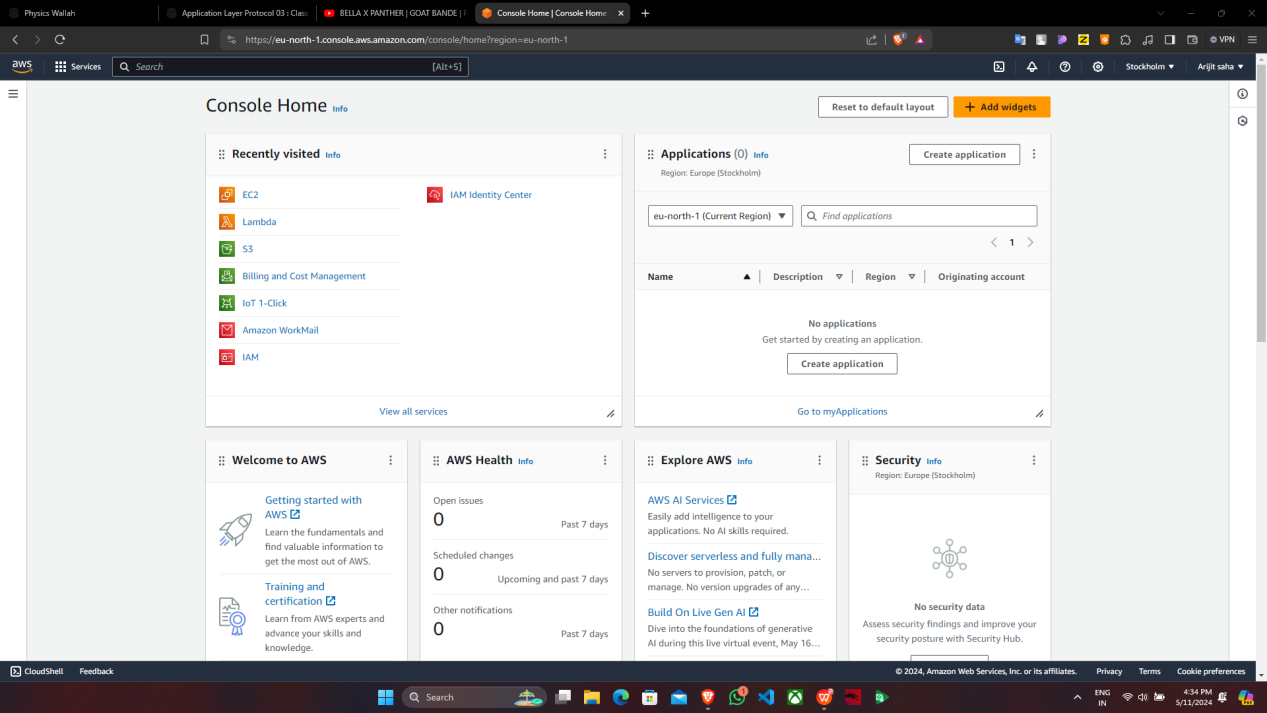
**Assignment No.-11**

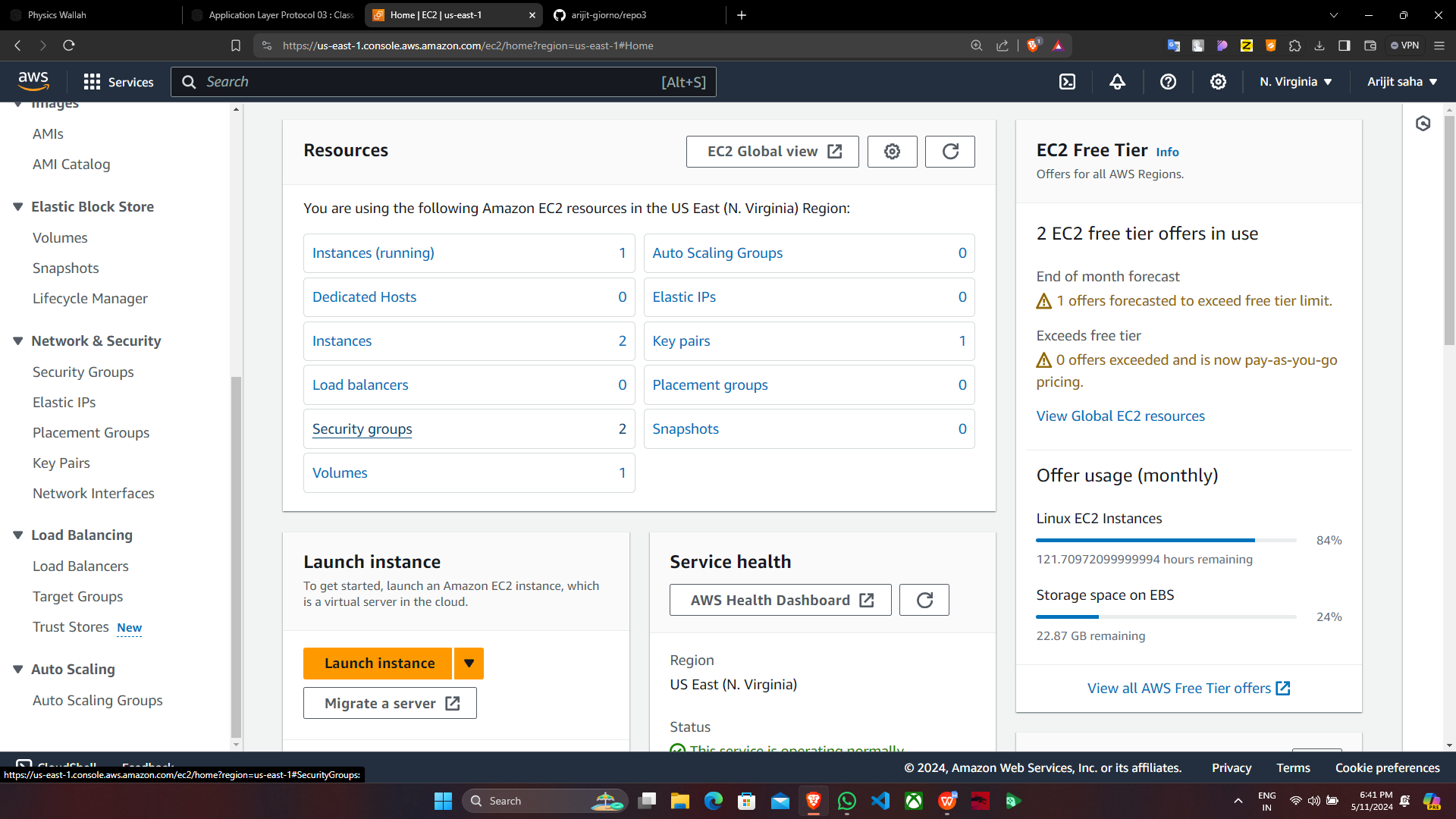
**PROBLEM STATEMENT :**  **Build scaling plans in AWS that balance the load on different EC2 instances.**

***Steps to build scaling plans :***

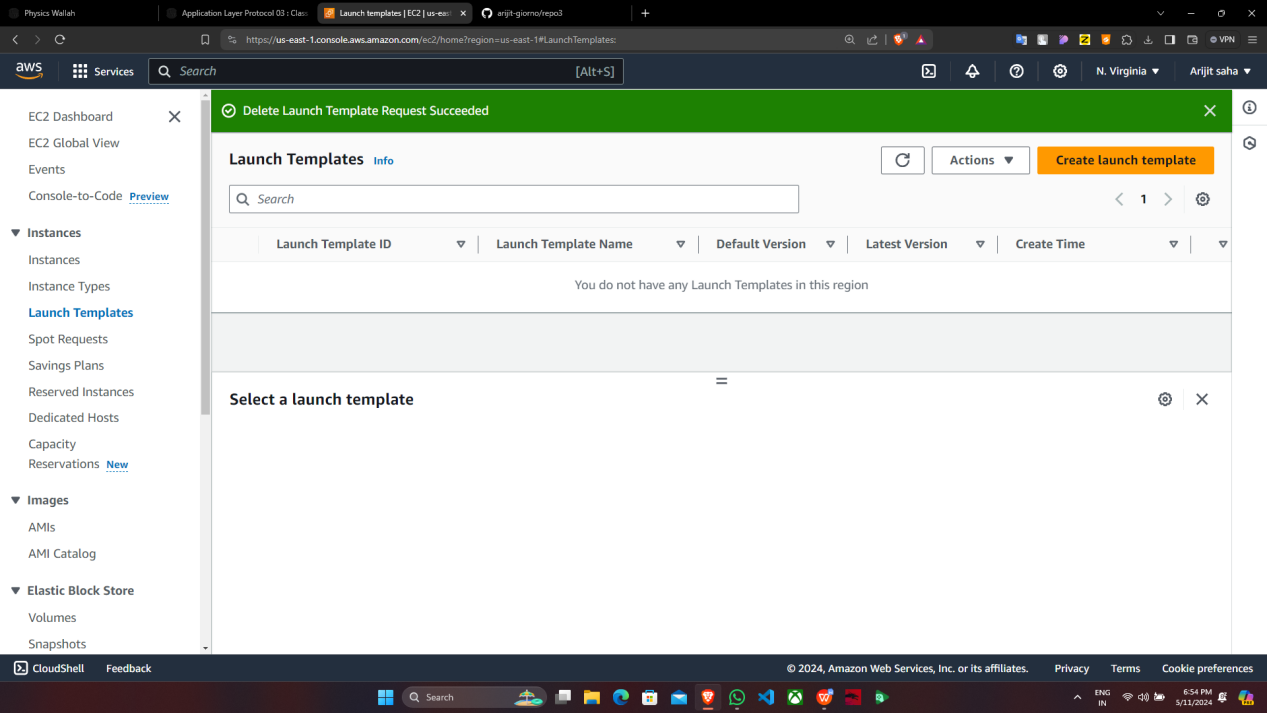
1. Sign up for an AWS account, search for ‘EC2’ then click on it.



2. If you have an existing Security group then no need to create it if not then create it.

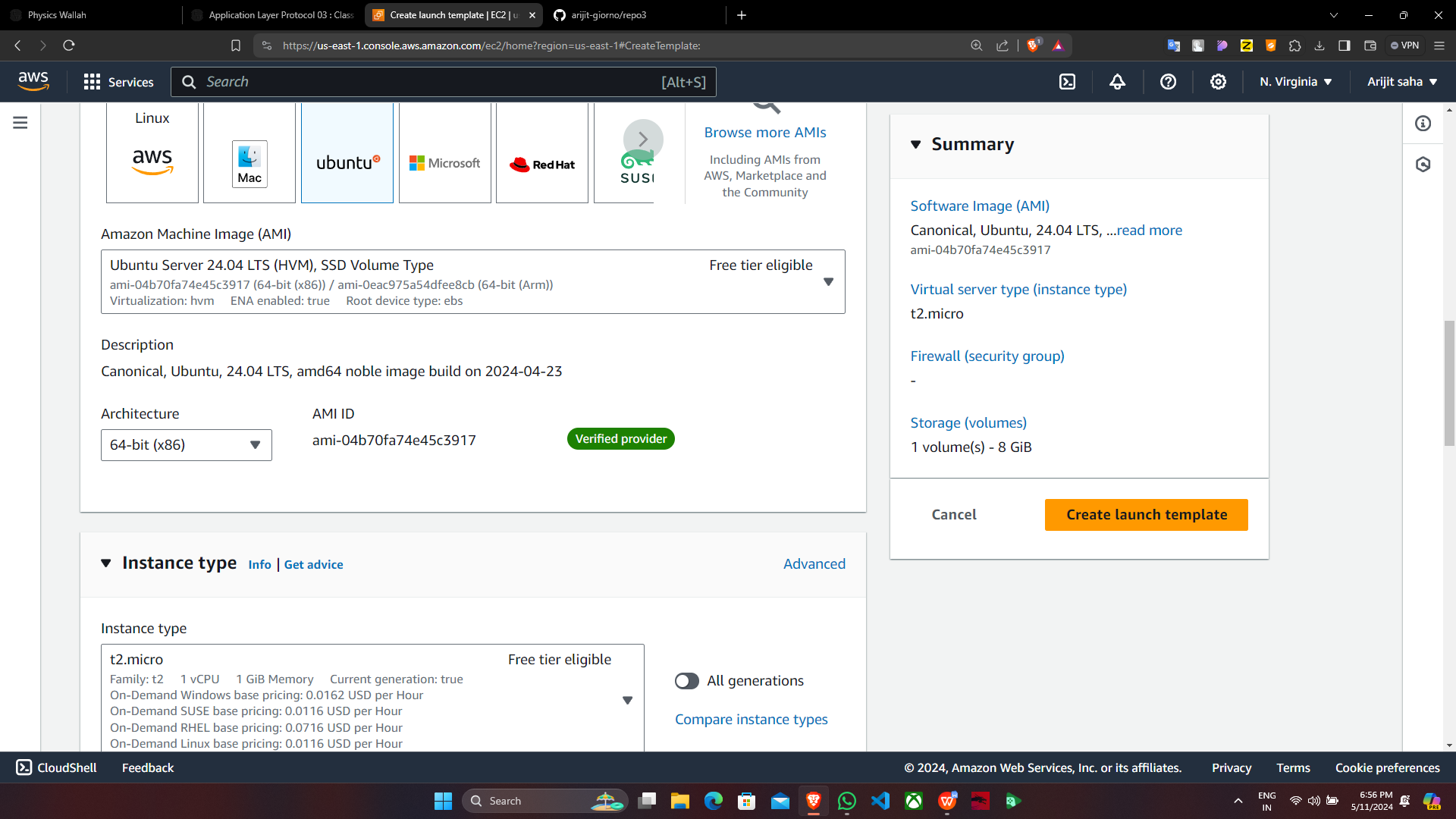


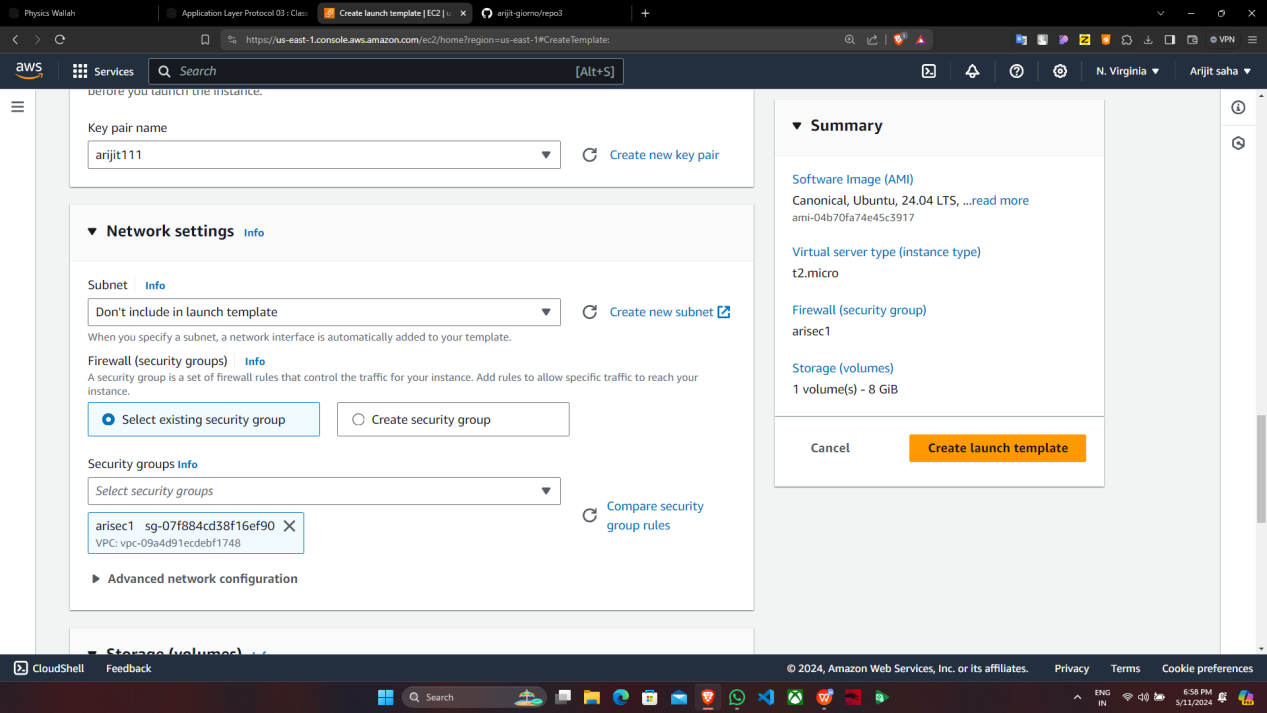
1. Create one template. Click Instances & there click on “Launch template”.



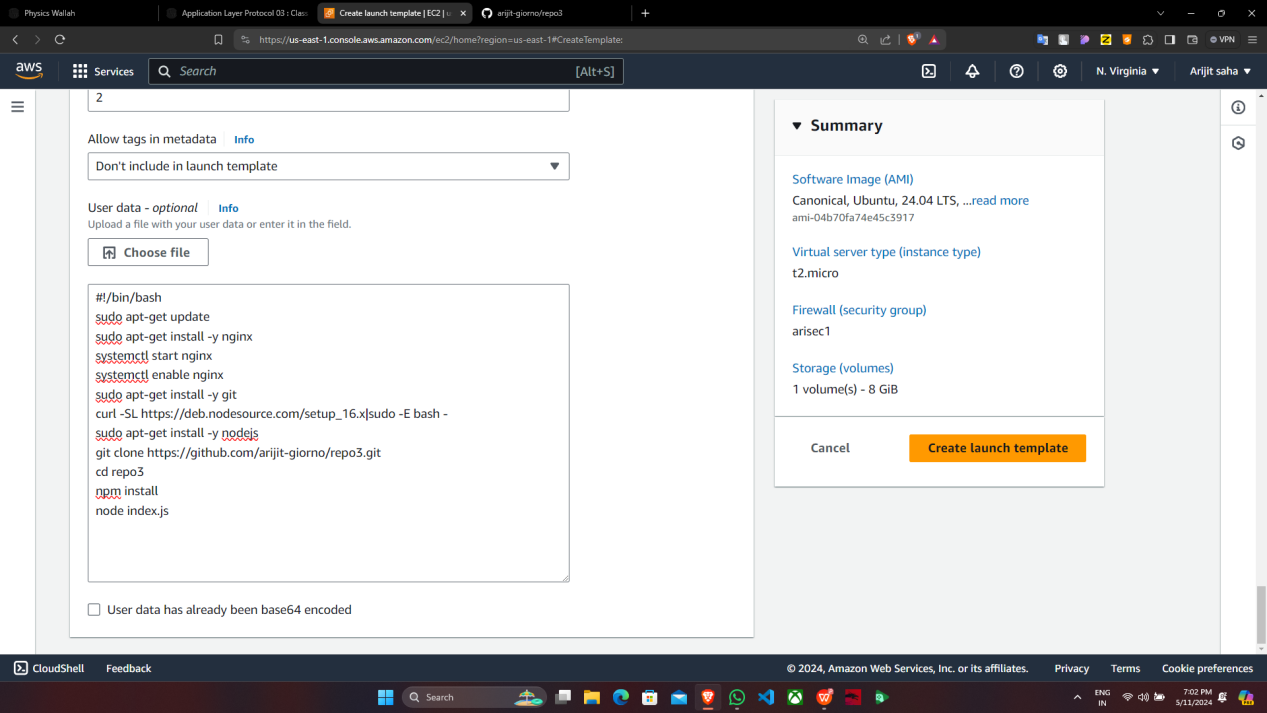
4. Now click on “Create launch template”.

5. Under “Create launch template”, give the following details.

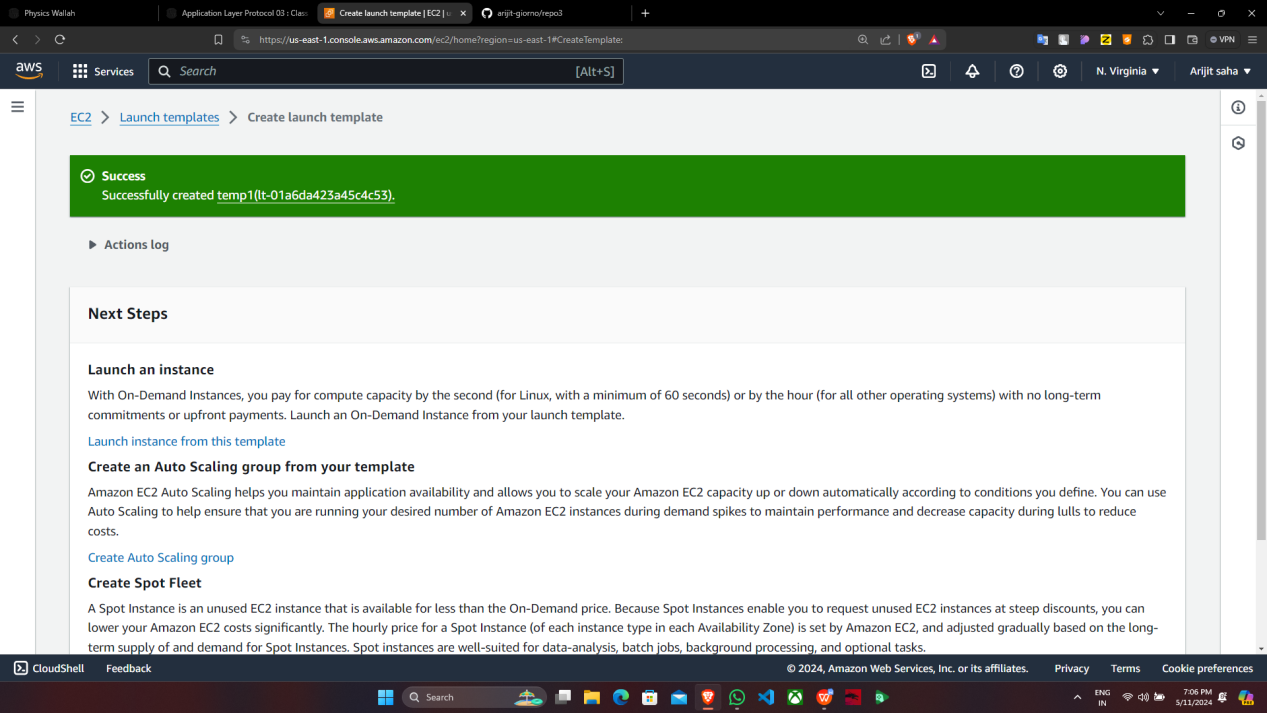
Click on “Quick start”->ubuntu & “Instance type”->t2.micro as it is free.Now select the key pair & the existing security group.



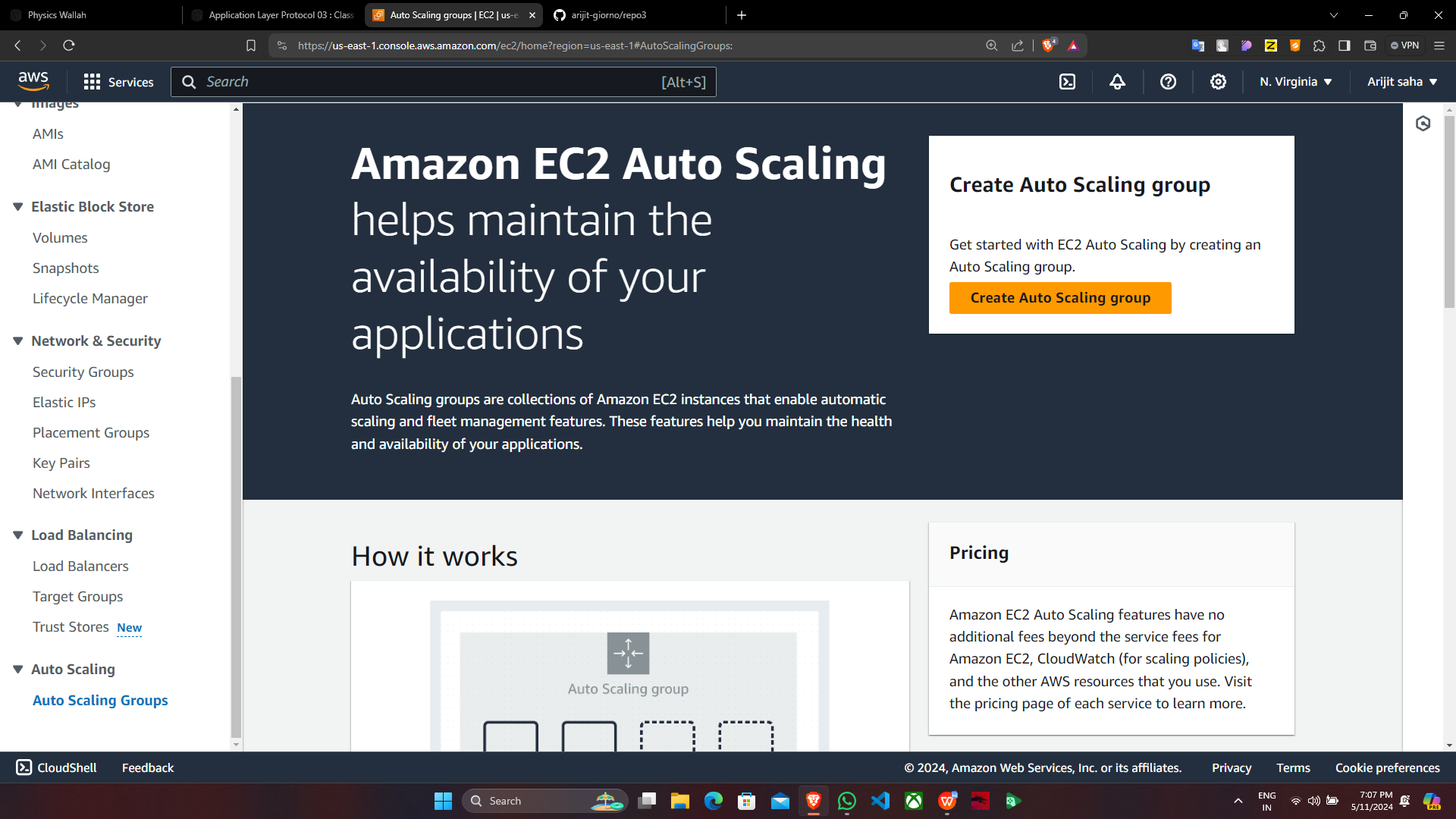
Expand the “Advanced Details” & scroll down to the bottom, in the bash console type the following commands, give the address & repository name from GitHub. Then click on “Create Launch template”



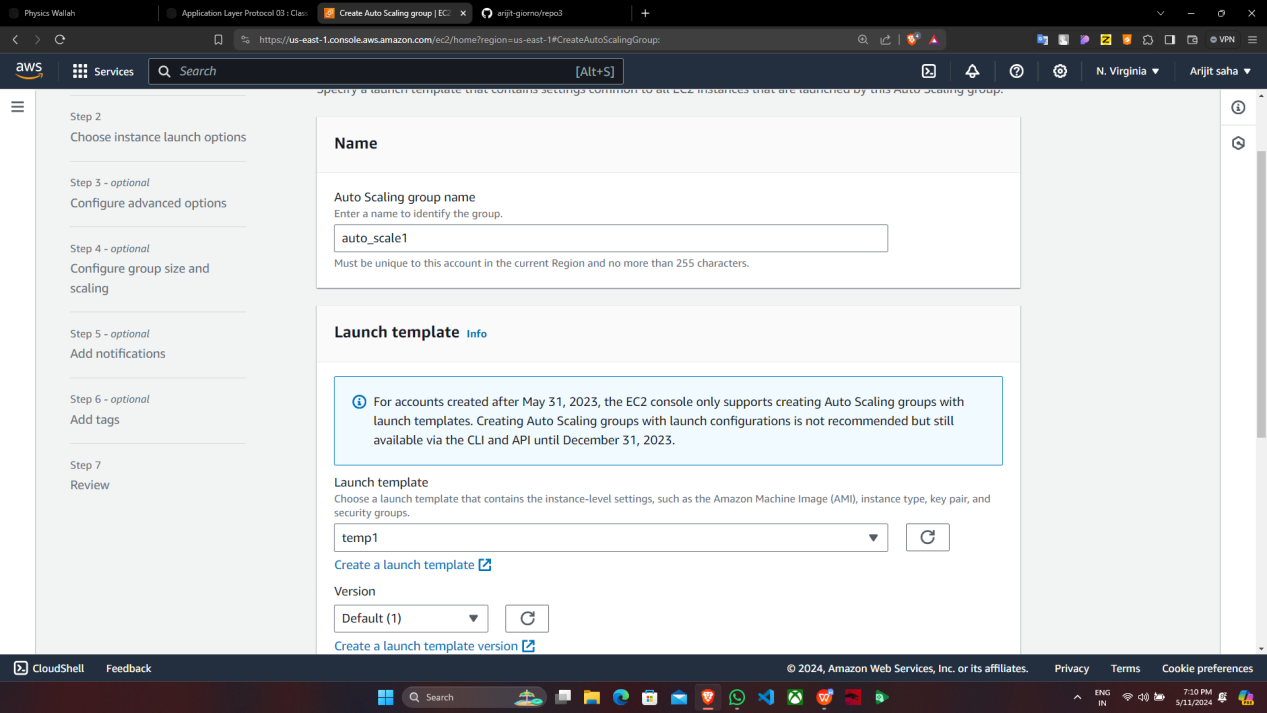
6. Template has successfully created & now click on “Auto Scaling Groups”.



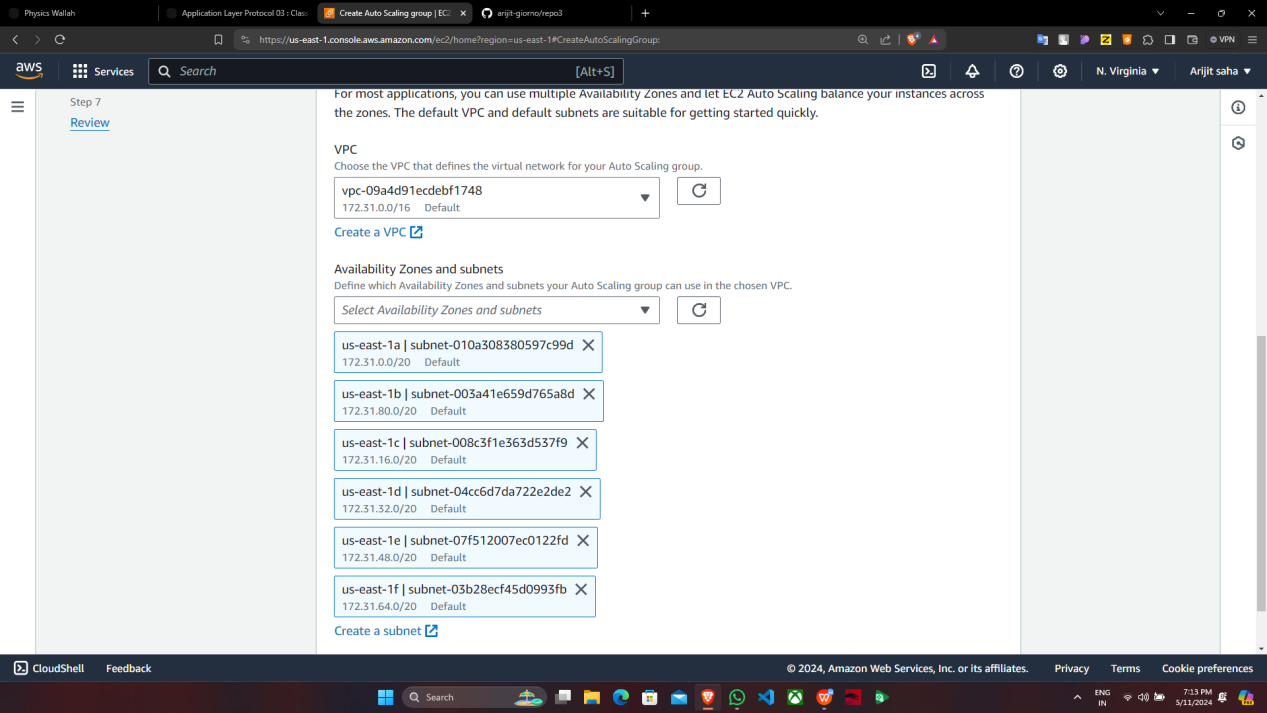
7. Click on “Create Auto Scaling group”.

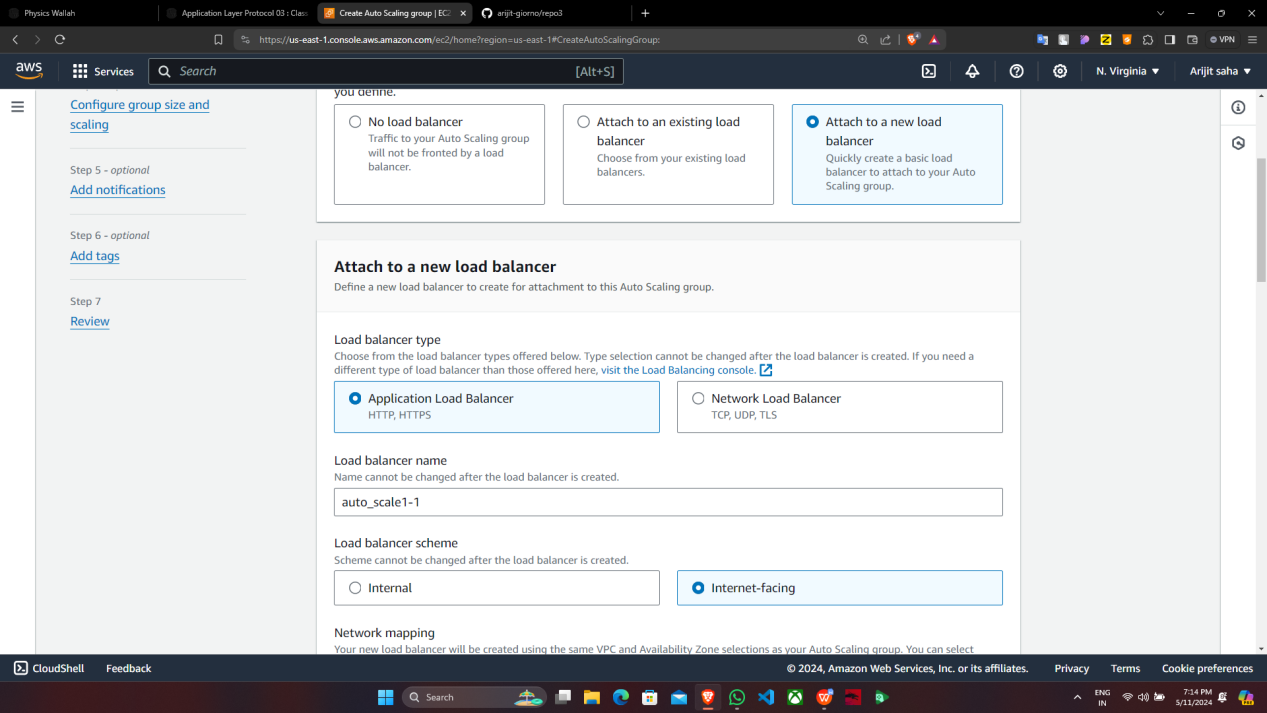


1. Under “Create Auto Scaling group”, give the name & choose the template that you have created then click on “Next”.

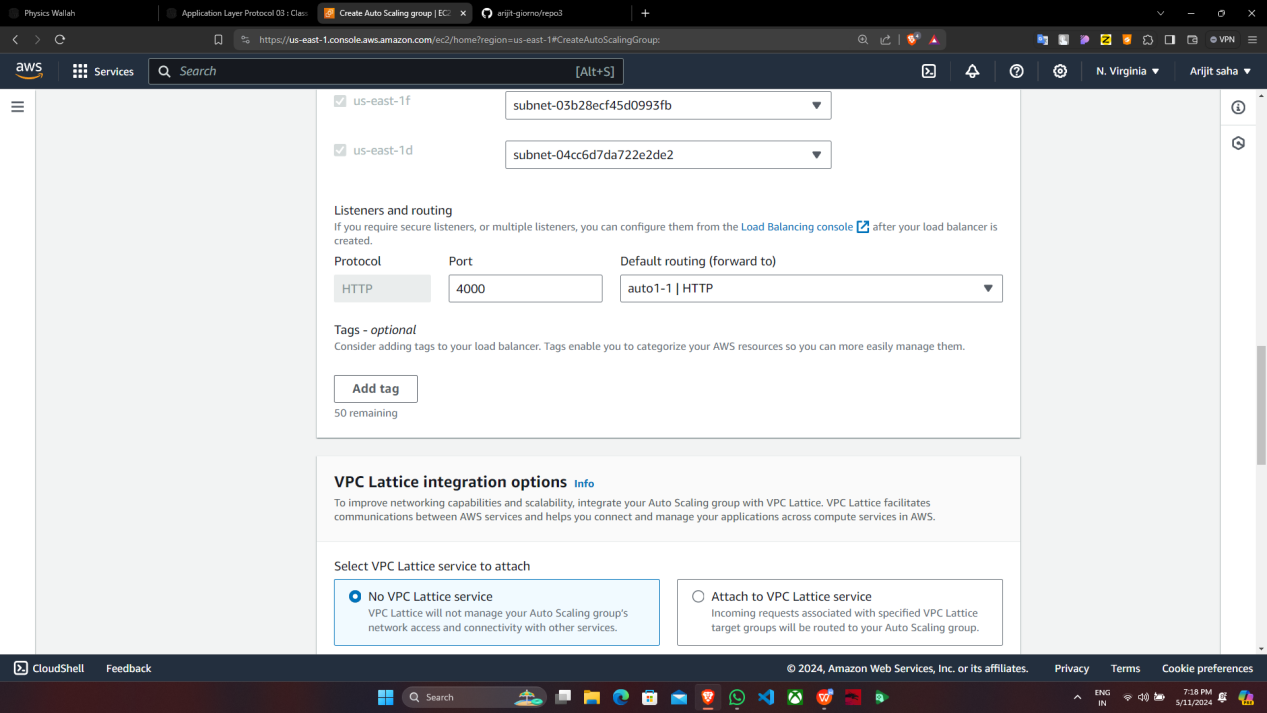


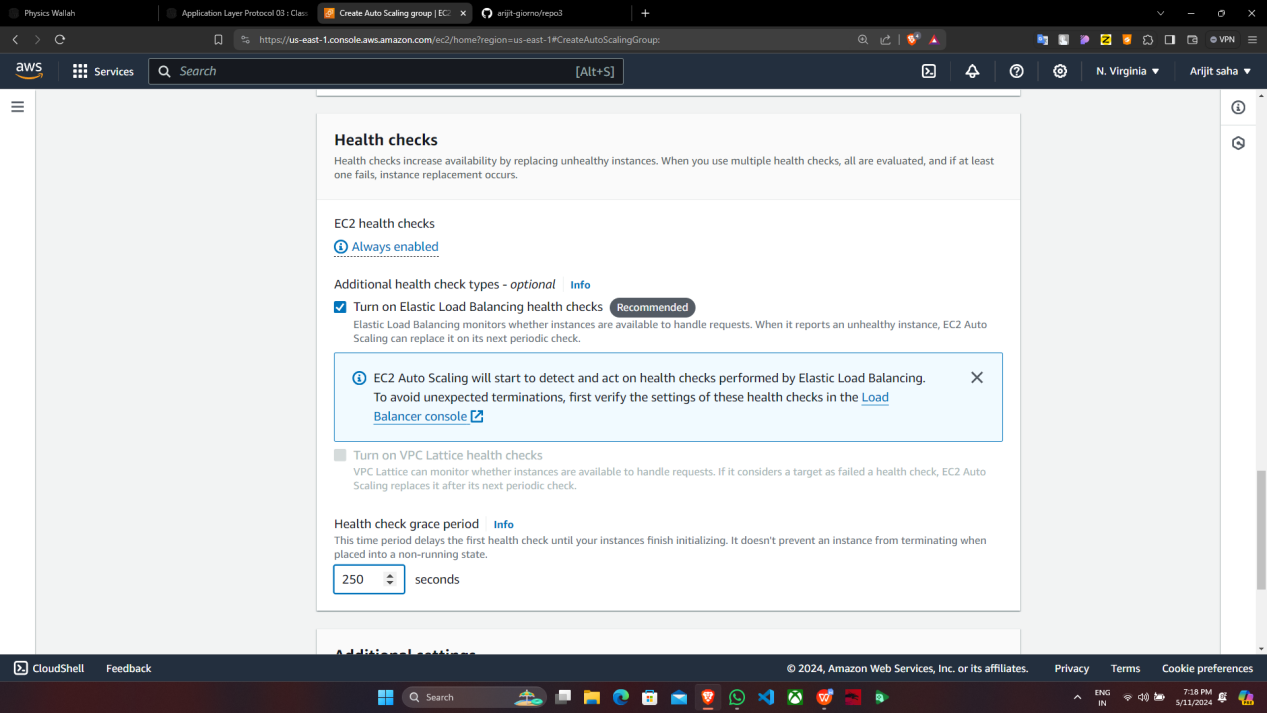
Then select all the “Availability Zones and subnets” & click on “Next”.



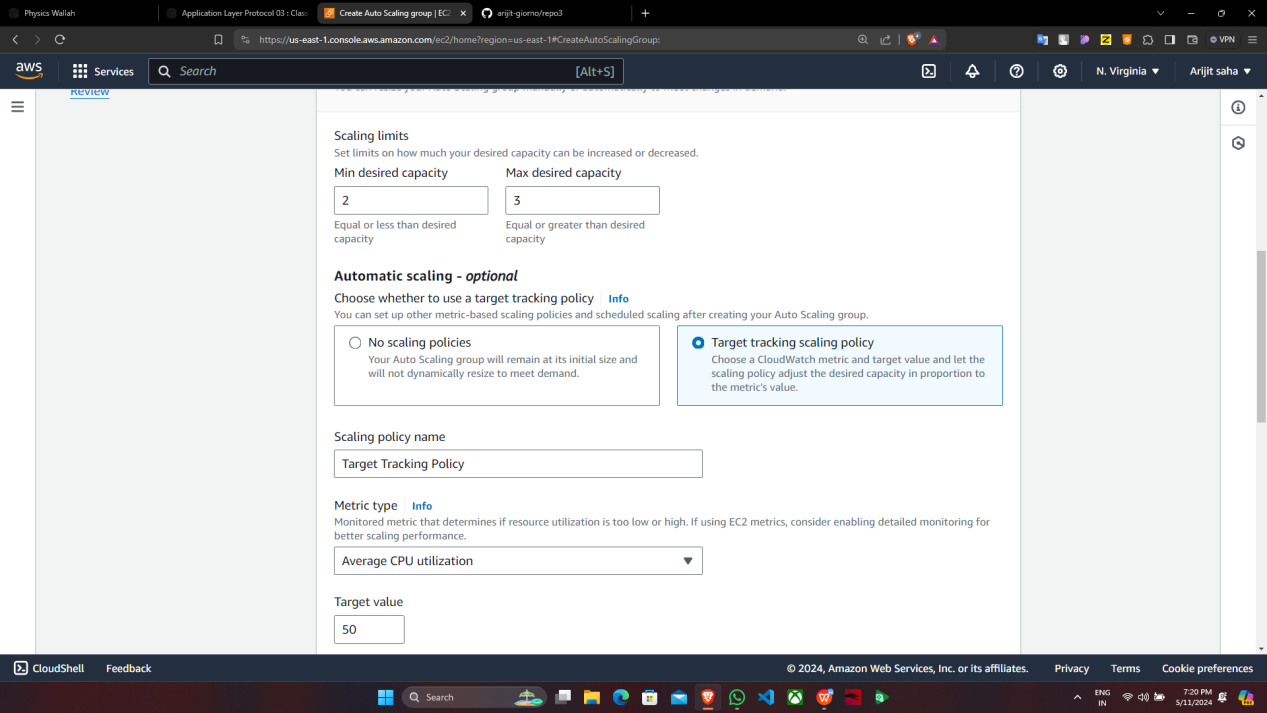


Click on “No VPC Lattice service”.In “Health checks”, click on the checkbox

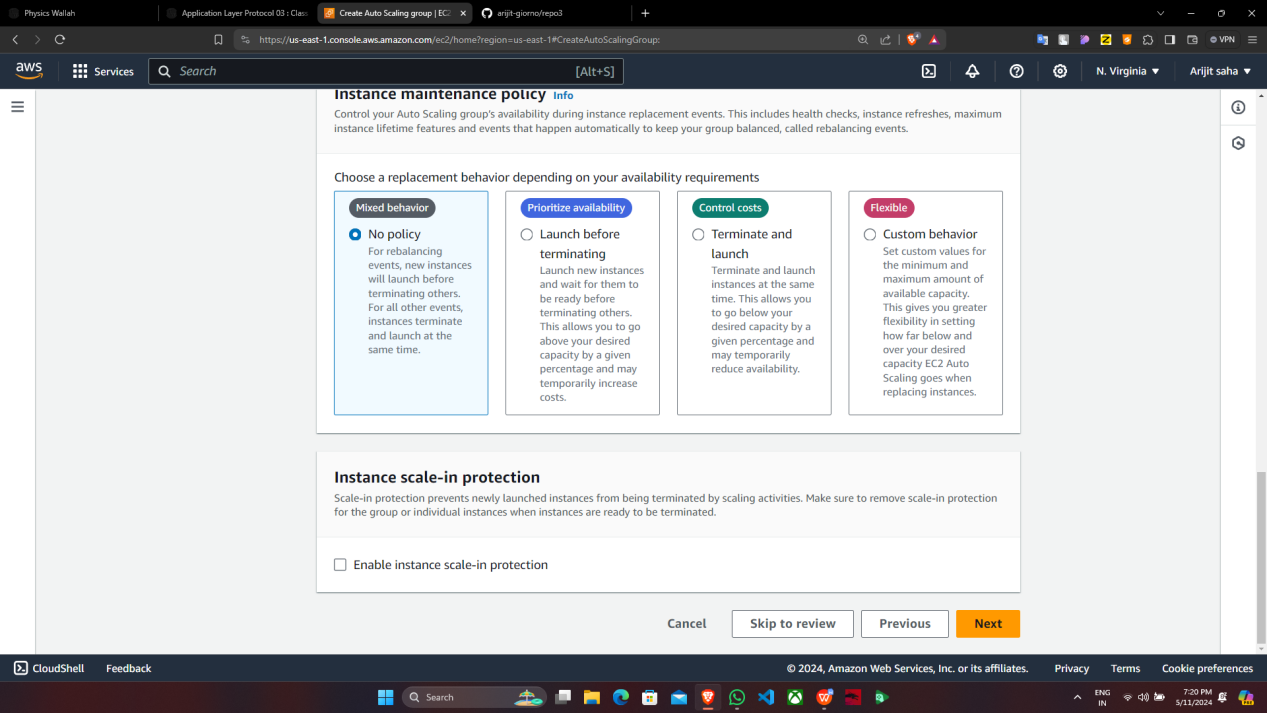


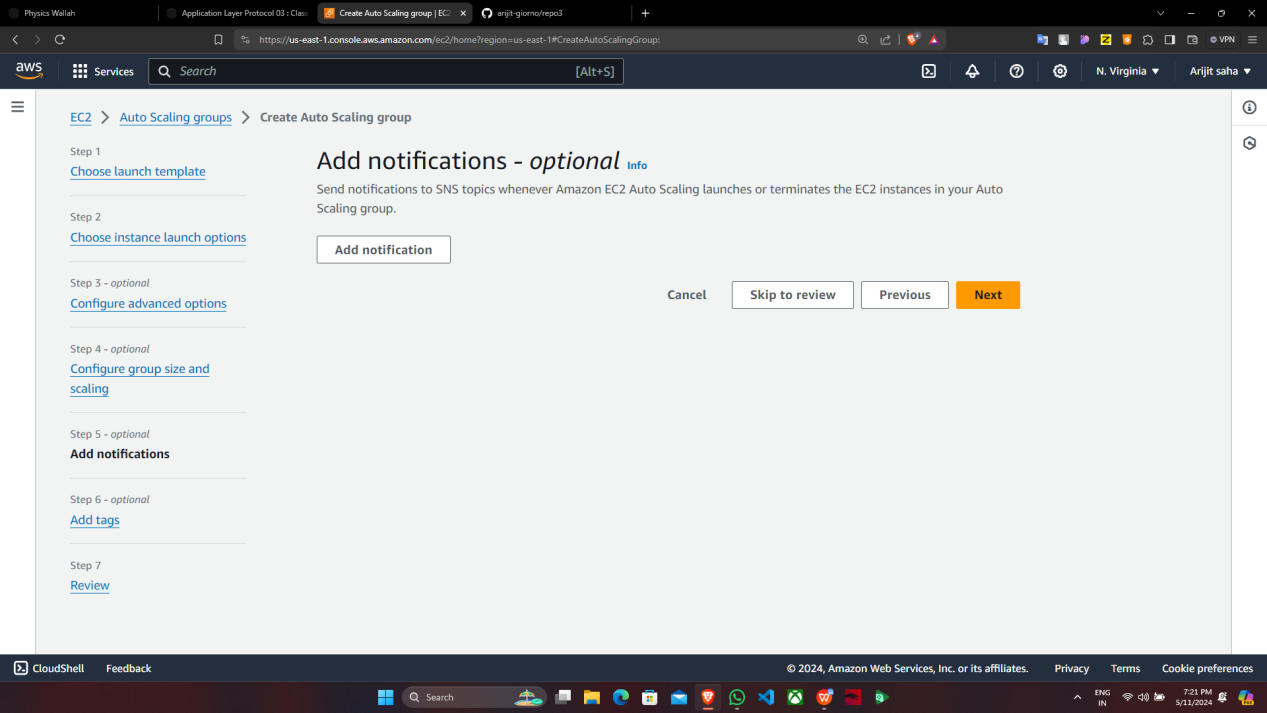


“Turn on Elastic load …” Then click on “Next”.

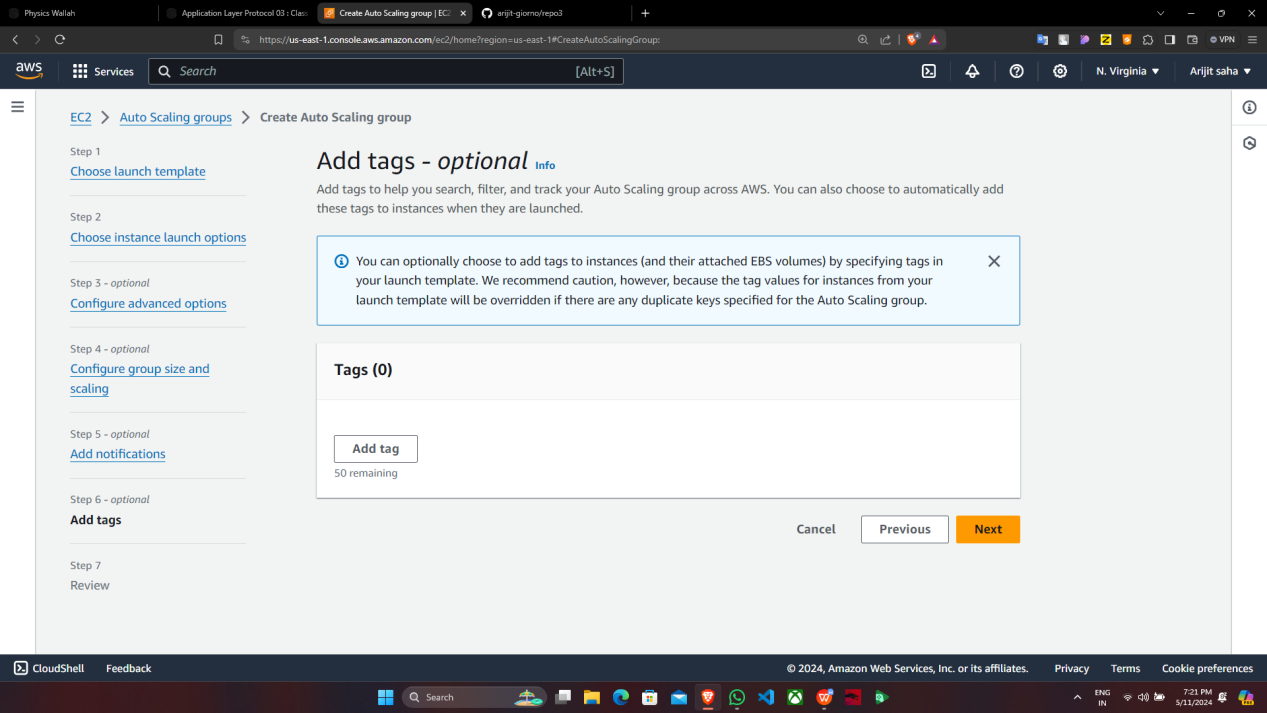


Click on “No policy” then on “Next”.Click on “Next”.

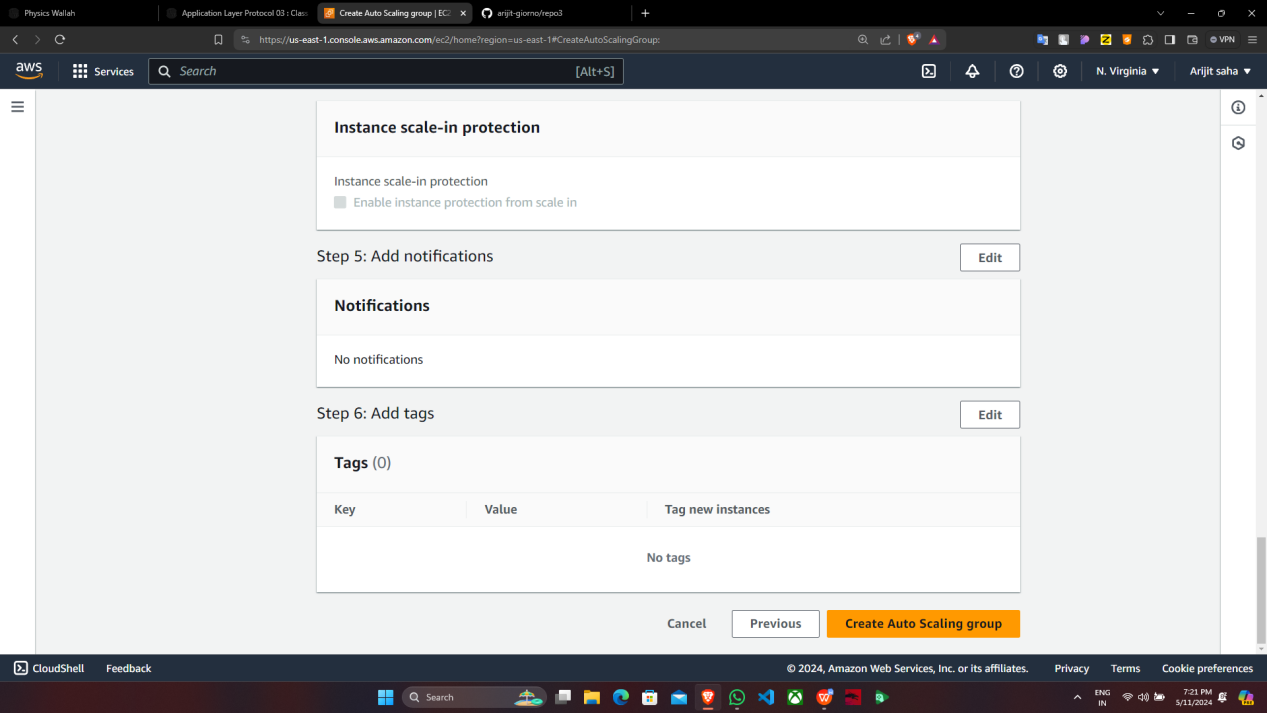




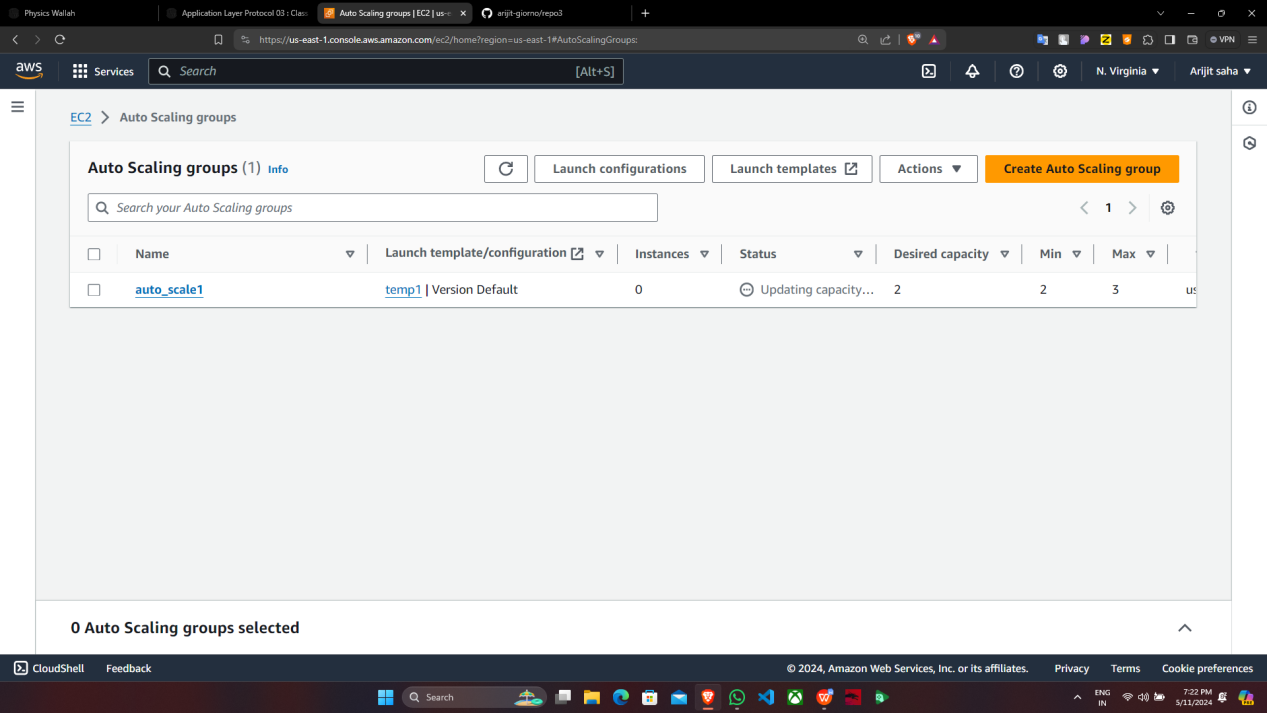
Click on “Next”.



Click on “Create Auto Scaling group”.

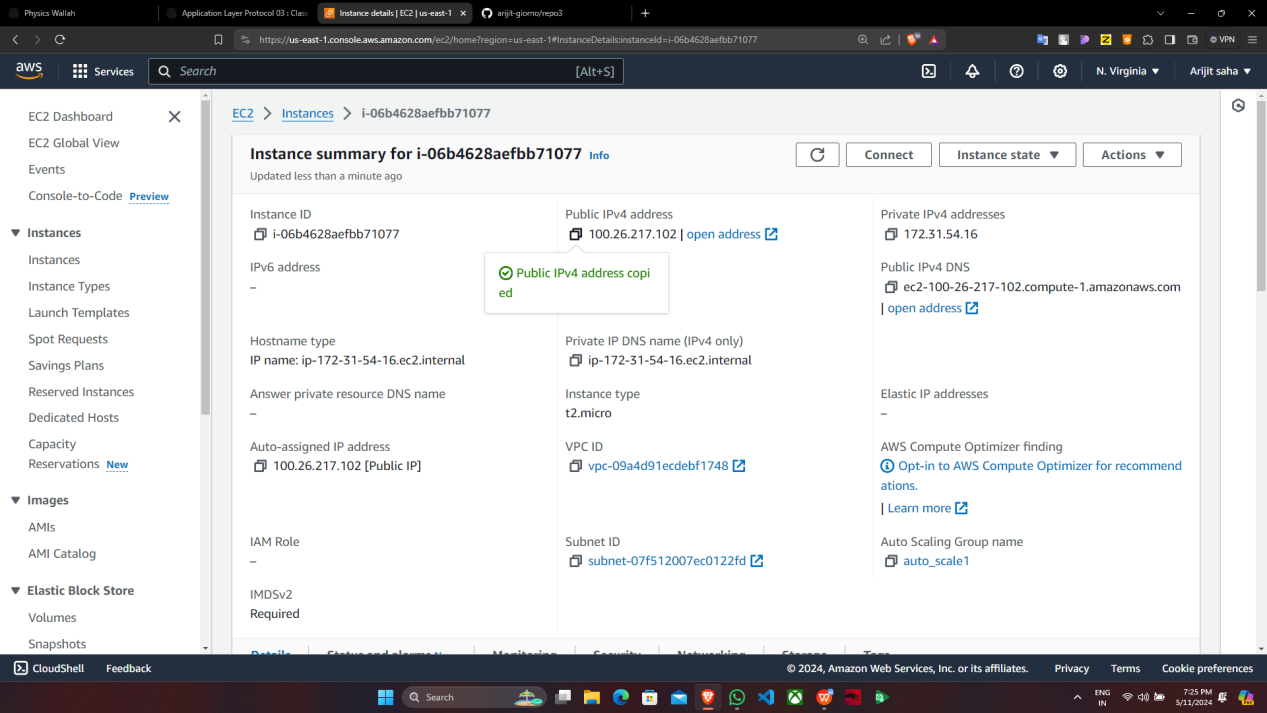


9. Auto scaling group is successfully created.

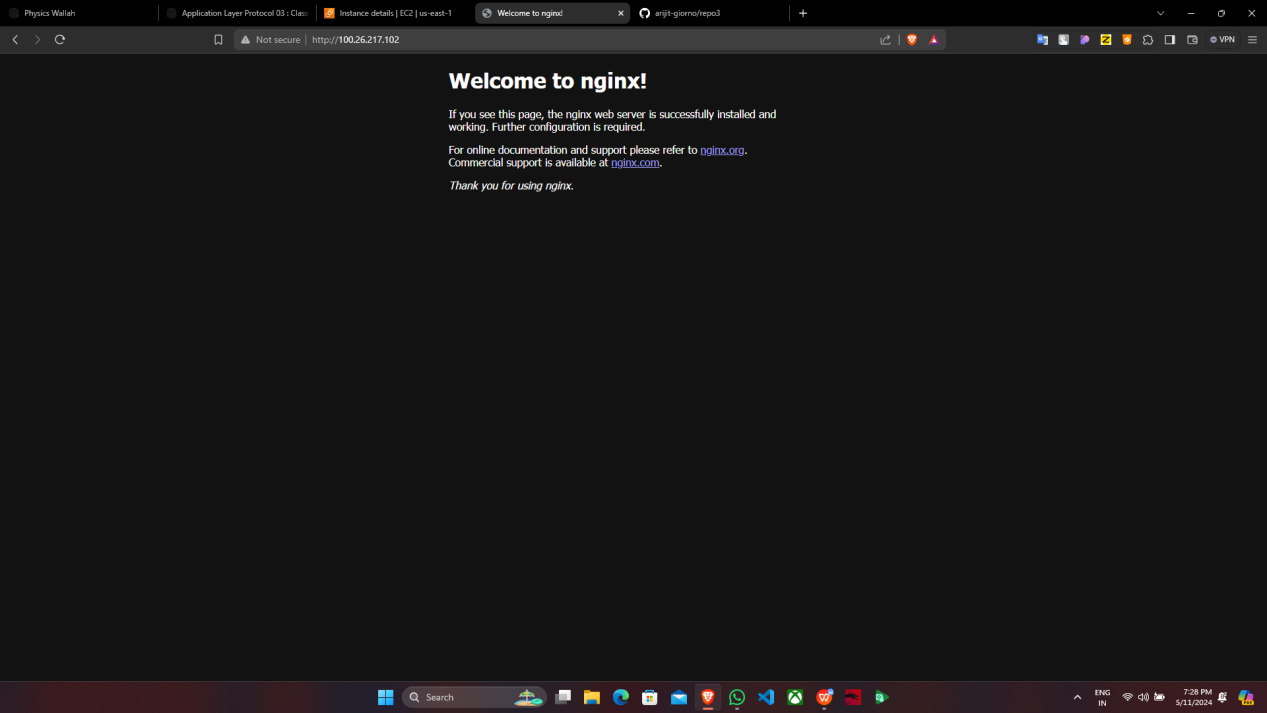


10. Now go to “Instance” and check for running instances with no name but then click on

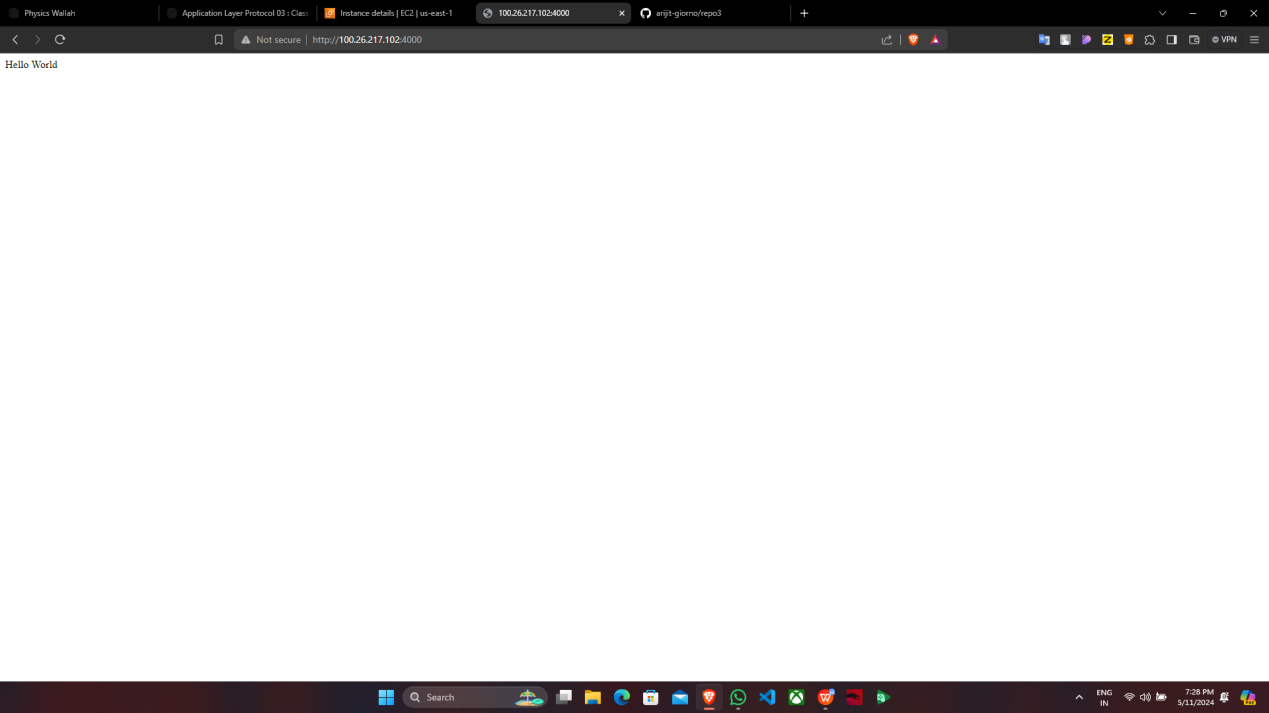
any one of the instance ID & copy the “Public IPv4 address”.



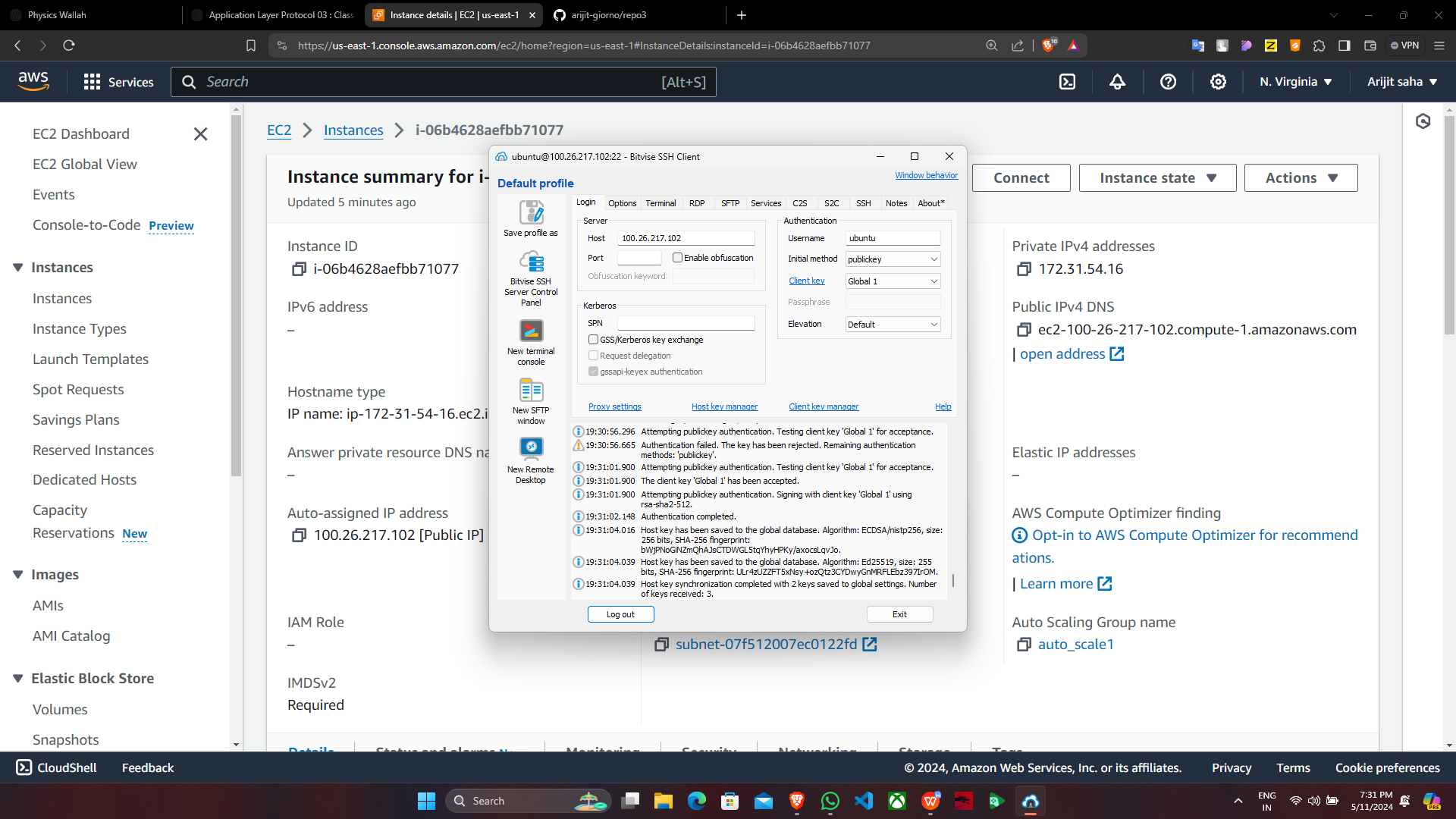
11. Paste the address in a new Window.



12. Now add “:4000” at the end of the IPv4 address and press enter.



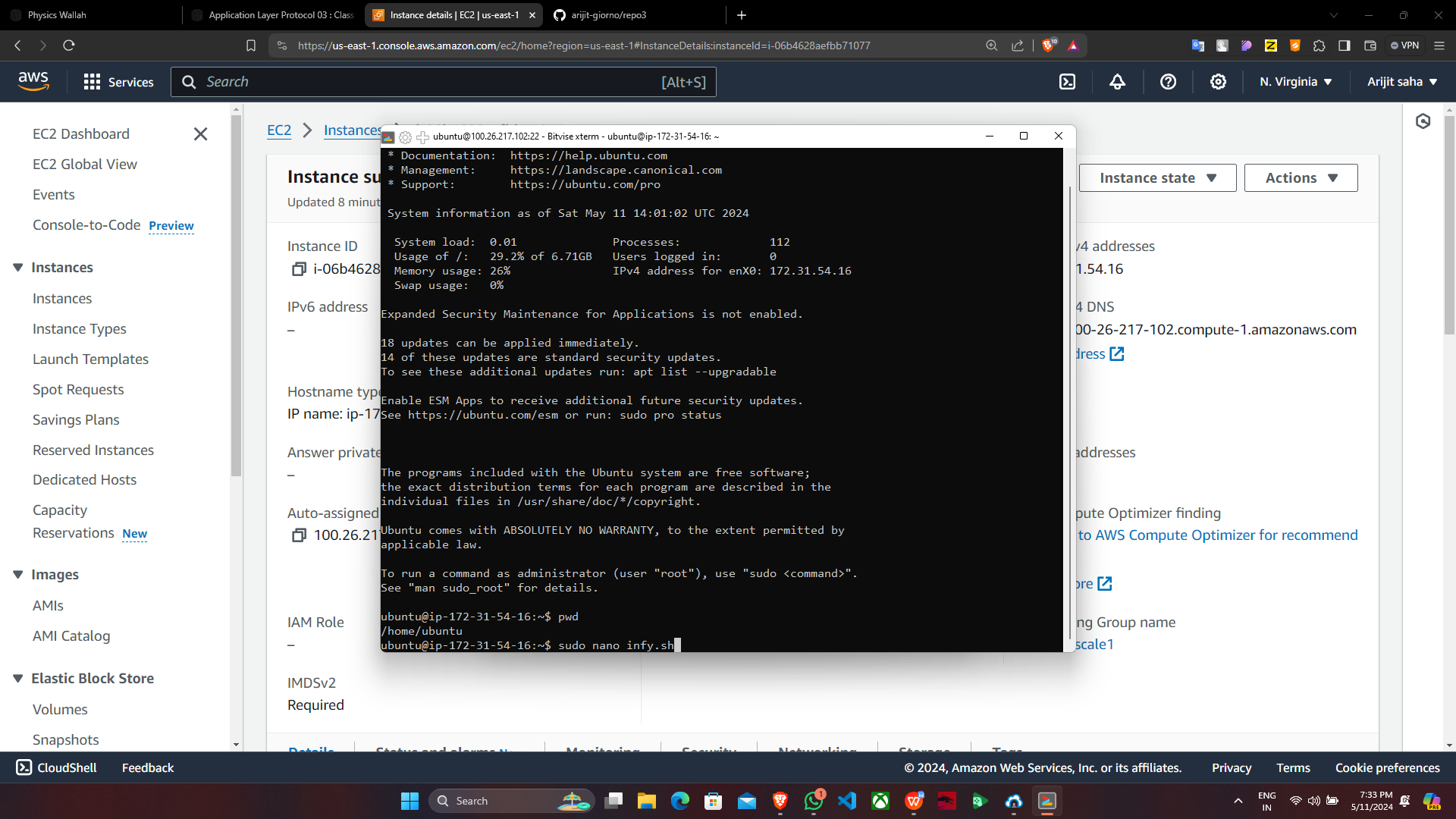
13. Then copy the address of any one of the instance & paste in the host of the “Bitvise SSH Client” then click on “Client key Manager”.



1. Then import the key & click on “Log in”.

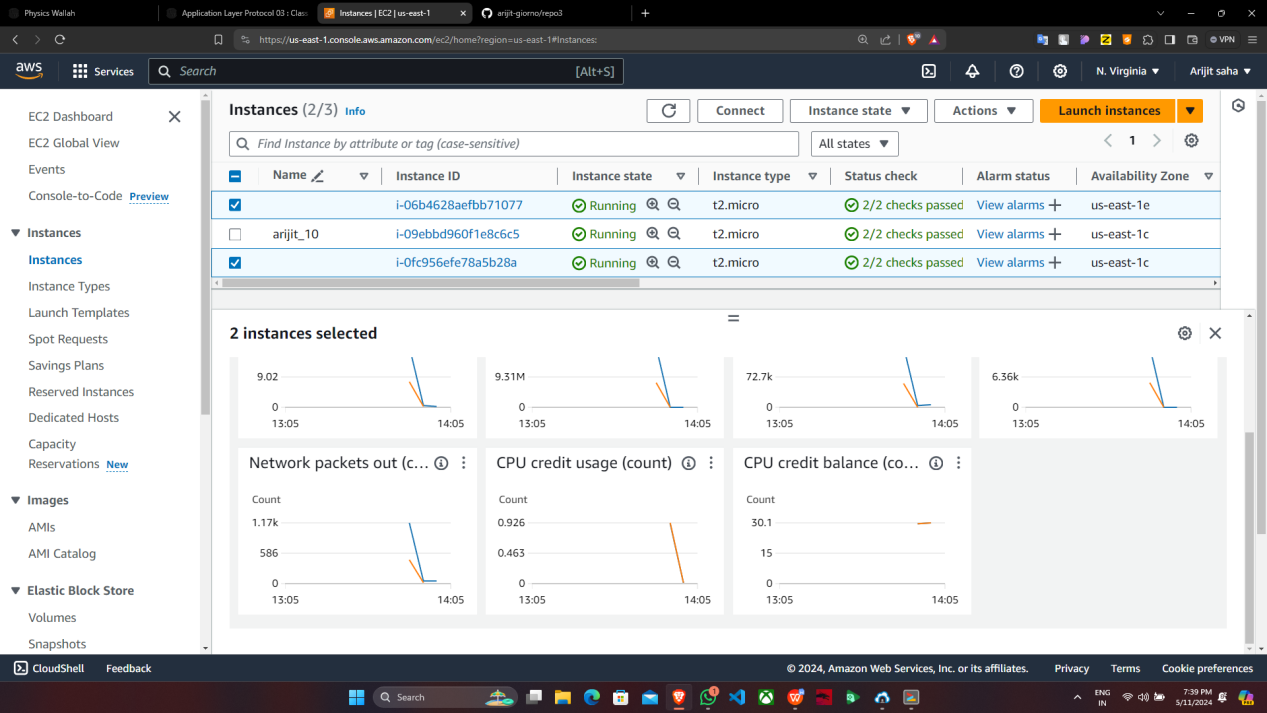
15. The “Log out” came means that is is successfully logged in.

16.Now in “New terminal console and type the following commands.

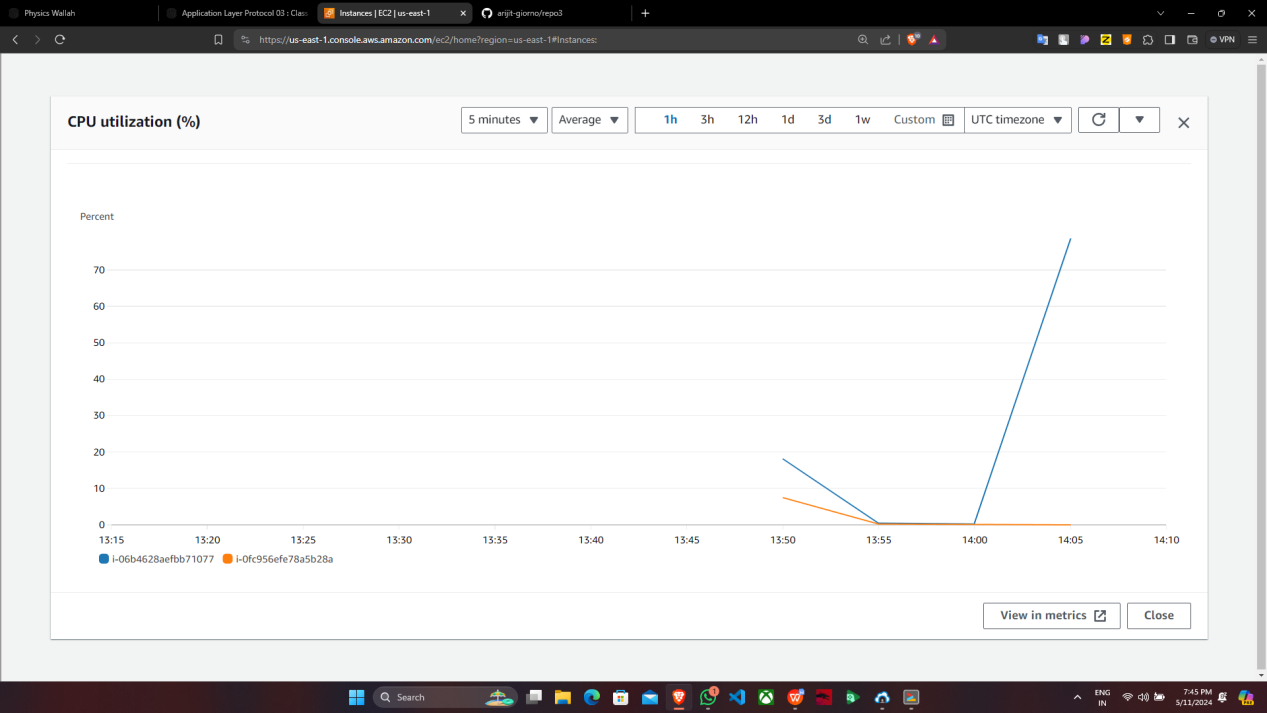




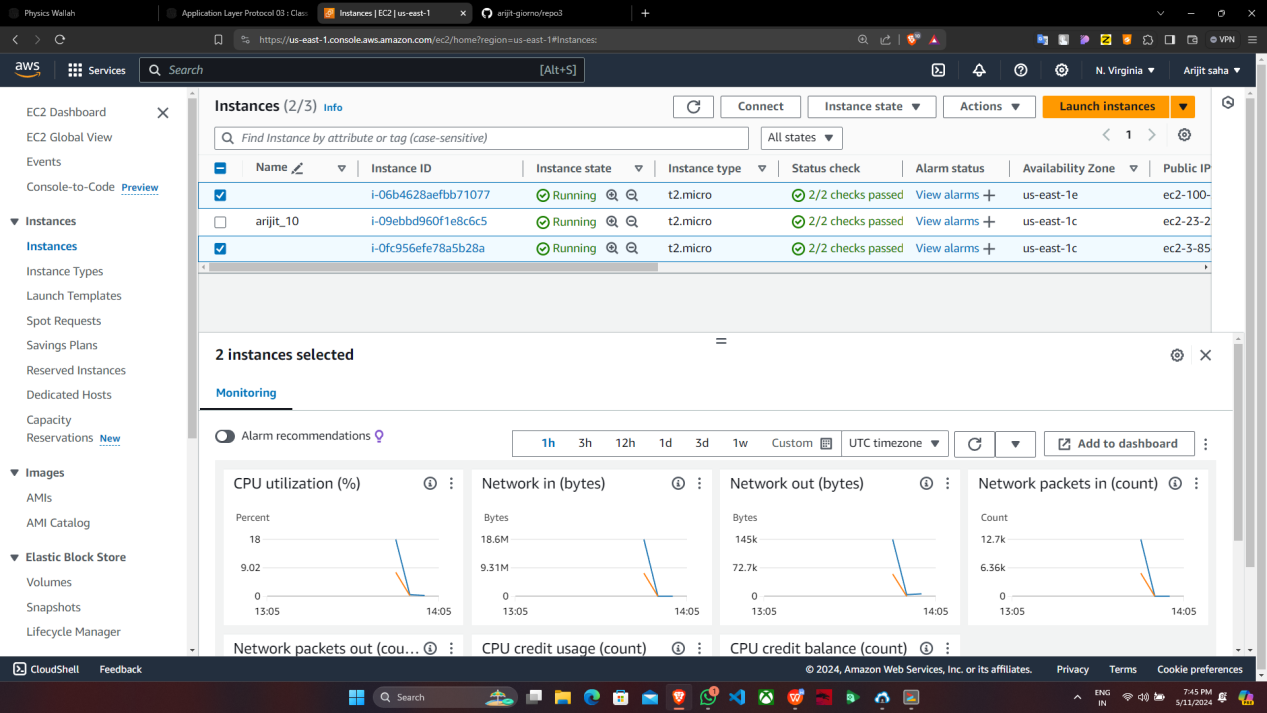
17. Now In Instances select both the unnamed instances & click on ‘Enlarge’.



18. Here, we see the graph of CPU utilization(%) & select “Local timezone”.



19. Here, we see that two instances are running along with the initialization of the third instance.



**INFERENCES:**

1. A launch template is created with a t2.micro instance type, security group, and GitHub initialization commands.

2. An auto-scaling group is set up using the template, covering multiple availability zones and enabling health checks and load balancing.

3. Instances are monitored for CPU utilization, triggering automatic scaling based on load demands.

4. Instances are accessed and verified using Bitvise SSH Client by importing the key pair.

5. Load is balanced across multiple instances, optimizing resource usage and maintaining application performance.