**Deployment of Website using Continuous Integration & Continuous Deployment (CICD)**

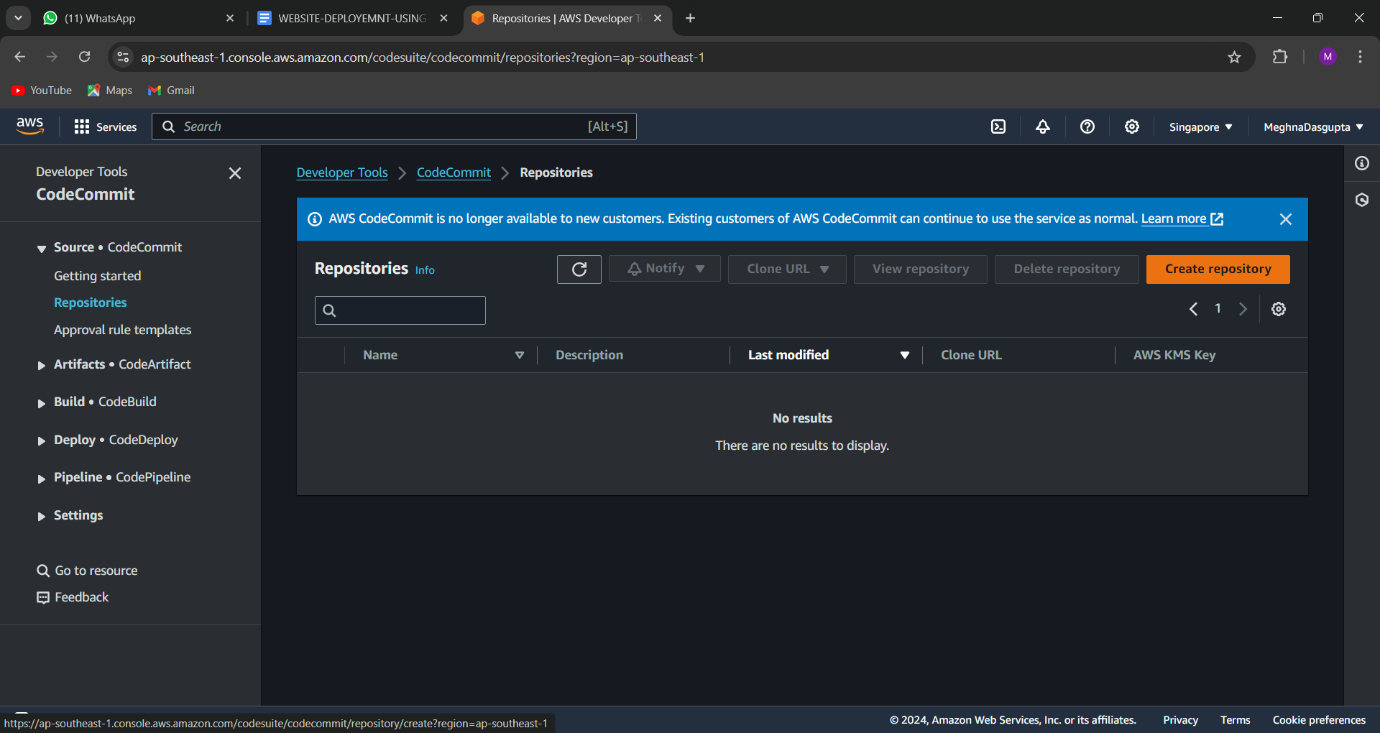
**Objective:**

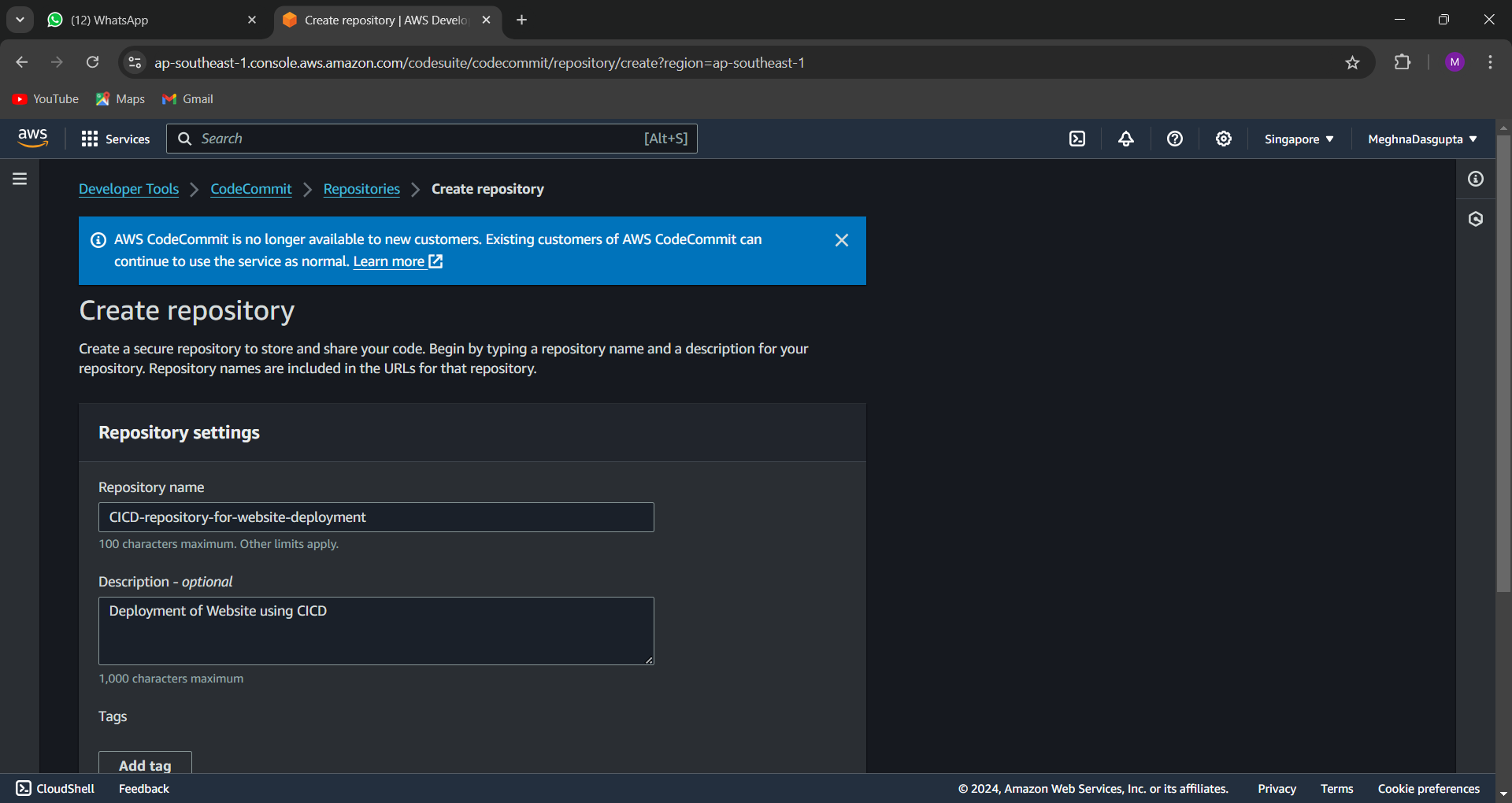
To set up a robust CI/CD pipeline for automating the deployment of a NGINX web server on an EC2 instance using AWS services. This pipeline integrates AWS CodeCommit for seamless source control, CodeBuild for efficient application building, S3 for secure artifact storage, and CodeDeploy for streamlined deployment.

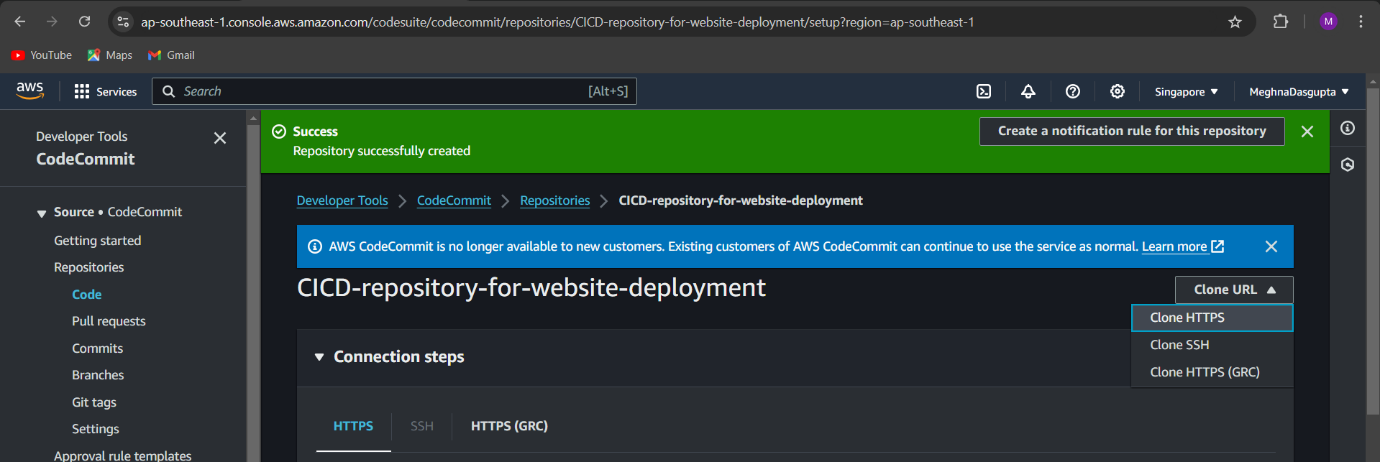
**Steps to Complete the Setup:**

**1. Set Up the CodeCommit Repository**

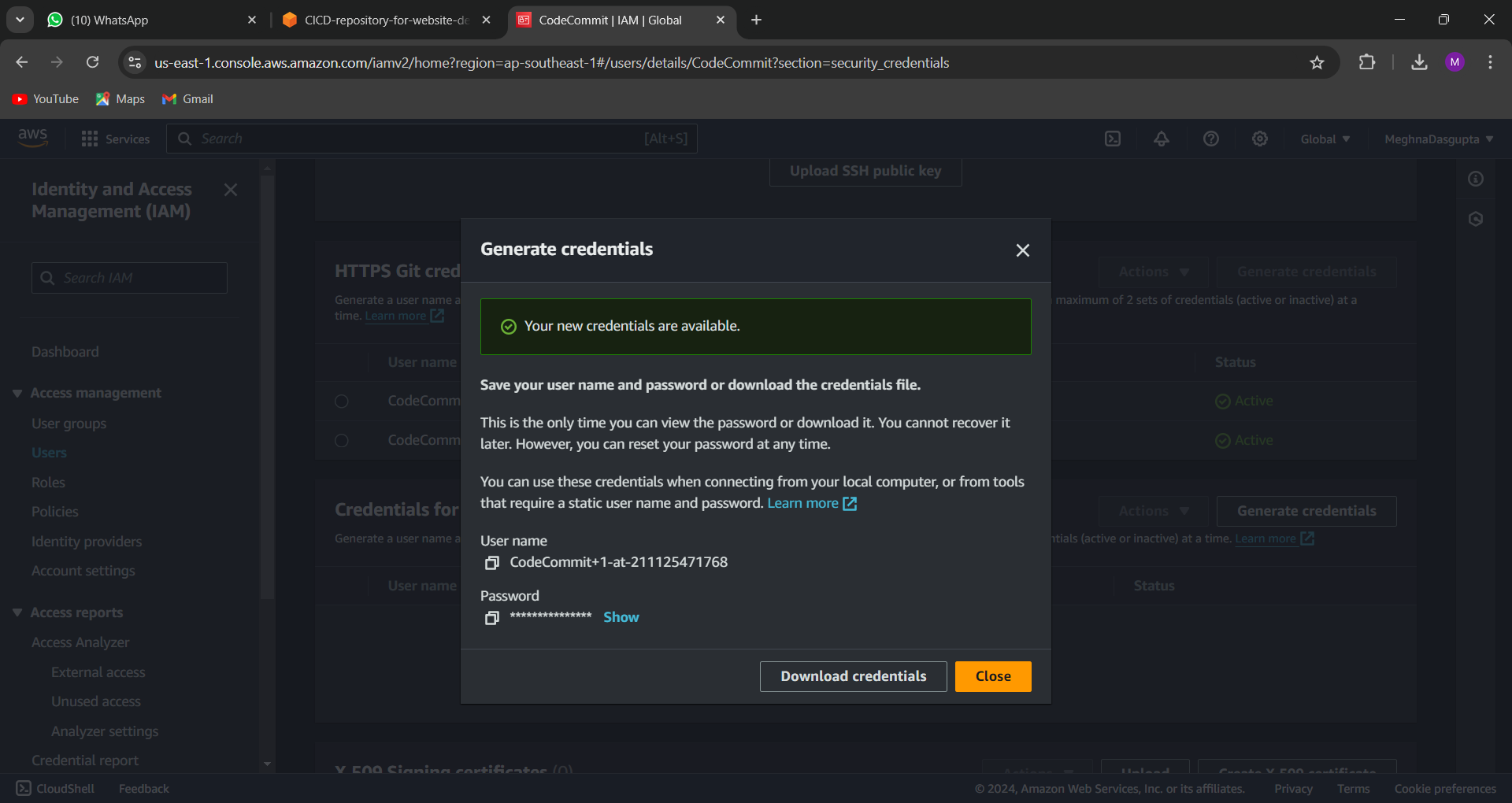
* Create a CodeCommit repository.



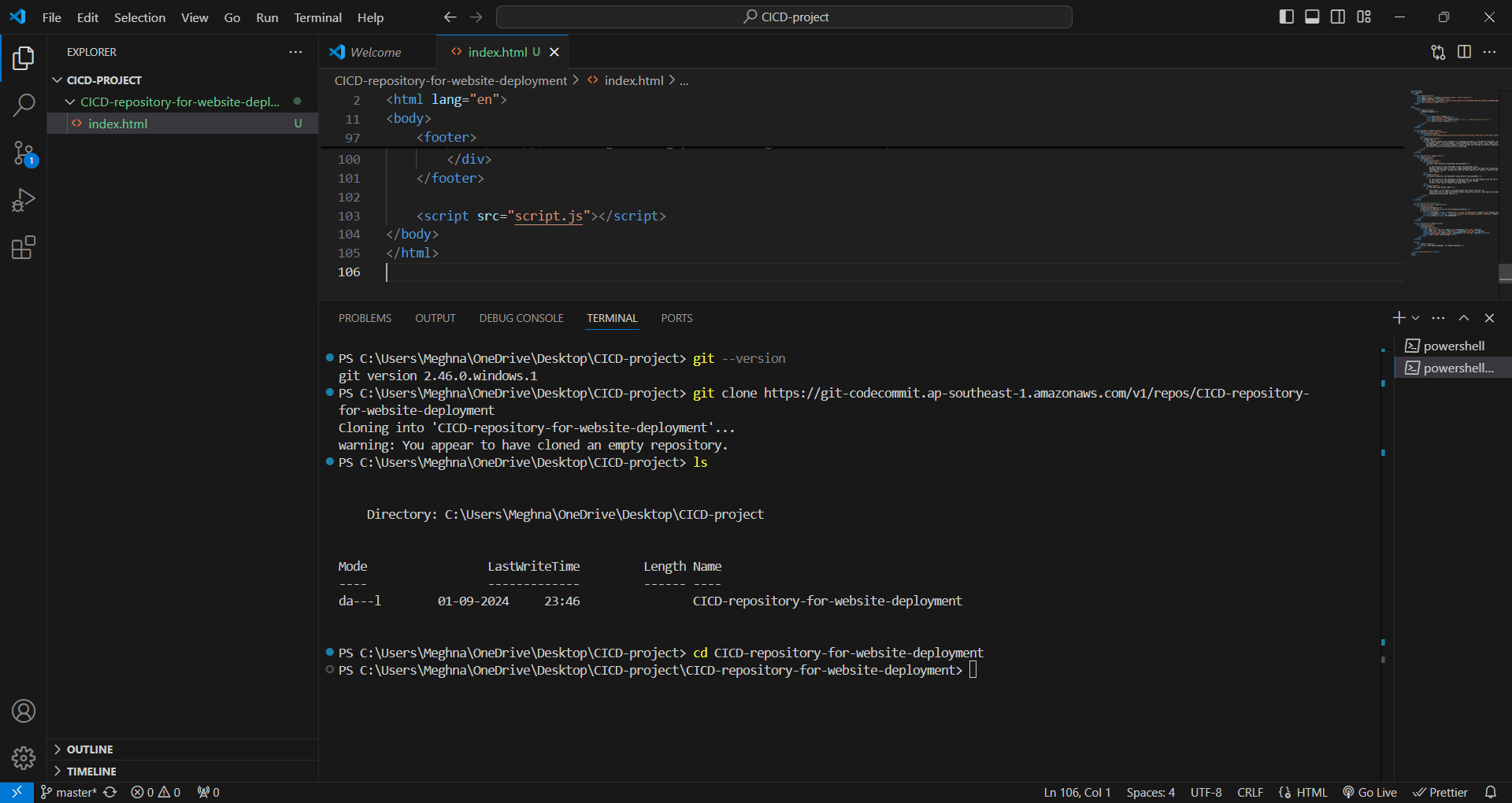


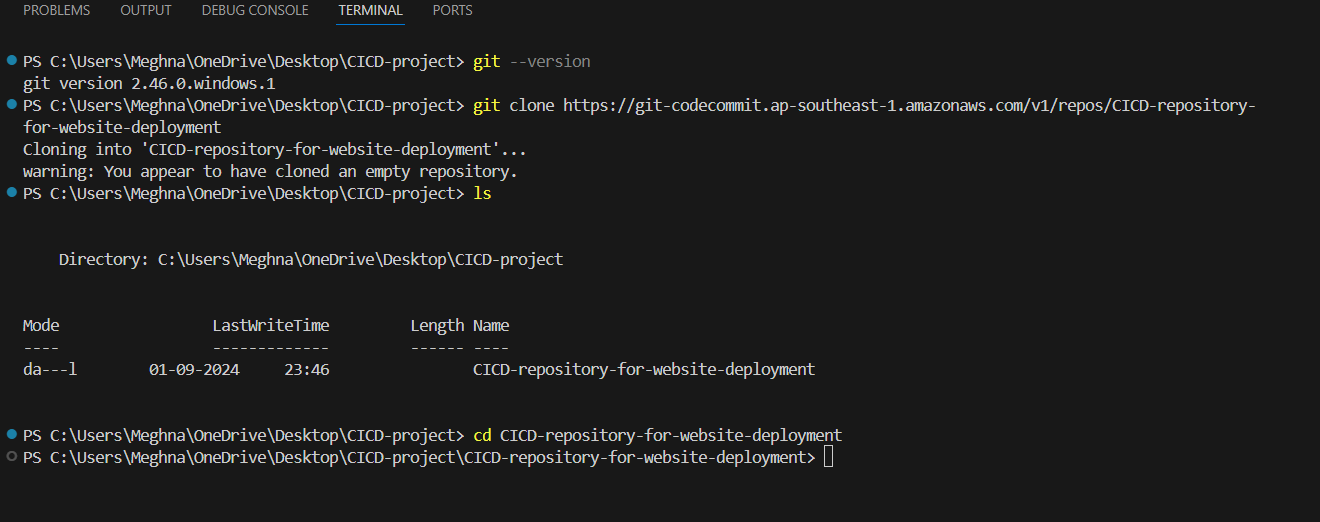


Clone the repository to your local machine:  
Command:: <https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/CICD-repository-for-website-deployment>

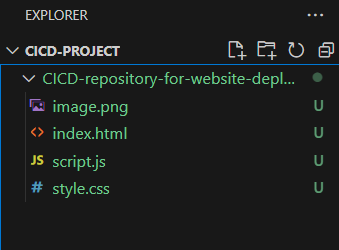




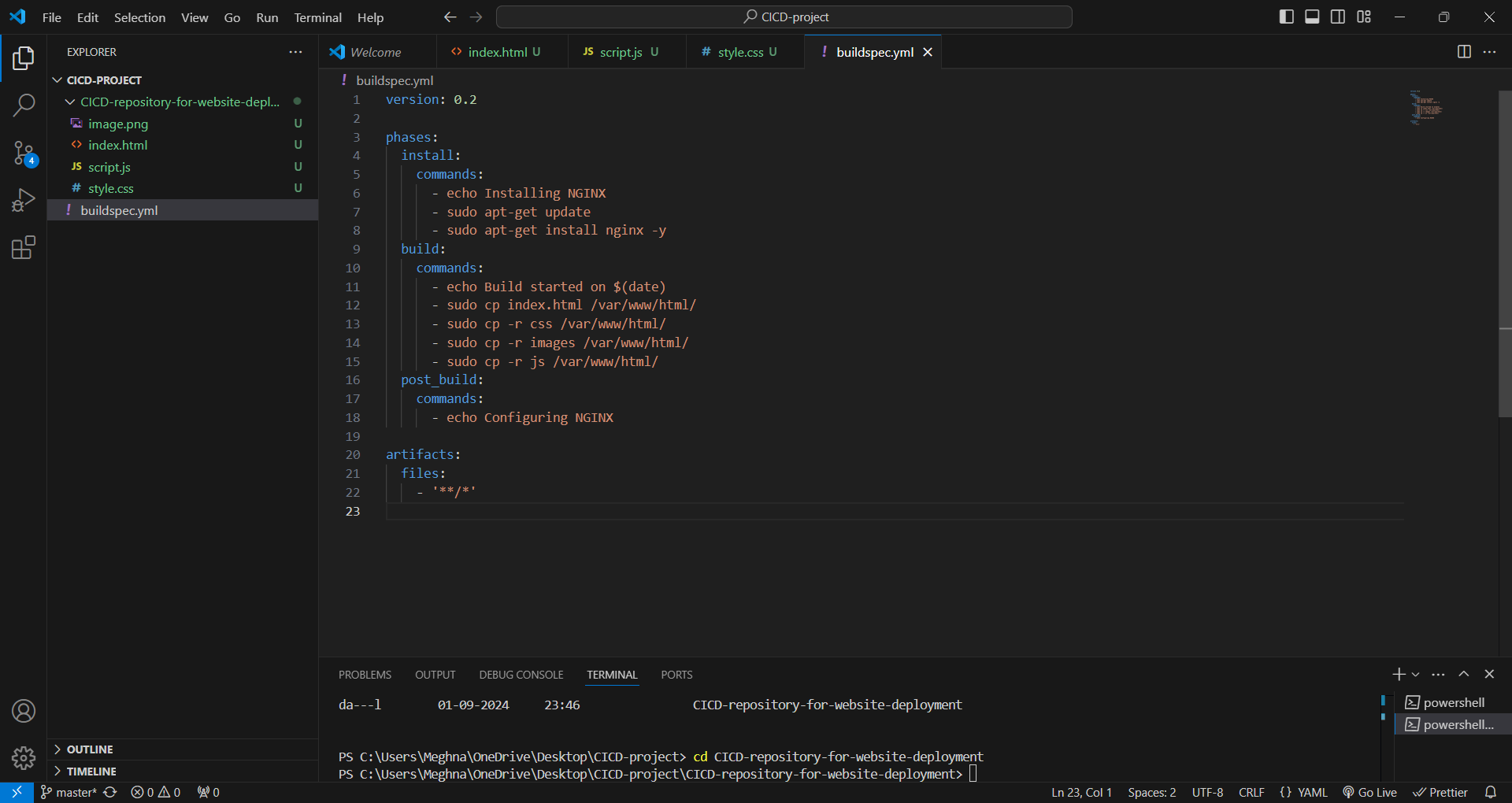




1. **Adding the html,css,images and js files in the CICD repository.**

****

**2. Create Necessary Files**

**buildspec.yml:** Defines the build process for CodeBuild.  
yaml  


version: 0.2

phases:

  install:

    commands:

      - echo Installing NGINX

      - sudo apt-get update

      - sudo apt-get install nginx -y

  build:

    commands:

      - echo Build started on $(date)

      - sudo cp index.html /var/www/html/

      - sudo cp -r css /var/www/html/

      - sudo cp -r images /var/www/html/

      - sudo cp -r js /var/www/html/

  post\_build:

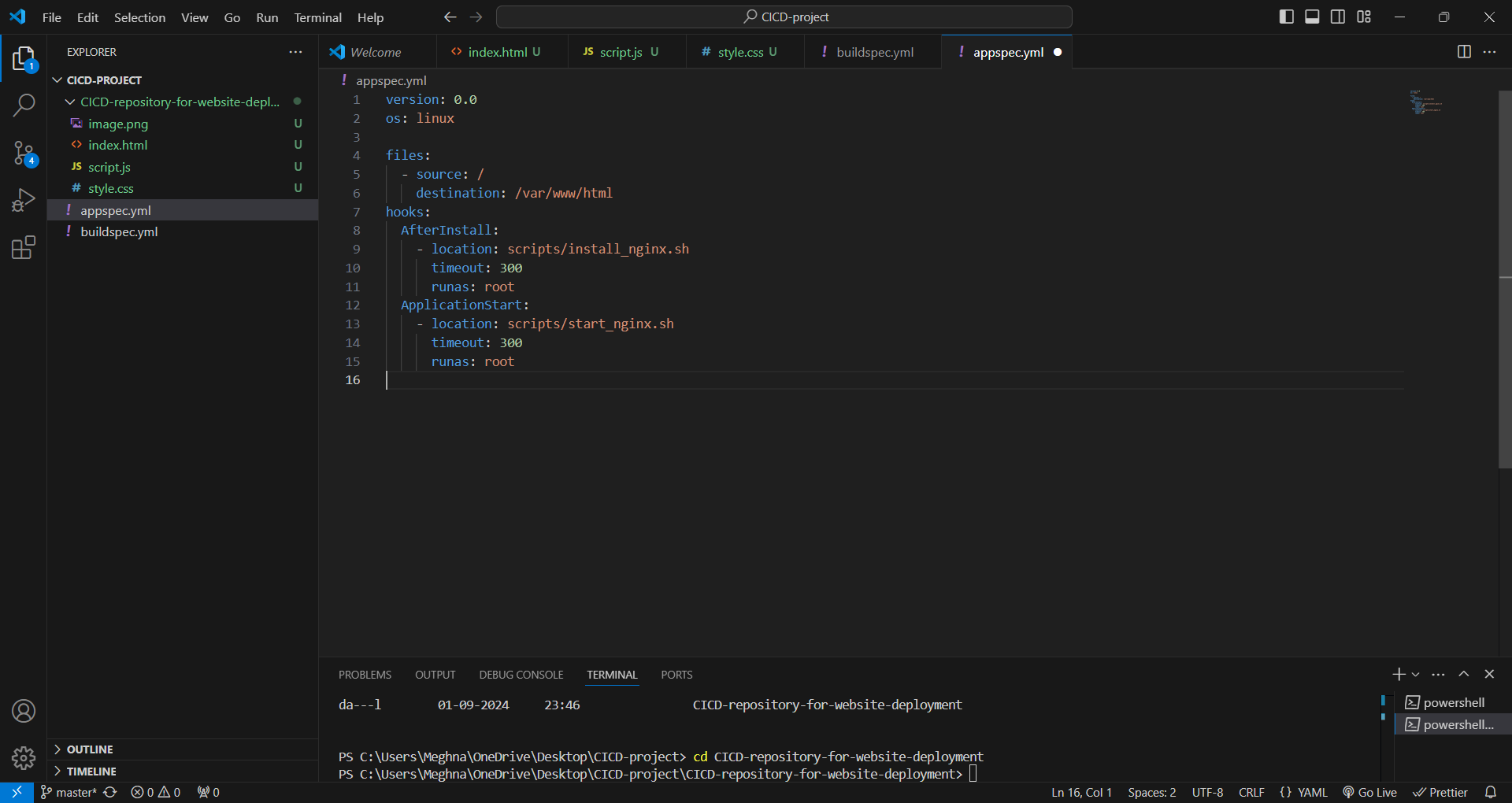
    commands:

      - echo Configuring NGINX

artifacts:

  files:

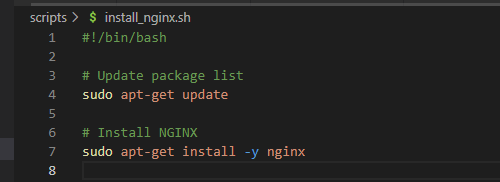
    - '\*\*/\*'

**appspec.yml:** Defines the deployment process for CodeDeploy.  
yaml  
  


**scripts/install-nginx.sh:** Installs NGINX on the EC2 instance.  
  
#!/bin/bash

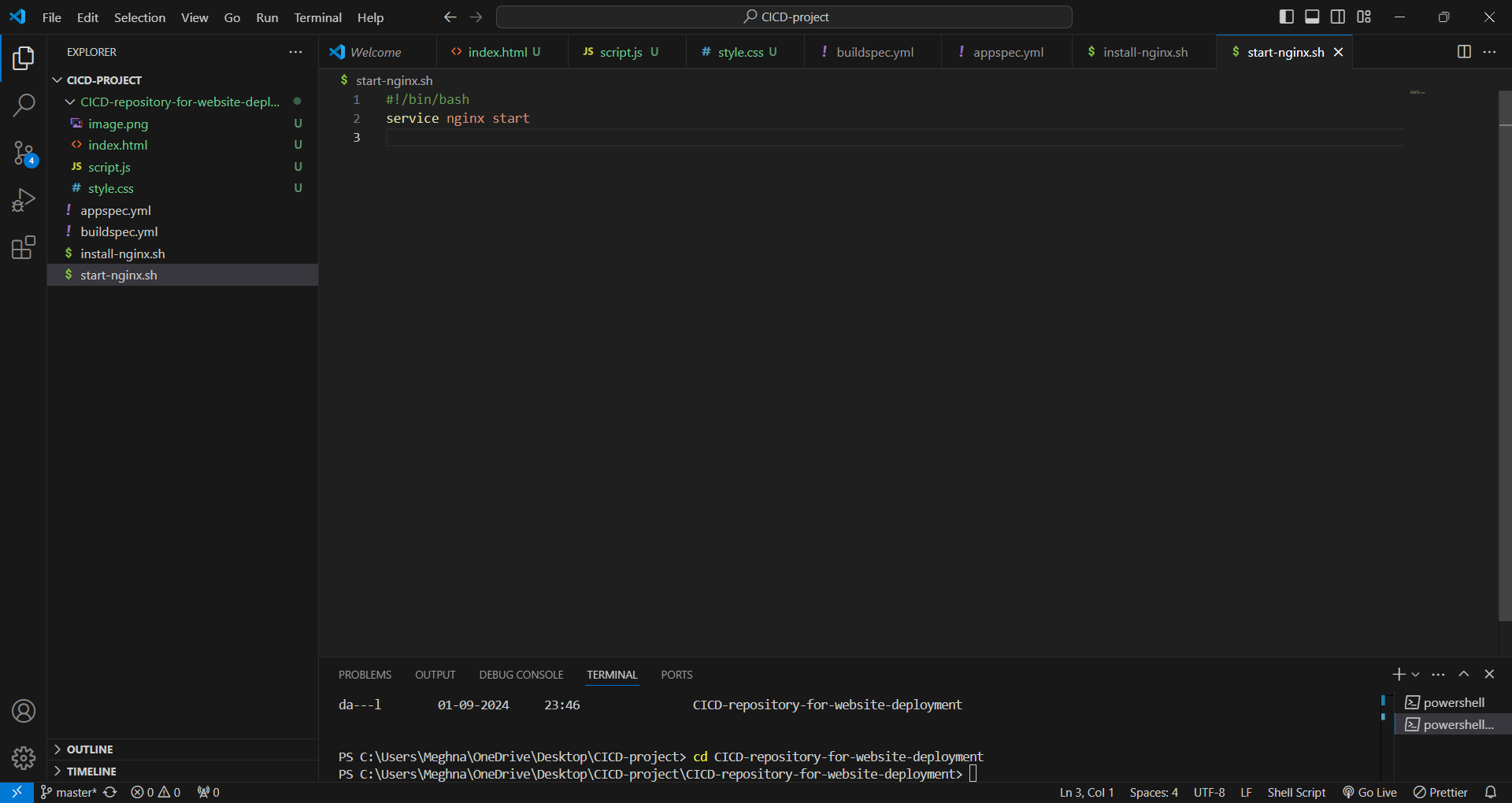
apt-get update

apt-get install -y nginx



**scripts/start-nginx.sh:** Starts the NGINX service.  
  
  
#!/bin/bash

service nginx start

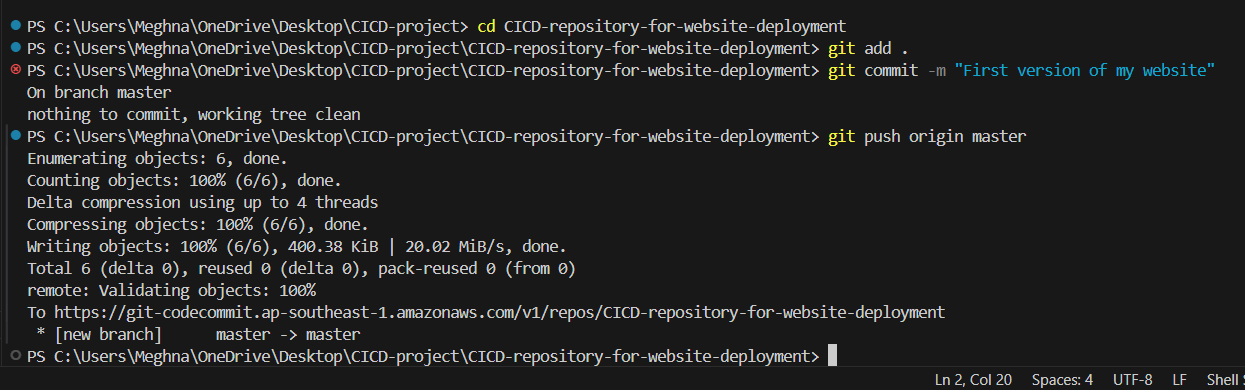


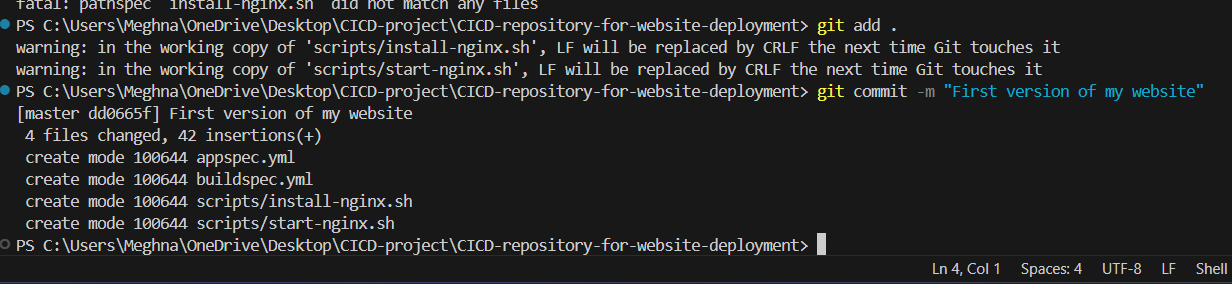
**3. Push Files to CodeCommit**

Add the files to your repository and push to the main branch:  
Command::  
git add .

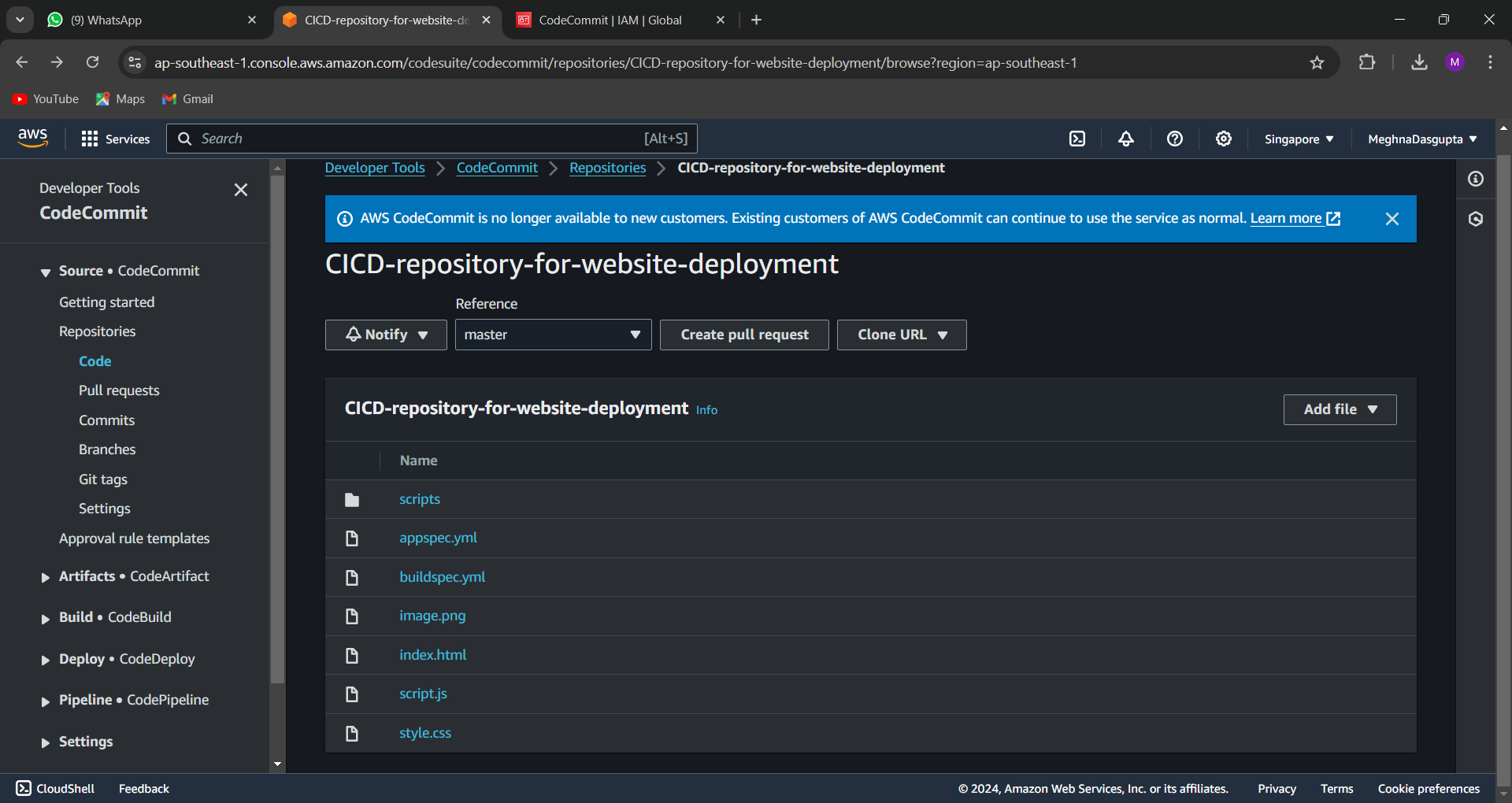
git commit -m "Initial commit"

git push origin main



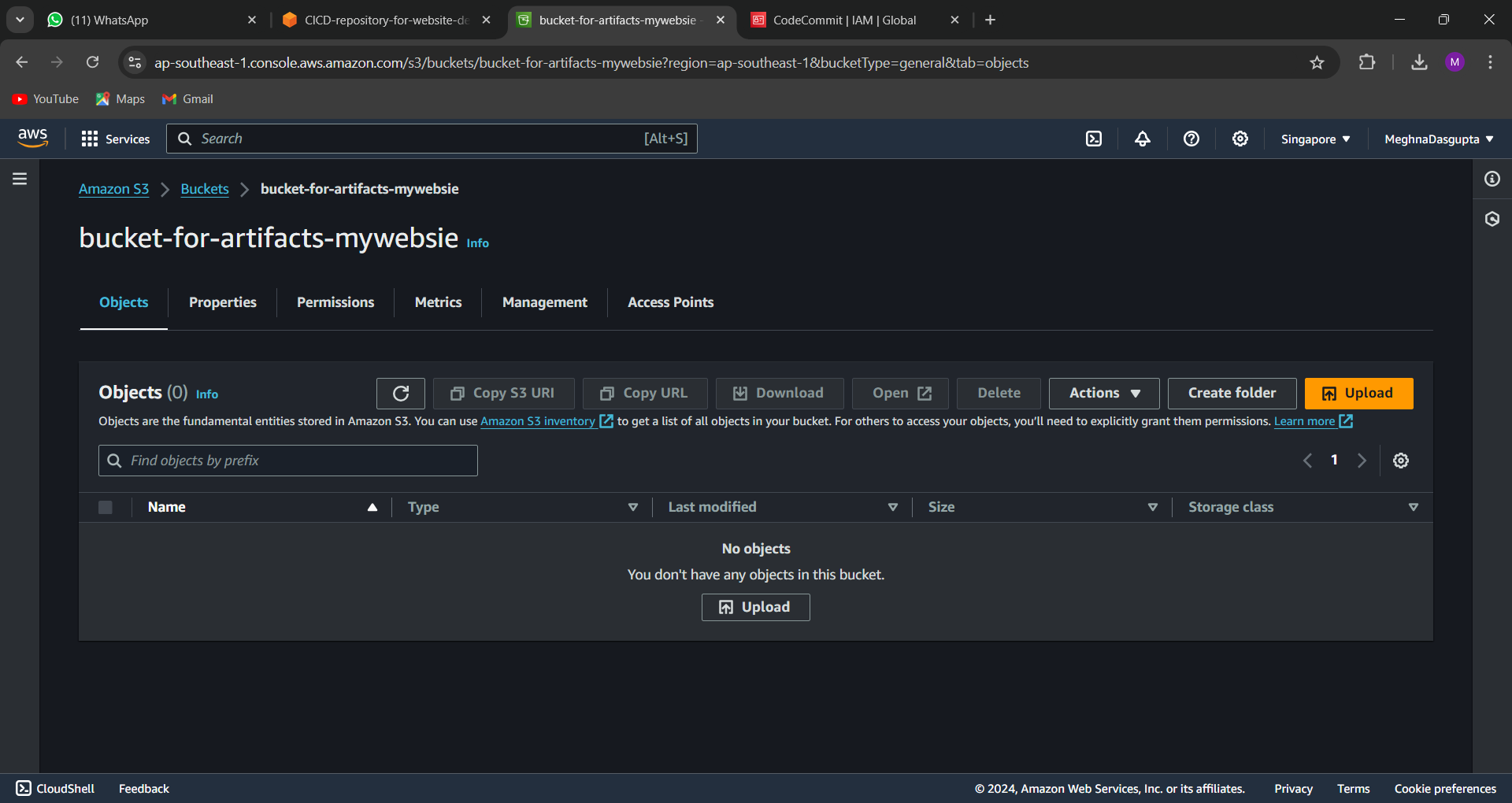


Checking the CodeCommit Repository



**4. Set Up S3 Bucket**

* Create an S3 bucket to store build artifacts.

****

**5. Configure CodeBuild**

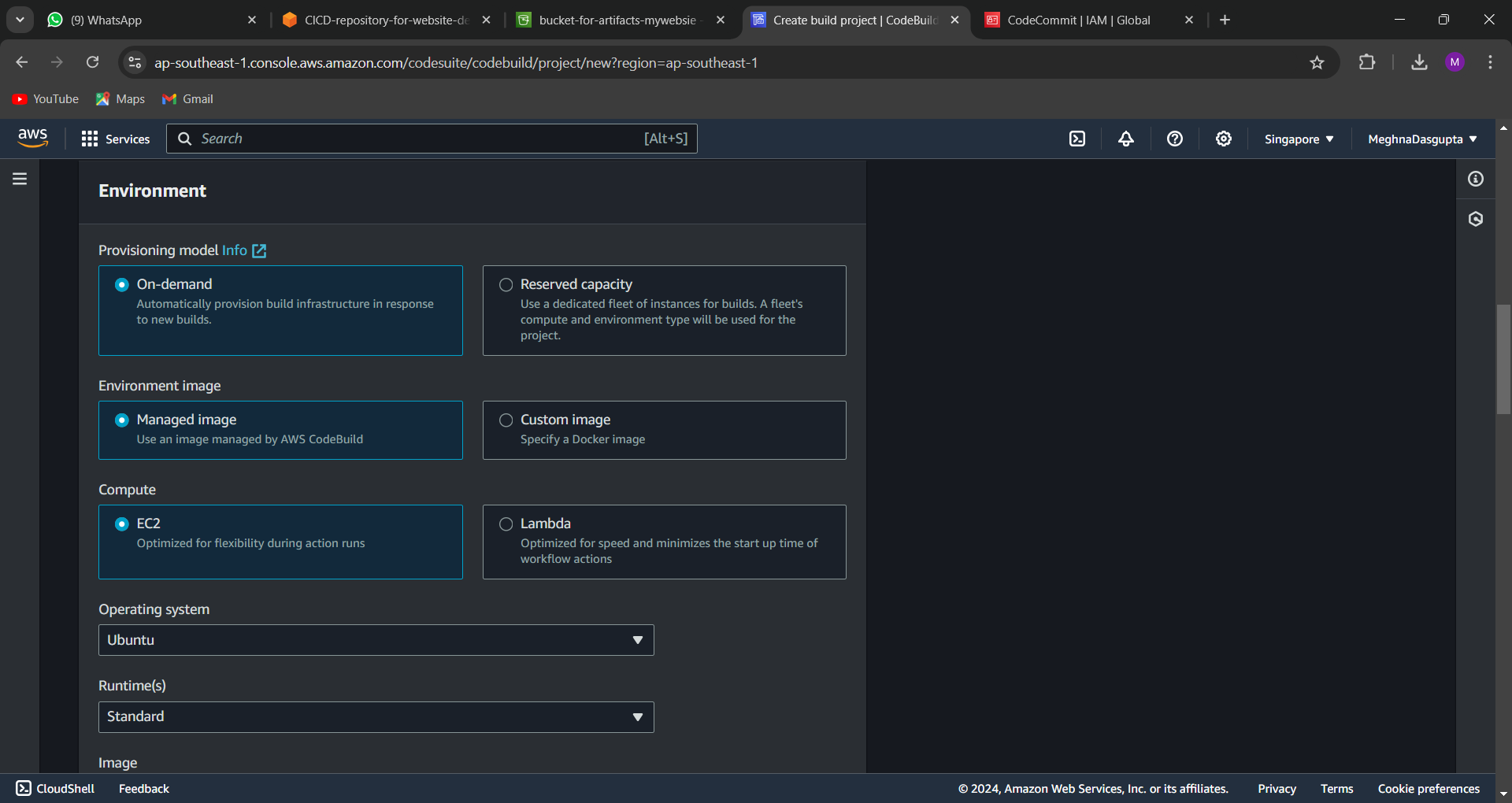
* Create a CodeBuild project:

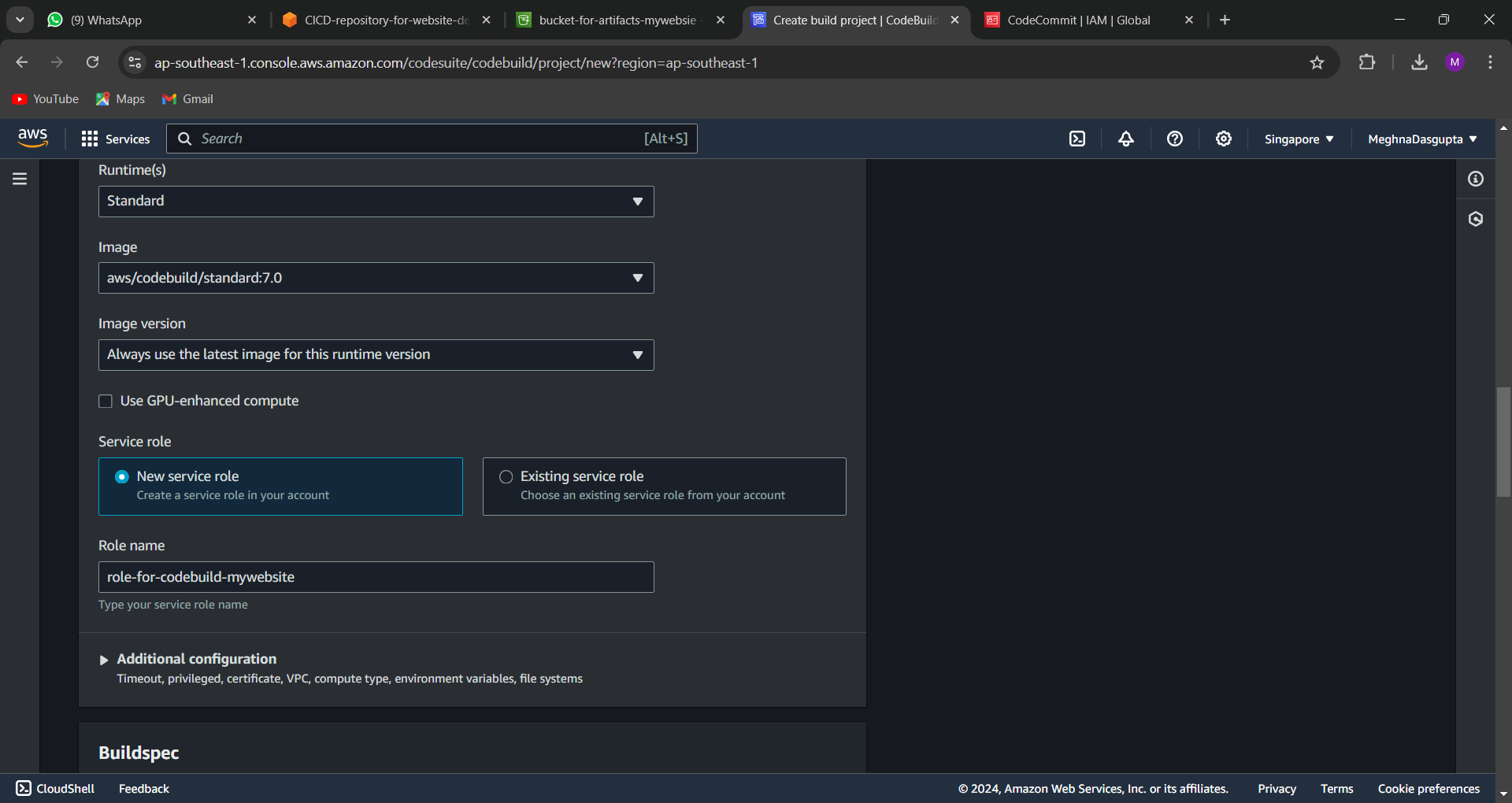
**Source:** Link it to your CodeCommit repository.

* **Buildspec:** Use the buildspec.yml from your repository.
* **Artifacts:** Store the output in the S3 bucket myapp-artifacts-bucket.

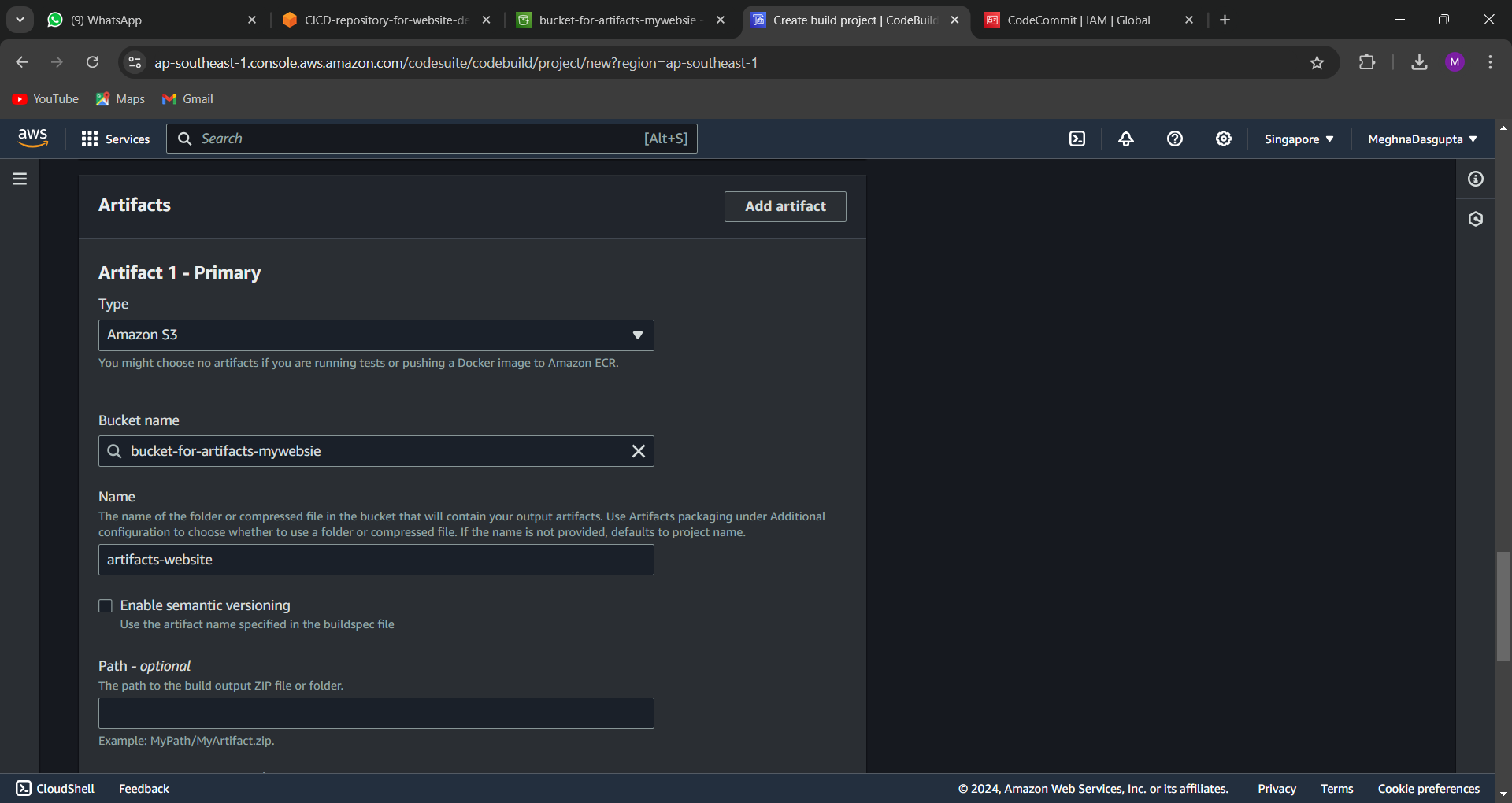
****

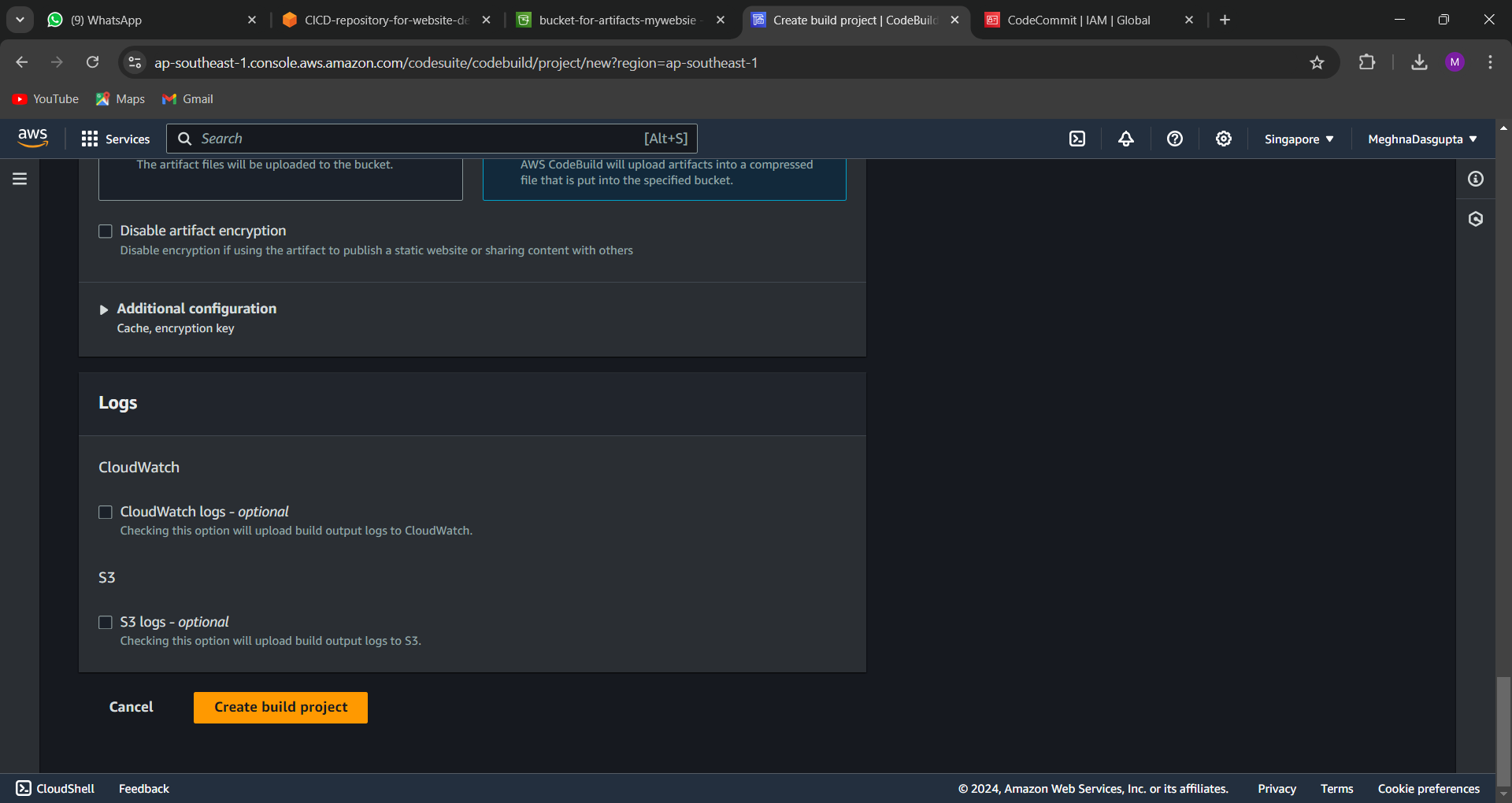
****

****

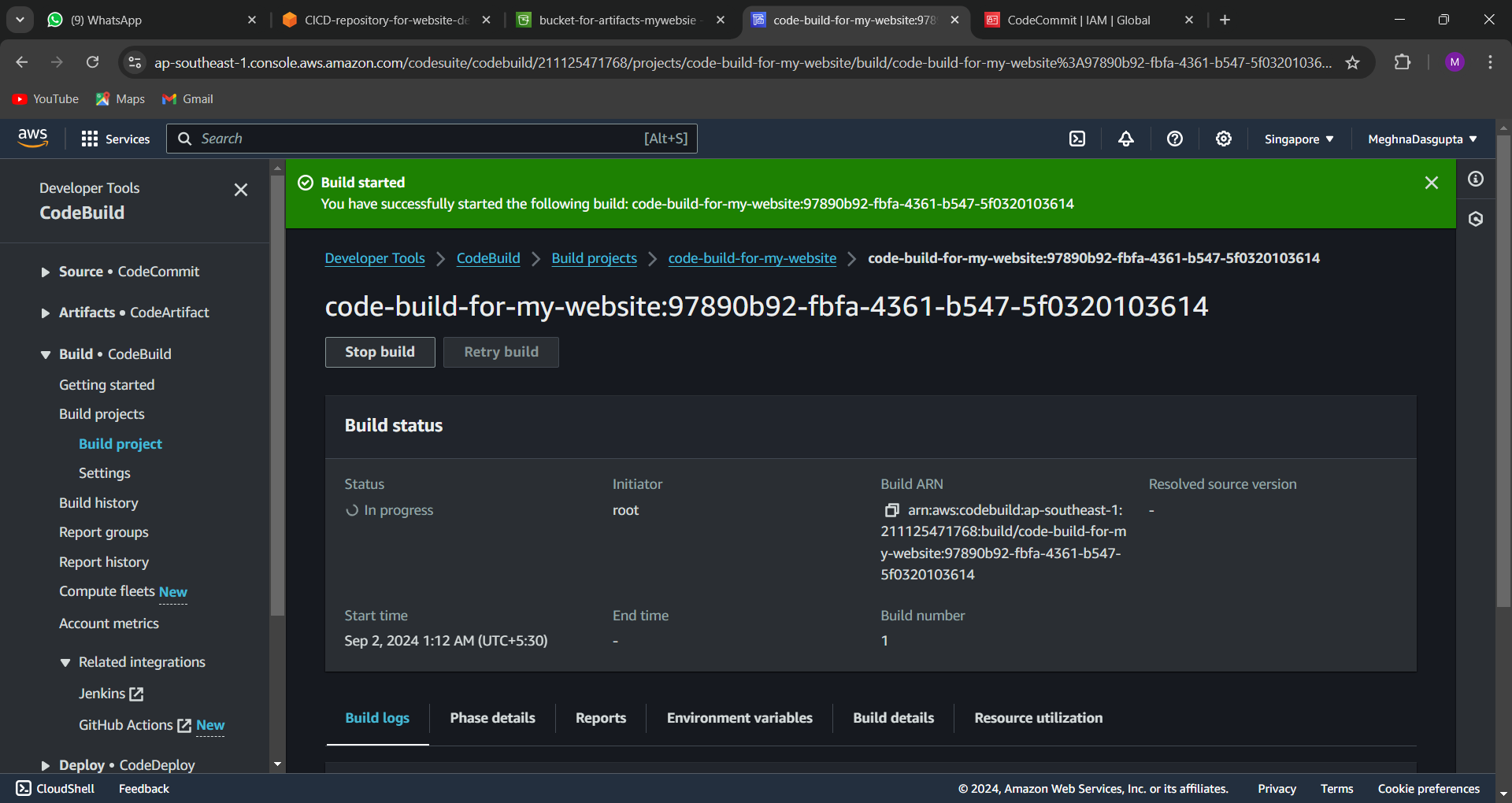
****

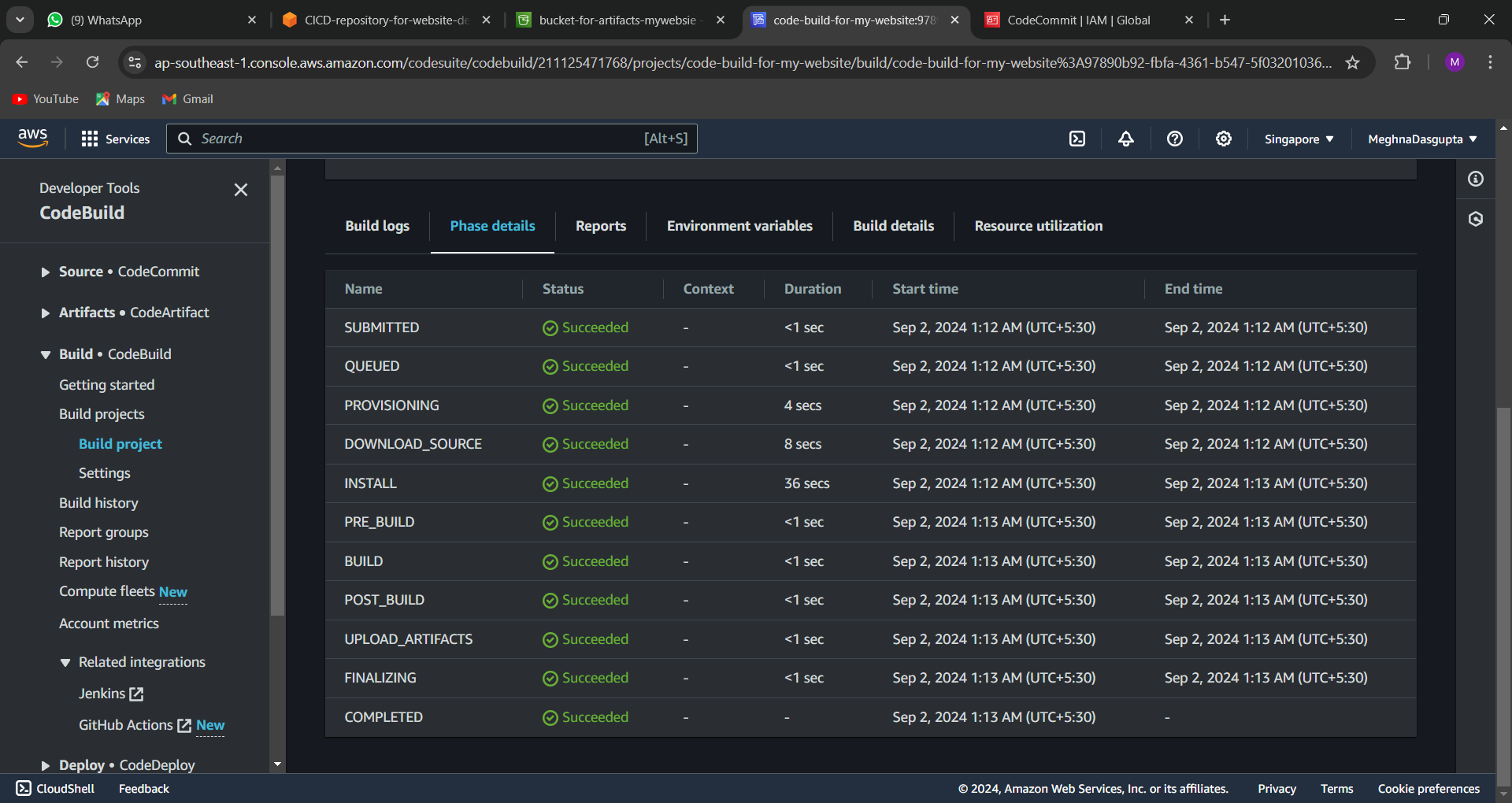
****

****

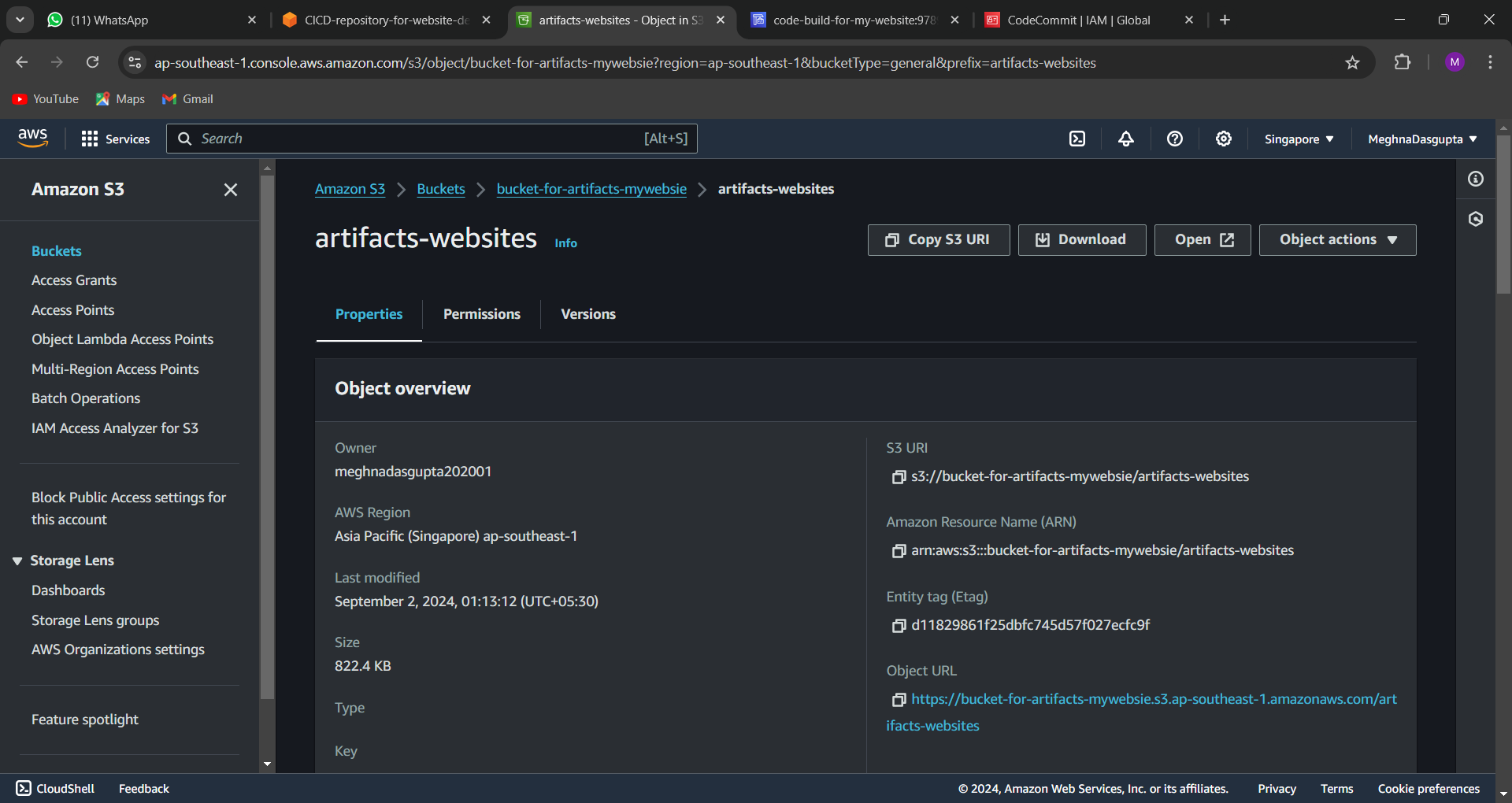
****

Start build :

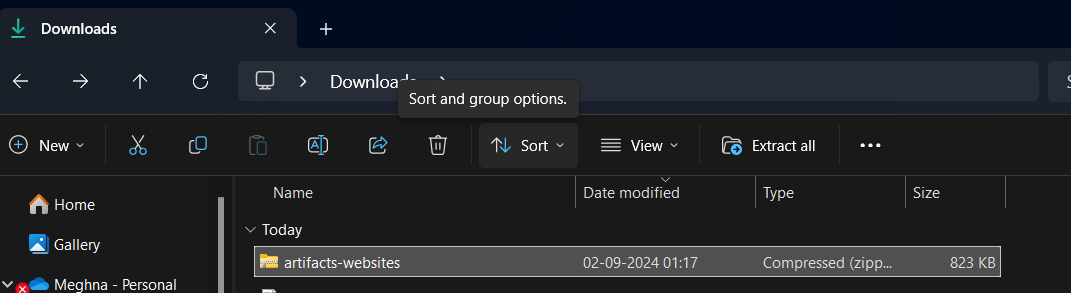


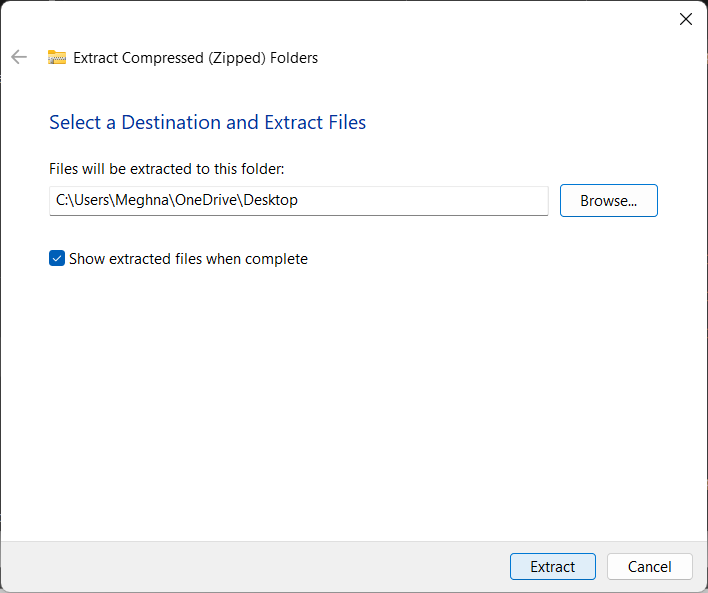


After completion check to the s3 bucket new folder artifacts will be created in the zip format



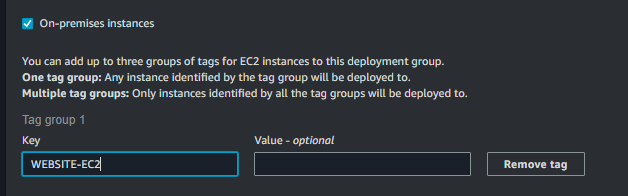
Check the files by downloading it



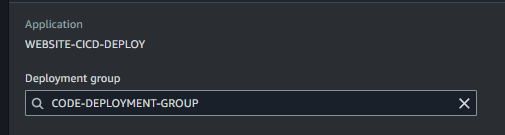


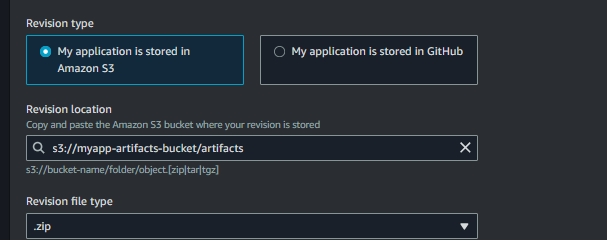
**6. Configure CodeDeploy**

* Create a CodeDeploy application and deployment group:
  + **Application Name:** MyAppDeploy
  + **Deployment Group Name:** MyAppDeployGroup
  + **Service Role:** Ensure it has access to S3, EC2, CodeDeploy, and SNS.
  + **Deployment Type:** EC2/On-premises.
  + **EC2 Tag:** Target instances by tag.

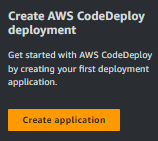


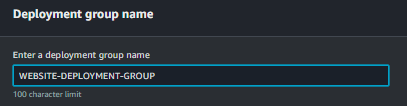
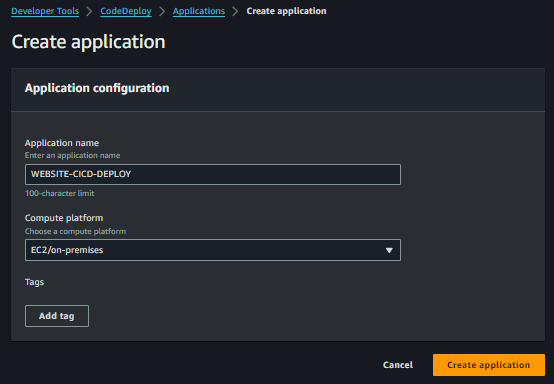




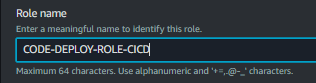
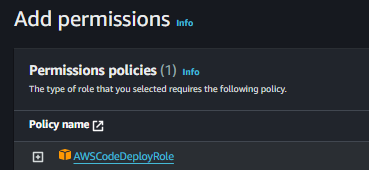
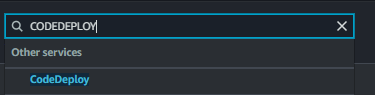


* **Deployment Config:** Use the CodeDeployDefault.OneAtATime configuration.

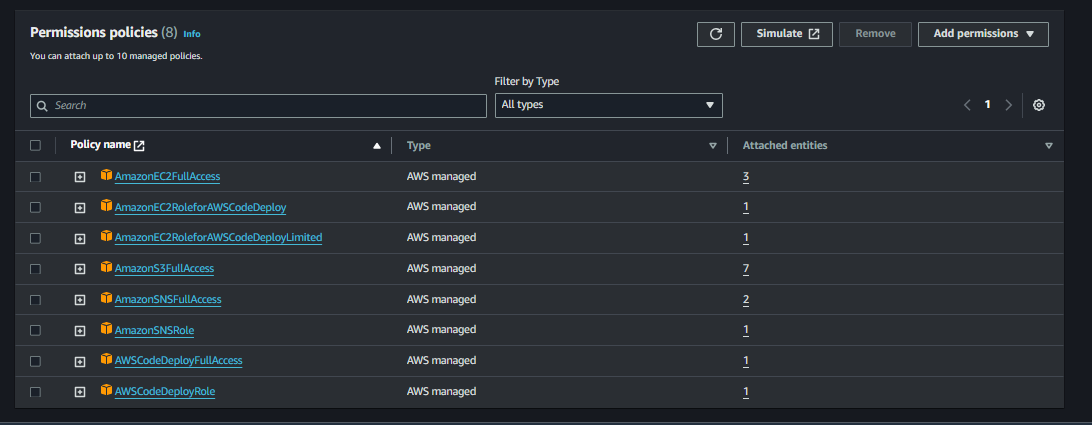




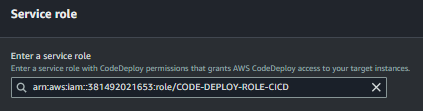
Create a Iam role for the CodeDeploy service

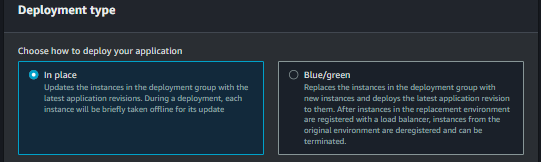


Open the role which you have created and add the necessary permissions



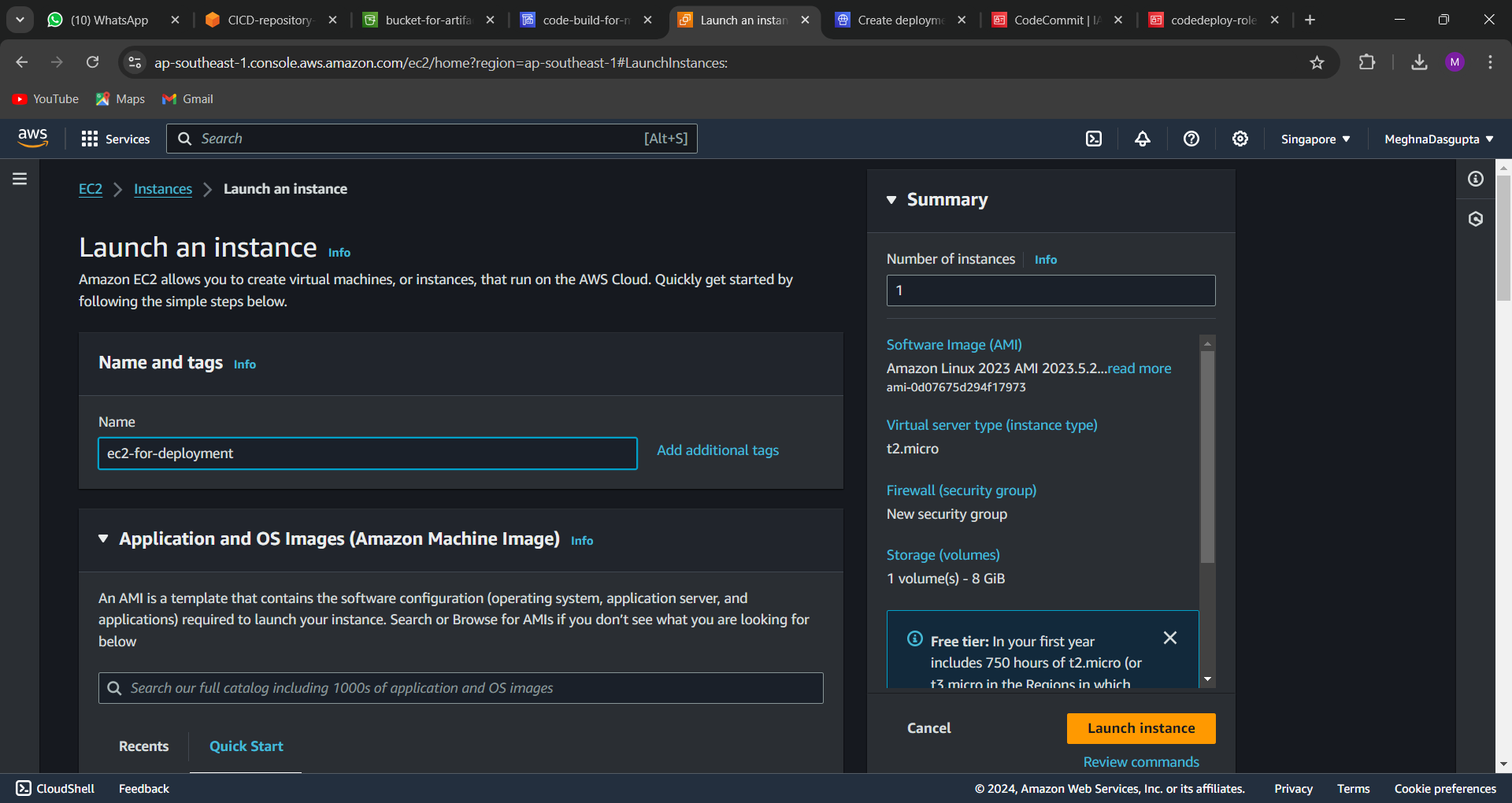
After that continue with your deployment group creation and add service role

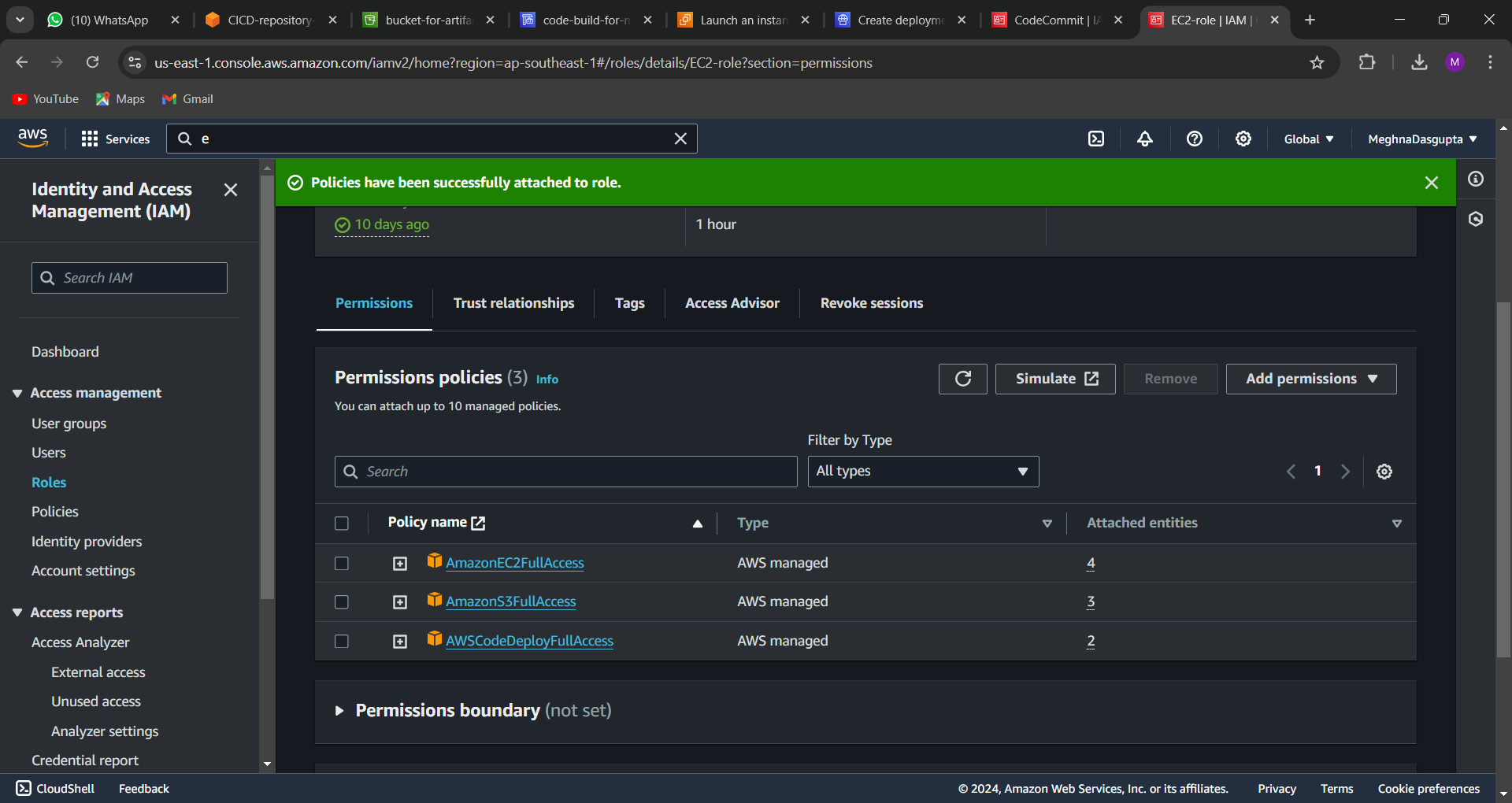


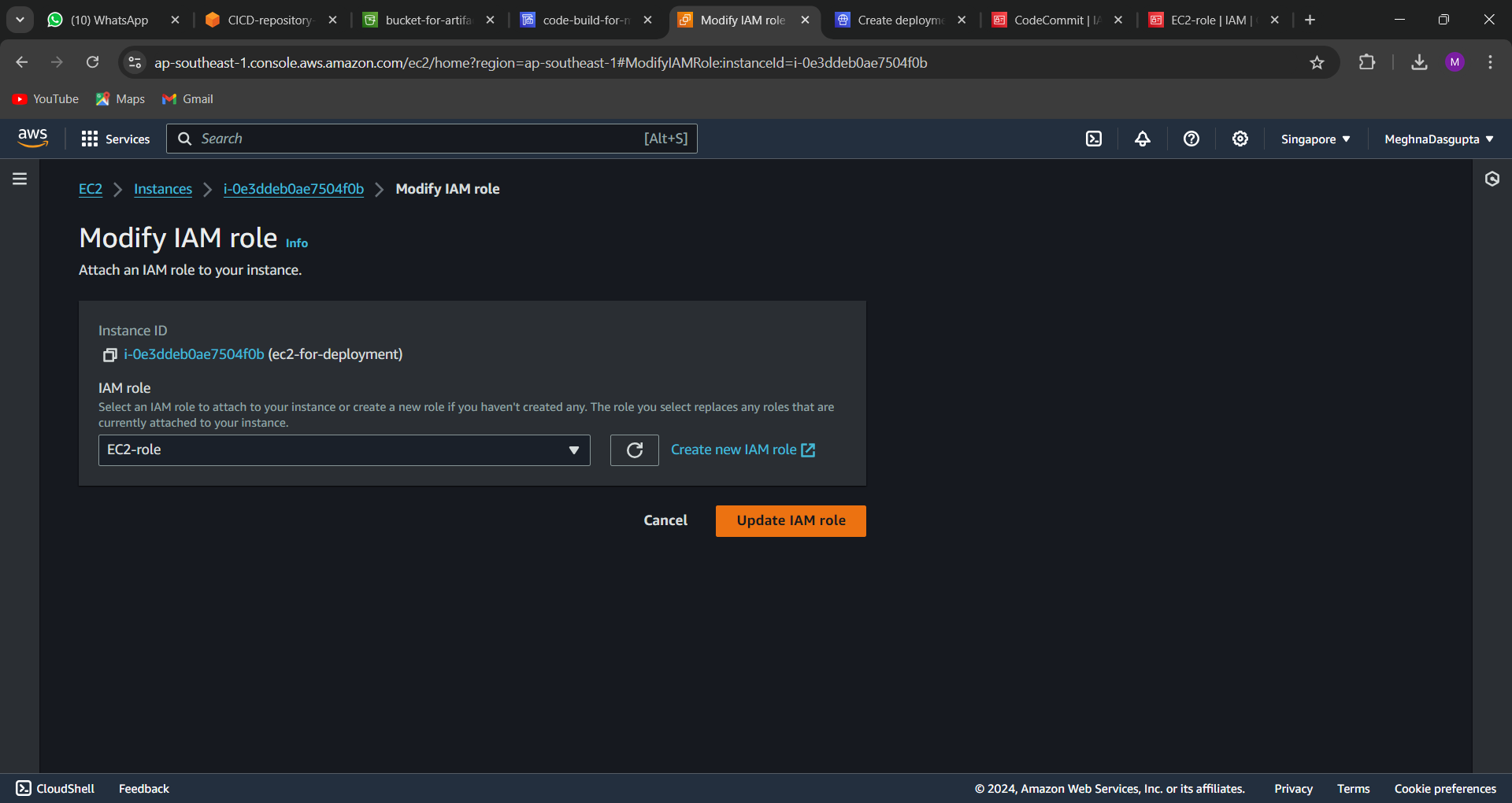


**7. Install CodeDeploy Agent on EC2**

SSH into your EC2 instance and install the CodeDeploy agent:  
Now create iam role for the ec2 instance



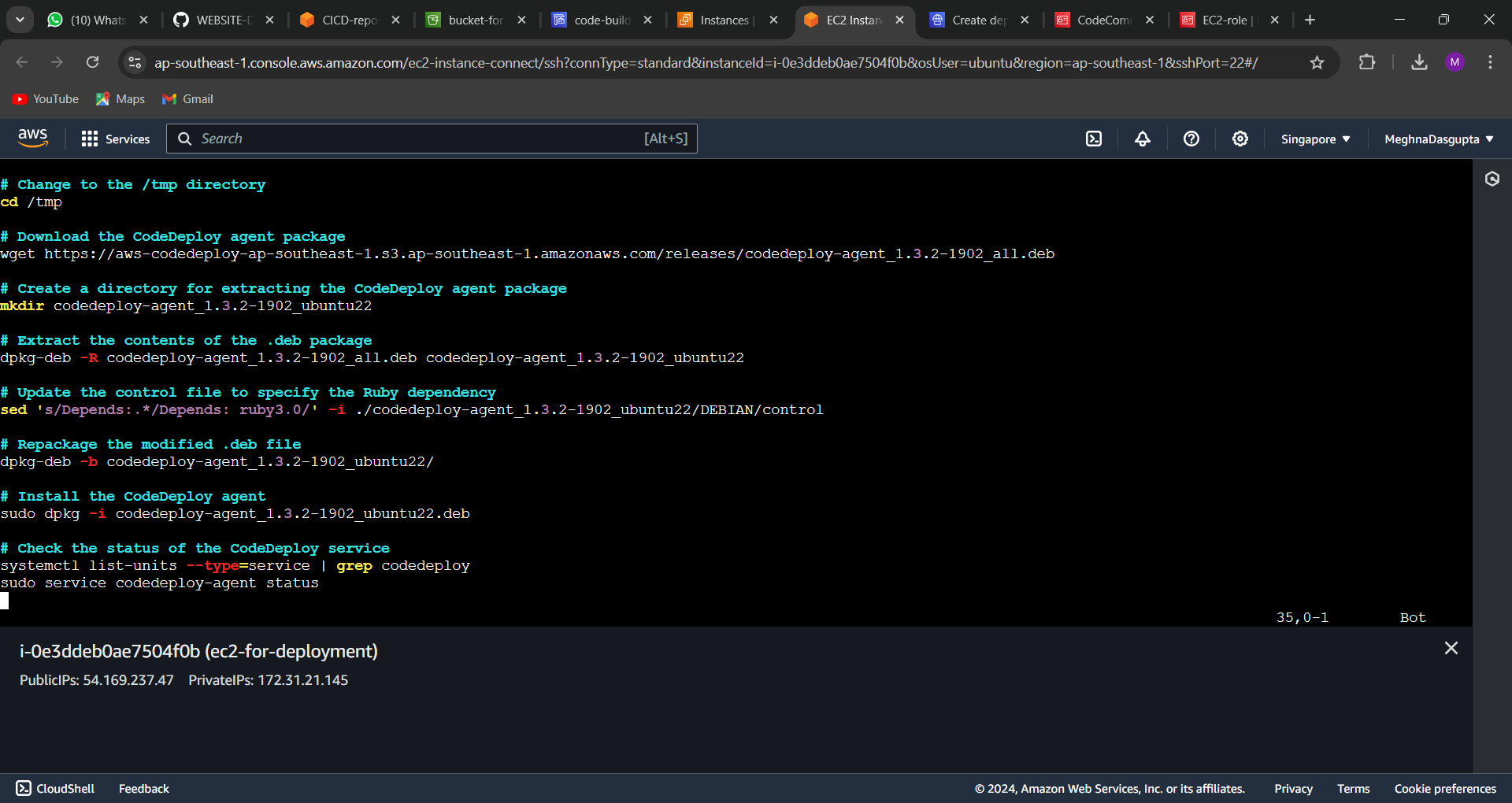


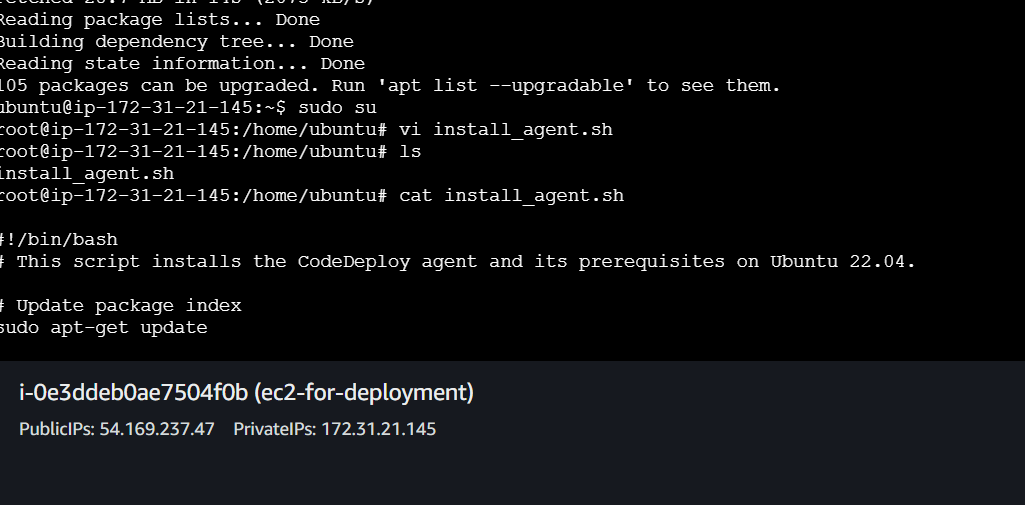


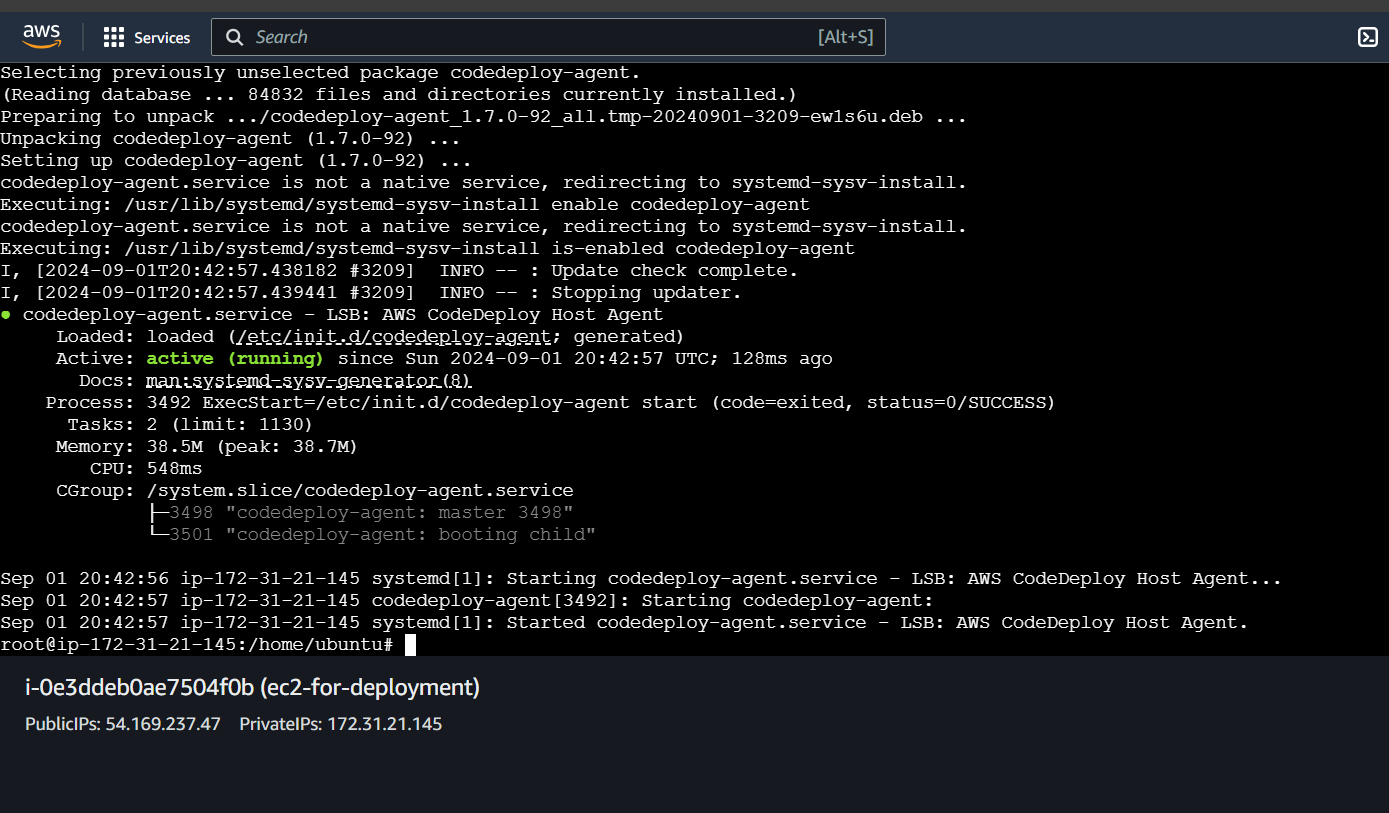
Connect to the instances

Run  the command sudo apt update -y

And vi install\_agent.sh

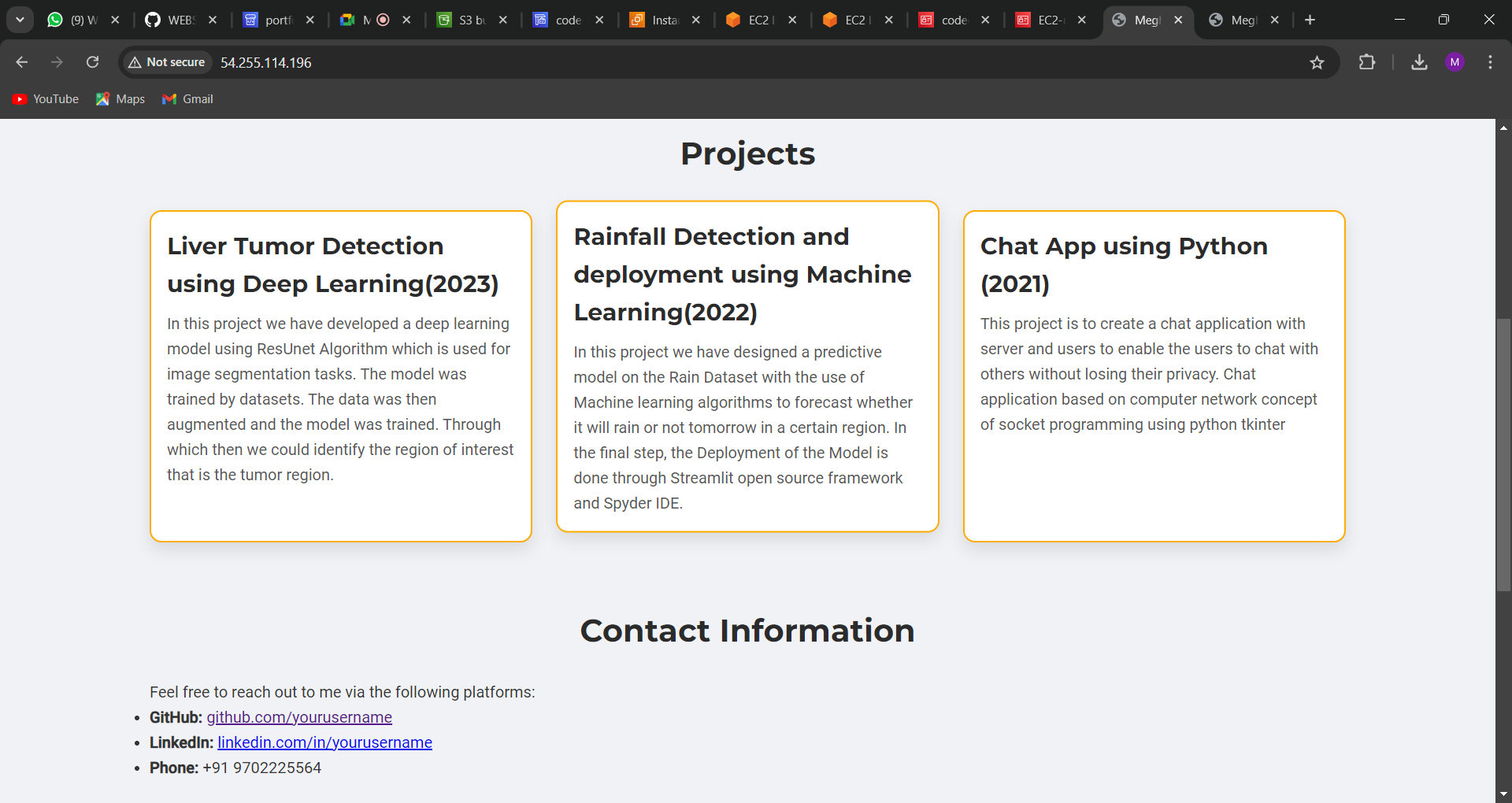




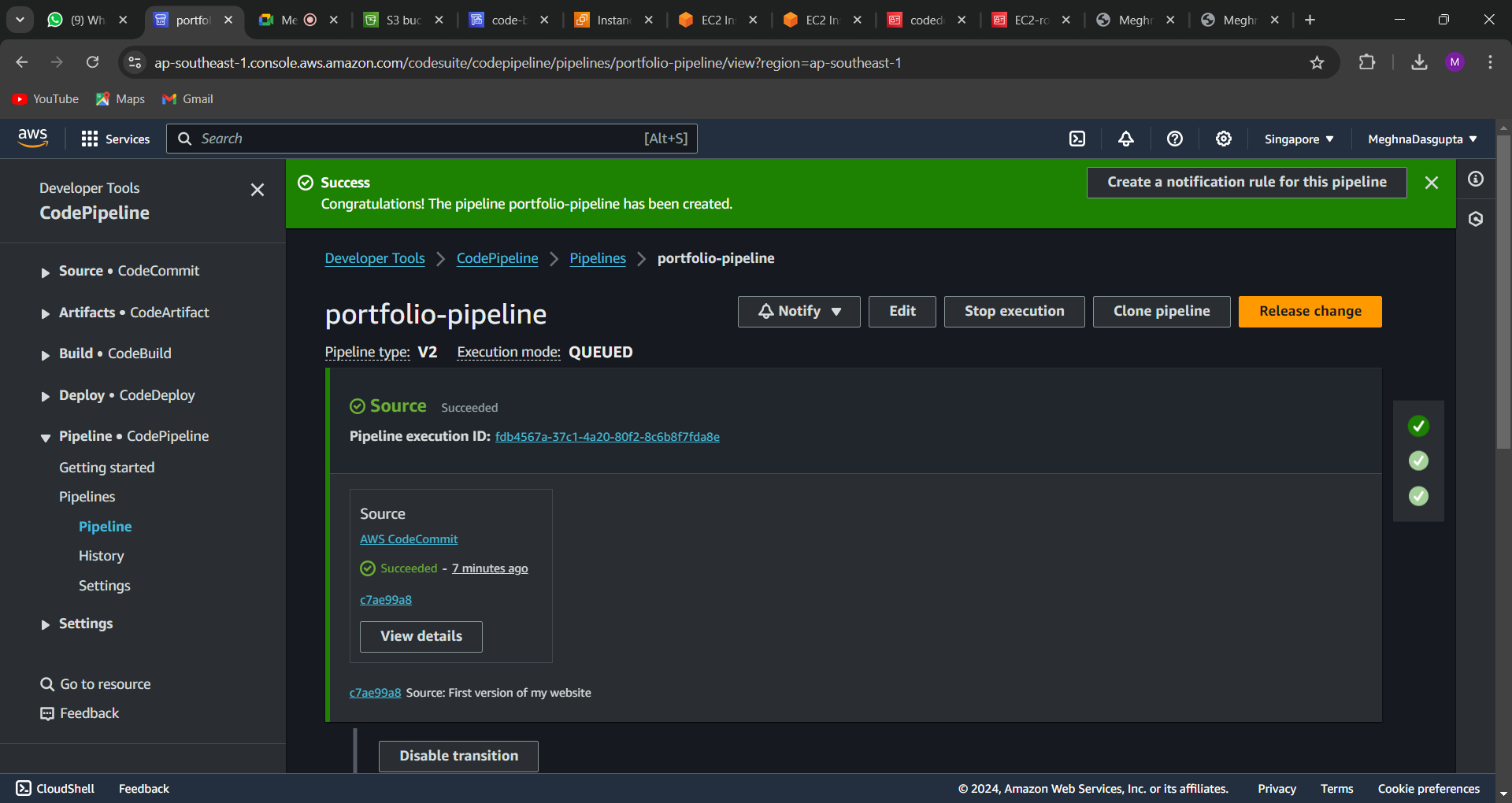
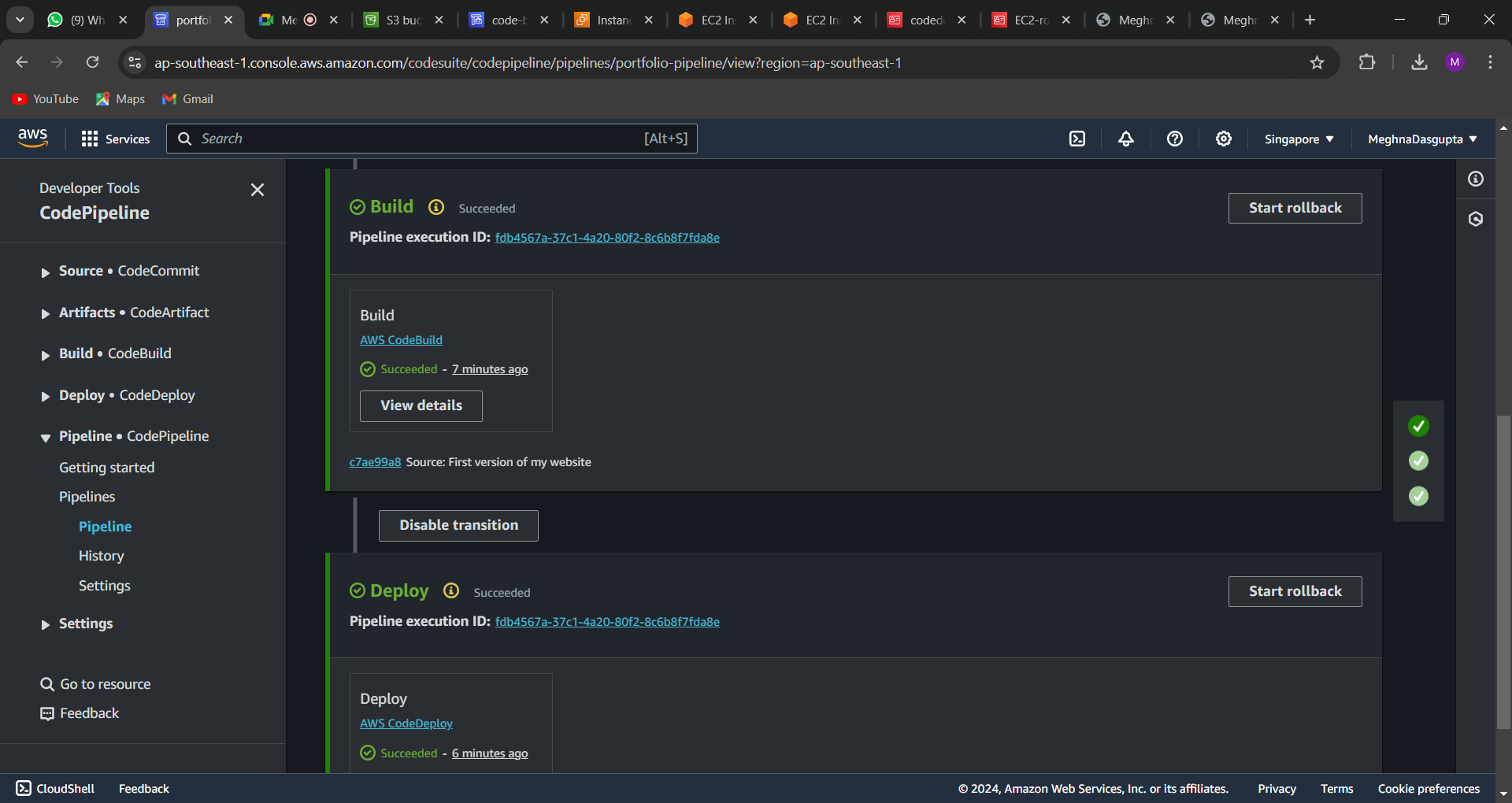


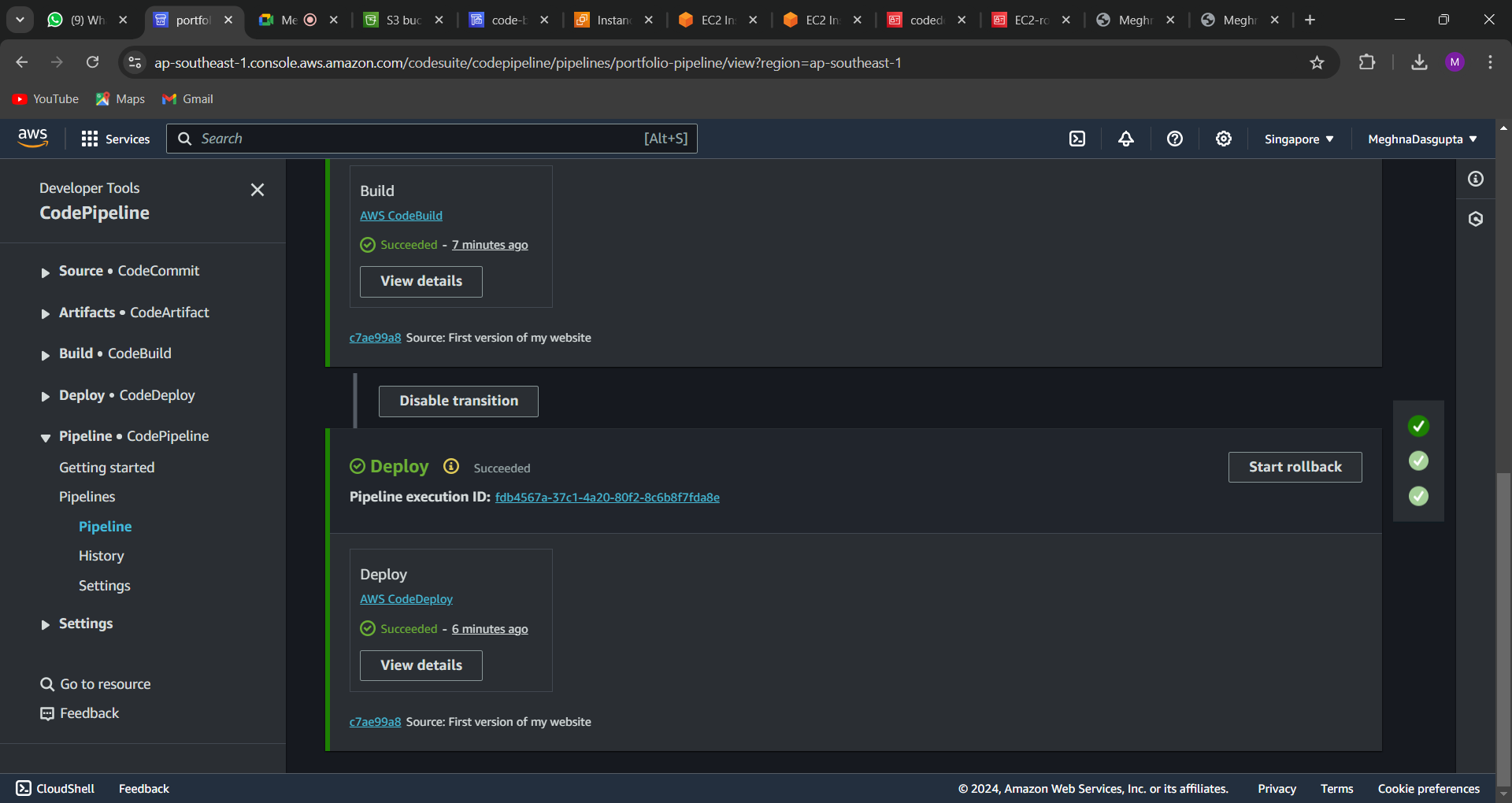
Now copy the ip and paste on your browser





Now create code pipeline



Make changes in version 2; and check your website





