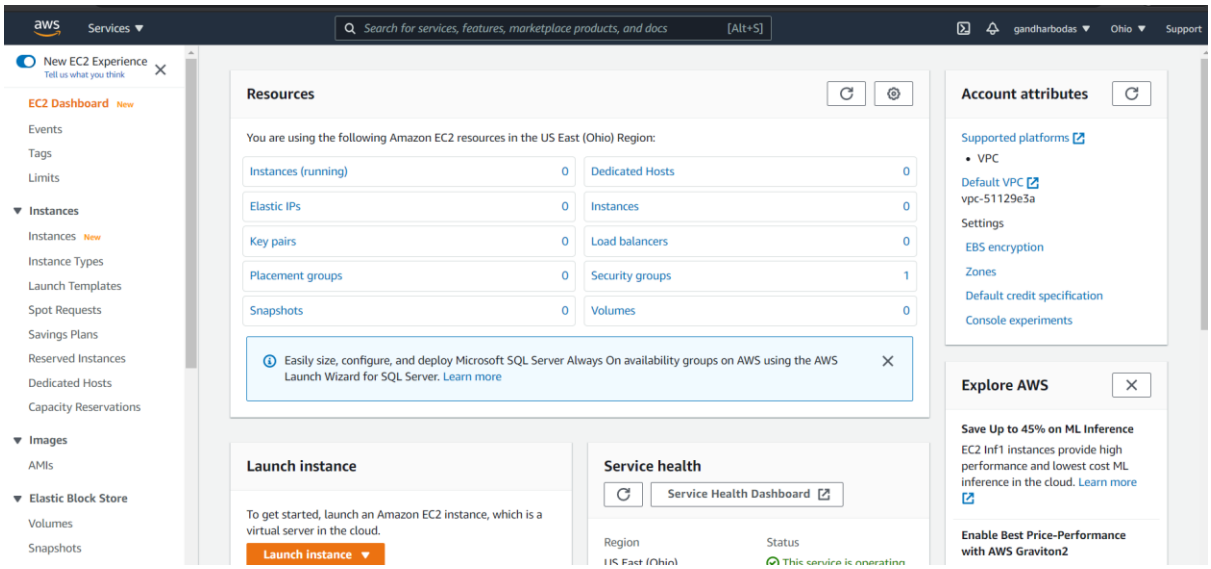
Step 1: *Create 2 EC2 Instances with ngnix web server installed*

Instance 1:

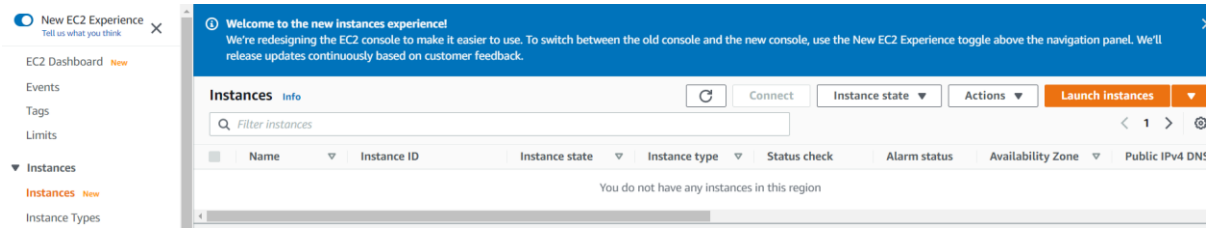
Go to the services --→ click on EC2-→



you will be able to see the below page

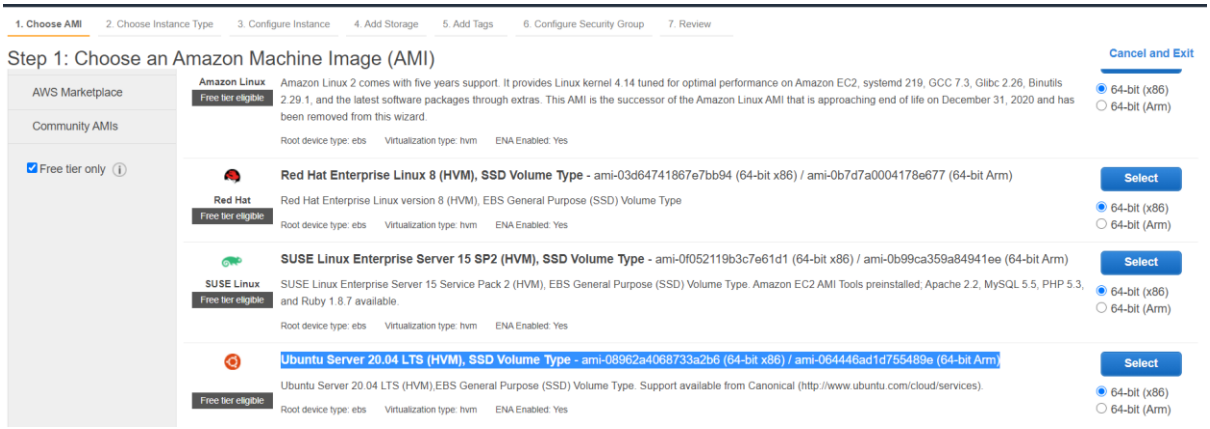


Click on instances option which is on left hand side panel:

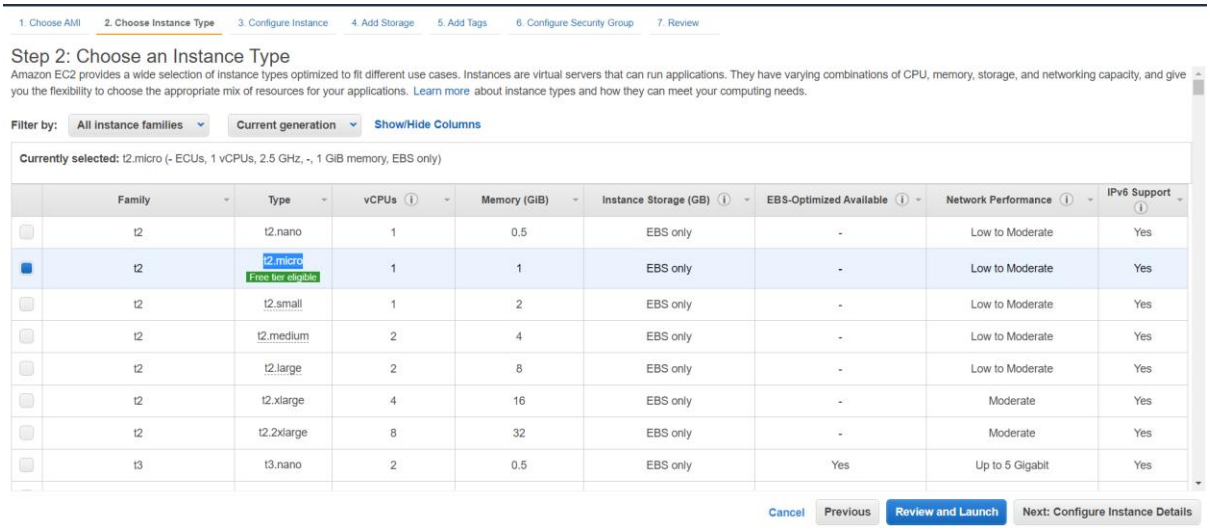


Click on “Launch Instances” and you will get the new page i.e., AMI page.

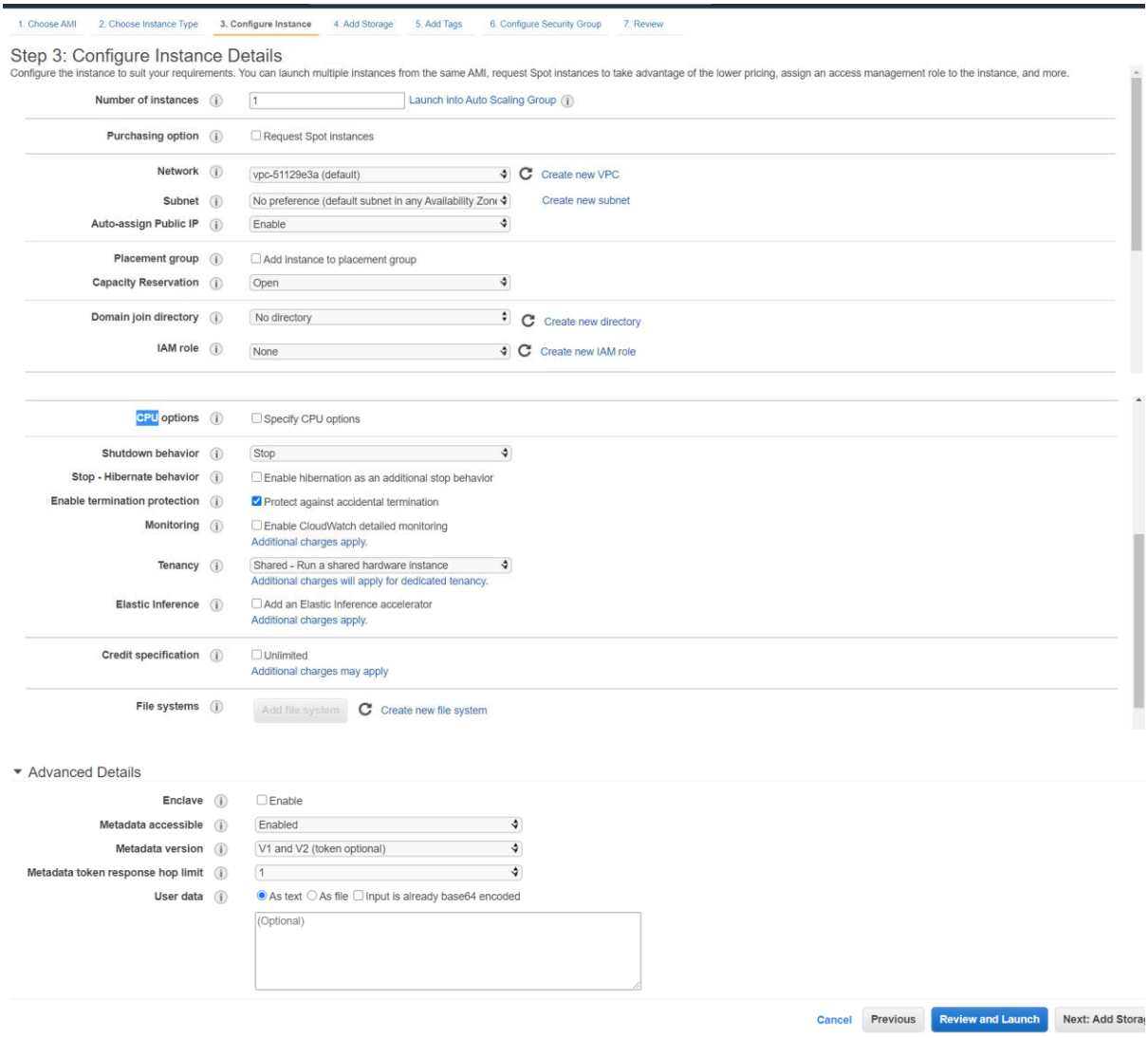
Click on check box: “Free the only” and select “Ubuntu **Server 20.04 LTS (HVM), SSD Volume Type** - ami-08962a4068733a2b6 (64-bit x86) / ami-064446ad1d755489e (64-bit Arm)” and click on “Select bttuon”.



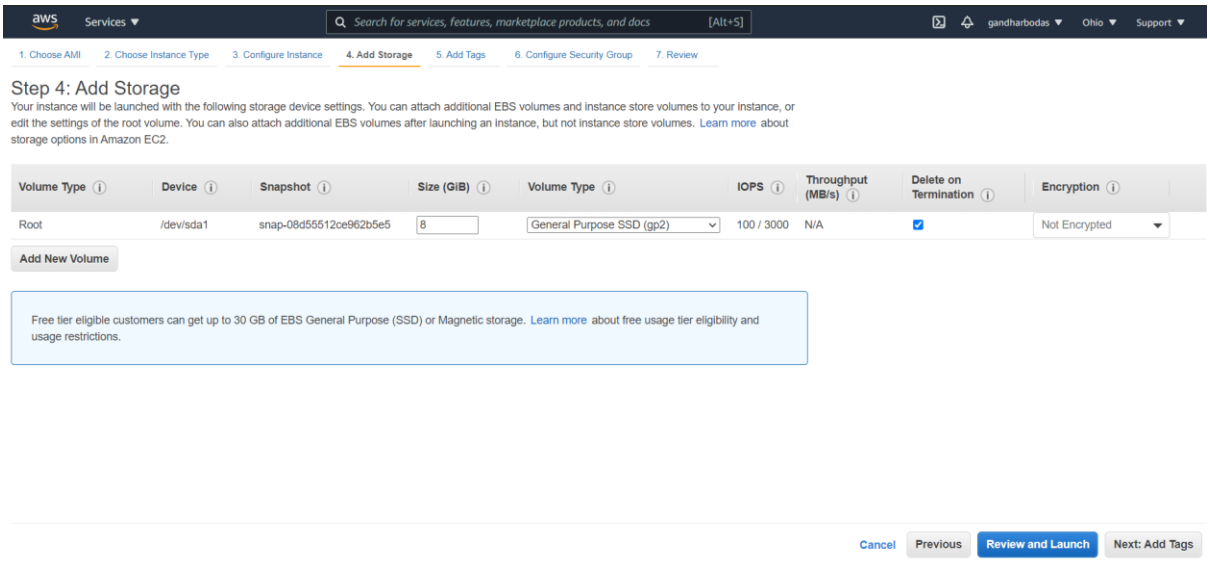
Select Instance type: t2. micro (Free tier eligible) and then click l “Next: Configure instance Detail’s button”.



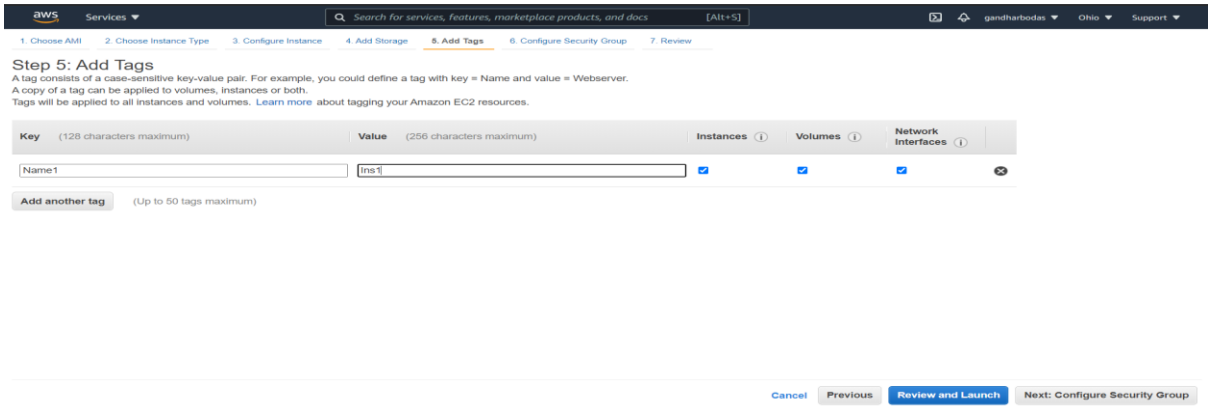
Configure Instance Details as **Auto-assign Public IP: Enable, enable termination protection: tick check box as “Protect against accidental termination” and later on click on Next: Add Storage.**



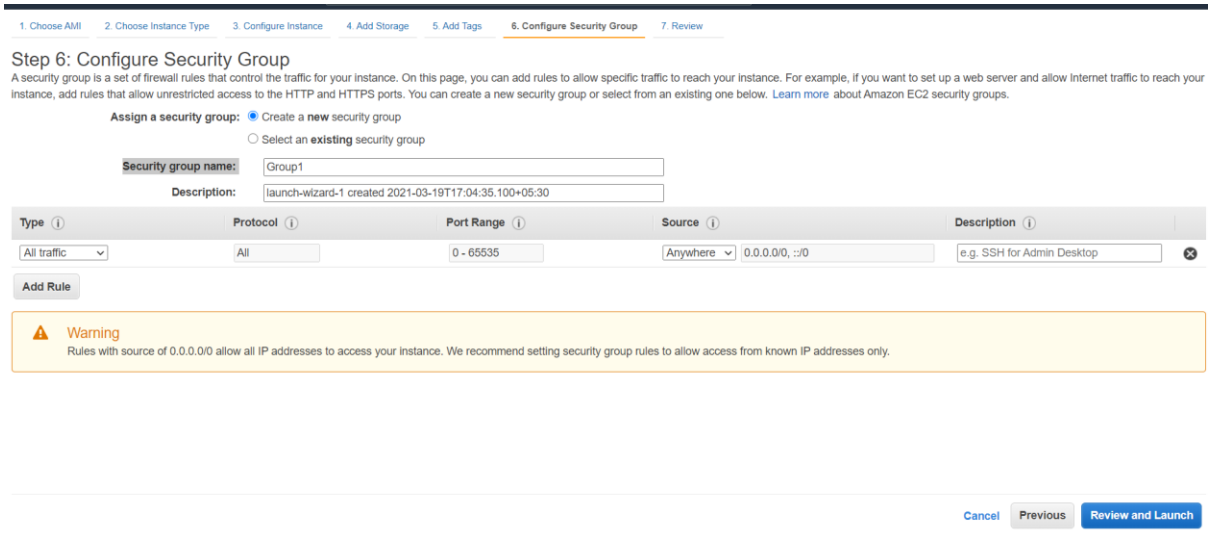
Add Storage: Keep as it is. Later on, click on Next: Add Tags



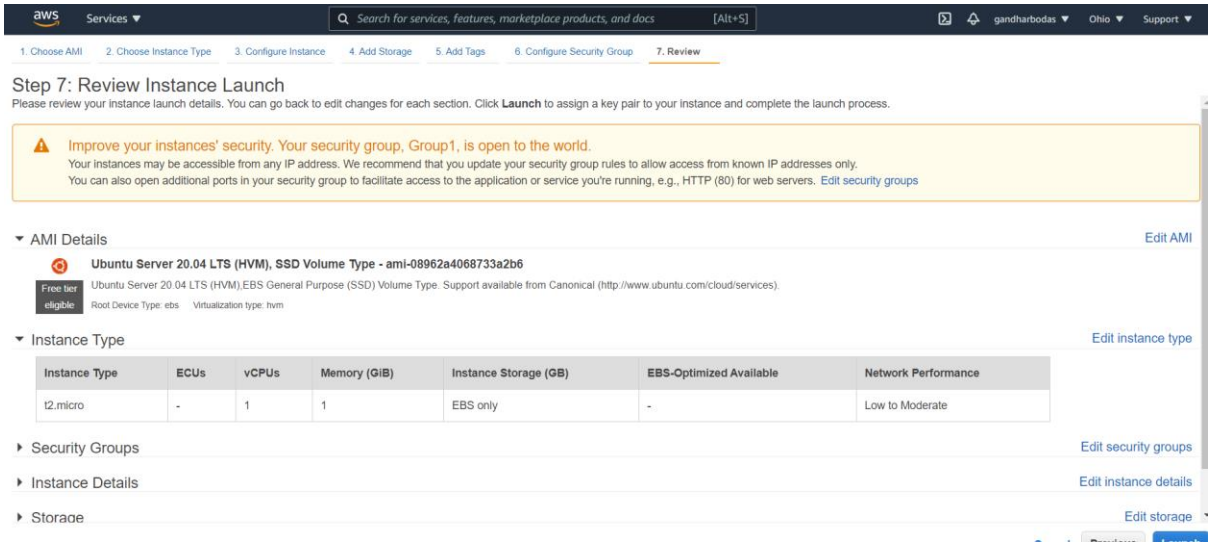
Add Tags: Key – Name1 and Value: Ins1 and click on Next: Configure Security Group button



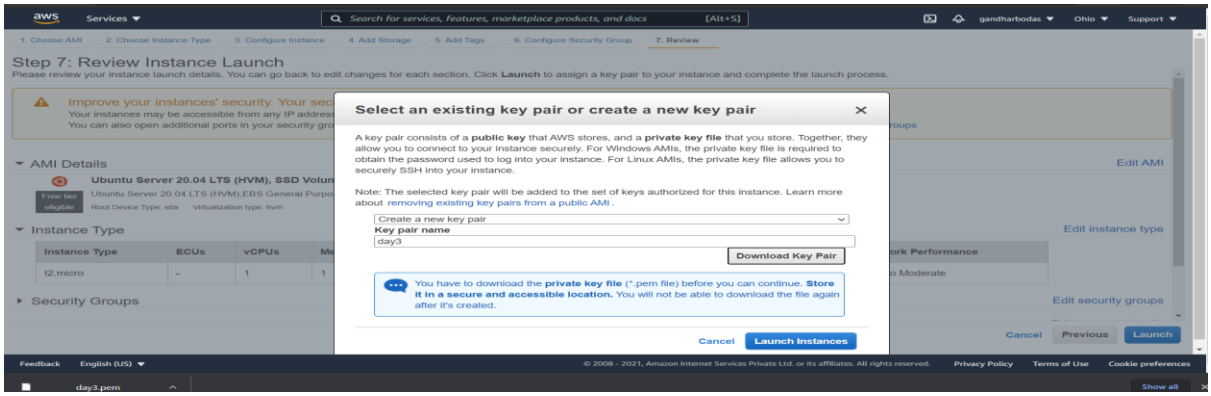
Configure Security Group: Security group name: - Group1, Type: All Traffic, Source: Anywhere and click on Review and Launch button.

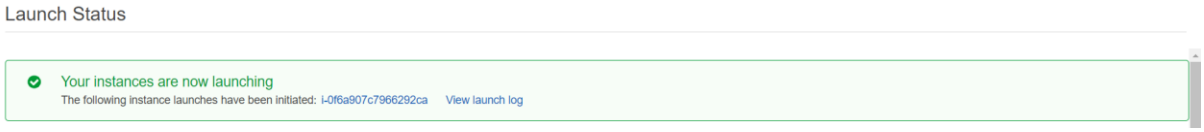


Review Instance Launch and keep as it is and later on Launch button:

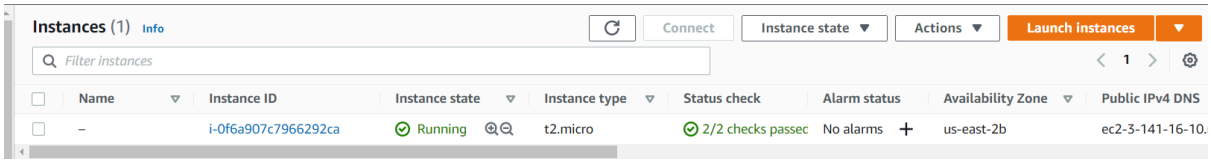


Creating a new key pair, name as: day3 and download key pair and click on “Launch Instances button”

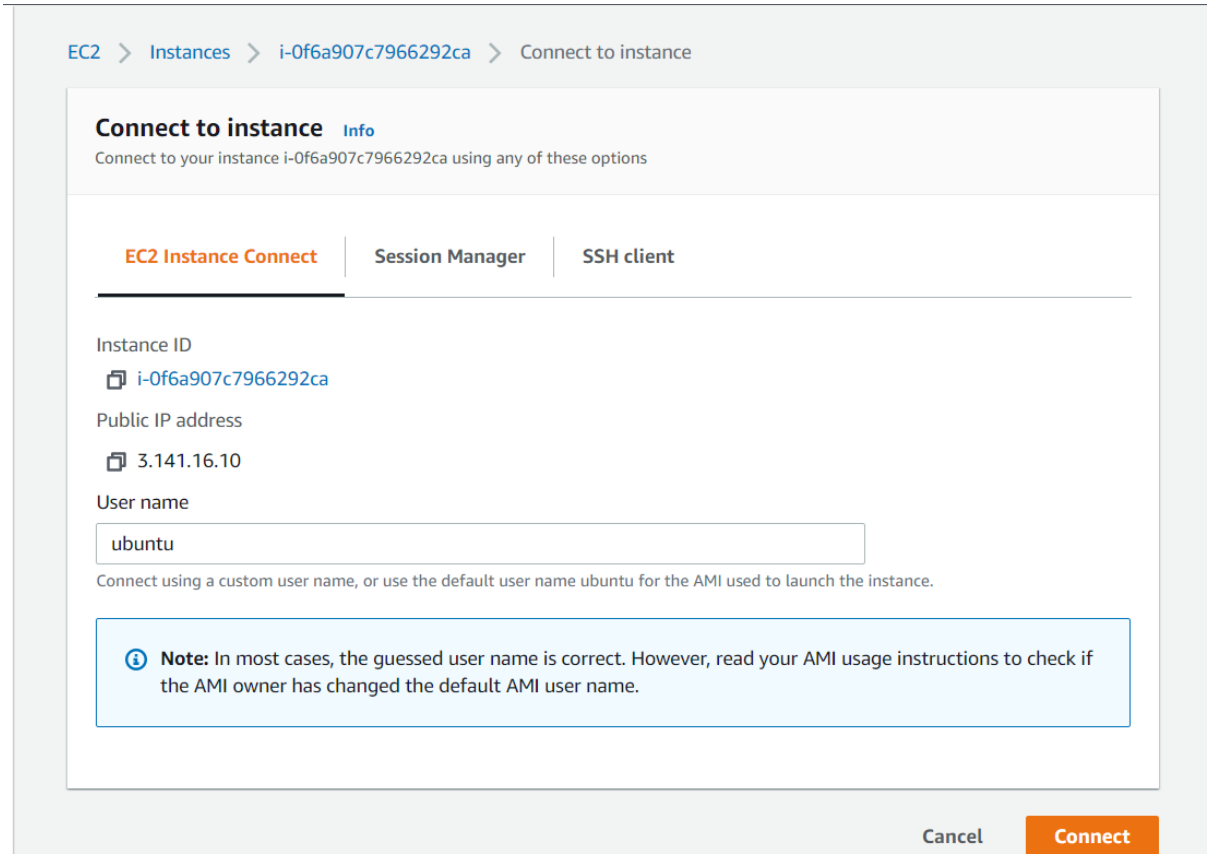




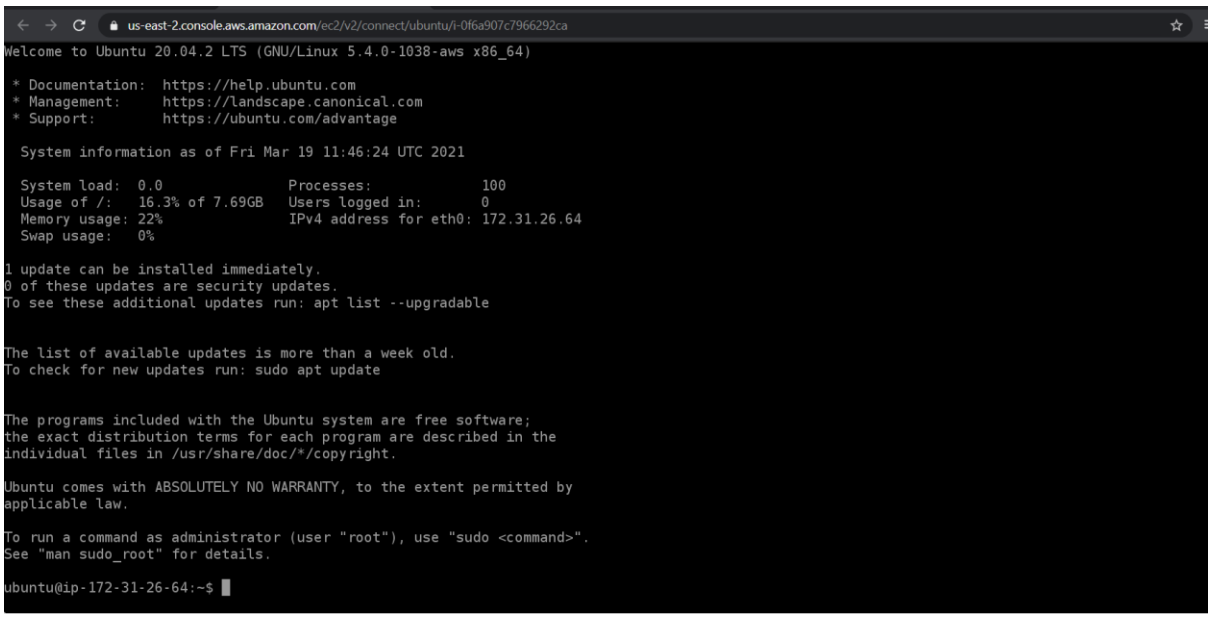
Click on View Instances button:



Now click on “Connect button and you will be able to see this page:



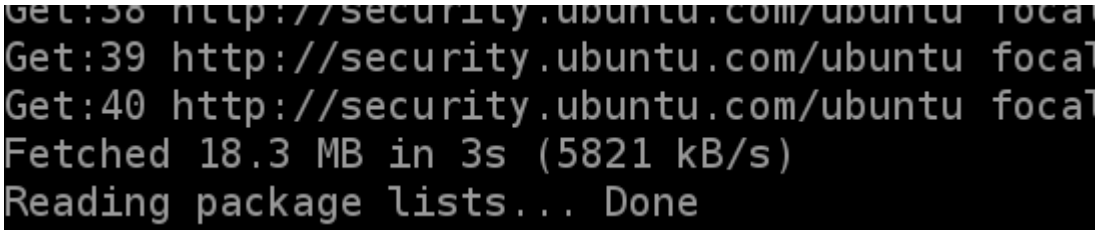
Click on “Connect” button and you will be able to see the black screen on new tab:

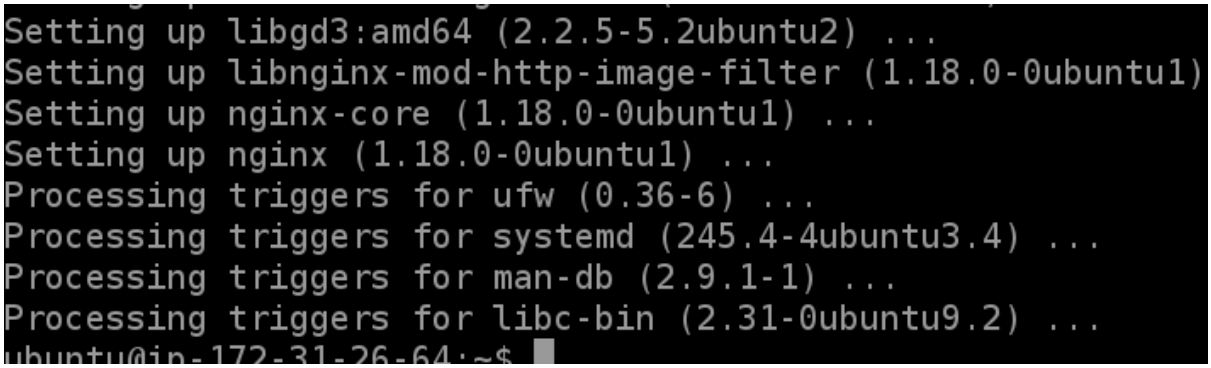


Now fire the below commands on the above screen:

Sudo apt-get -y update

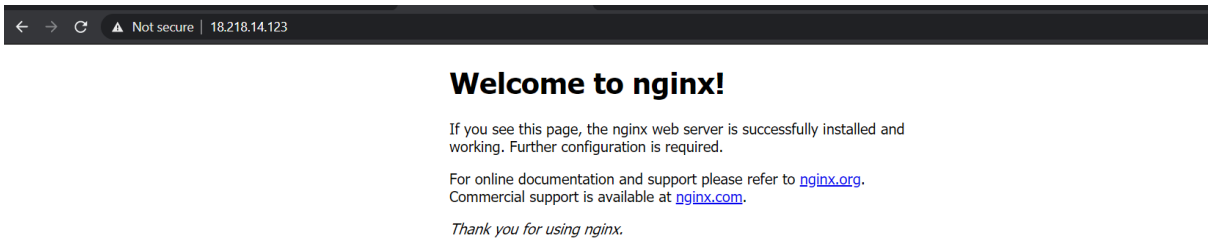
Sudo apt-get -y install nginx





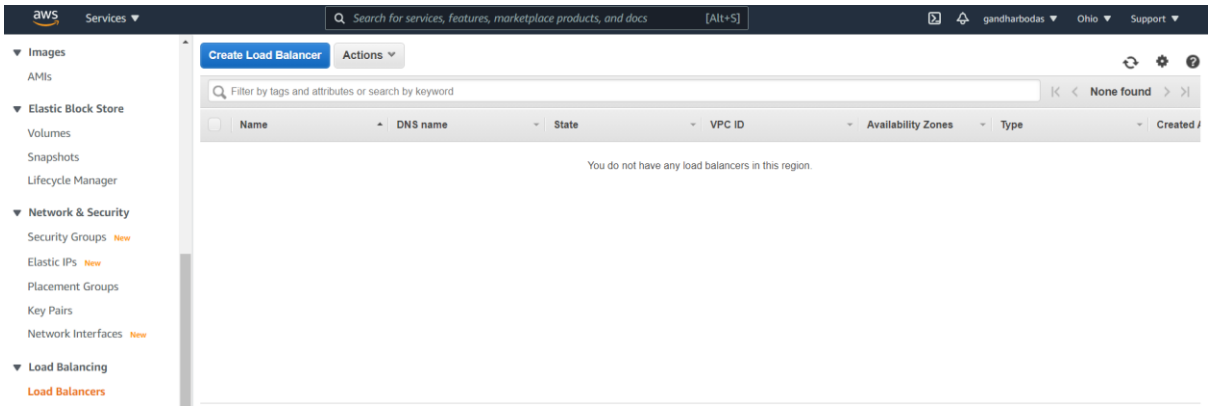
Once update and installed nginx web server then copy the public ip : 3.141.16.10 and paste it on a new tab and you will be able to see the output:

Instance 2: Follow the above same procedure to create another Instance

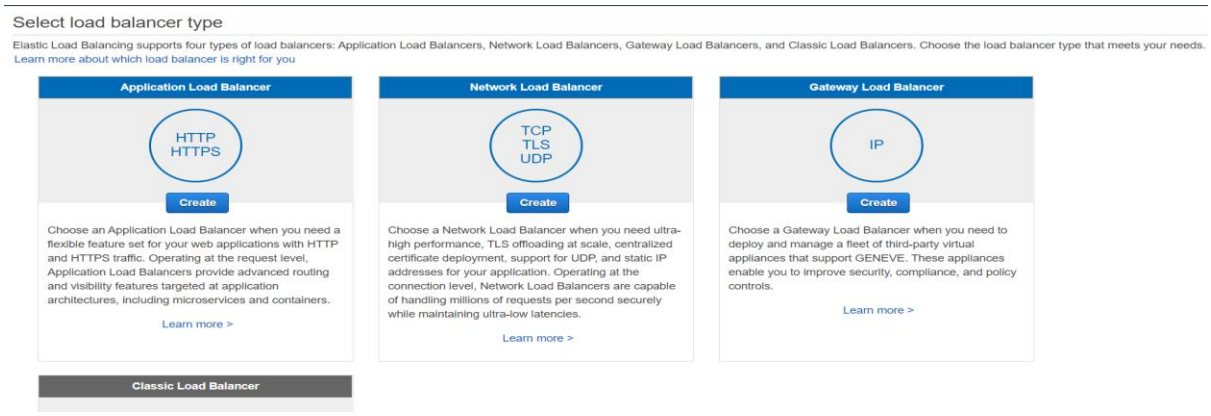


Step 2: Create Elastic Load Balancer:

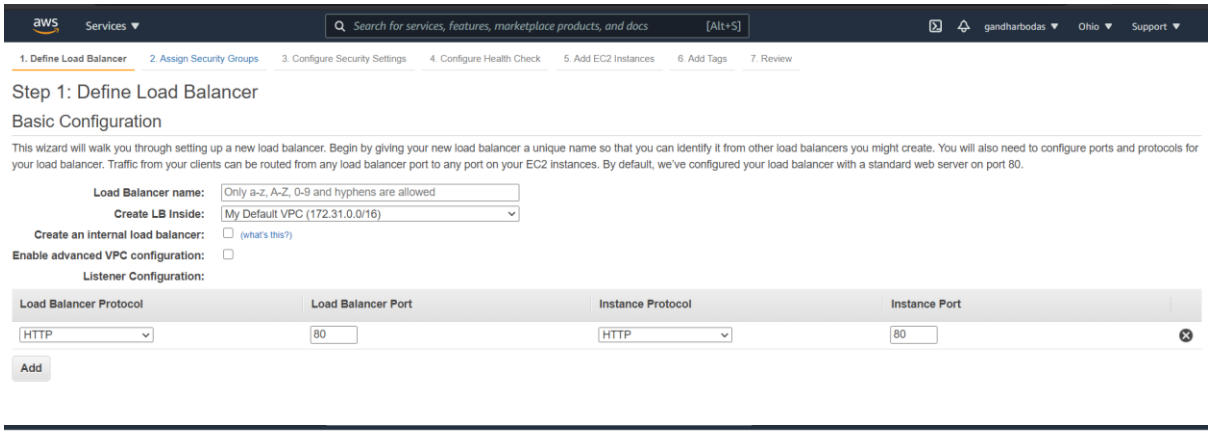
Click on “Load Balancer” option which is on Left Hand Side.

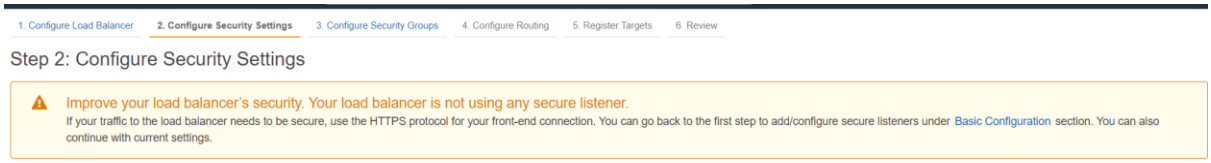


Now click on “Application Load Balancer button”



Select “Application Load Balancer” and click on Create button.Now Basic configuration: Load Balancer Name- Demo, Load Balancer Protocols – HTTP, Port: 80, Availability Zones: tick on all check boxes, and now click on “Next: Assign Security Groups button” and you we be able to see the below screen:





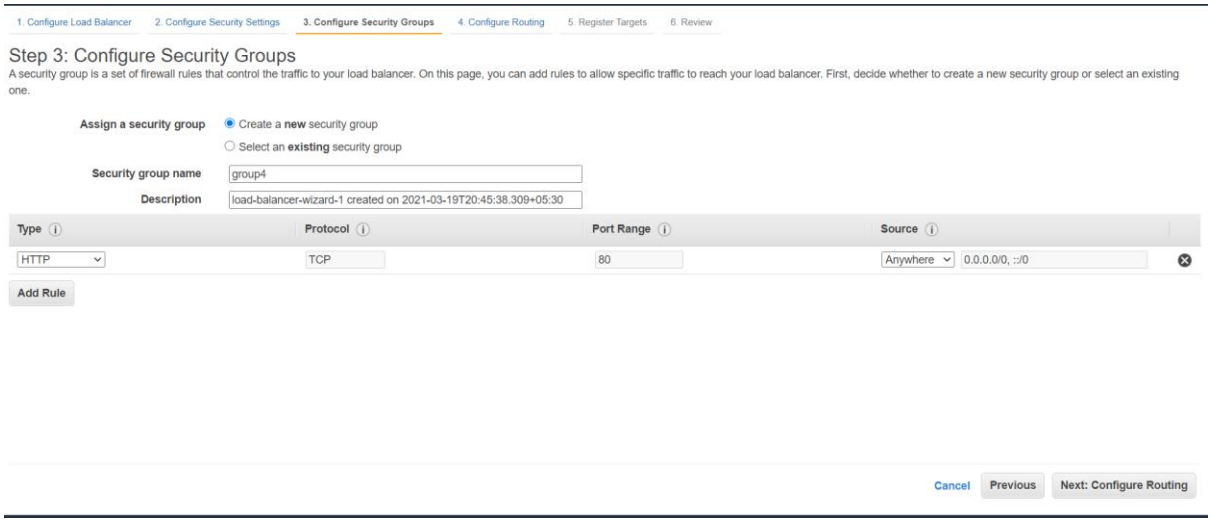
Click on “Next: Configure Security Groups” then Configure Security Groups page will be appearing.

Assign a security group: Create a new security group.

Security group name: group4.

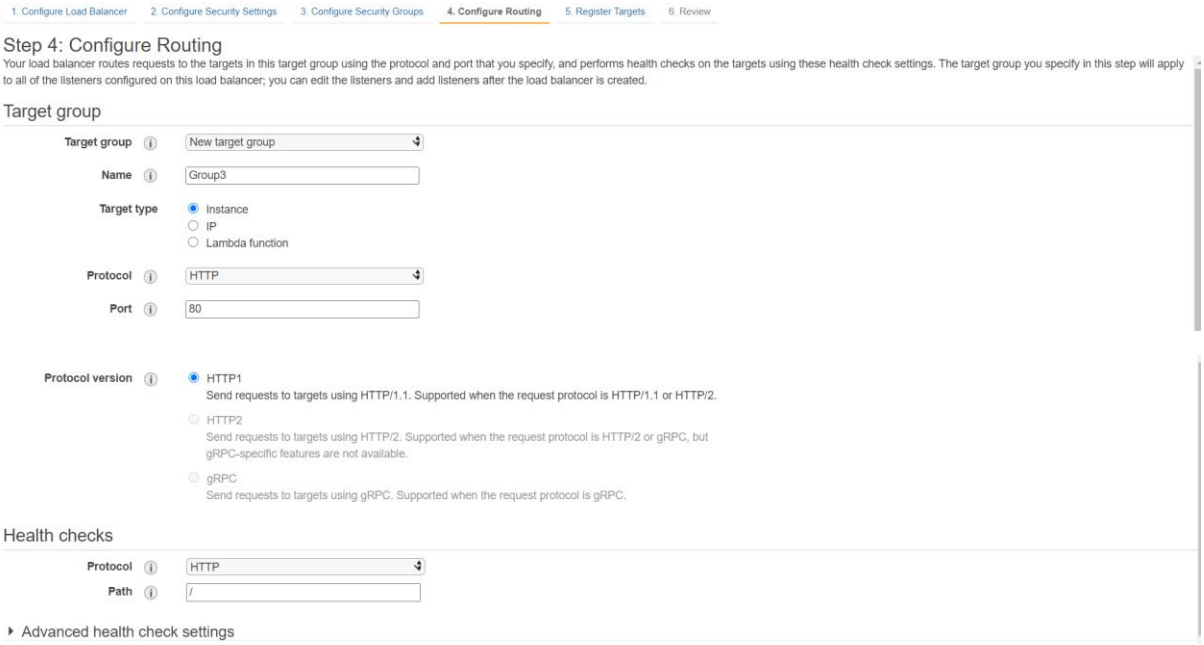
Type: HTTP, Protocol: TCP, Port Range: 80, Source: Anywhere.

Once selected these all click on “Next: Configure Routing” button.

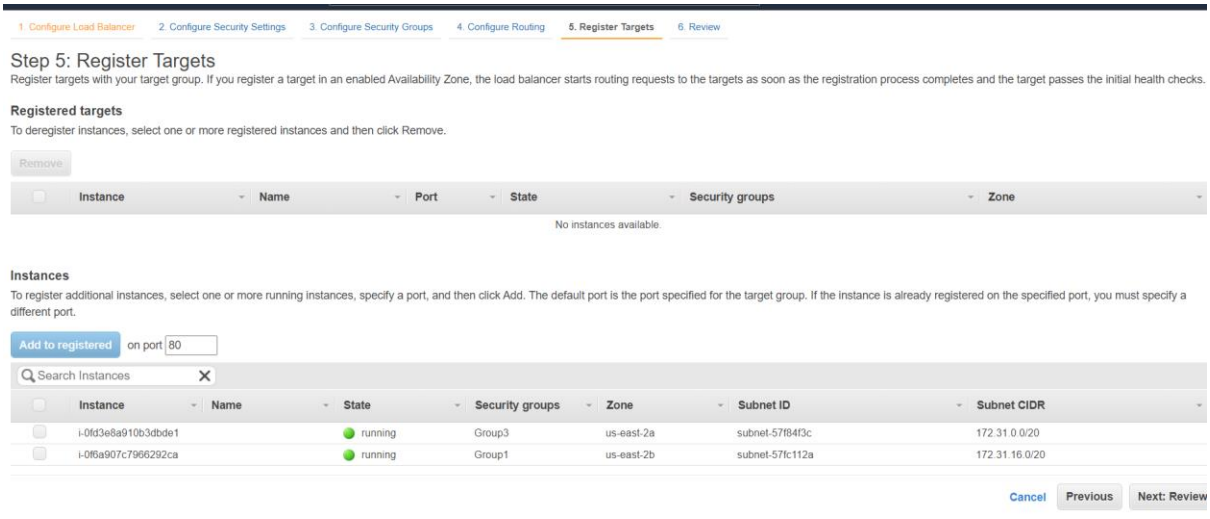


Once clicked on “Next: Configure Routing” button you will be able to see the below page:

Here, you will select Target group: New target group, Name: Group3, Target Type: Instance, Protocol: HTTP, Port: 80, Protocol version: HTTP1, and rest of the details keep as it is.

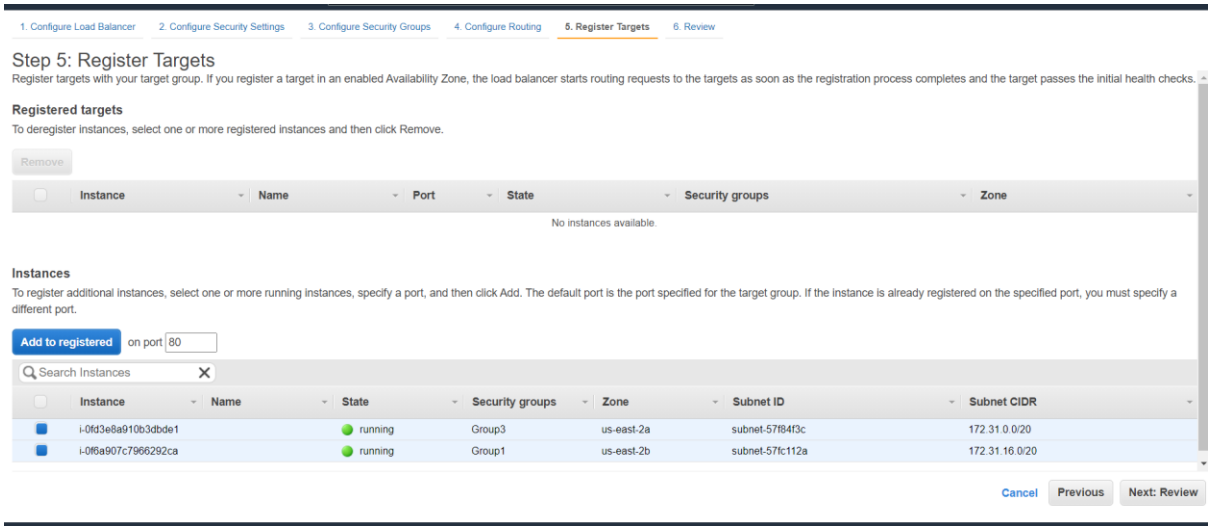


And later on, click on “Next: Register Targets button”. After clicking you will be able to see the Instances which we have created previously as such:

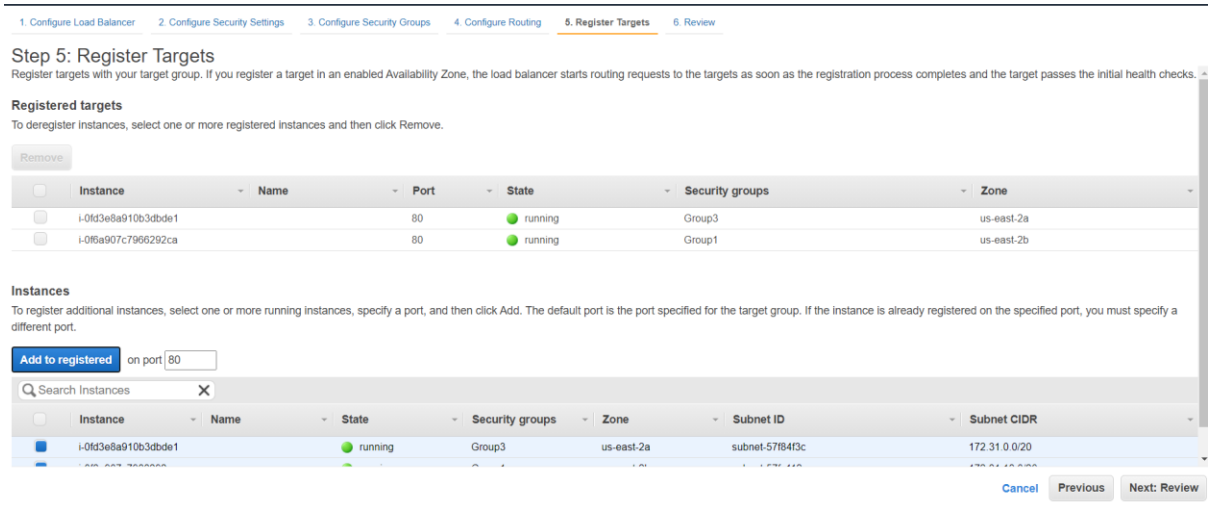


Step 3: Add these two instances into the ELB's target group:

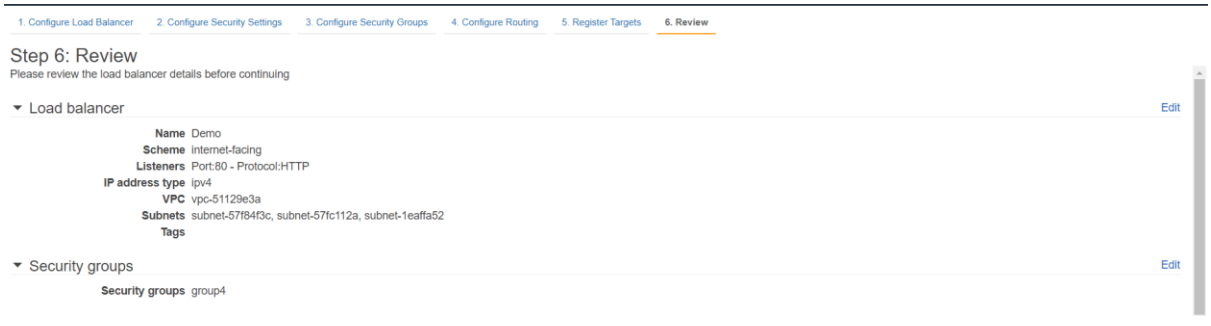
So, now click on two check boxes instance 1 and instance 2 and then click on "Add to registered".

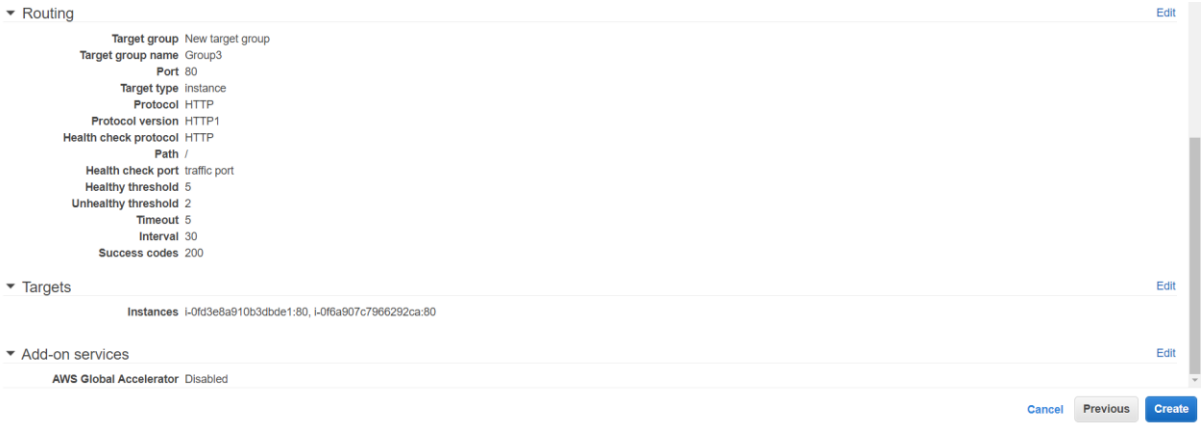


The moment I clicked on "Add to registered" button it will appear under "Registered targets"

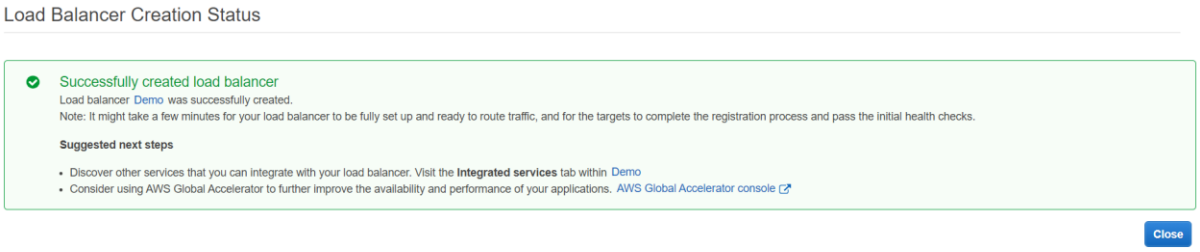


Now, appeared under "Registered targets" click on Next: Review button" and you will able to see the below Review page:

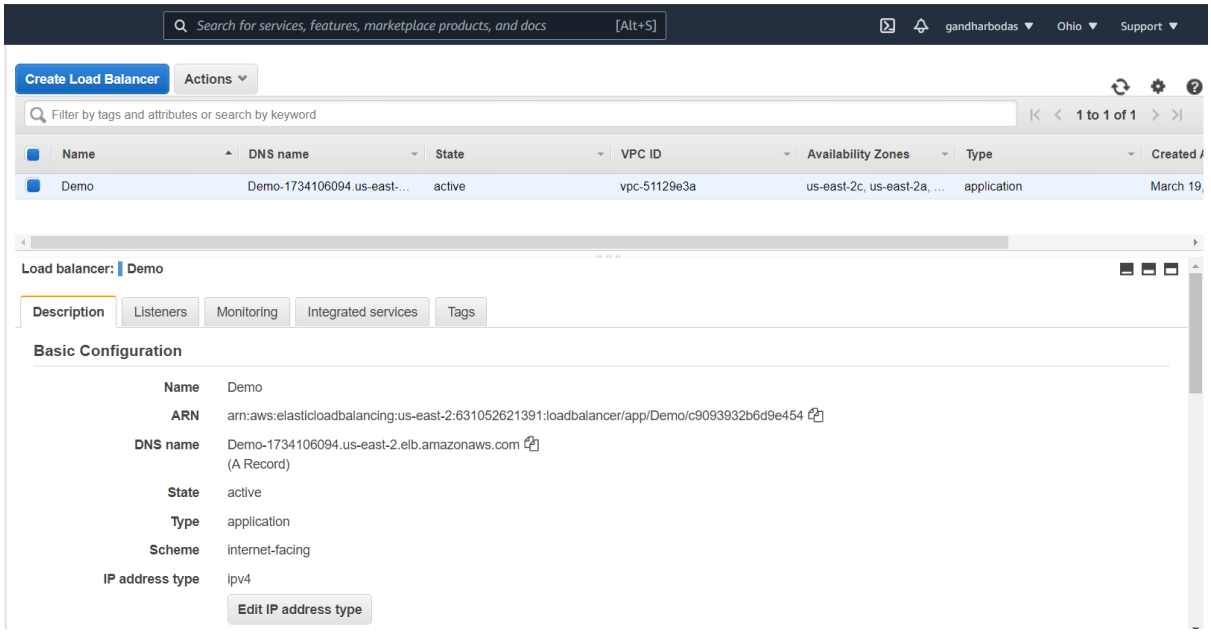


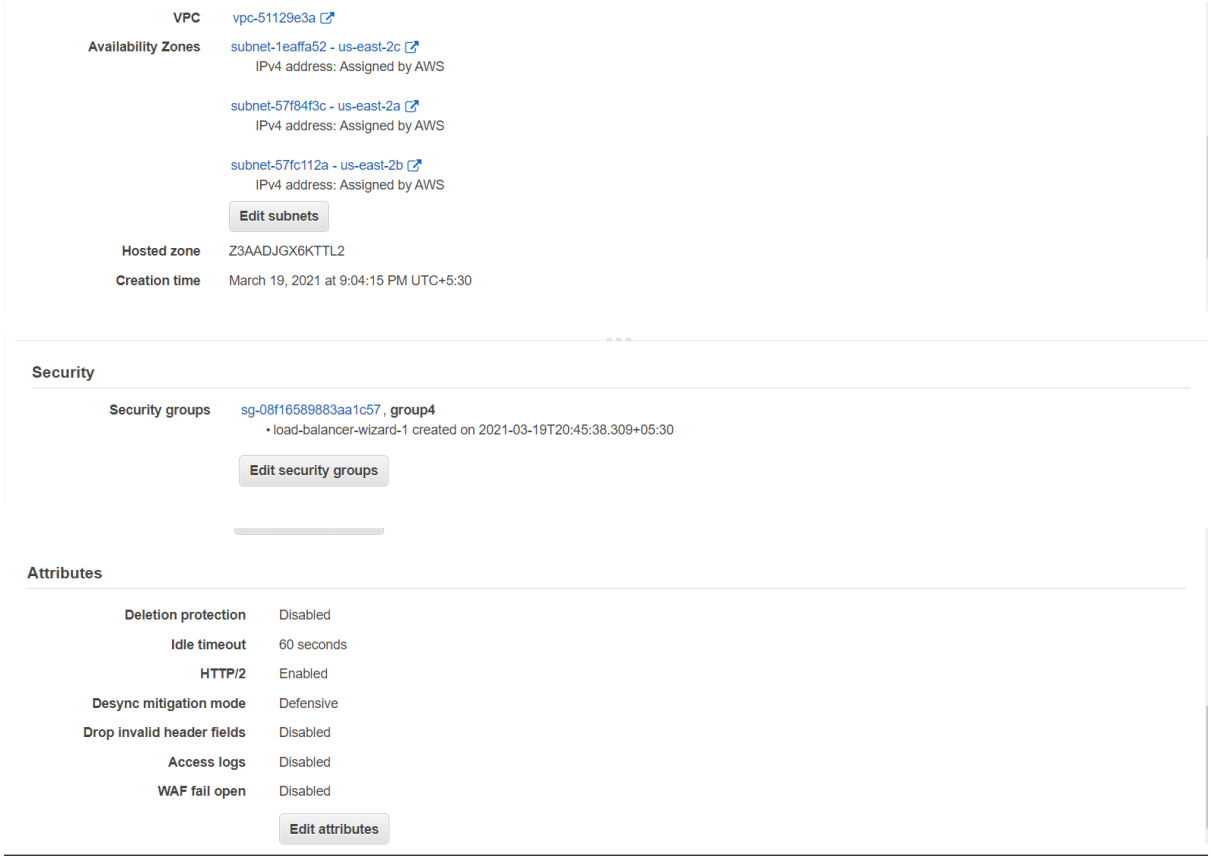


Once reviewed then click on “Create button”. Once done you will be able to see the below page:

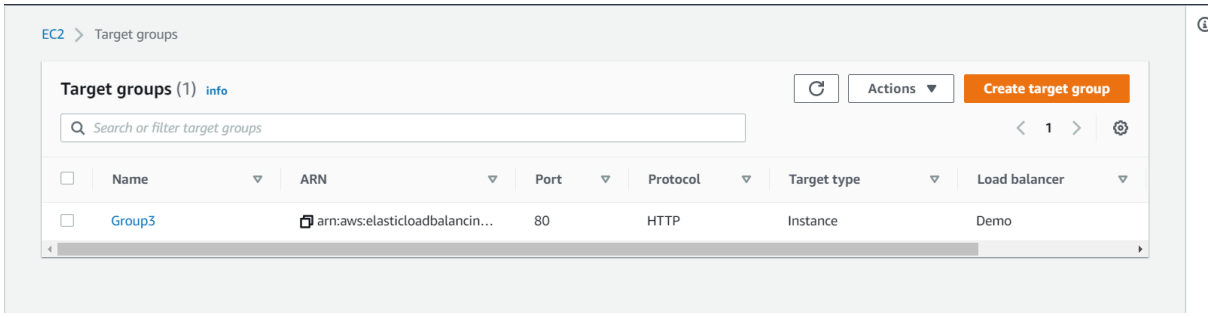


Once done then click on “Demo link” and it will take you to the home page as below:

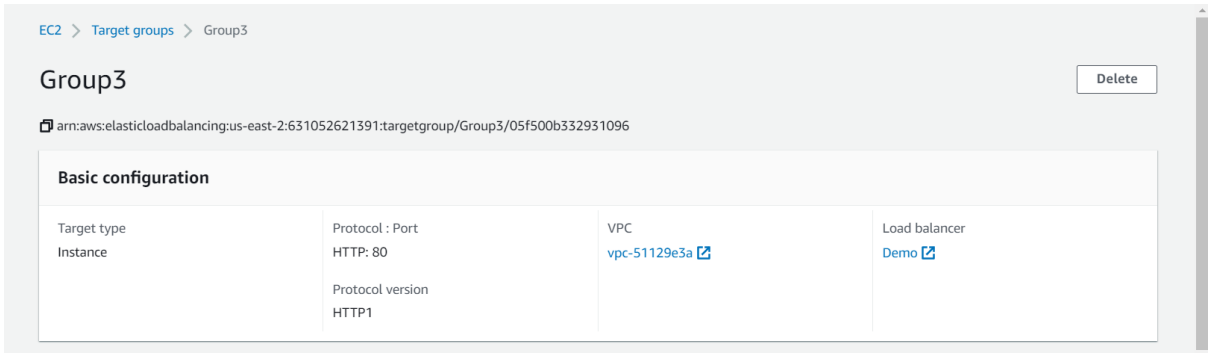


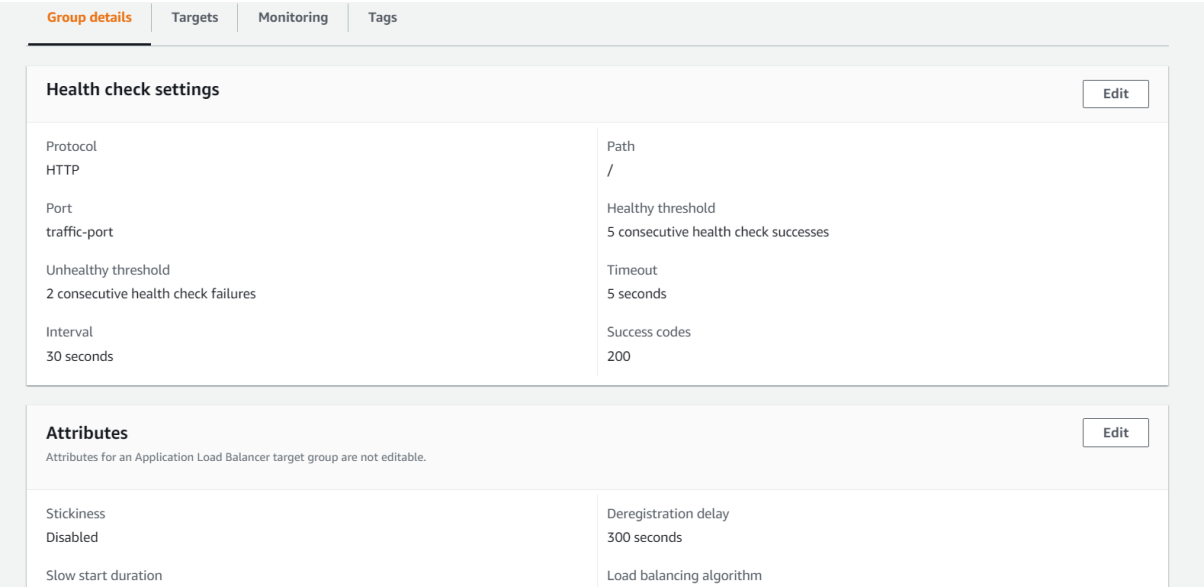


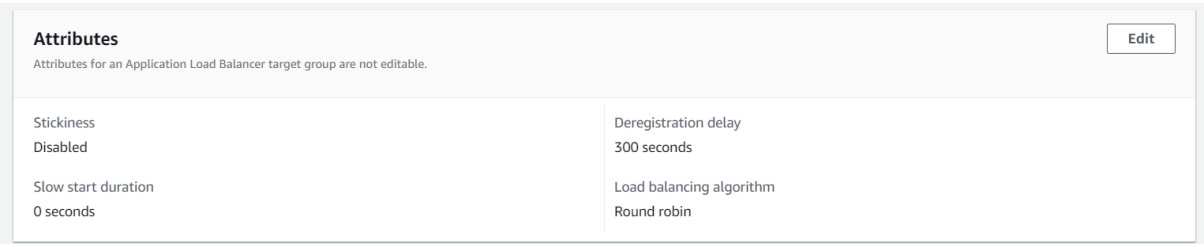
Now, as part of the Load Balancer creation we created a Target Group as below:

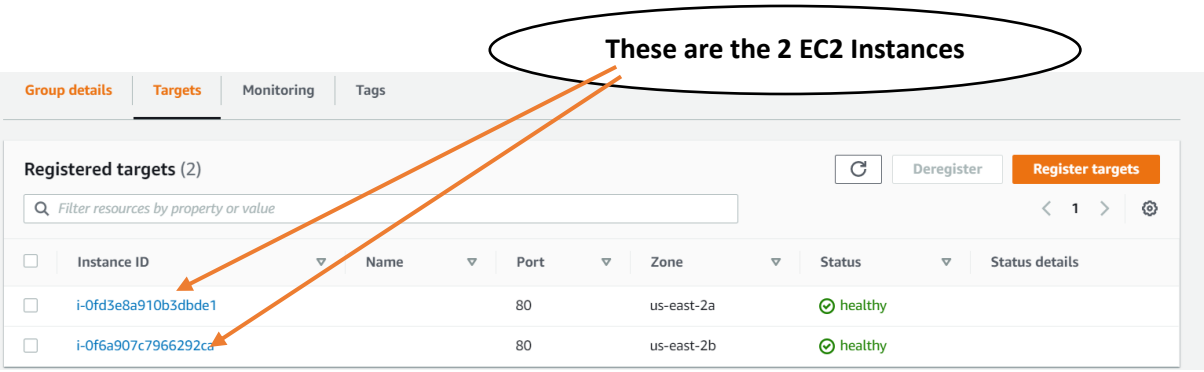


Now, go the home page of Name: Group3:









Now again to the “Load Balancer” and you can able to see the State: active”.

Copy “DNS name: Demo-1734106094.us-east-2.elb.amazonaws.com” and paste it on a new tab. Once pasted then you will be able to see the below output:

OUTPUT:

END