**DevOps project to deploy a running application in AWS ECS service.**

1. Run the Node application in local PC
2. First install the nodejs in ec2 by using

$ curl --silent --location https://rpm.nodesource.com/setup\_6.x | sudo bash

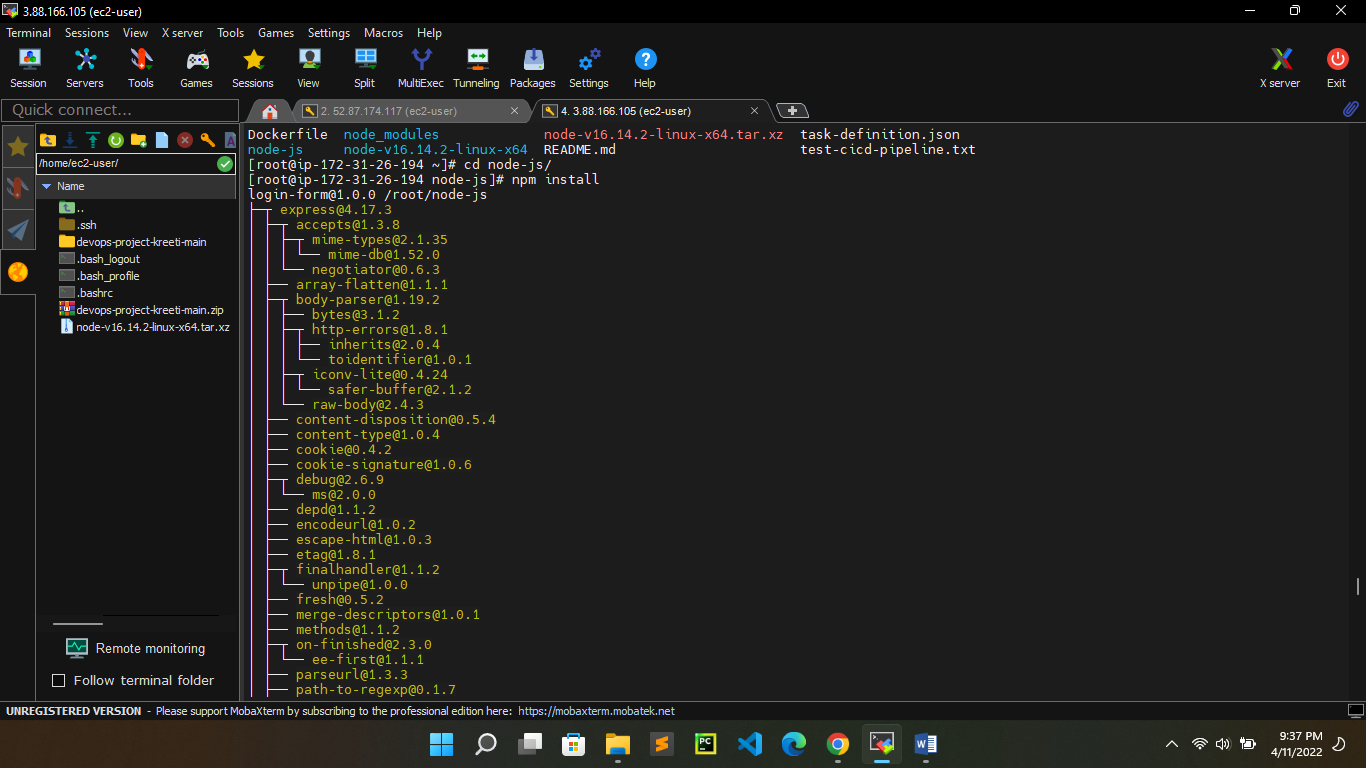
$ yum install –y nodejs

1. Check npm version

$npm –v

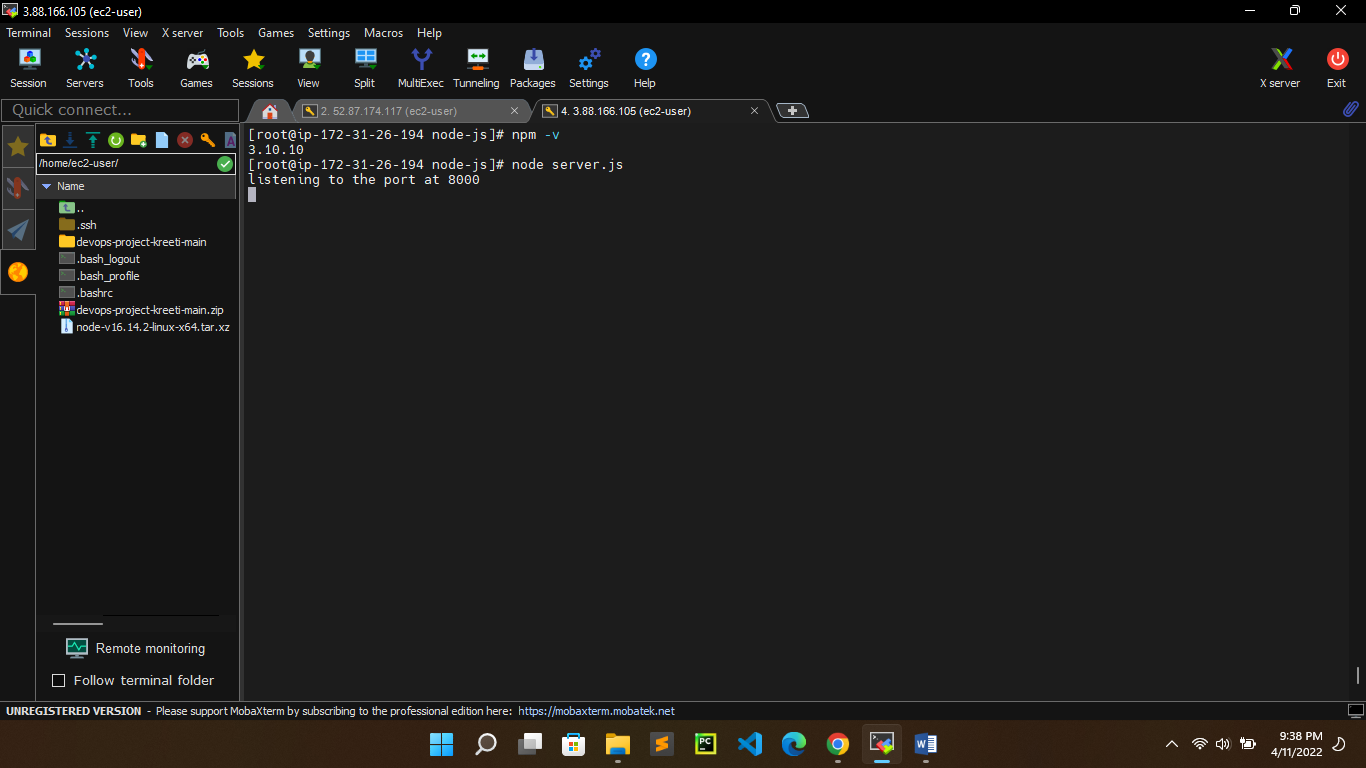
1. Install npm modules

$npm install

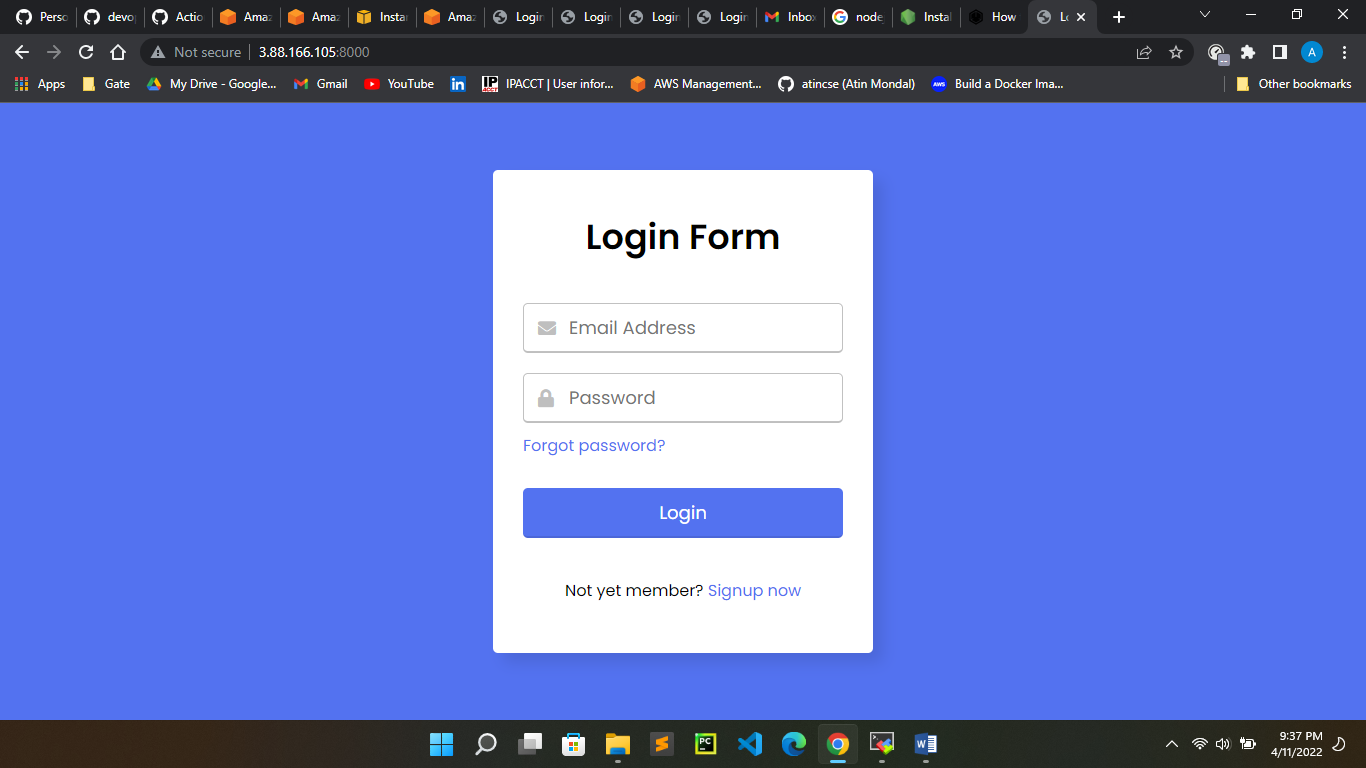


1. Start the server

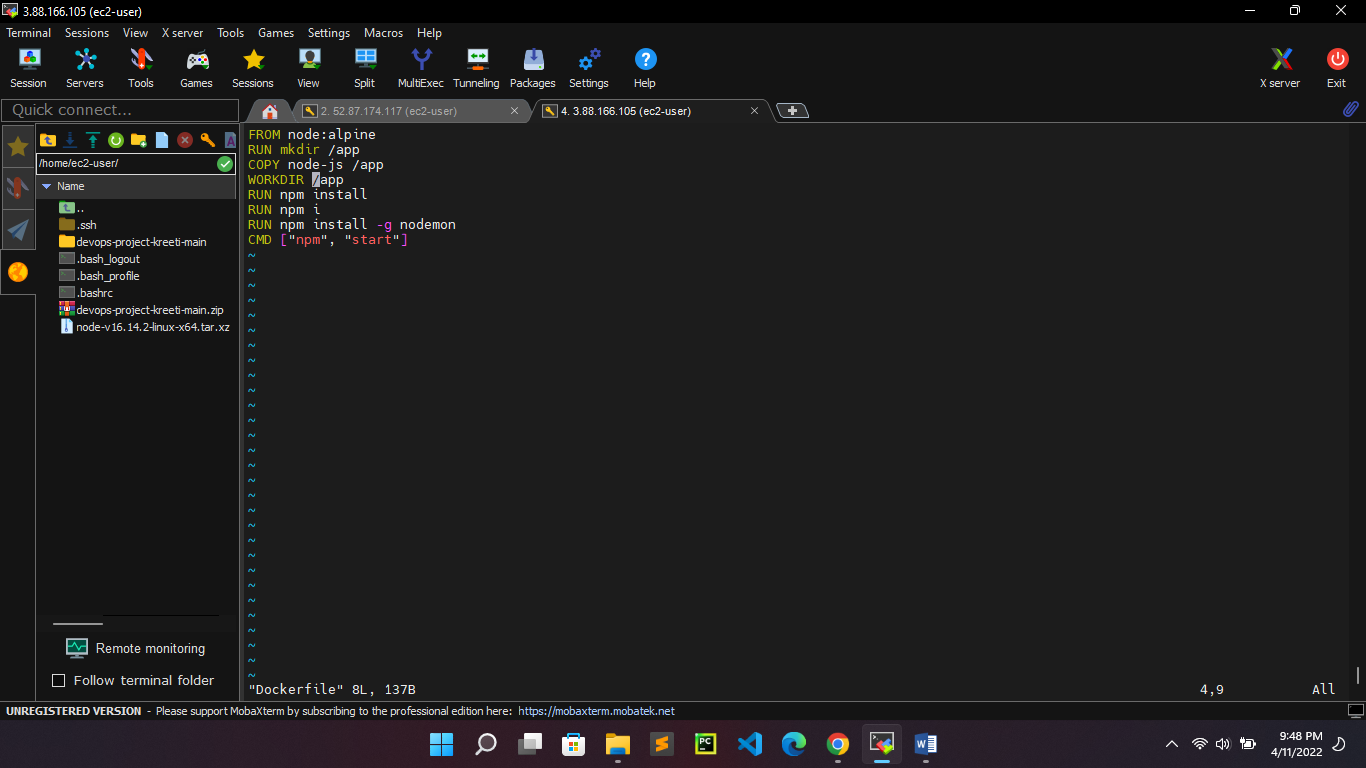
$node server.js



Application image in the local browser (ec2 public ip - 3.88.166.105:8000)

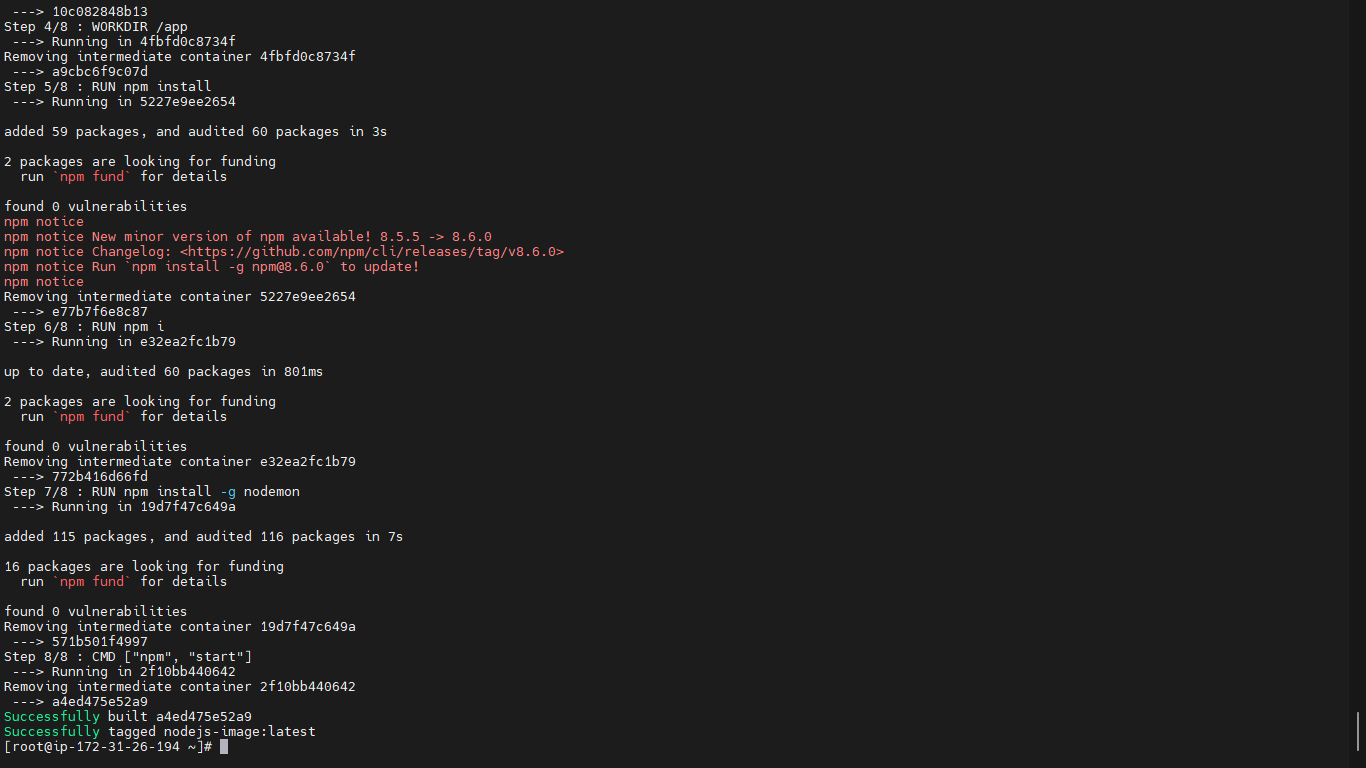


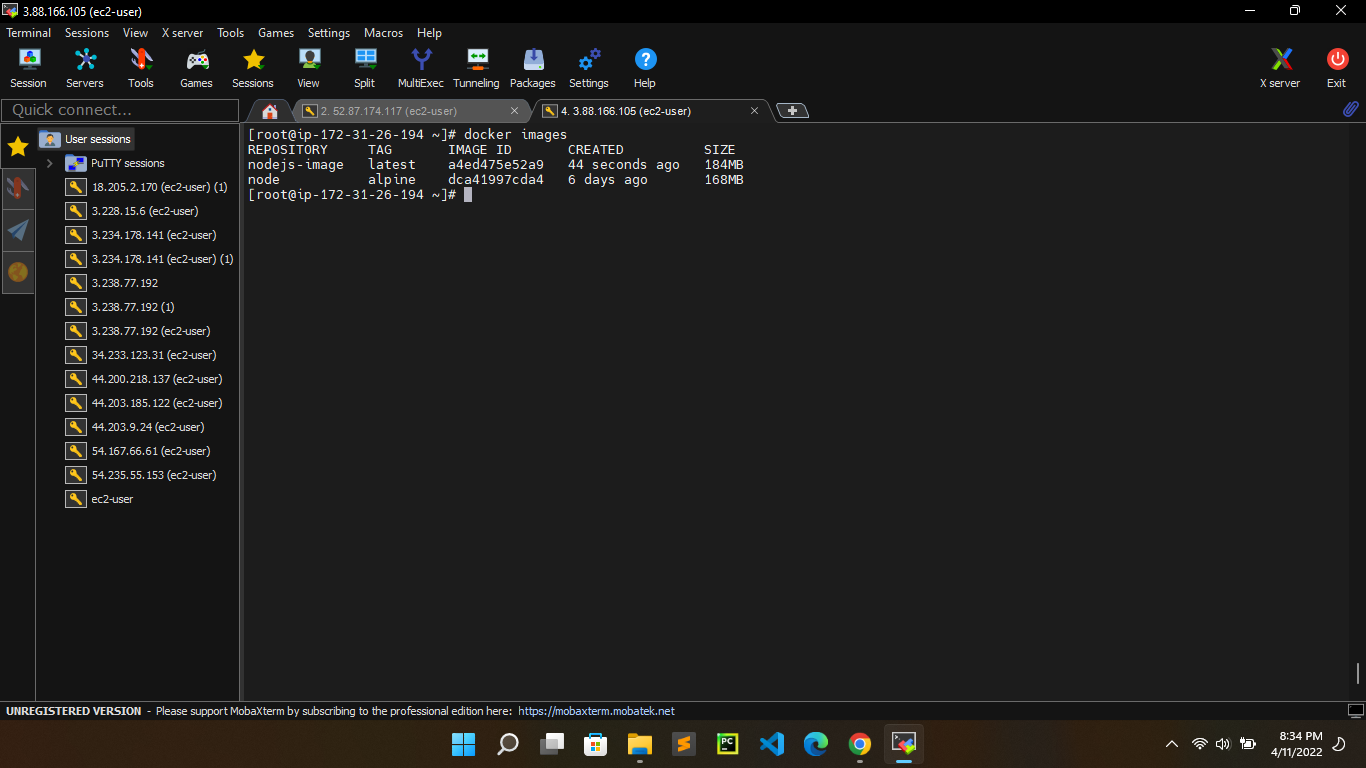
1. Create Docker environment
2. Create Docker file $ vi Dockerfile



1. Build docker image for our nodejs application

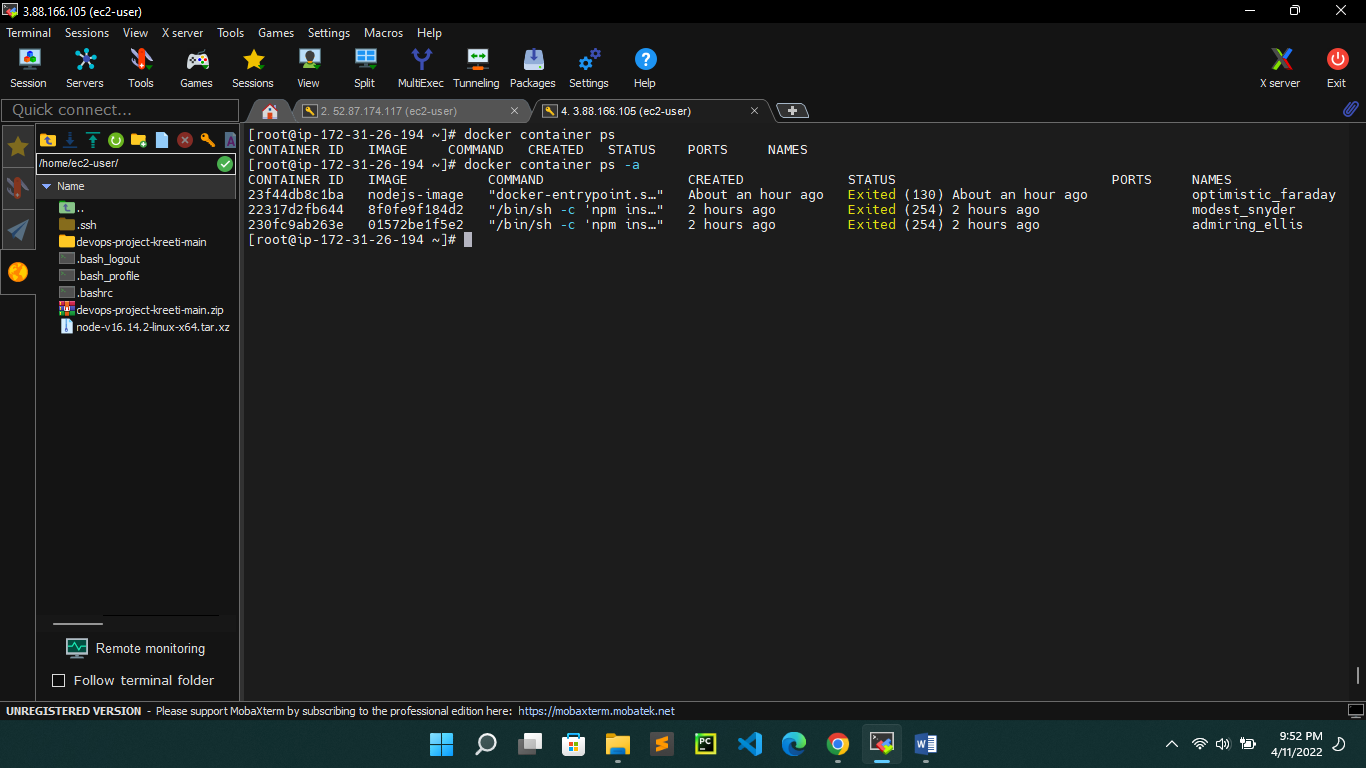
$docker build -t nodejs-image .

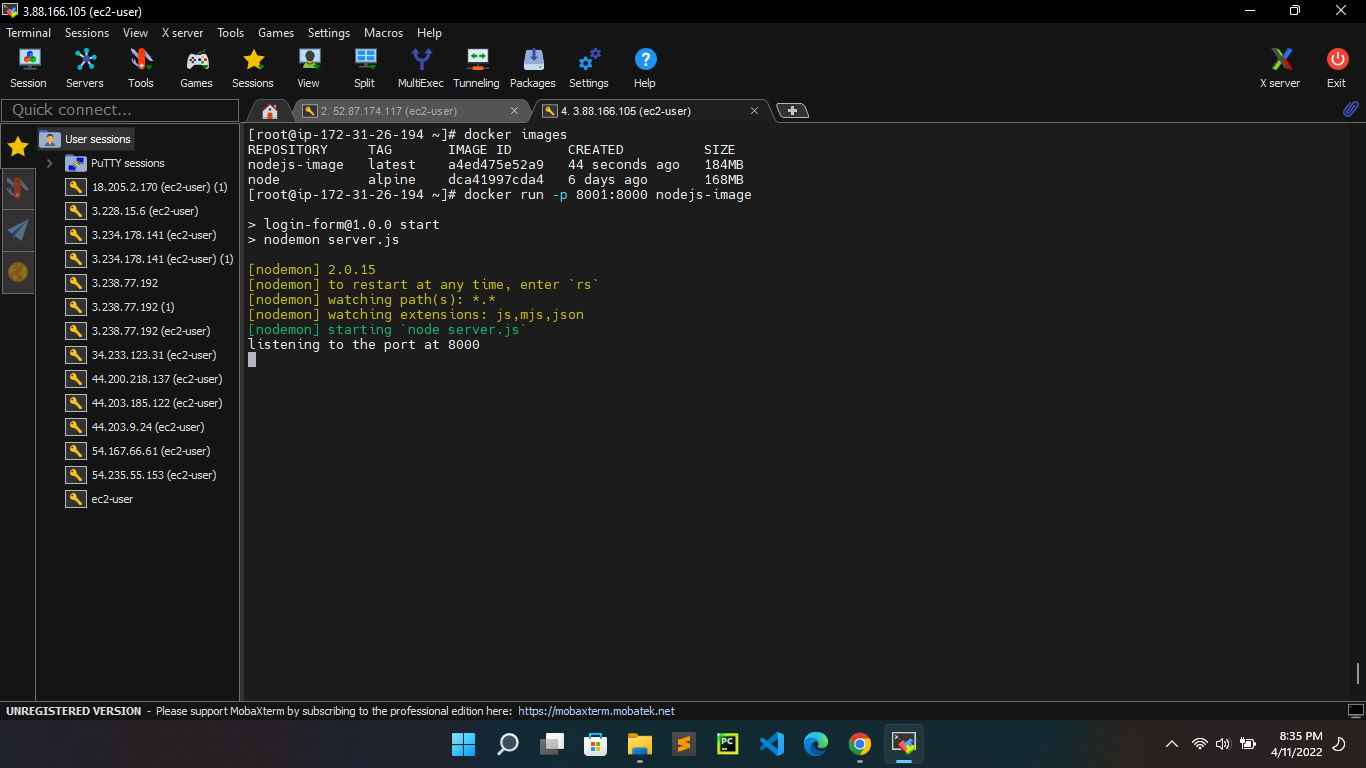




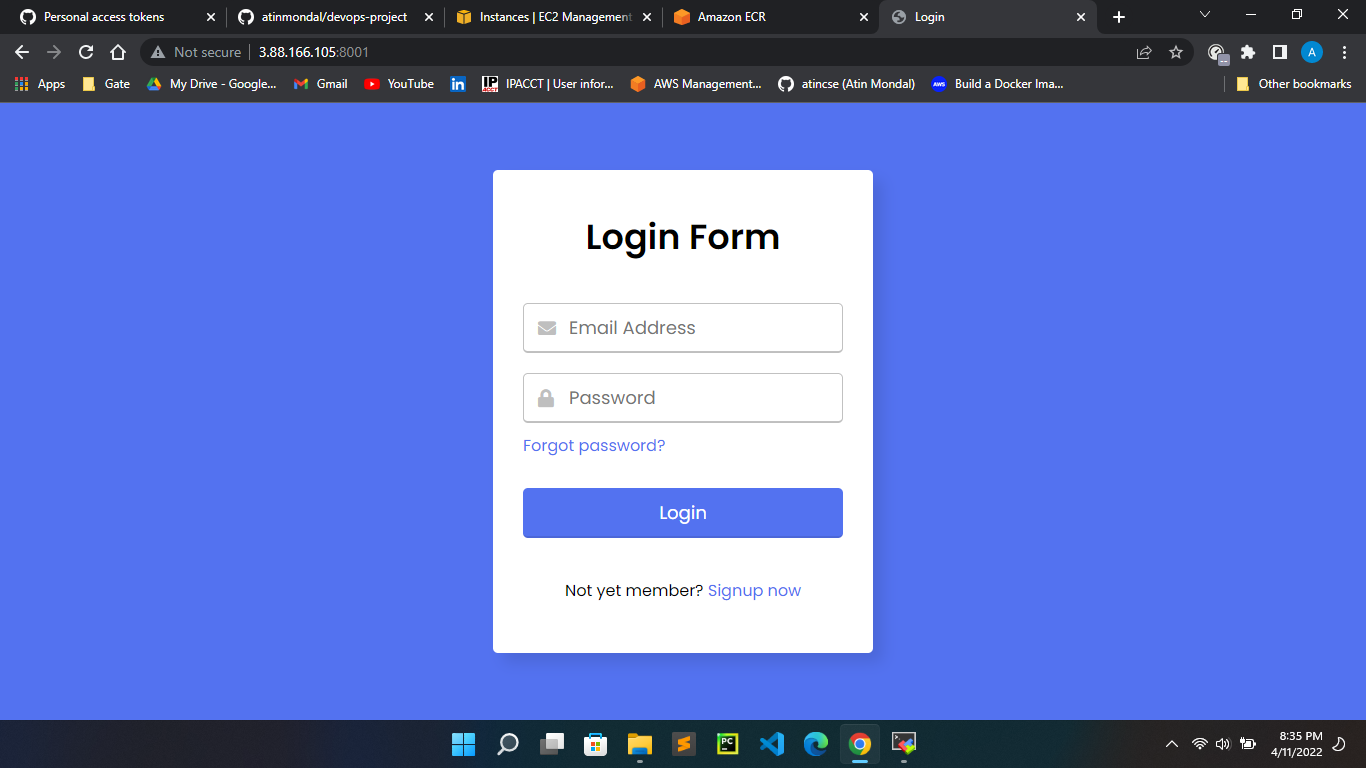
1. Run docker container and see the o/p in the browser

$ docker run -p 8001:8000 nodejs-image



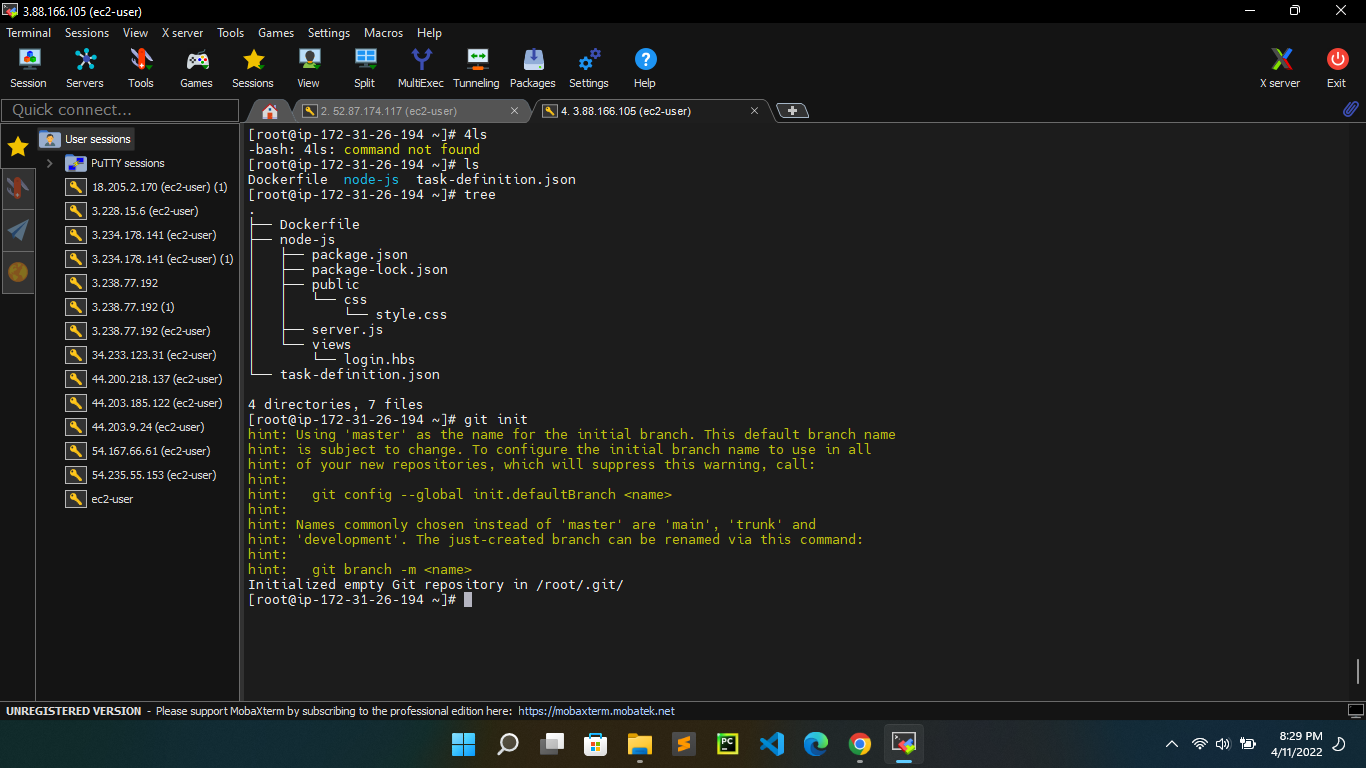


Application image in the browser (ec2 public ip - 3.88.166.105:8001)



1. Push the Node application project to Github repository.
2. Initialize git repository and push our project to github

$git init

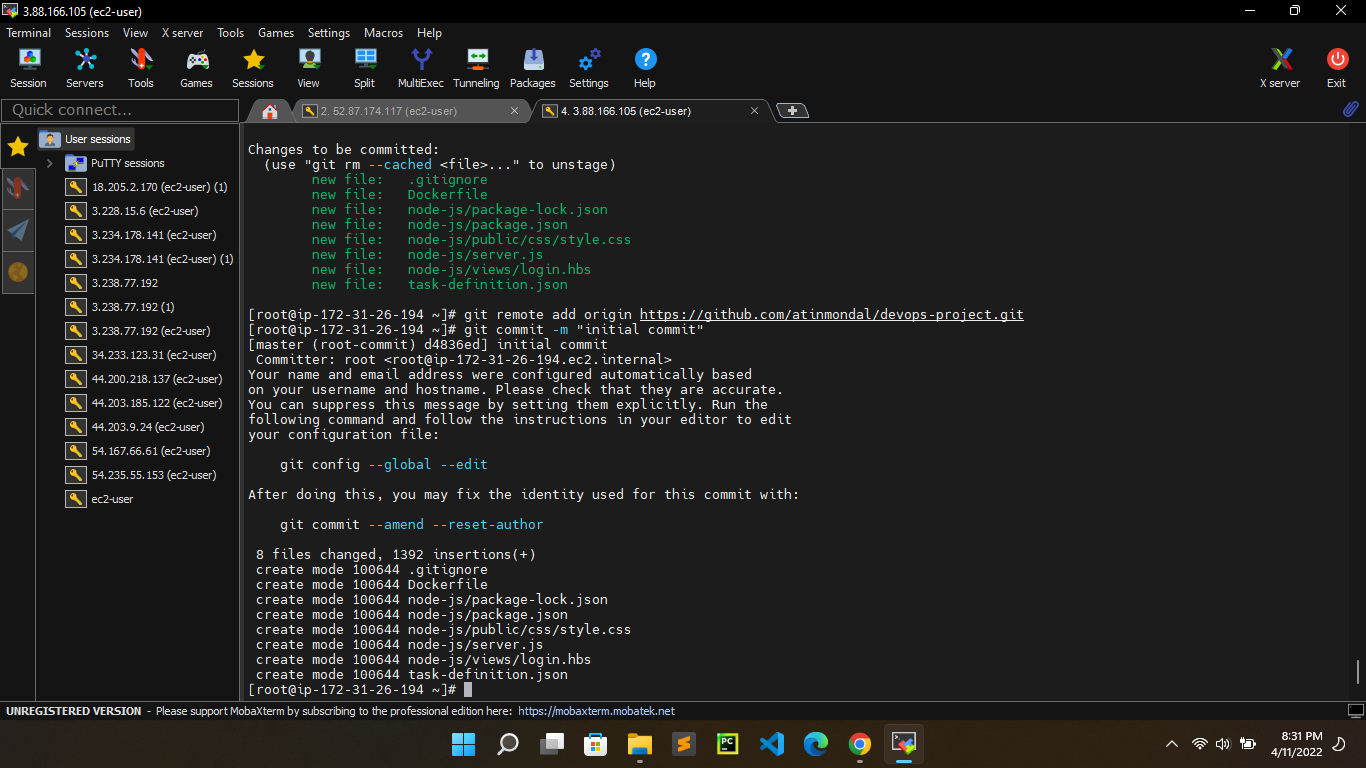


$ git add .

$ git status

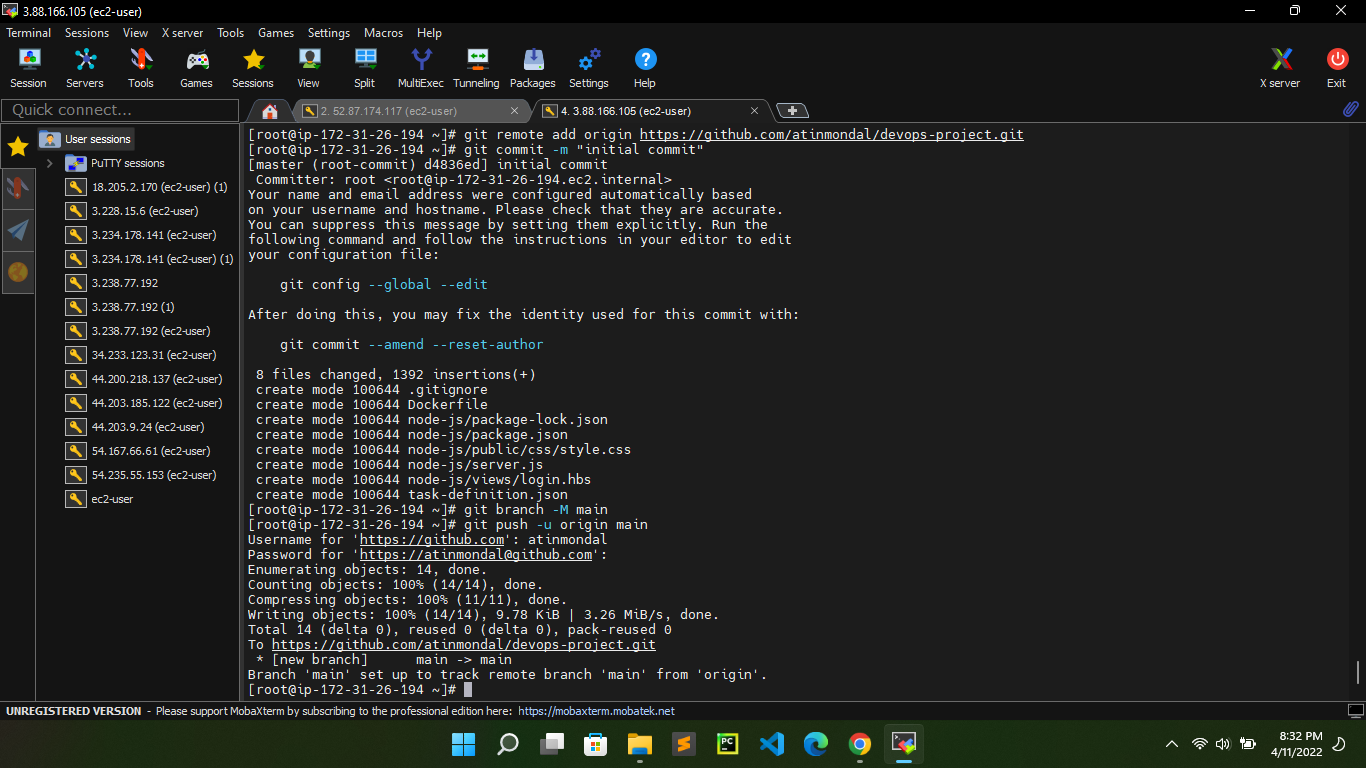
$ git remote add origin <https://github.com/atinmondal/devops-project.git>

$ git commit –m “Initial commit”

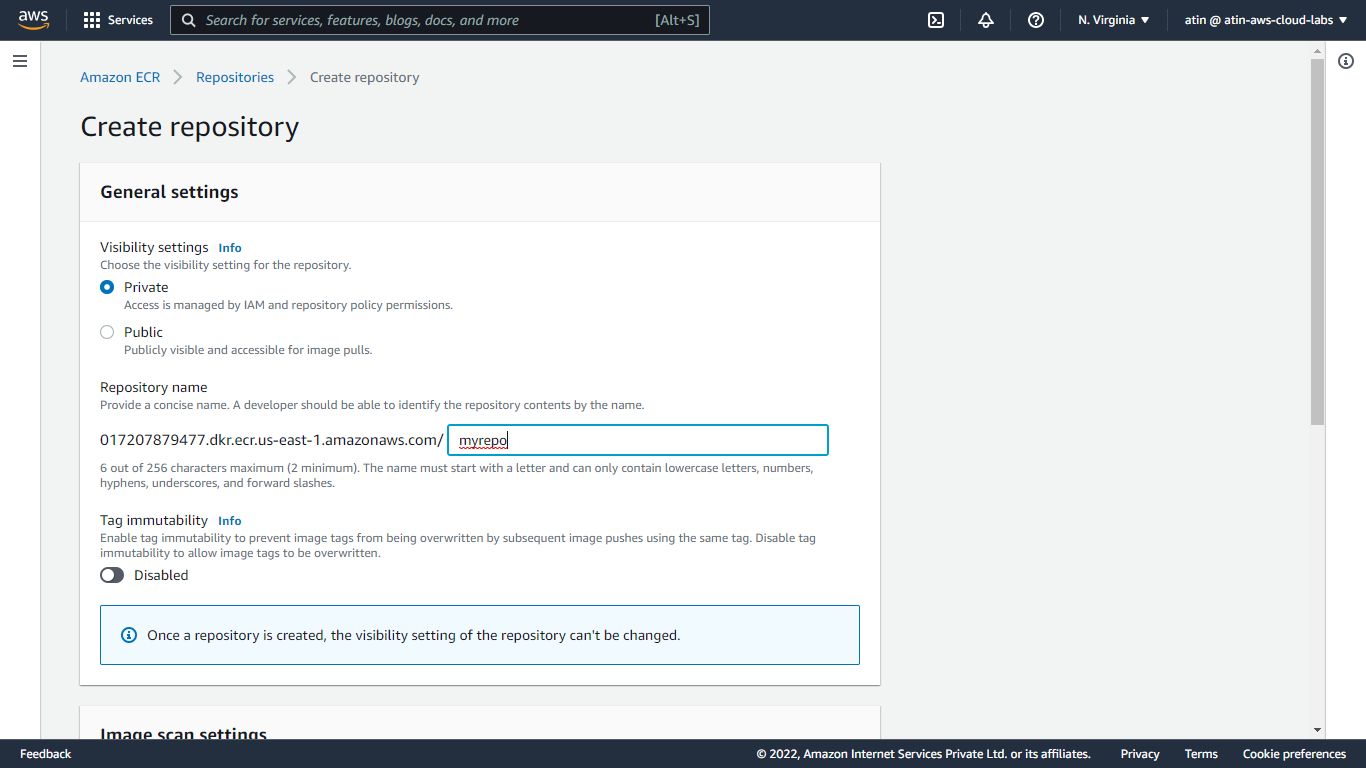


$ git branch -M main

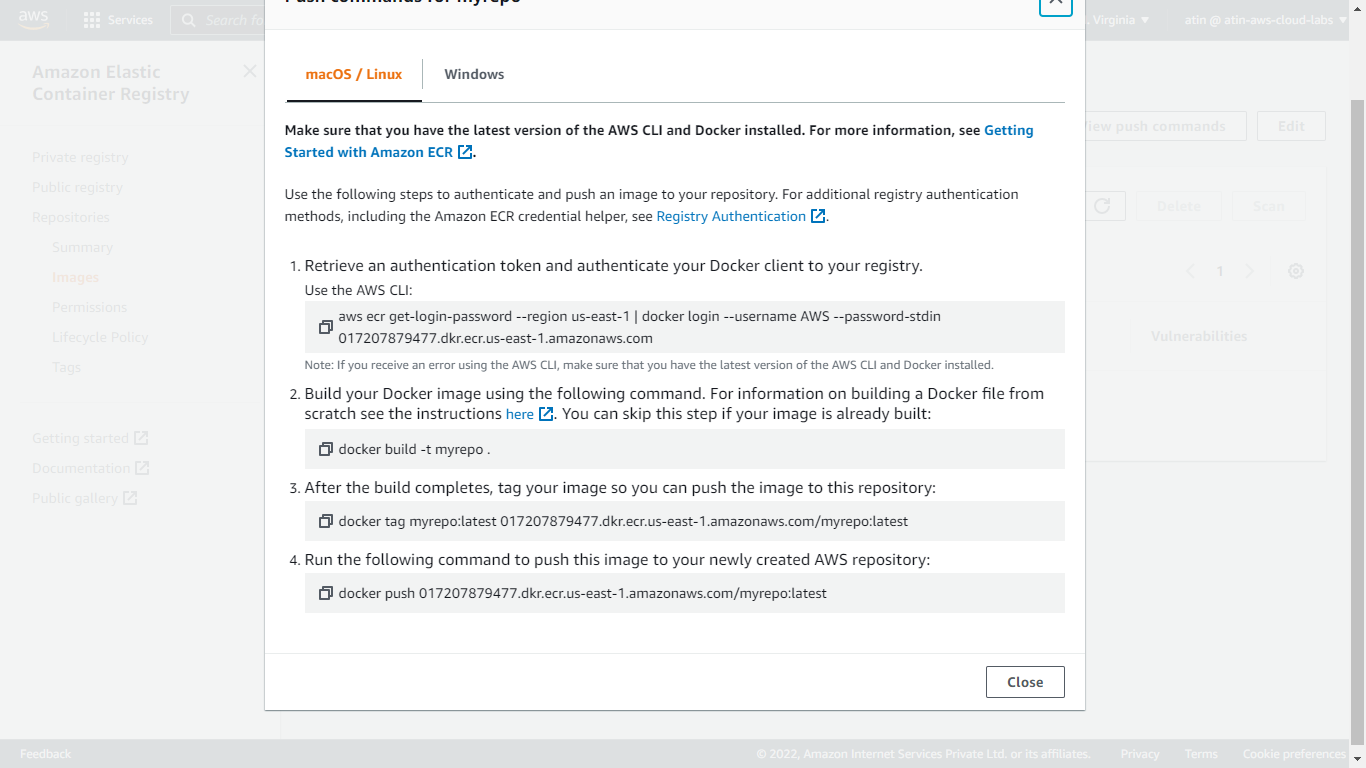
$ git push -u origin main



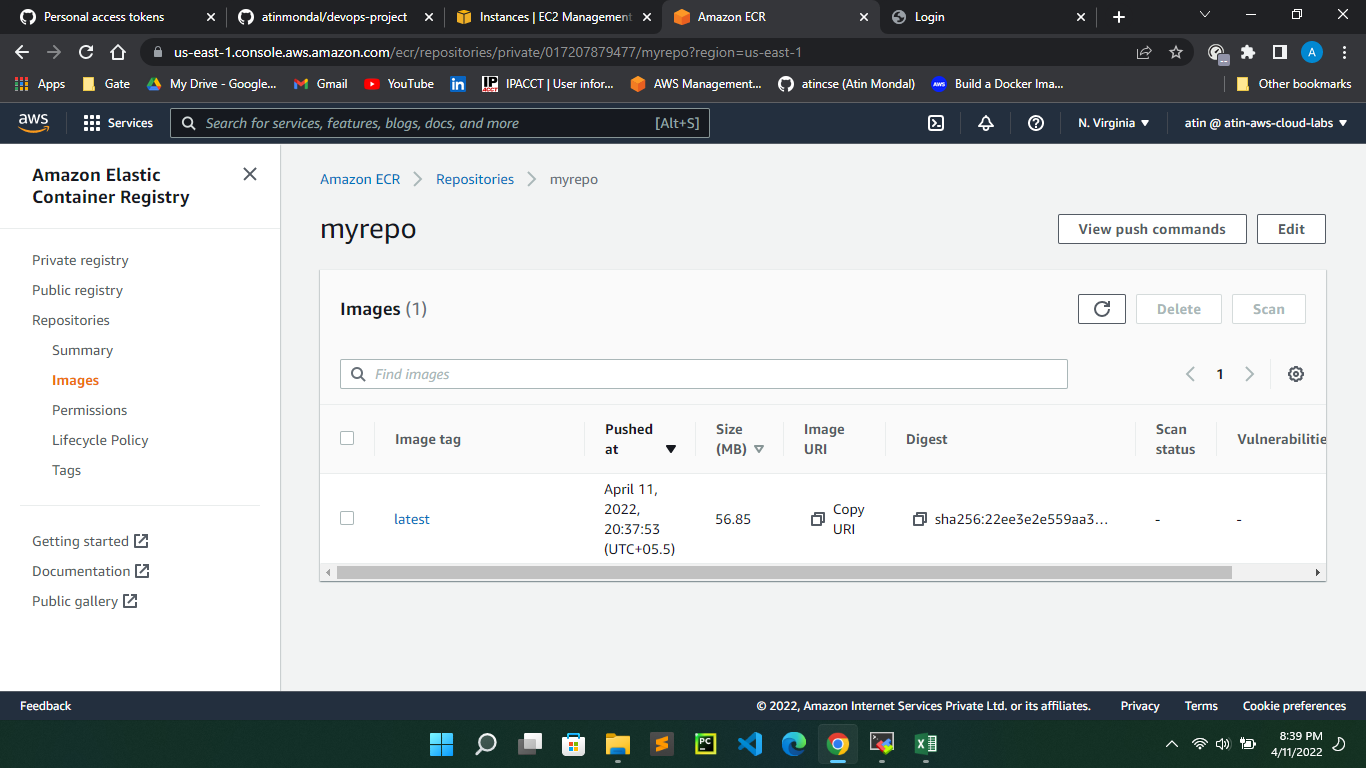
1. Create ECS environment
2. Create repository in ECR(name of the repository – myrepo)



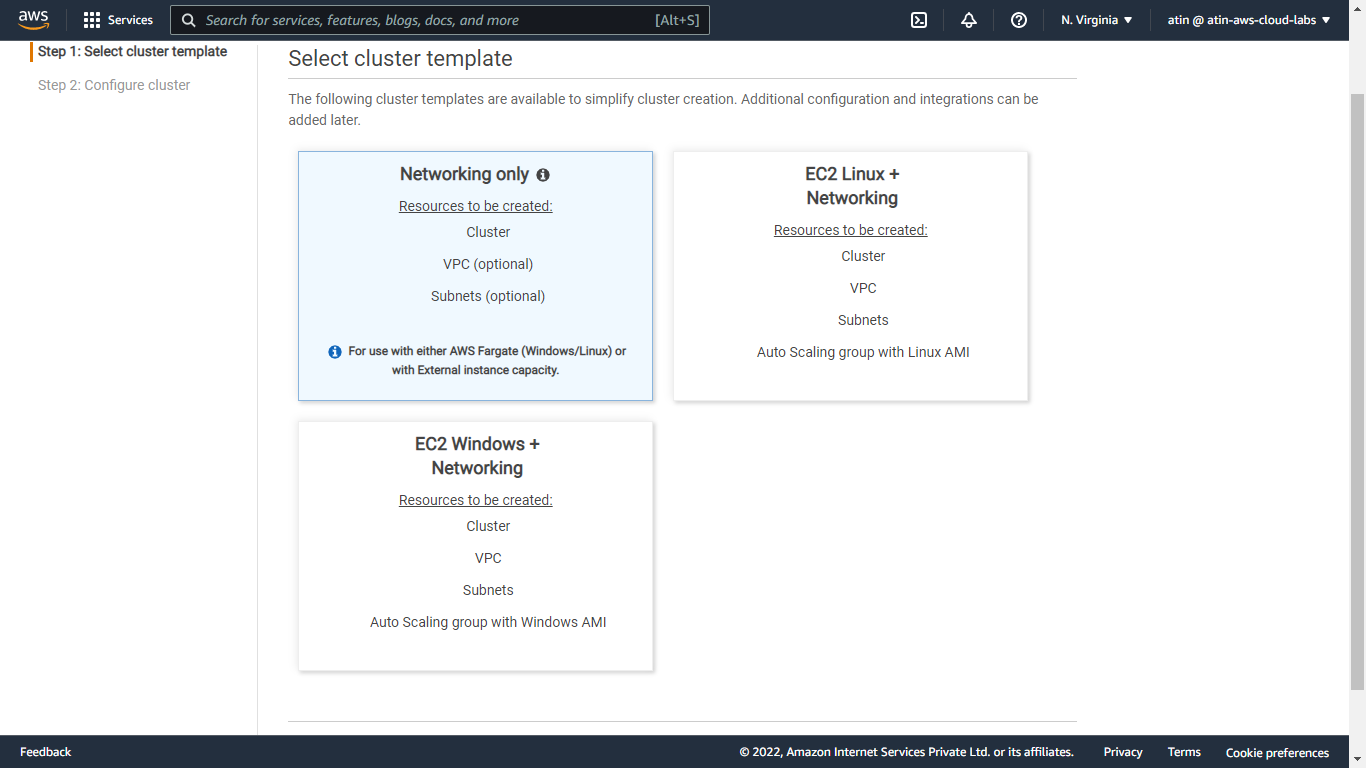
Push image from ec2 to ECR repository

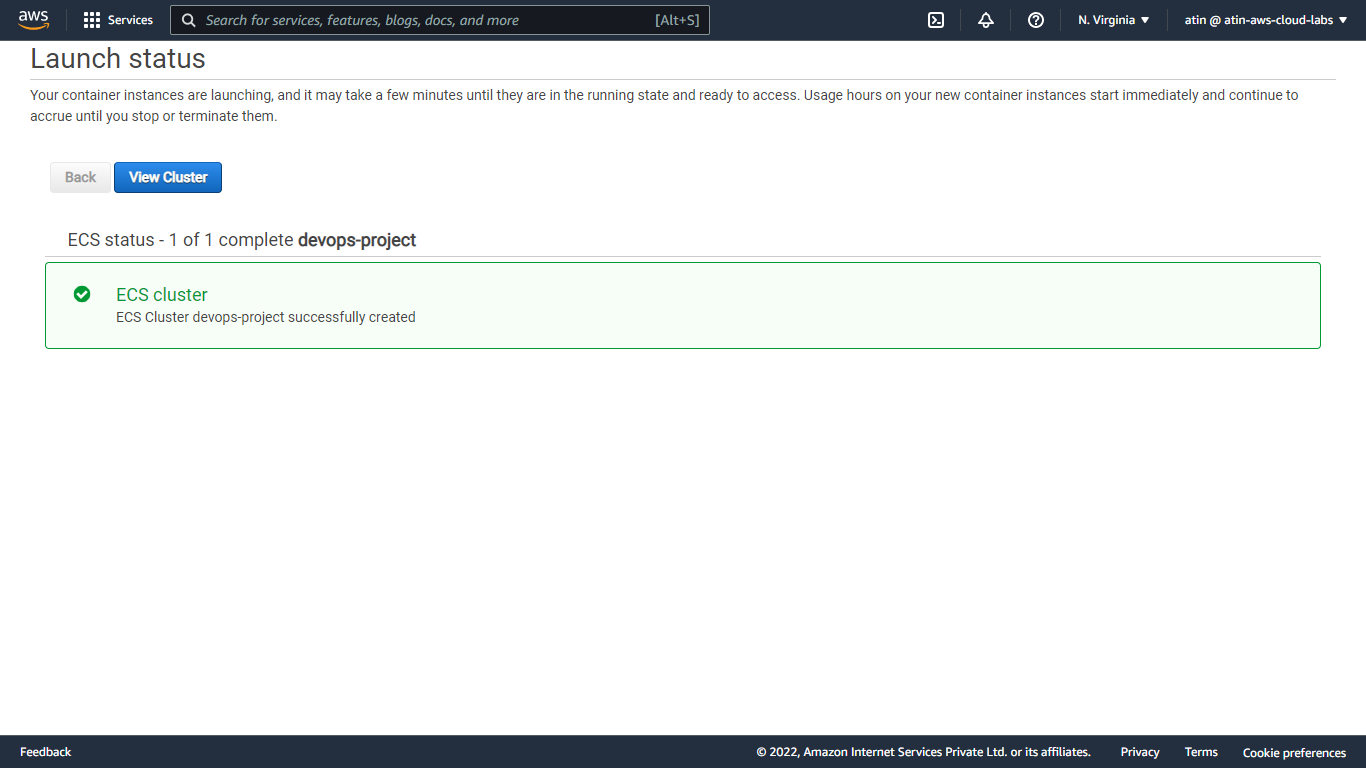


This is the docker image pushed from ec2 via AWS push commands

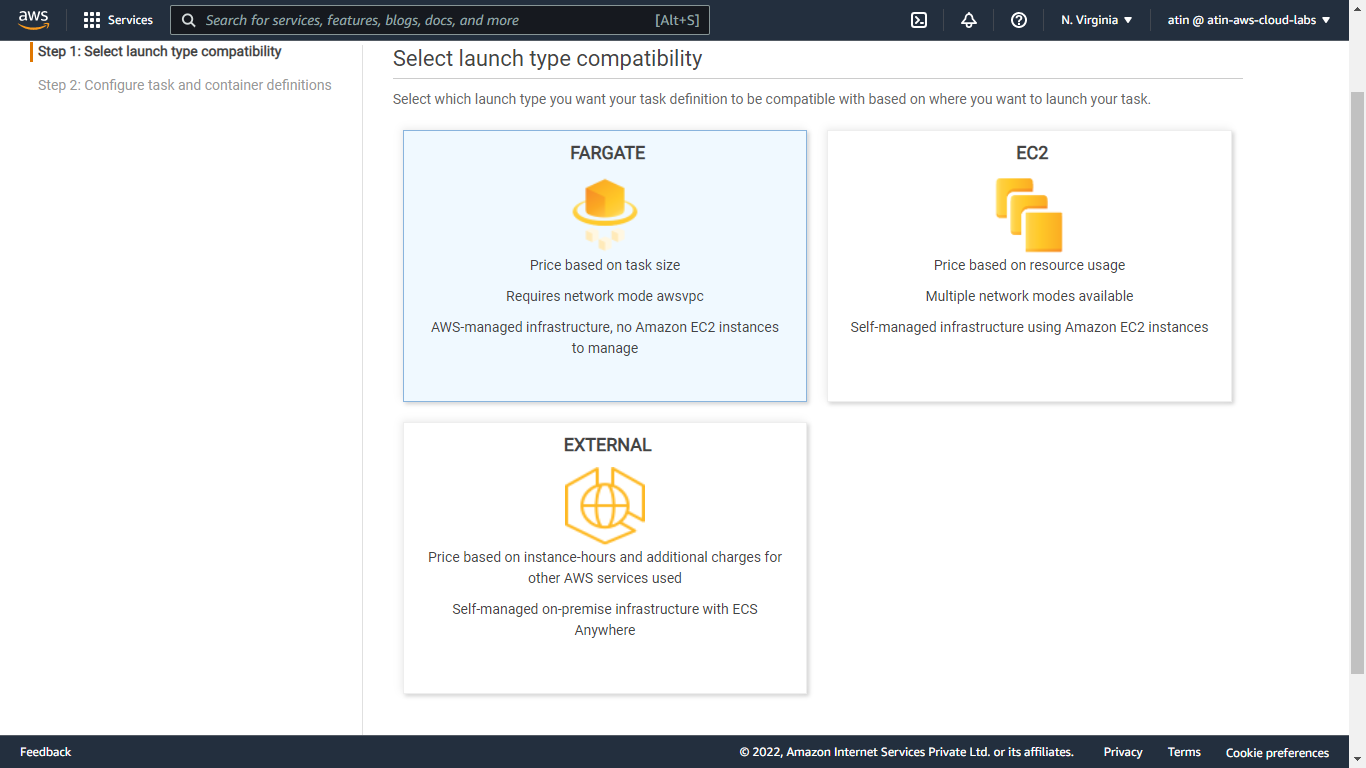


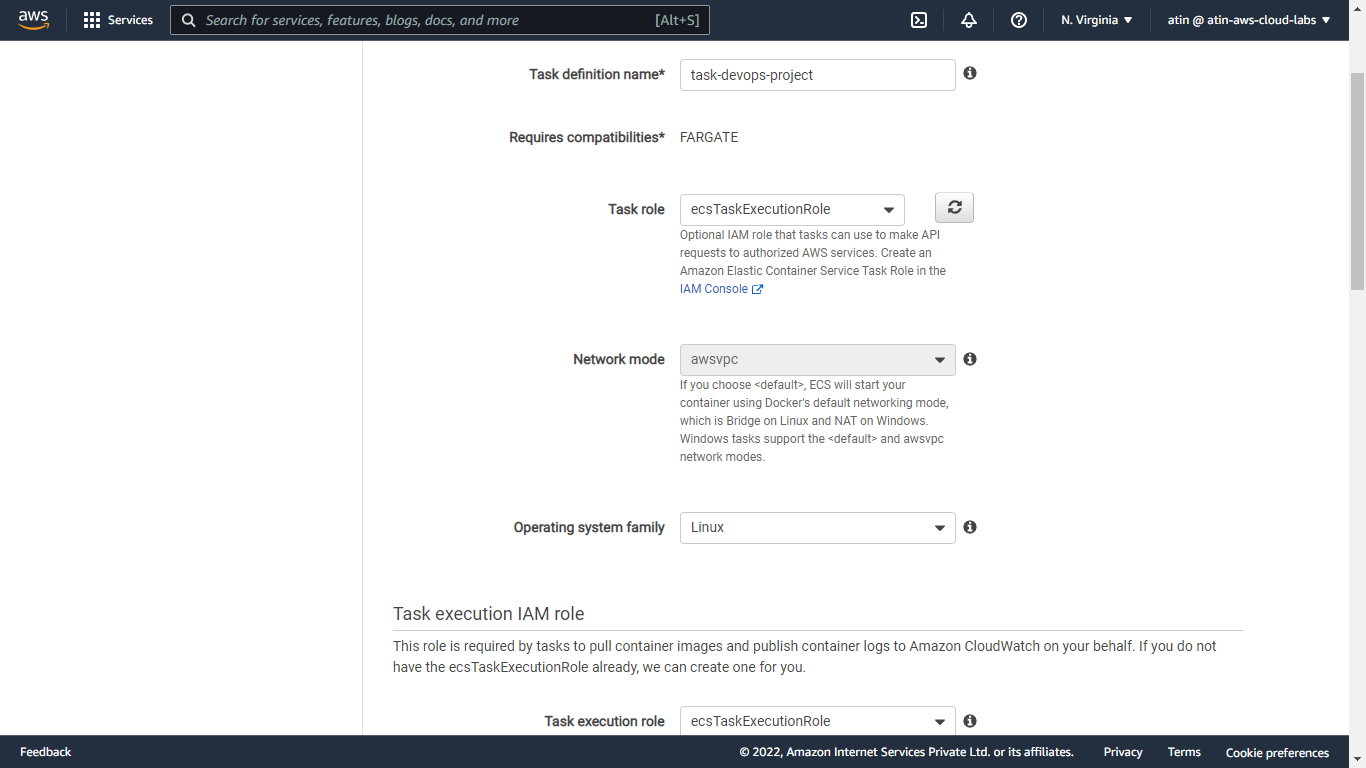
1. Create cluster in AWS ECS (cluster template – fargate & name – devops-project)

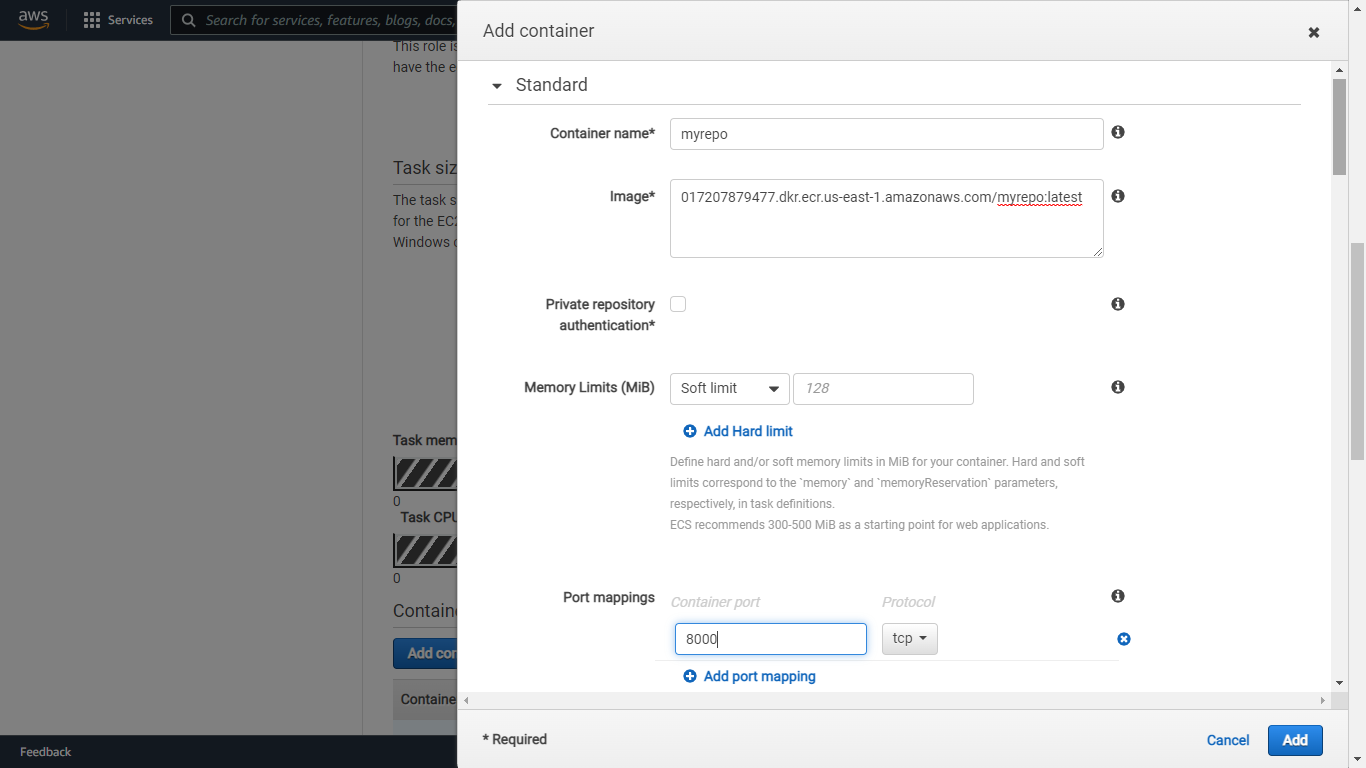




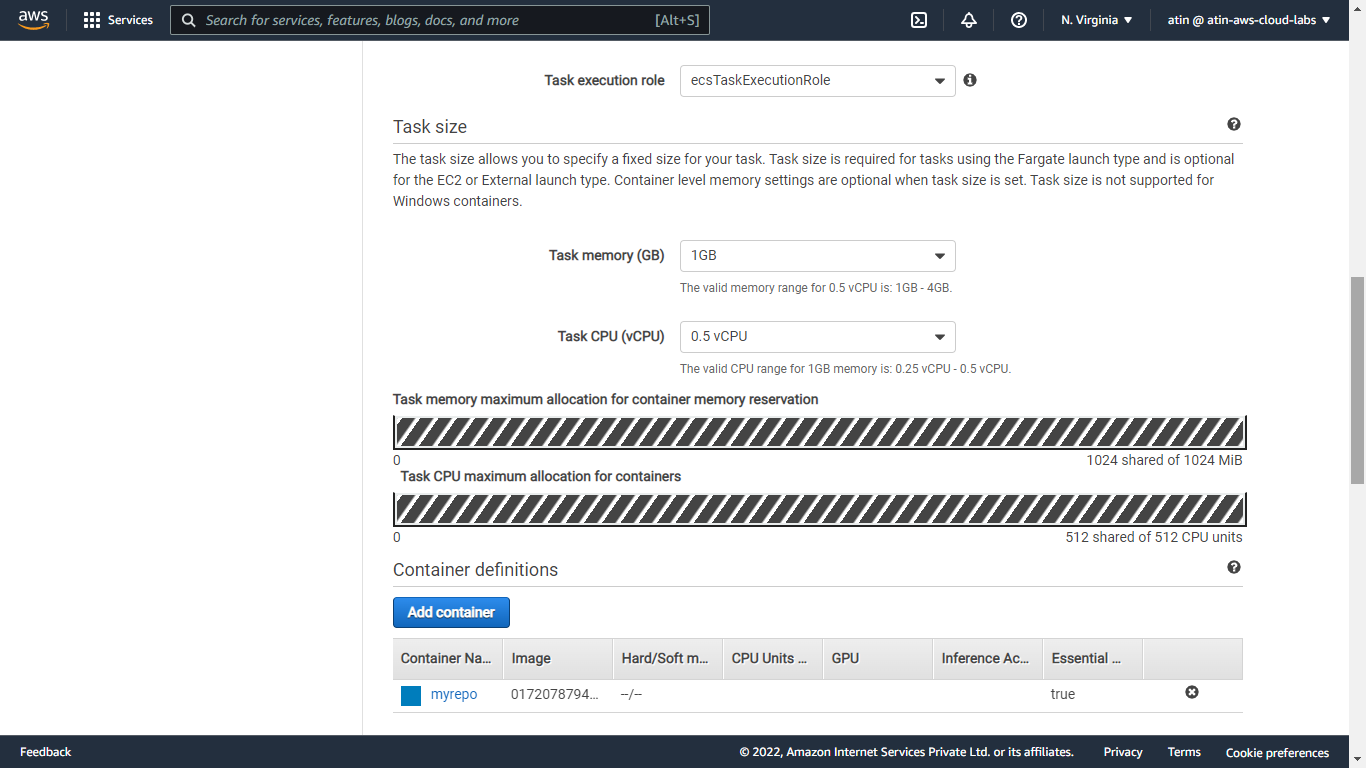
1. Create task definition for our cluster(Launch type – fargate & name- task-devops-project)





Add container name ,image URL and port map to 8000 

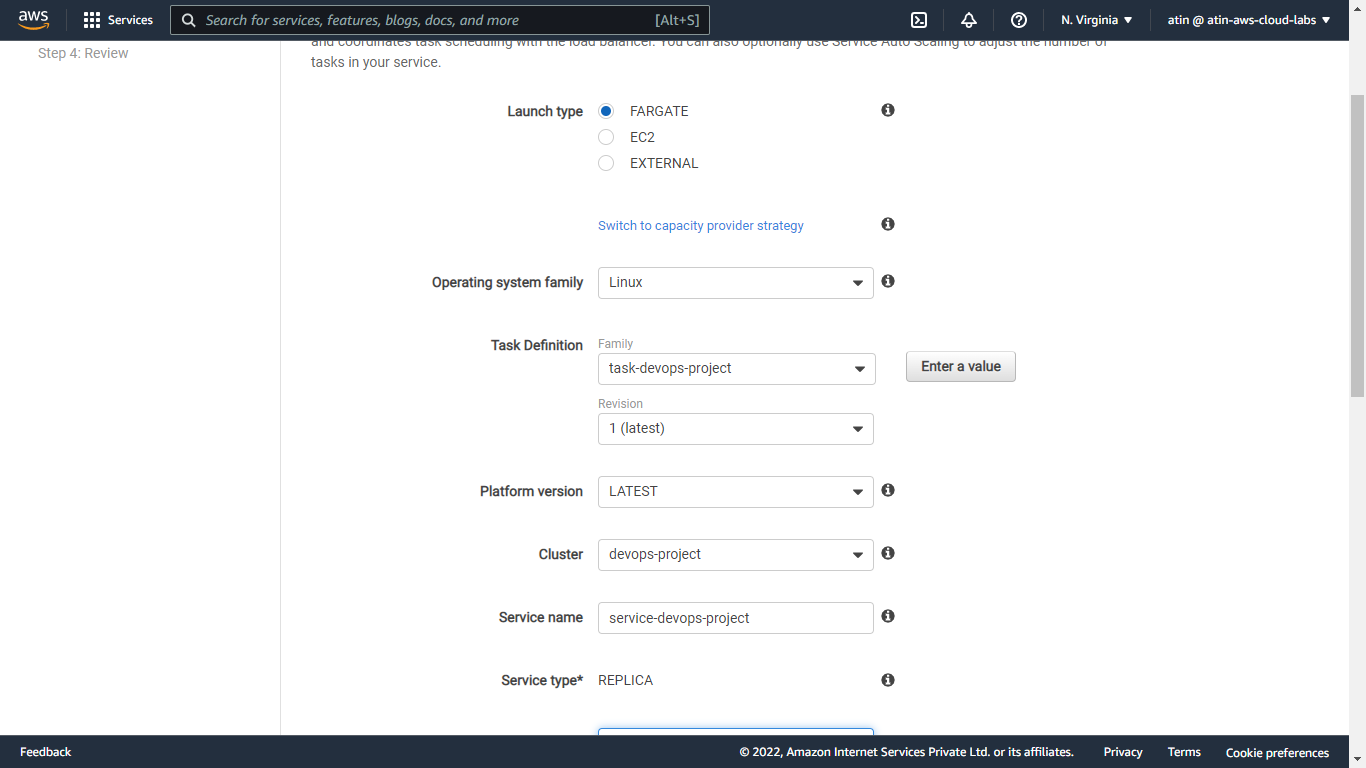
Task capacity – 1GB & Task CPU – 0.5 vCPU



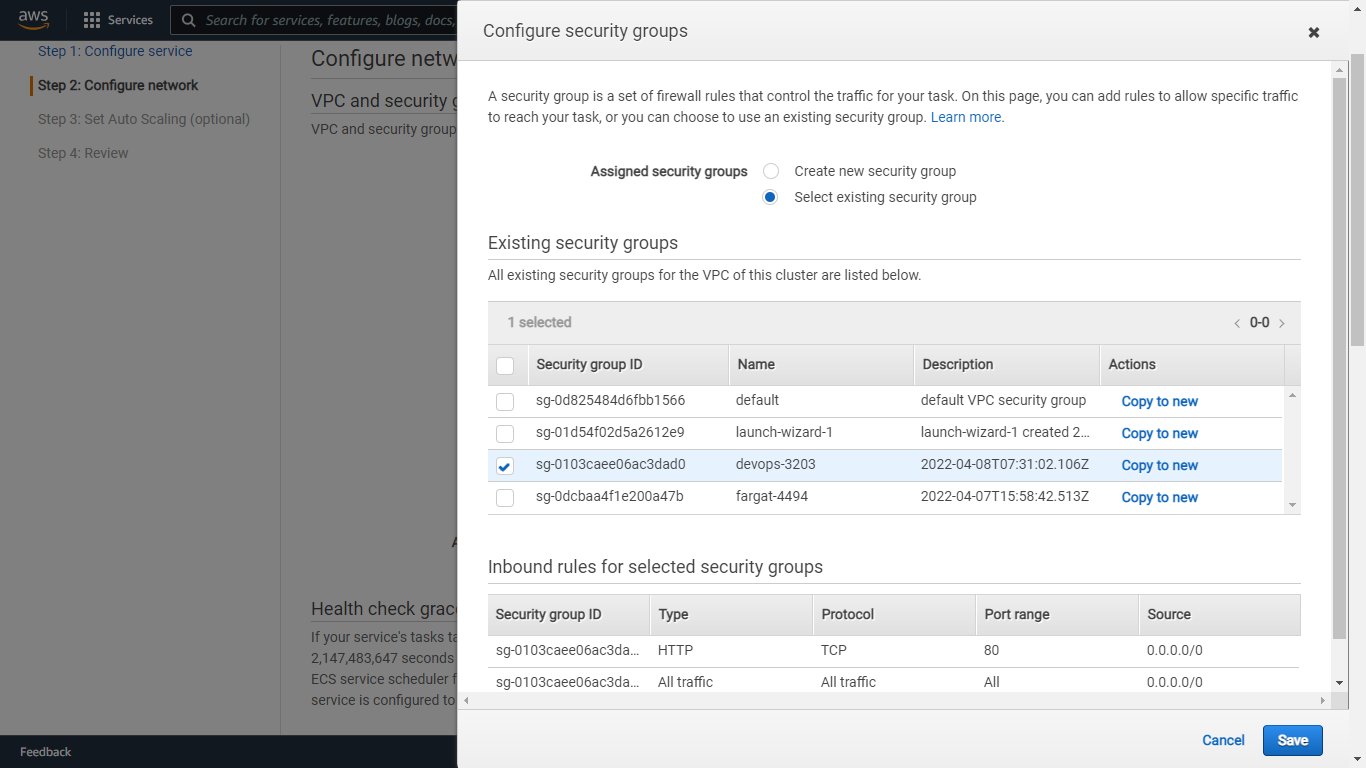


1. Create service for our ECS cluster

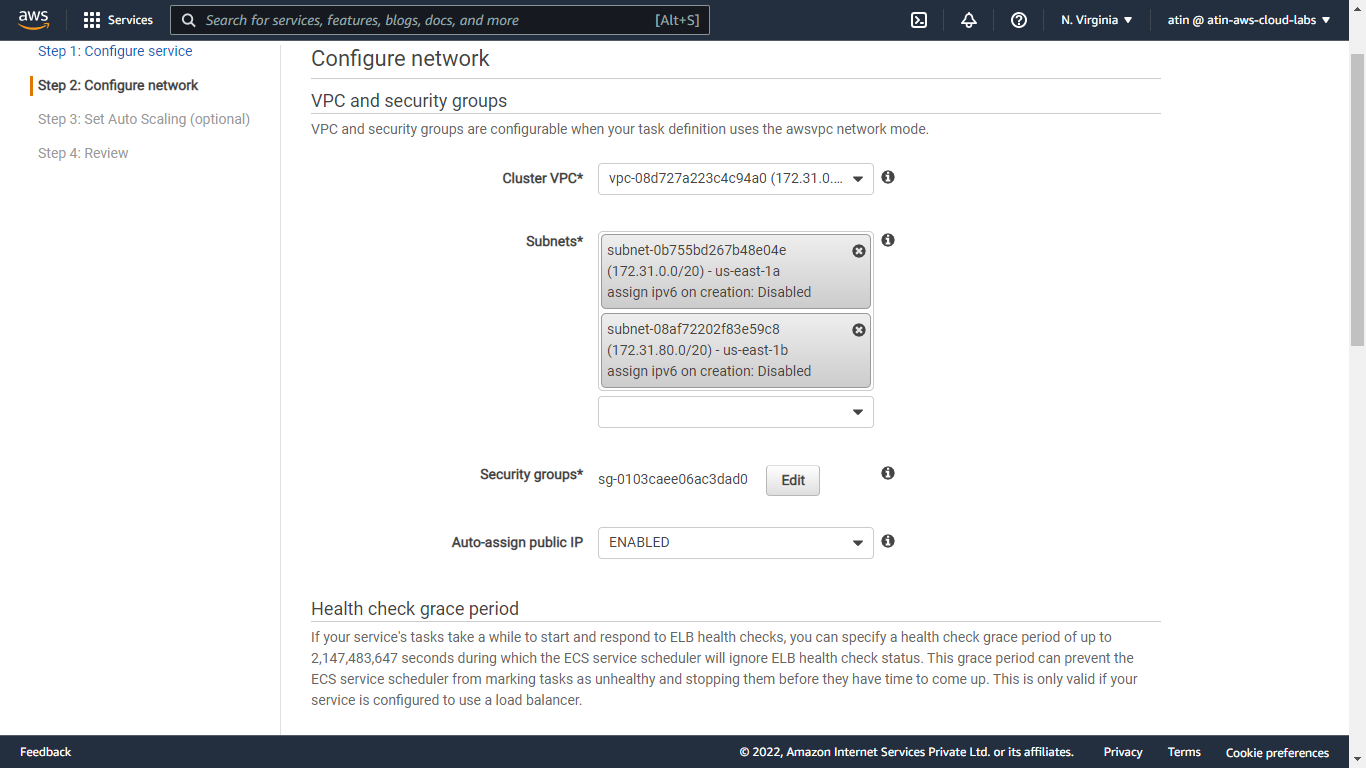
(Launch type –fargate & name- service-devops-project)

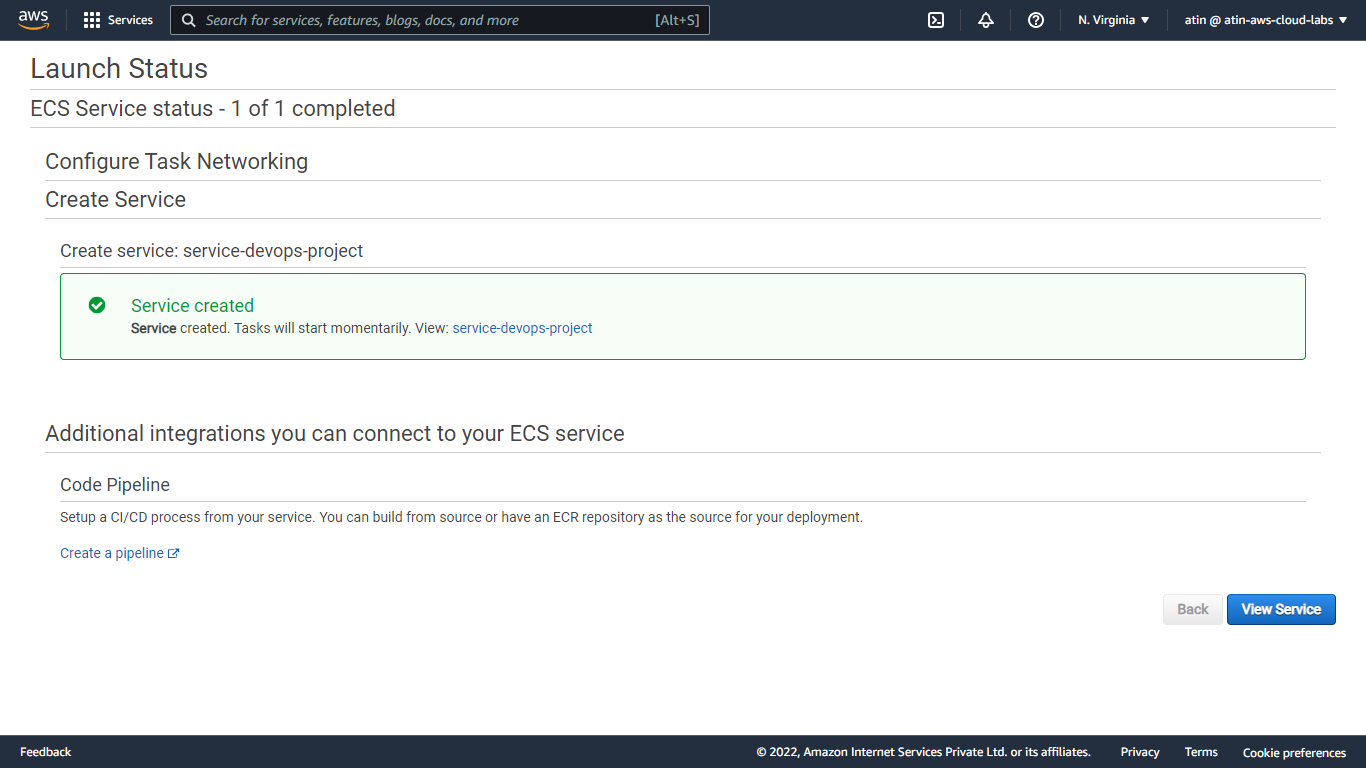


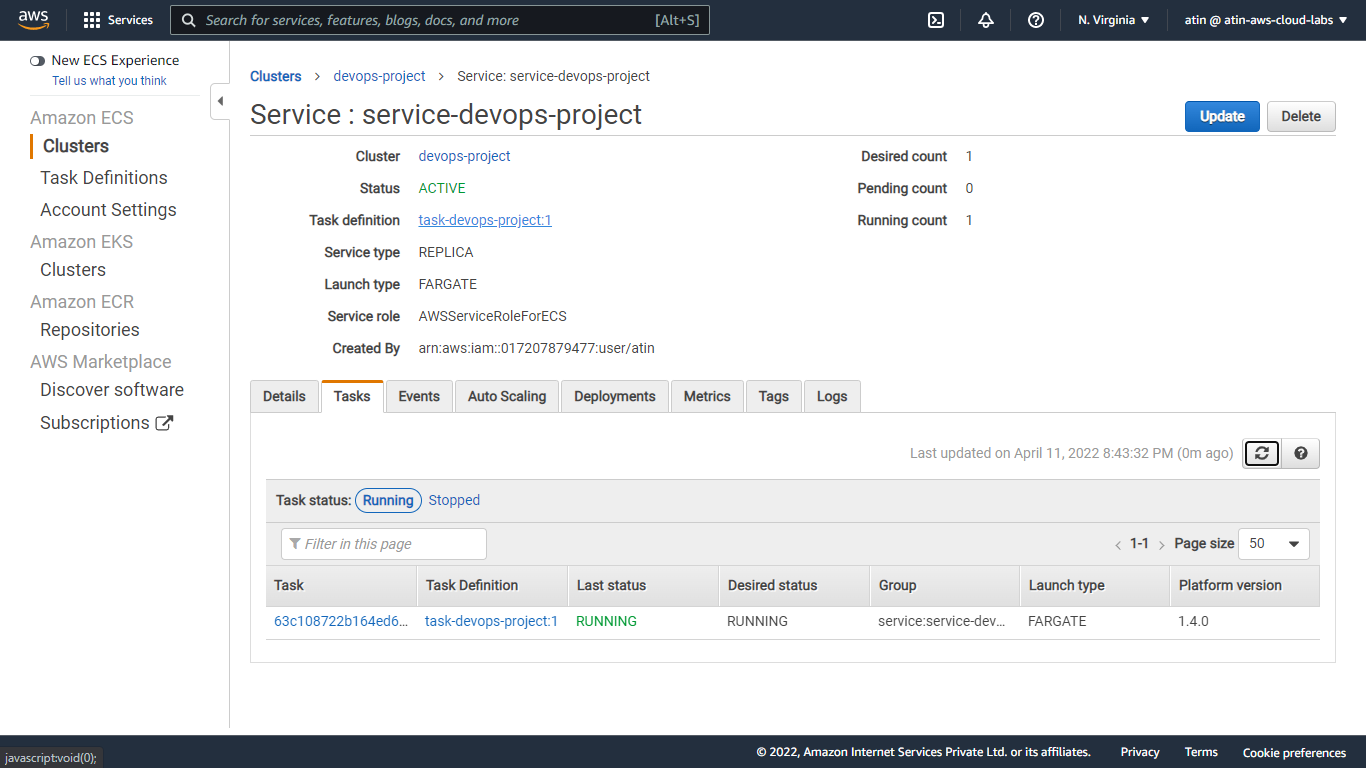
Configure security group



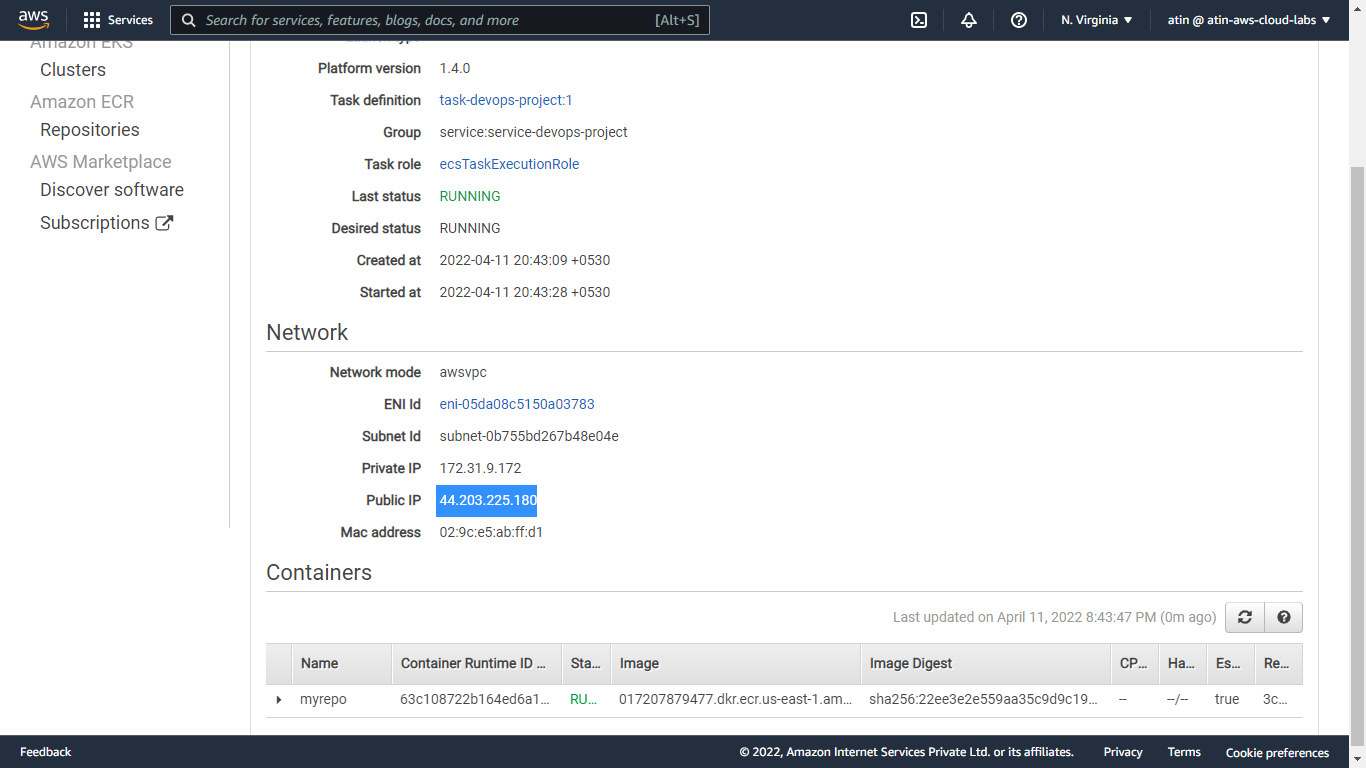
Keep vpc as default – 172.31.0.0/16 and subnets are – us-east-1a and 1b

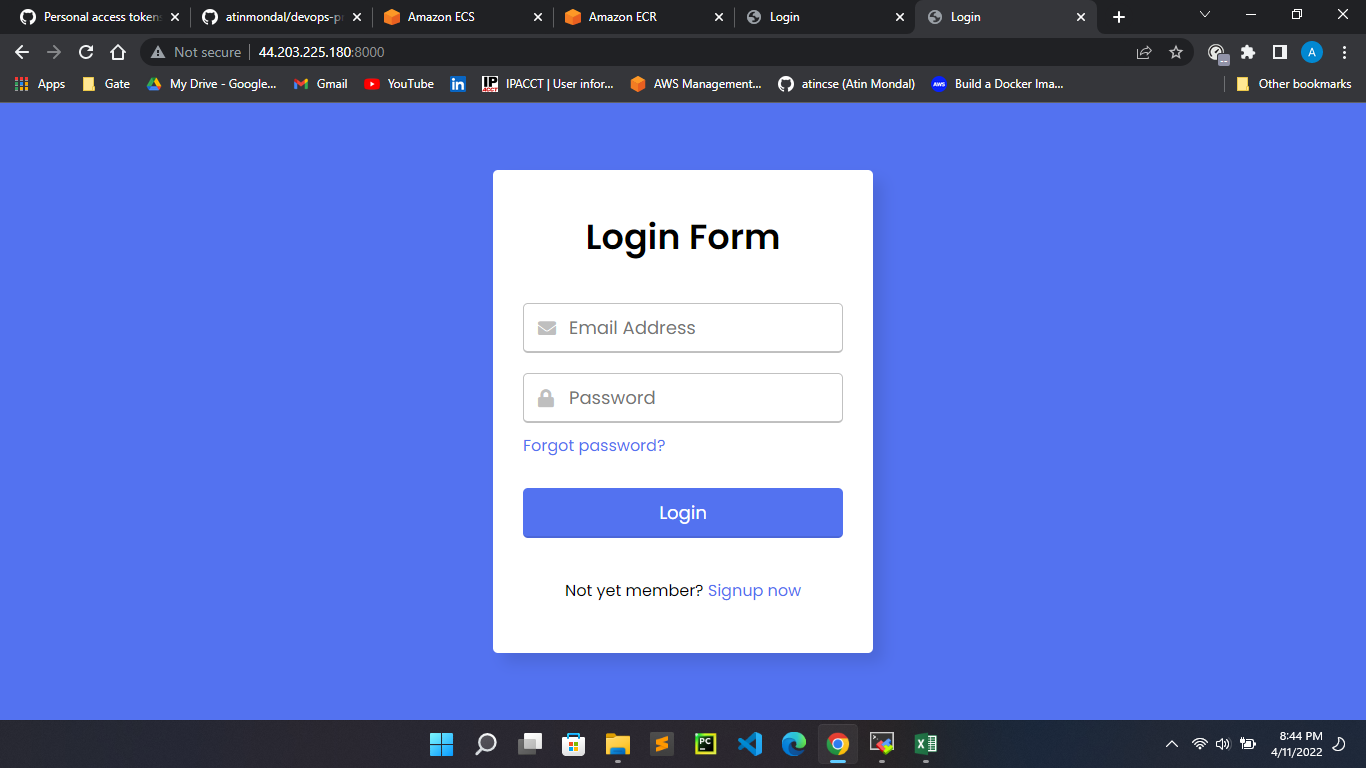




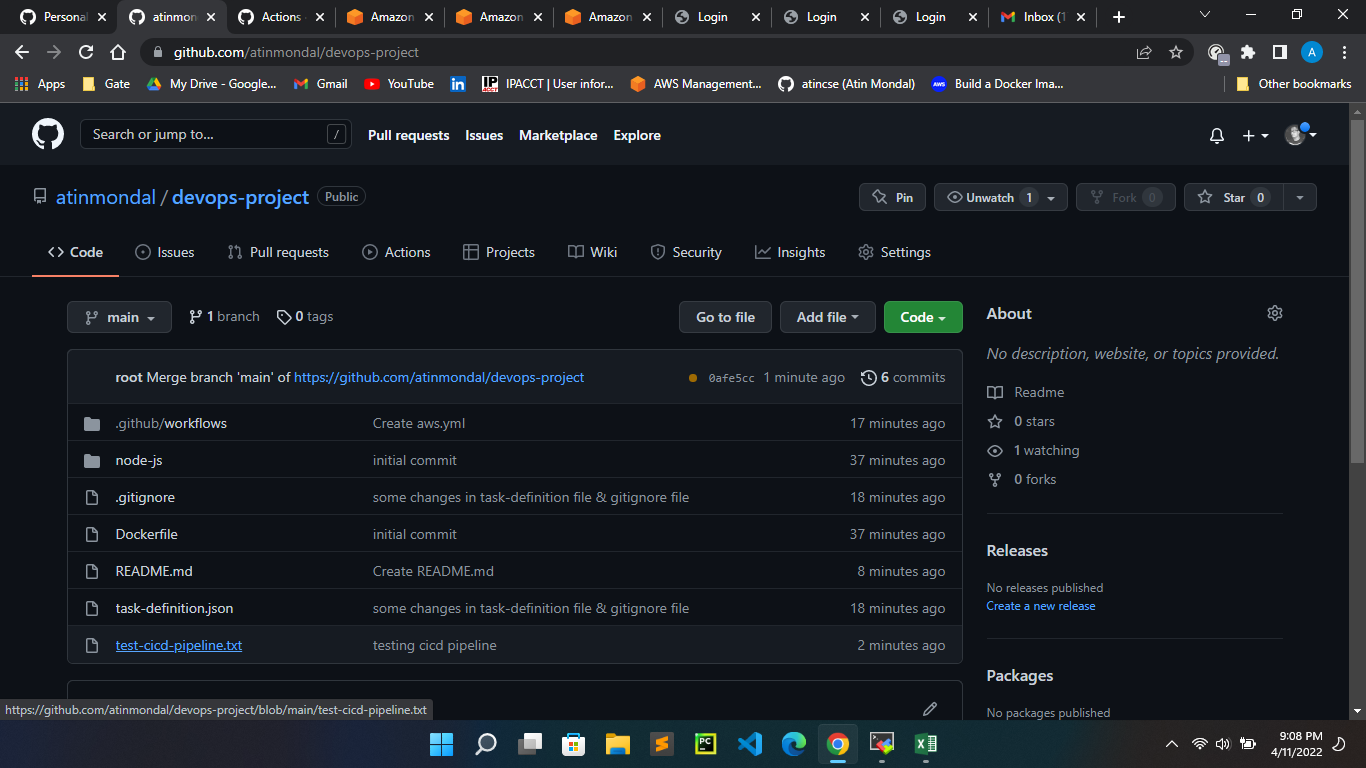
Our task is running

Next click on the task id then copy public ip (44.203.225.180:8000) and see the o/p in the browser

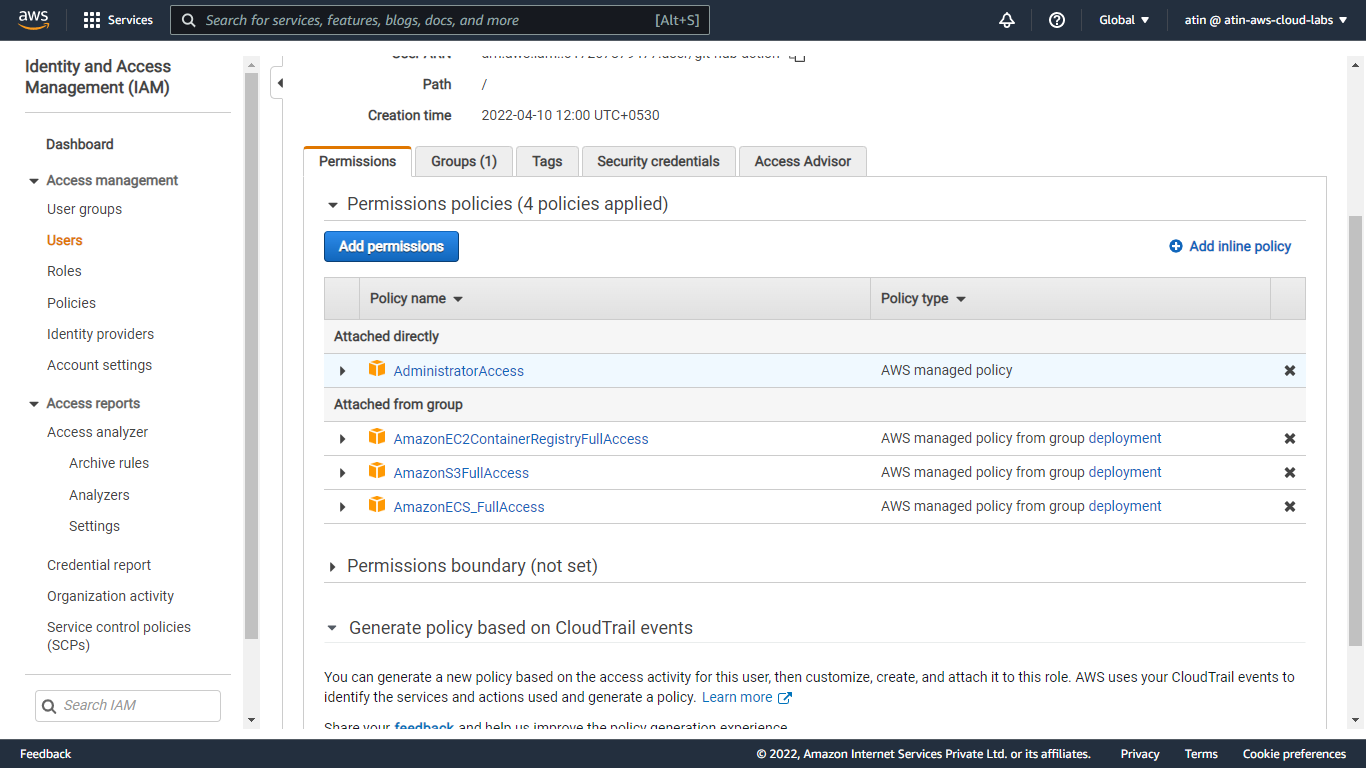




1. Use Github Actions workflow to build an image, push it to ECR, and then download and push an updated task definition to AWS to deploy the latest code to ECS.
2. Store our Amazon ECS task definition as a JSON file (task-definition.json) in our GitHub repository.



1. Create IAM user in AWS and attach policies like - [AmazonECS\_FullAccess](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonECS_FullAccess) , [AmazonS3FullAccess](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonS3FullAccess) , [AmazonEC2ContainerRegistryFullAccess](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonEC2ContainerRegistryFullAccess) etc.



1. Create aws.yml workflow file for github action

environment variables:

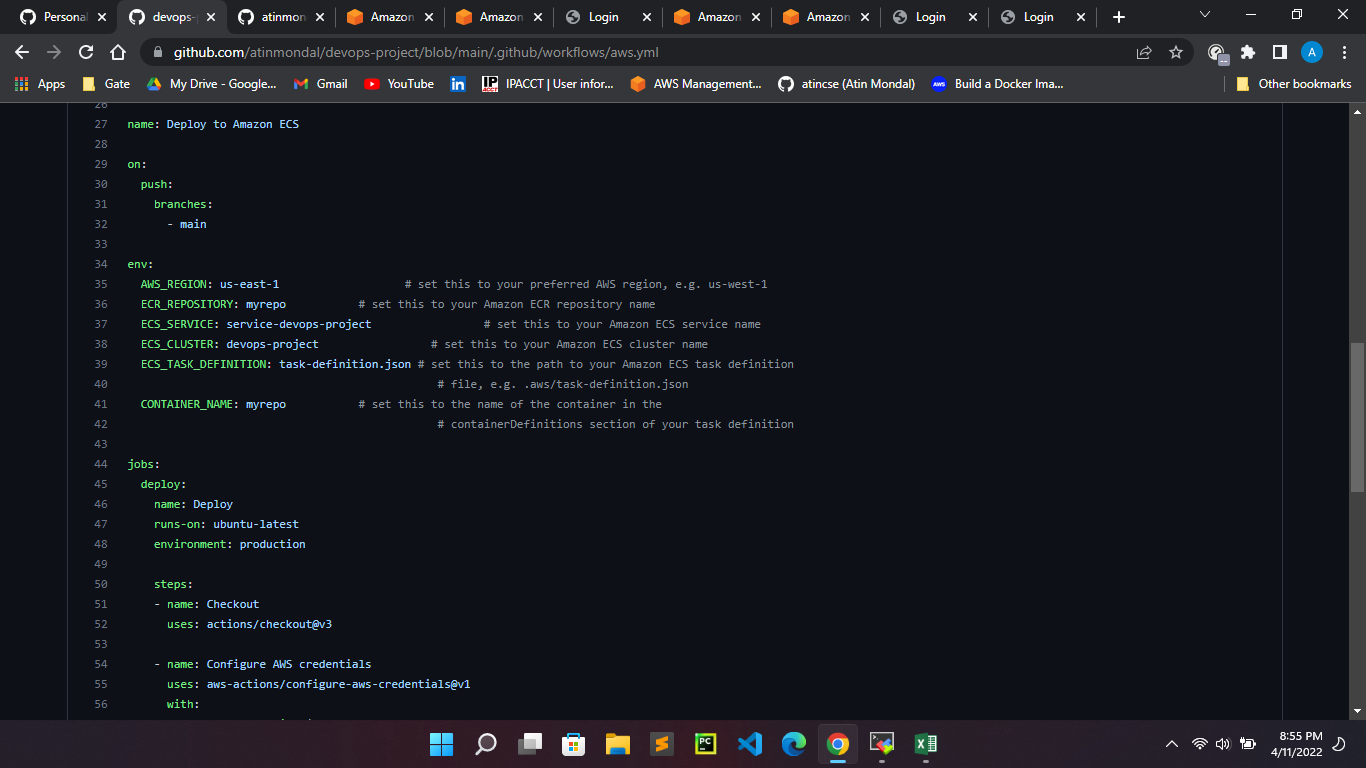
AWS\_REGION: us-east-1

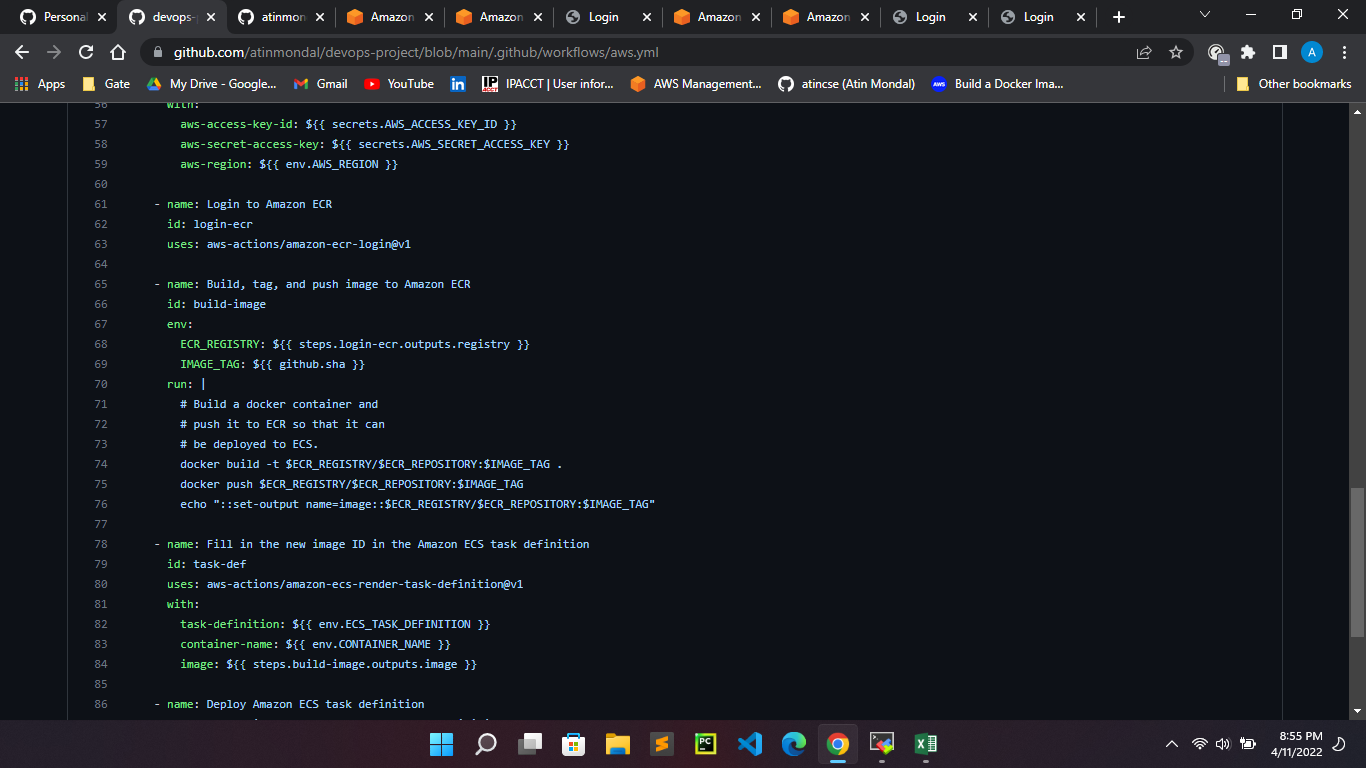
ECR\_REPOSITORY: myrepo

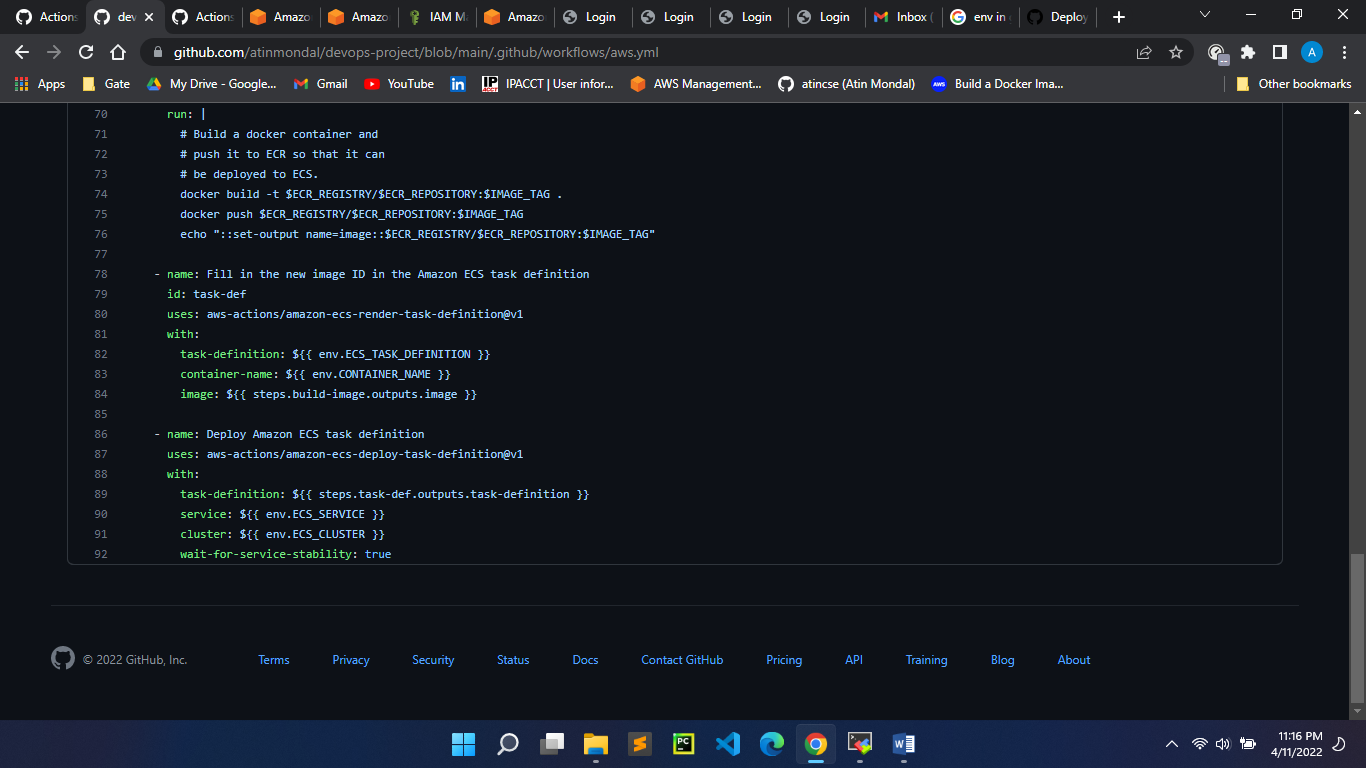
ECS\_SERVICE: service-devops-project

ECS\_CLUSTER: devops-project

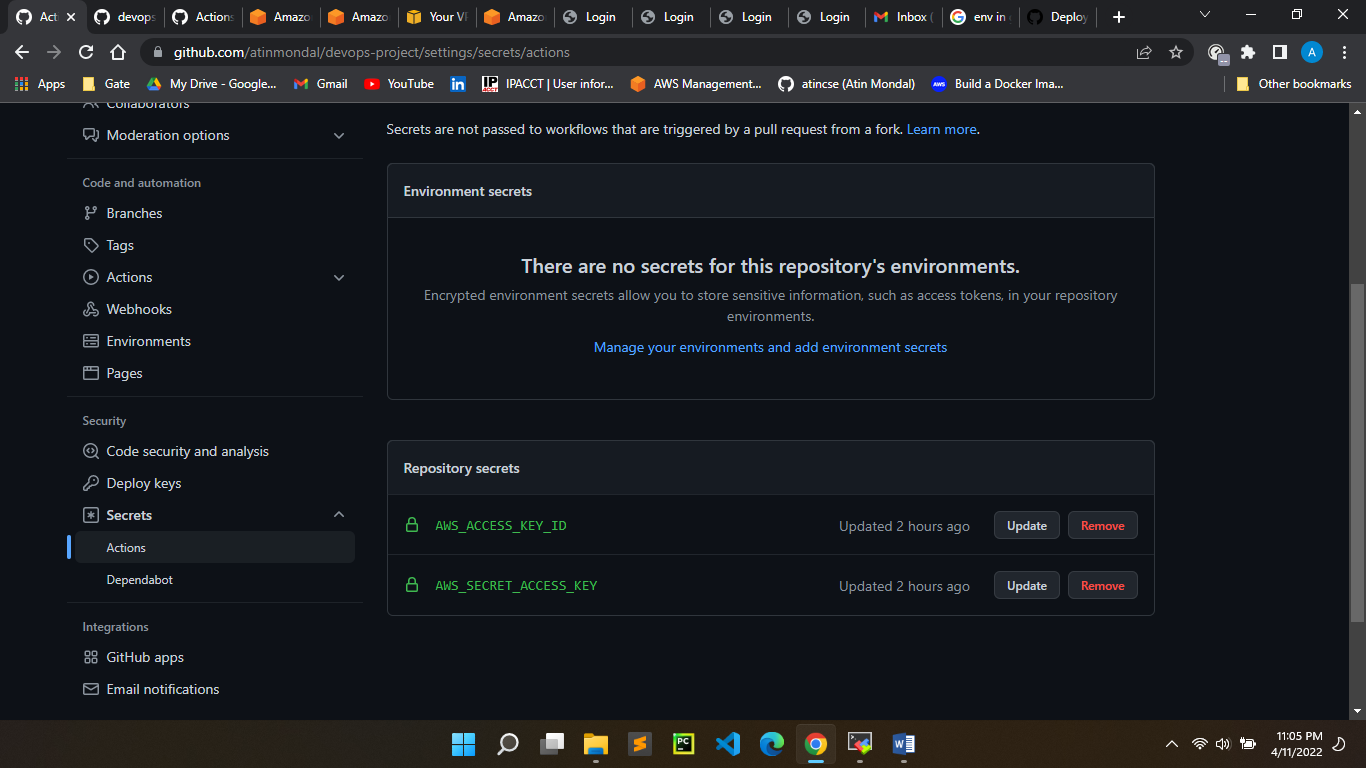
ECS\_TASK\_DEFINITION: task-definition.json



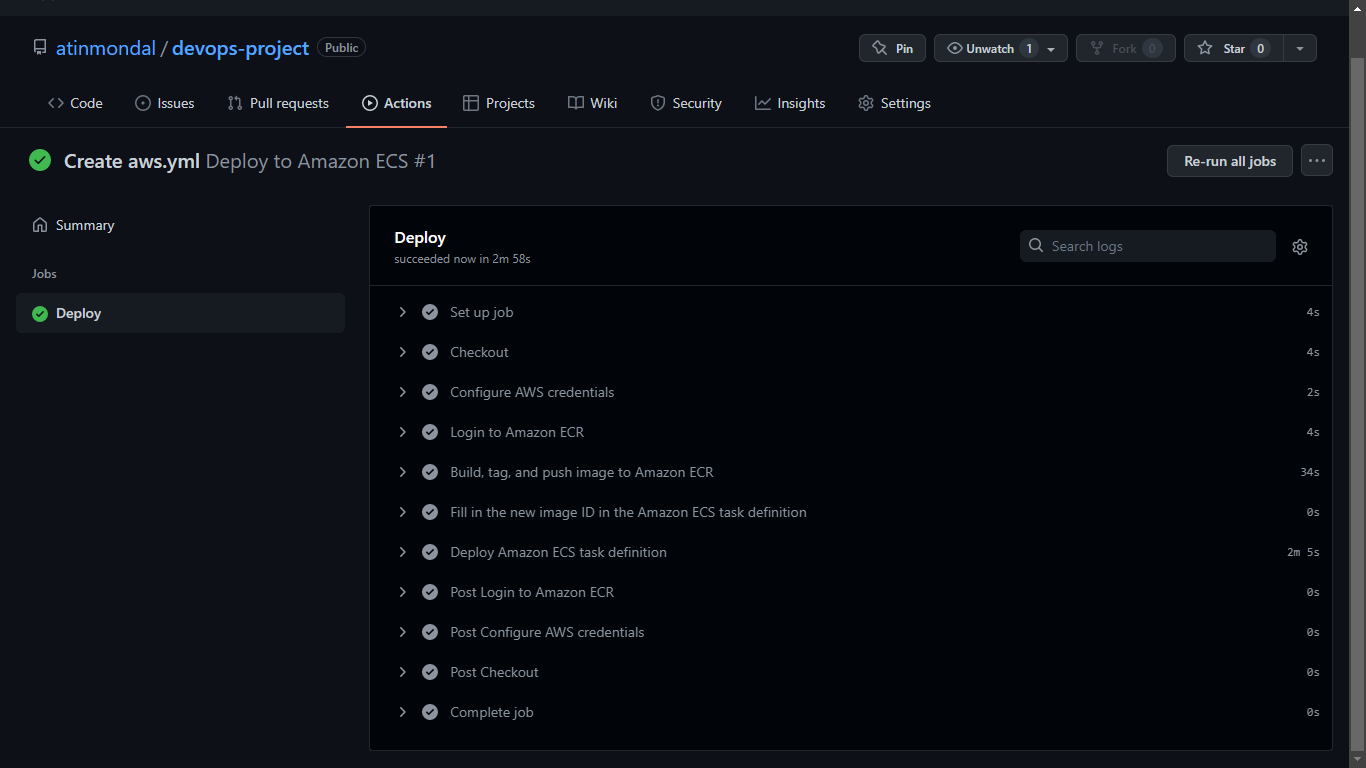




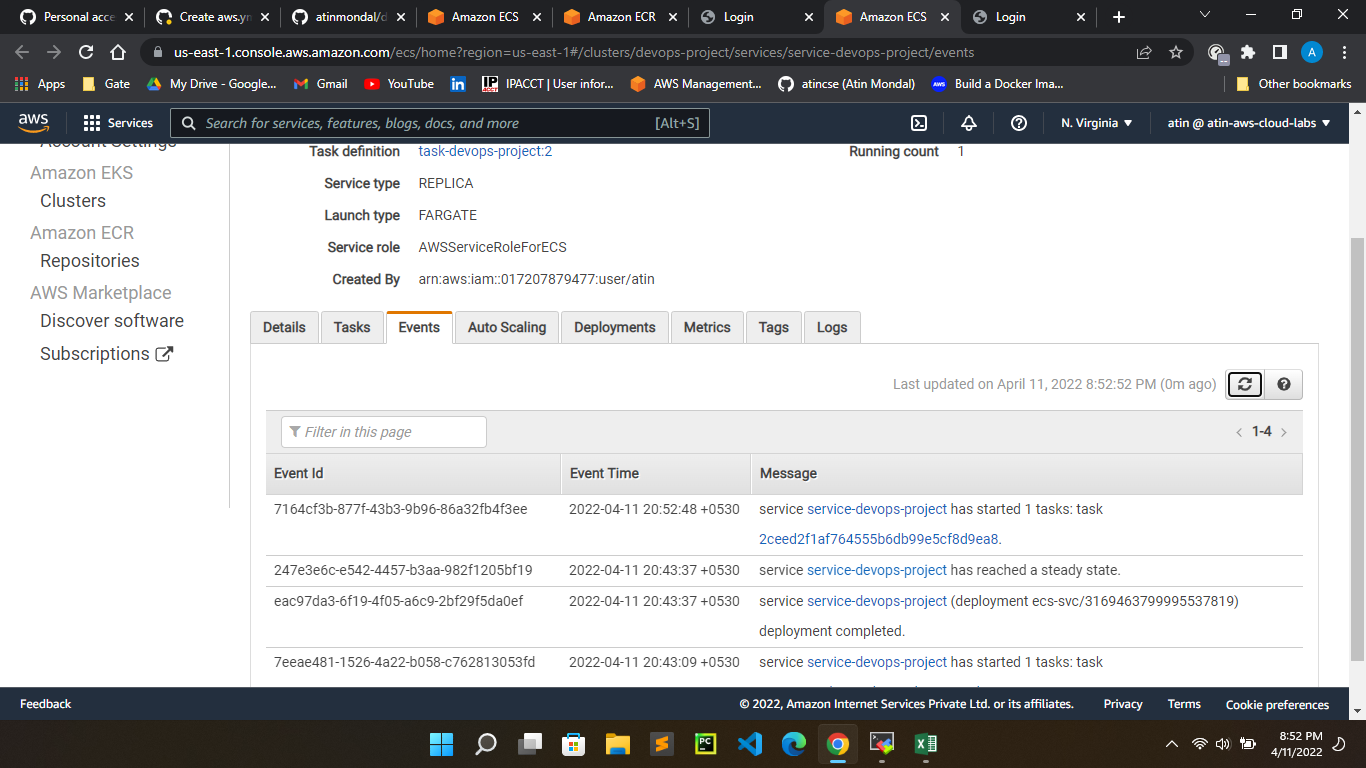
Store IAM user AWS\_ACCESS\_KEY\_ID & AWS\_SECRET\_ACCESS\_KEY in the repository secret action

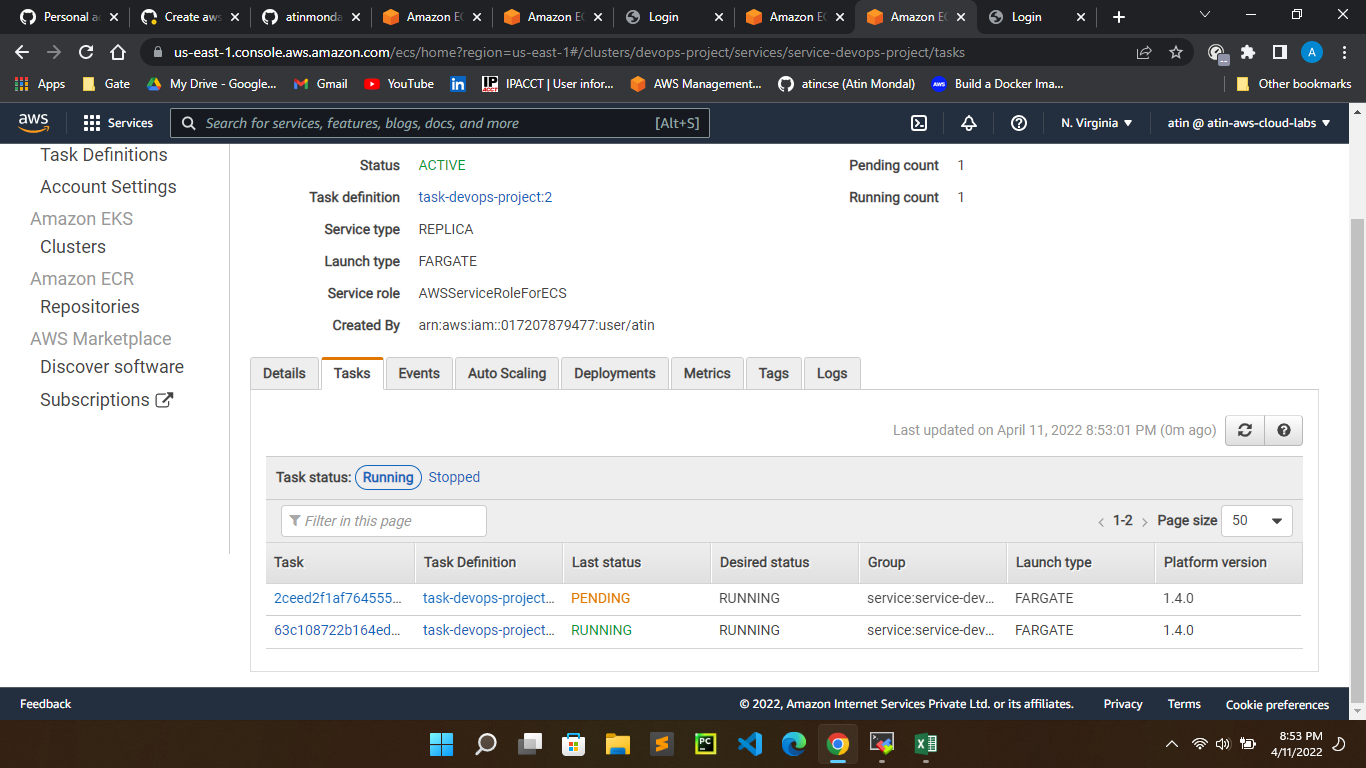


1. Start commit of our aws.yml file

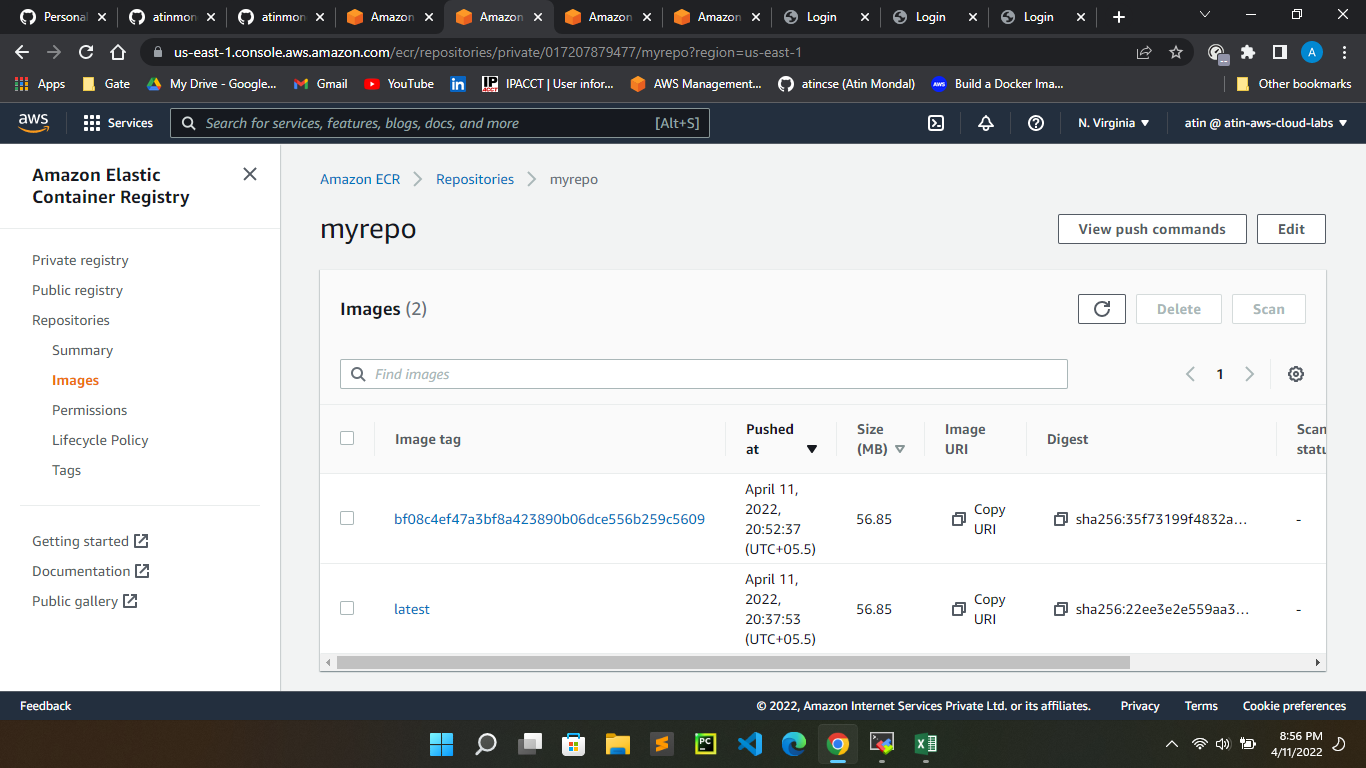


New task definition has been created and its running

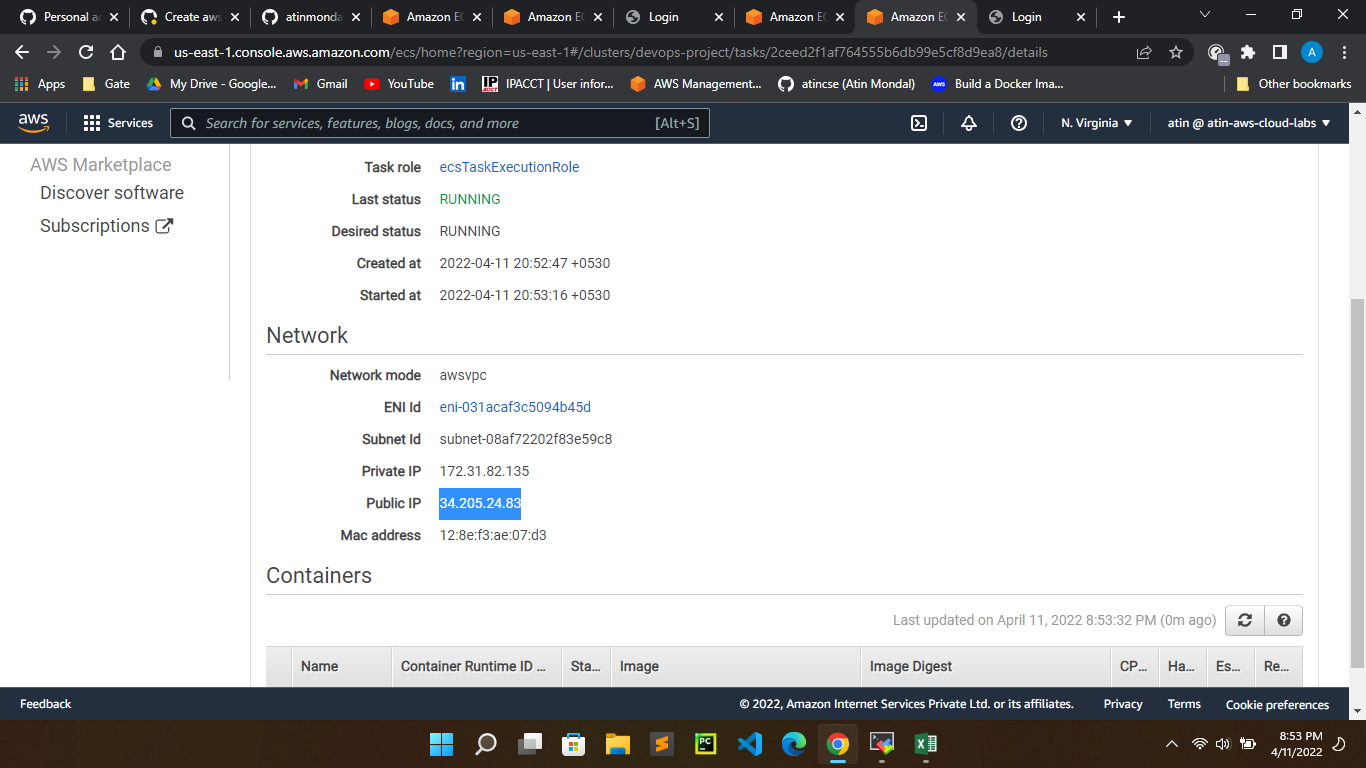


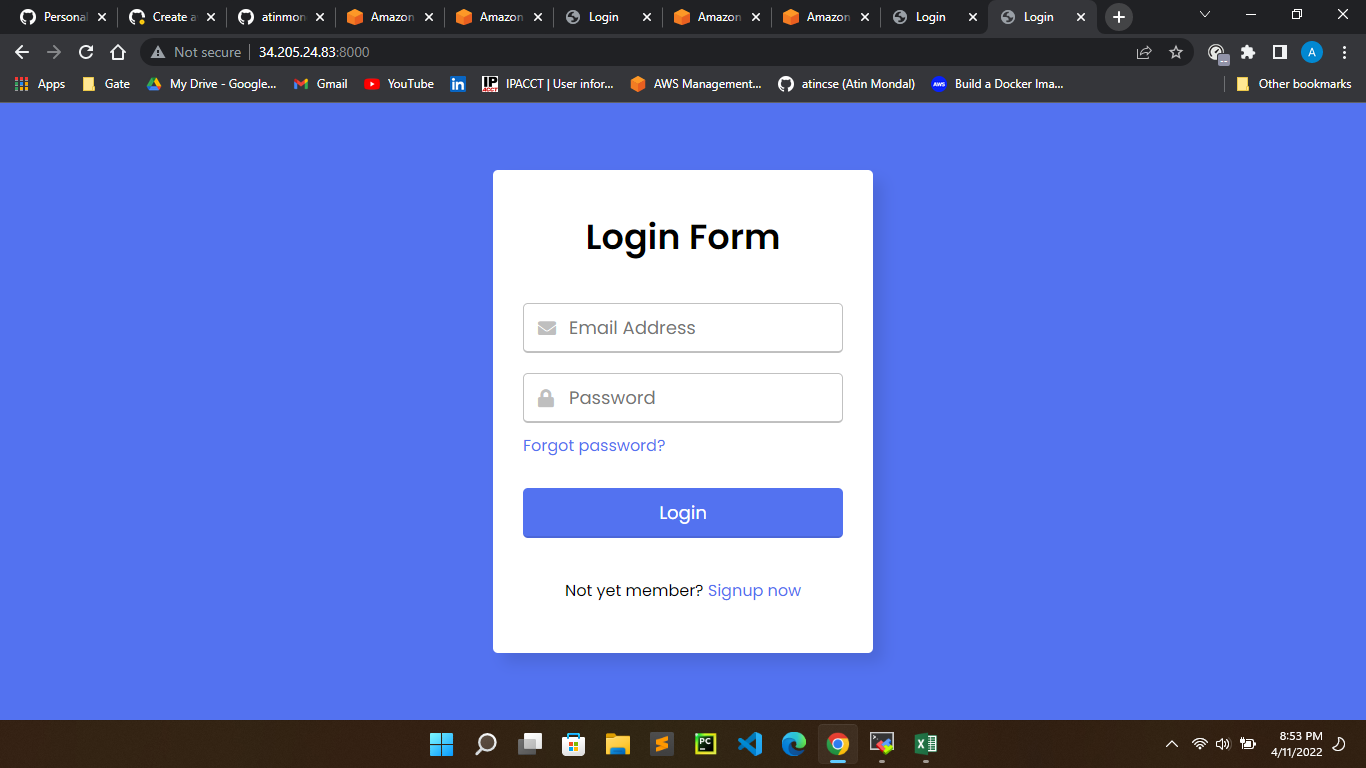


New image has been pushed to ECR



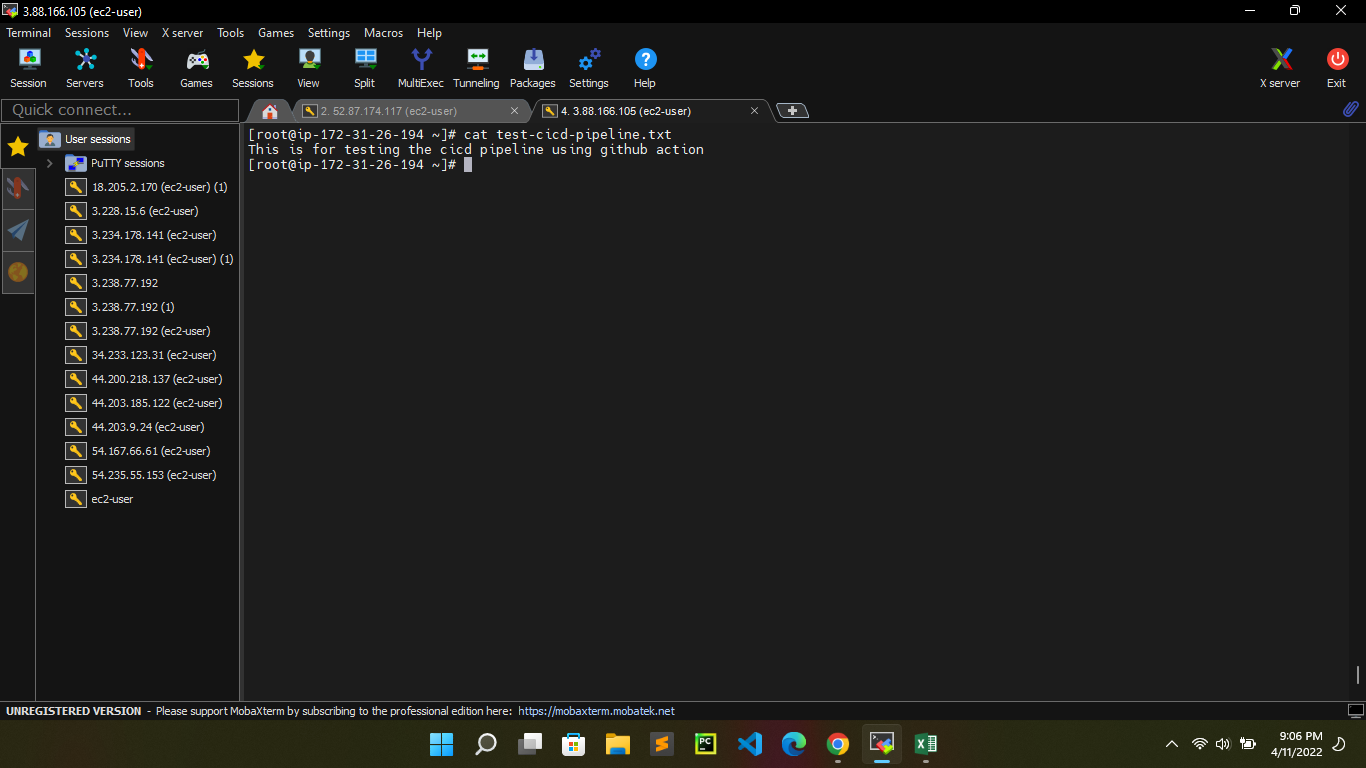
Copy new task public ip (34.205.24.83:8000) and see the o/p in the browser

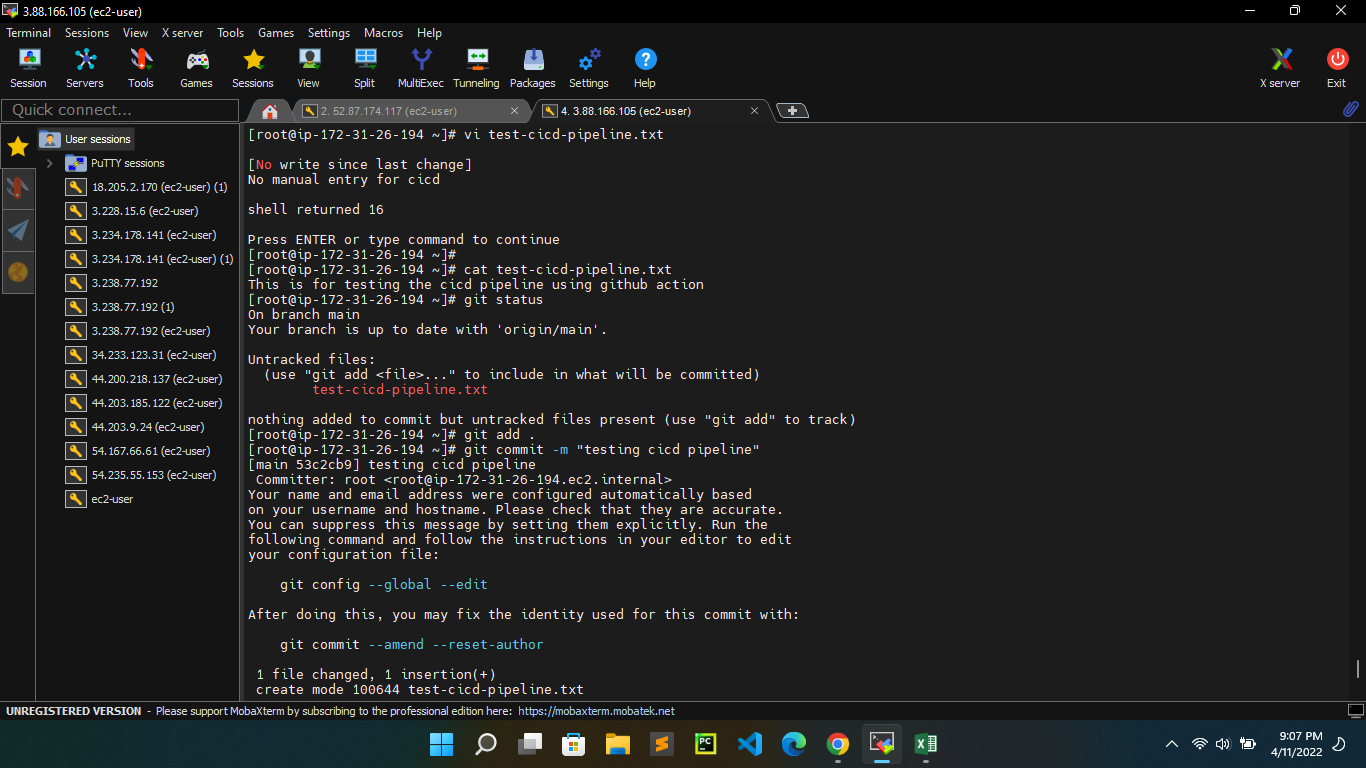


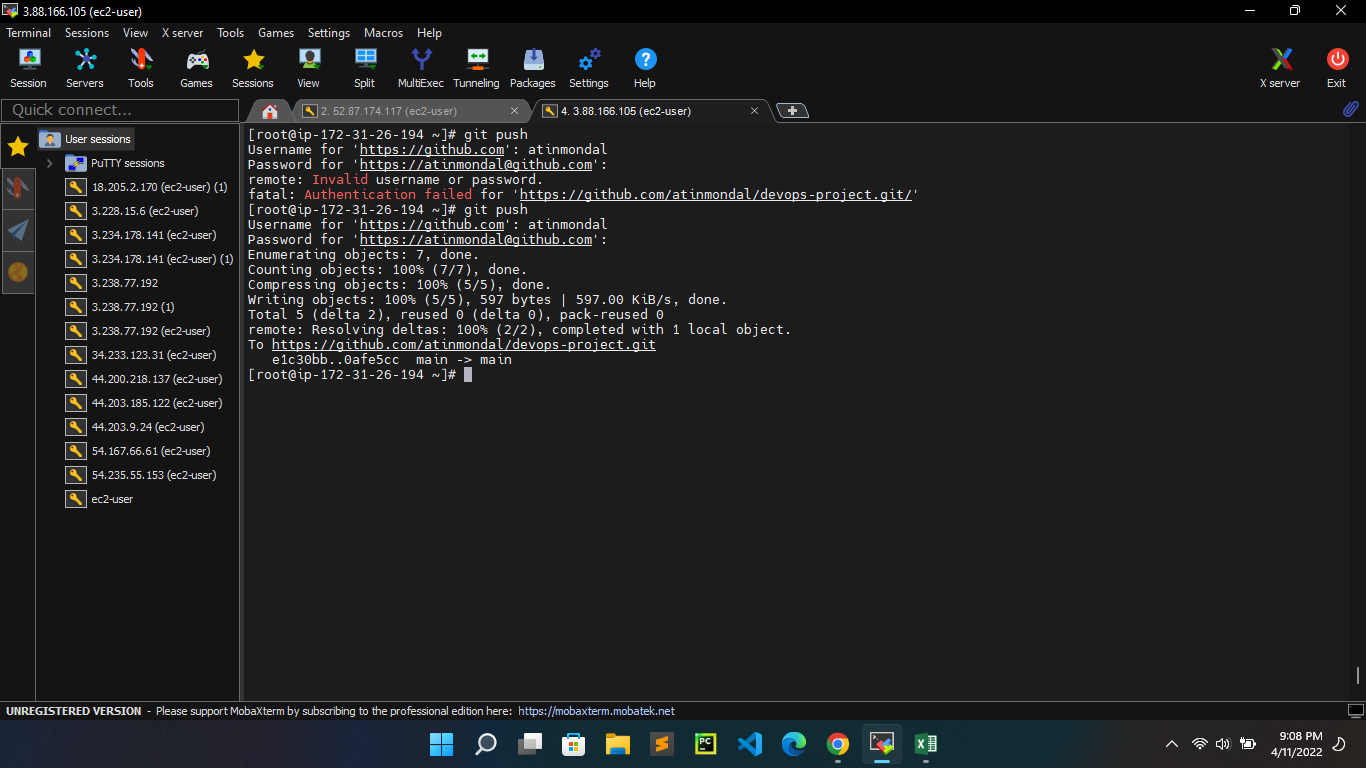


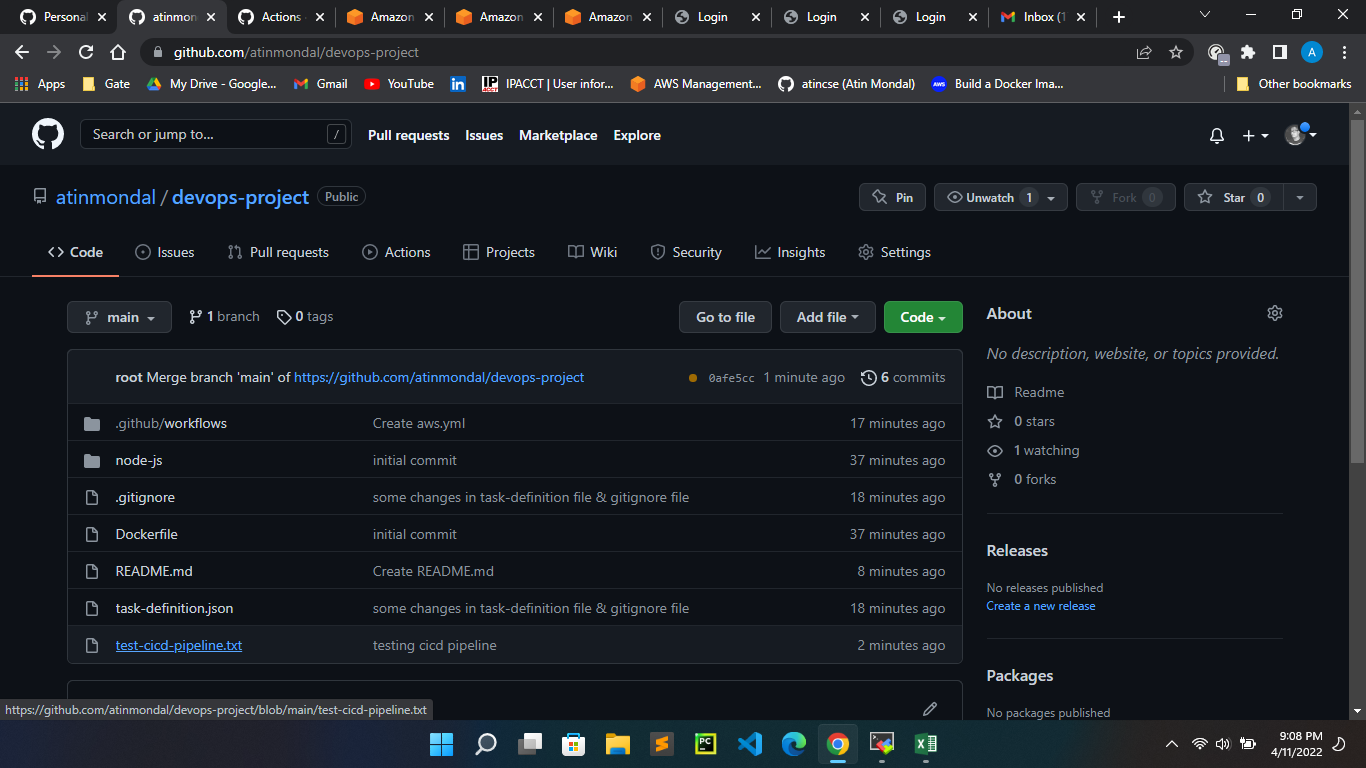
**Lets test our pipeline (github action) is working or not for every push operation into github repository**

Create a [test-cicd-pipeline.txt](https://github.com/atinmondal/devops-project/blob/main/test-cicd-pipeline.txt) file and push it to our remote repository

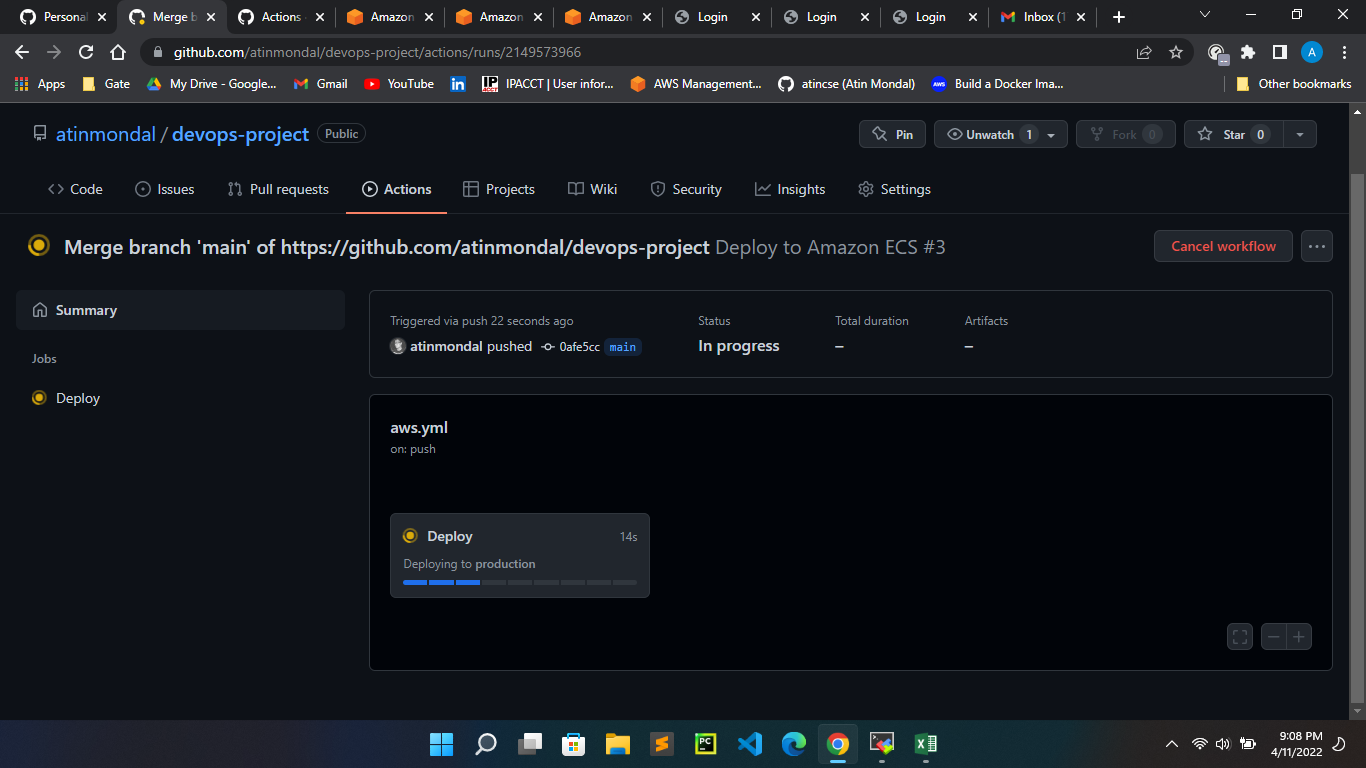


****

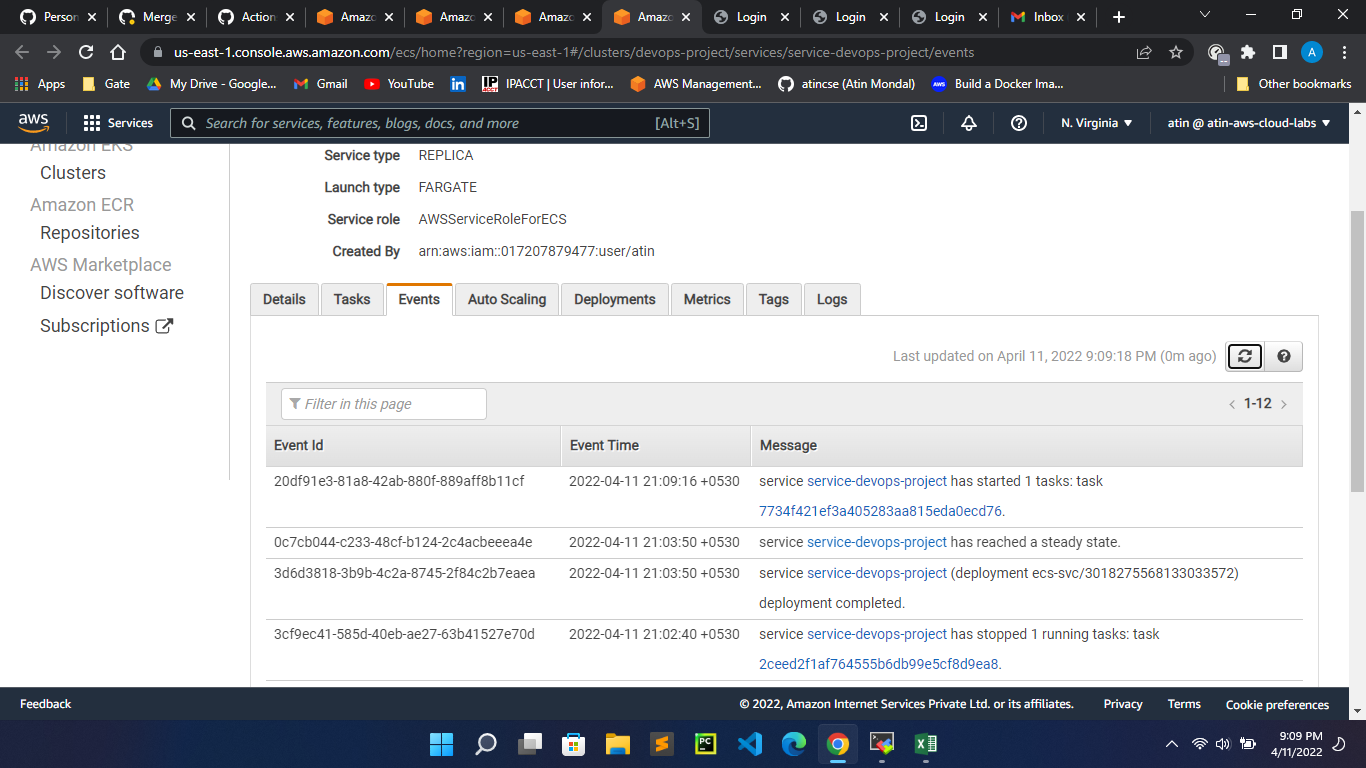
****



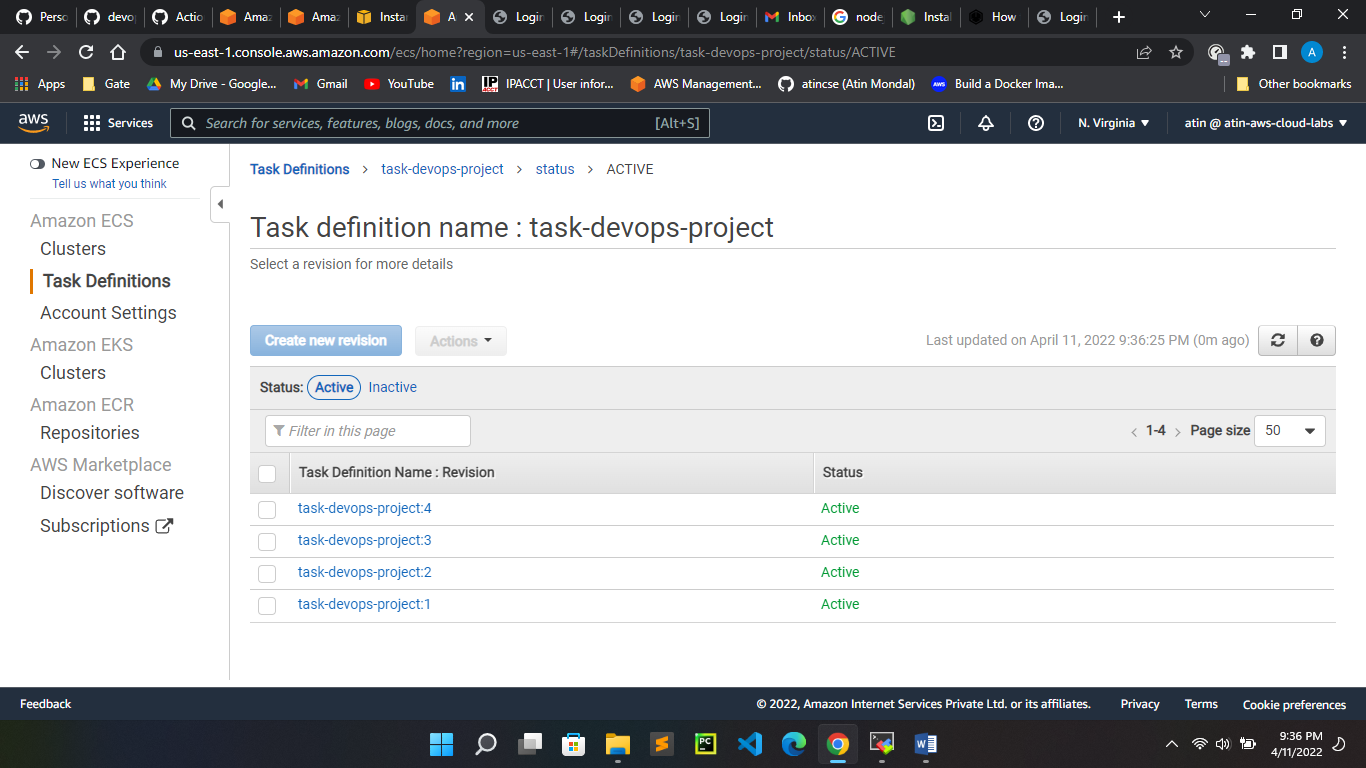
So, our git action triggered automatically after the push operation

****

New service started



New task definition created



Copy public ip (54.208.152.150:8000) of our newly running task and see the o/p in the browser

