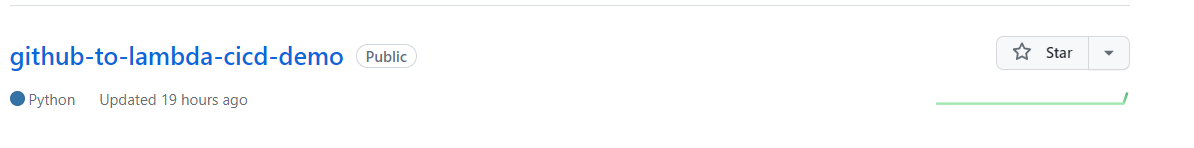
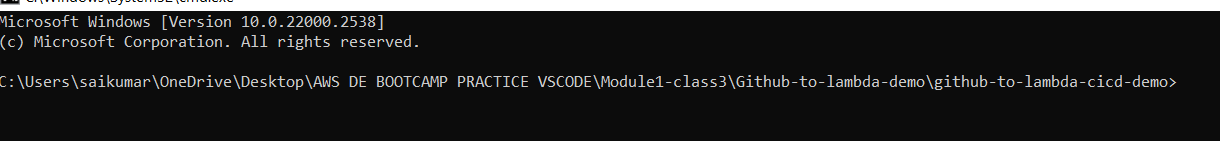
**GitHub-to-Lambda-demo**

In this document we are going to see entire code development circle . Like here we add vscode file to github merge it , we will also see how to create pull request, merge from test branch to Main branch and etc..

1. We will create a new repository in GitHub 🡪 add gitignore next -> add readme 🡪 create



1. Next we will create directory file (i.e) Github-to-lambda-demo
2. Next in that create another directory github-to-lambda-cicd-demo
3. Next go to cmd where directory is there



1. Next git clone the repository created to cmd prompt by command
2. Git clone -m <https://github.com/12saikumar8/github-to-lambda-cicd-demo.git>.

Here only main or master branch is there as a developer we cannot commit directly code changes in main or master branch so we will create a new branch called test we do code changes here and we will commit or push changes in that test branch and we will raise pull request (PR) that will be reviewed by seniors and they will merge code from test to main branch in some companies.

6.now create new branch by ( git checkout -b test) command here test is branch name we can give anything .

Next in case of production grade if we have external packages like modules/libraries required for code . we need to define the all packages in some requirements.txt file , and when we spin up our code the all packages should be downloaded during runtime itself.

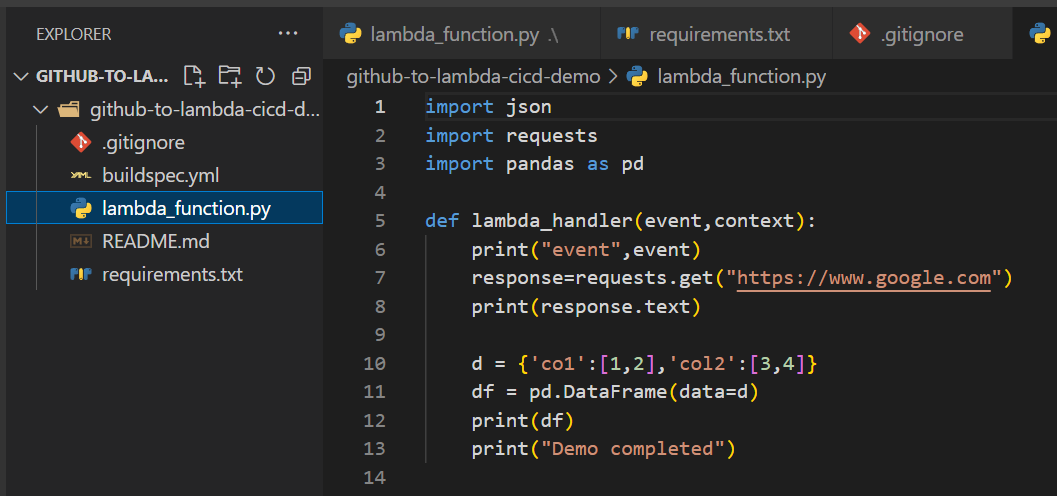
1. In our directory github-to-lambda-cicd-demo we create a new file lambda\_function.py and create requirements.txt file for dependencies.

The way we will write like cicd components which will package my code their I will define 1 step i.e during the execution it needs to pull first particular dependencies

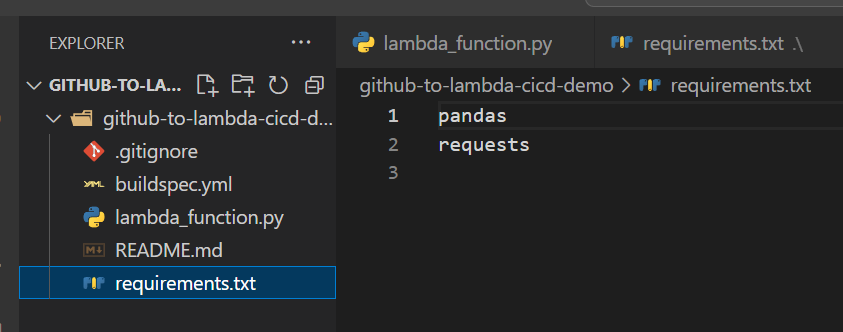
So we create requrements.txt -> add required libraries -> if we want we can add version of libaraies

Lambda\_function.py we added some code like its hould get some response by making api call to go to website and we hard codded some code for creating pandas dataframe.

We need to code as per requirement this is demo.



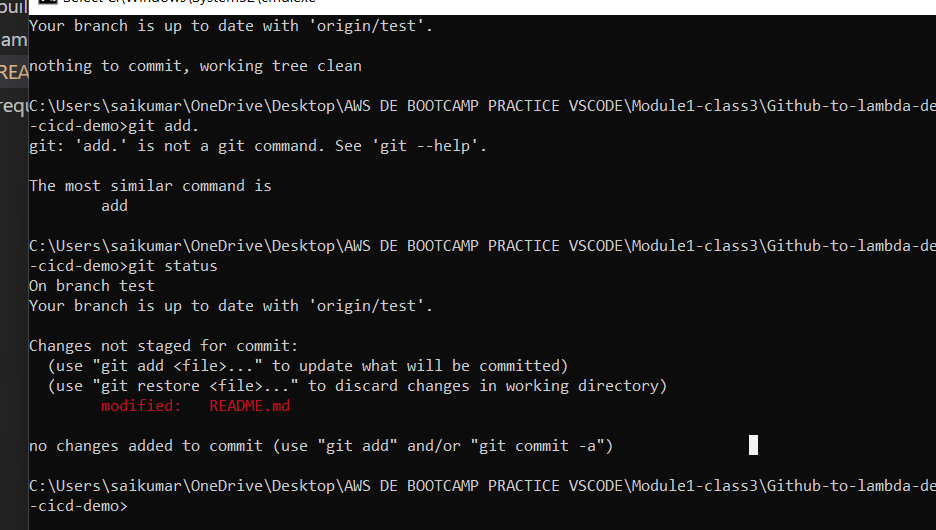
Below is requirements.txt



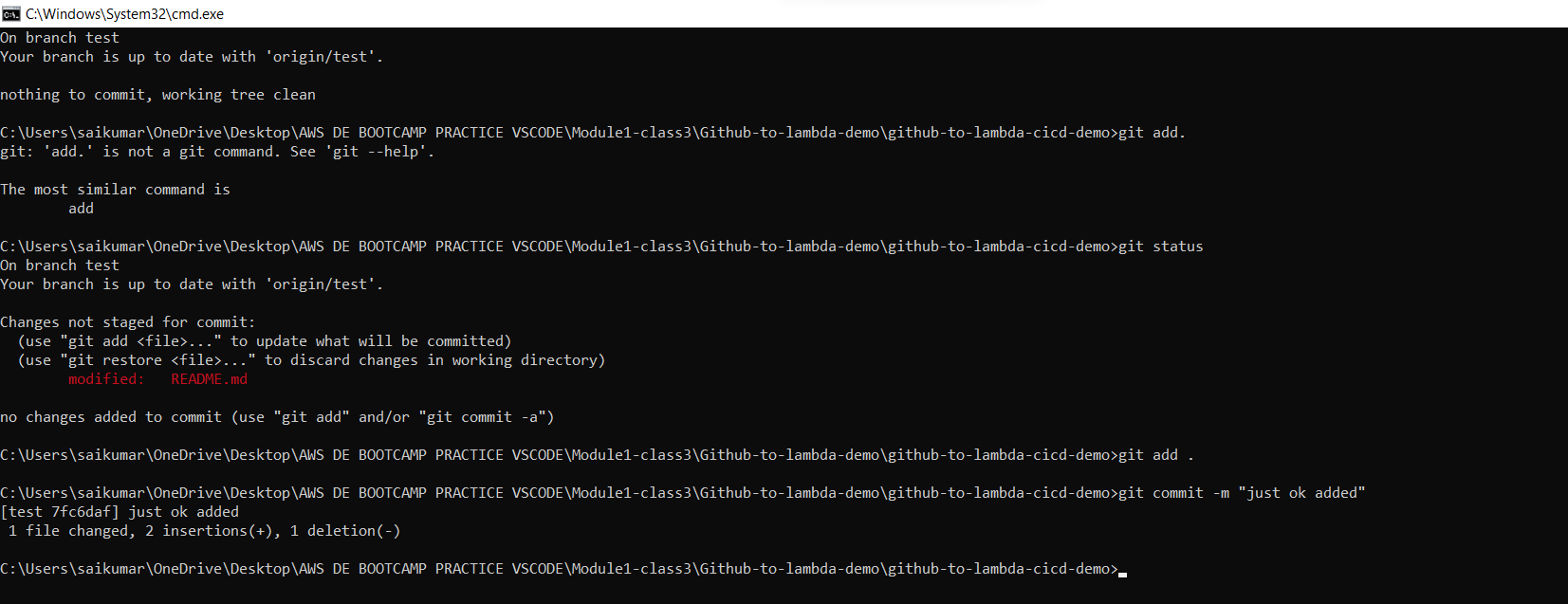
* In case of production or real-life here so far code changes made are simply commit changes to test branch next will Raise Pull request 🡪 next will merge to master

Here it’s just learning purpose so no other teams so ww will only merge from test to main branch

1. Now in cmd see git status for changes (git status) read colour file are changes made in screenshot it’s just an example but here it will show lambda\_function.py and requirements.txt



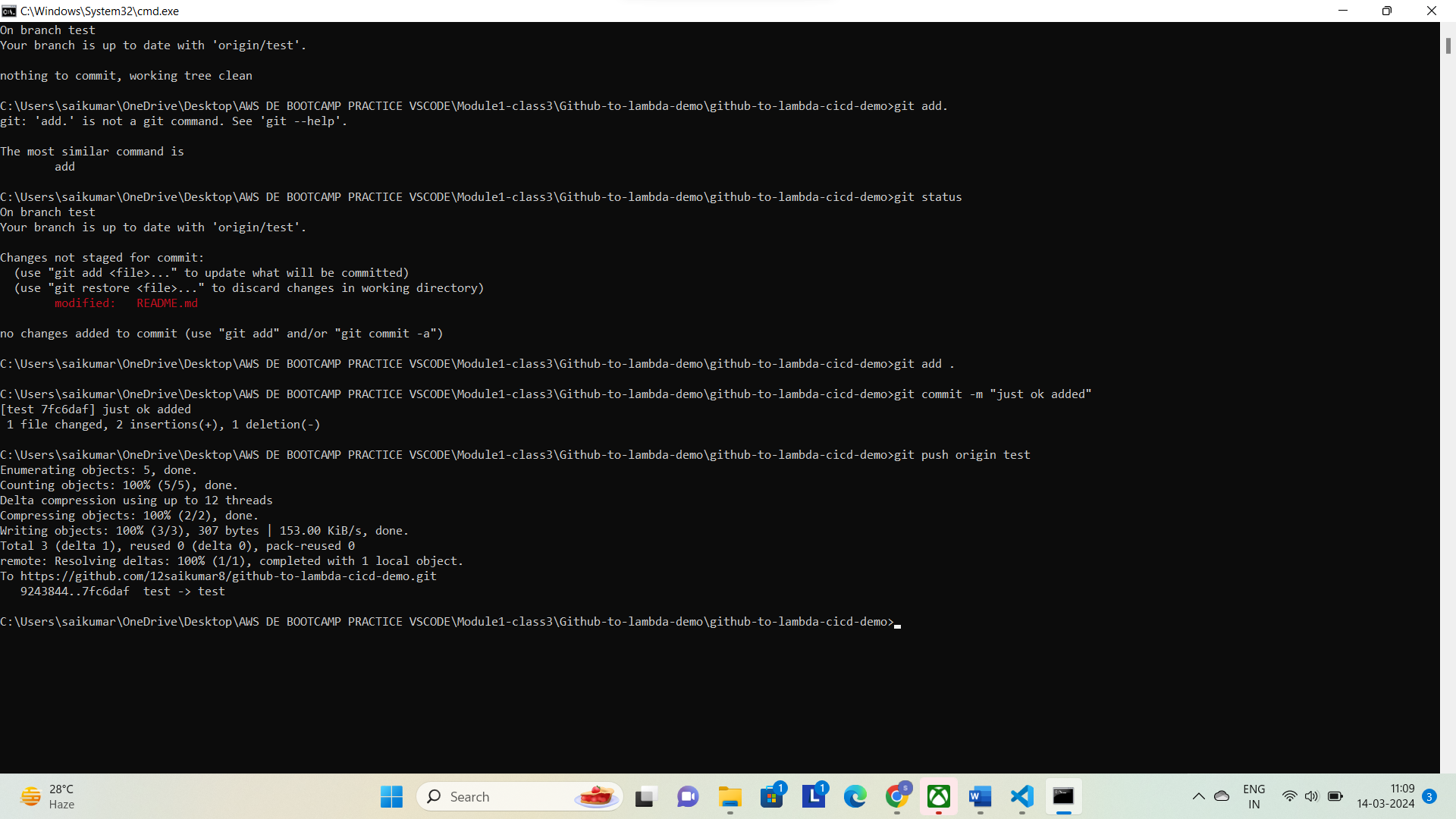
1. Next add these changes using (git add .) here . means all changes we made if we want only 1 file means we can add that file (git add filename)
2. Next commit by git commit -m “ message “ , in message we can give anything here message is developed lambda code for cicd demo



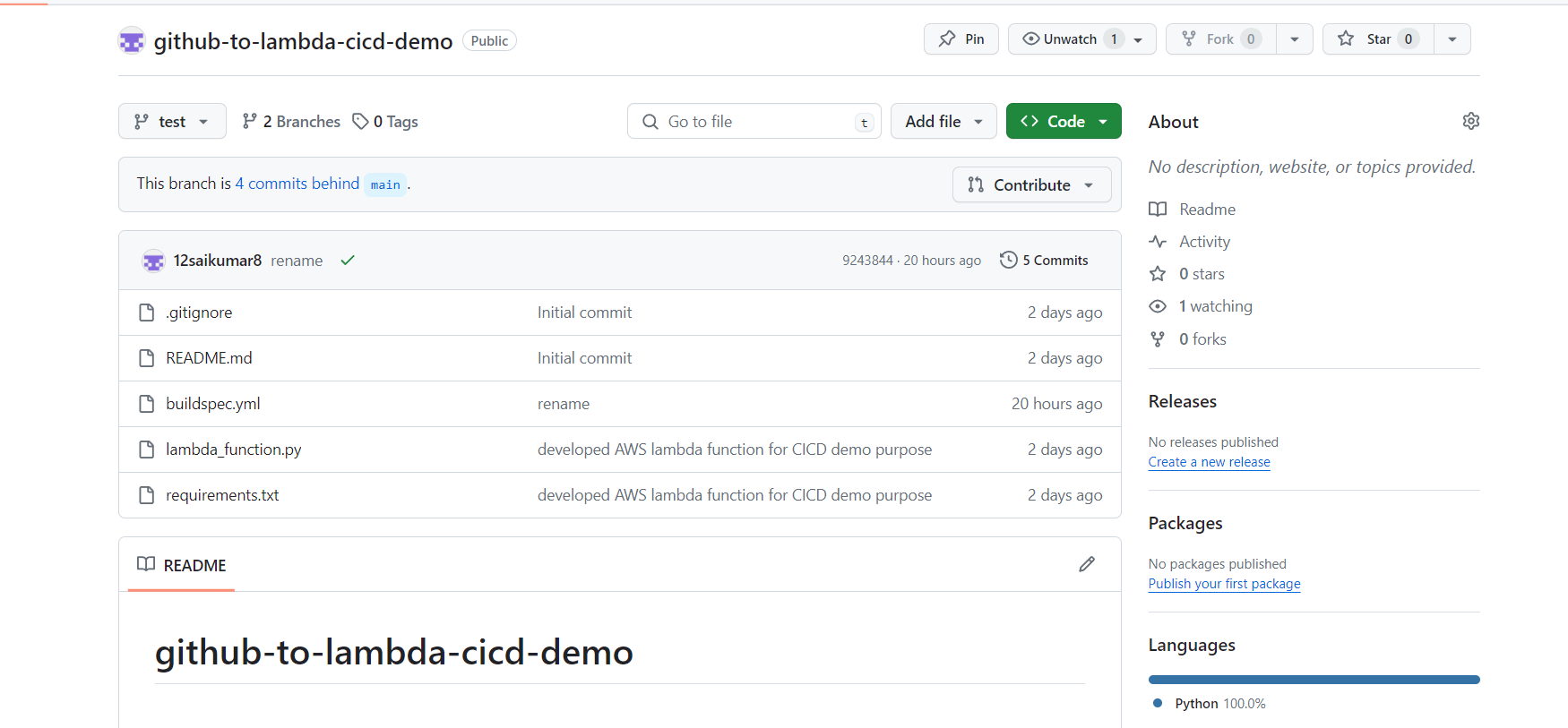
We can see what happened

1. All changes in current branch are commited.
2. Next git push origin test

We can see what happened after this in screenshot

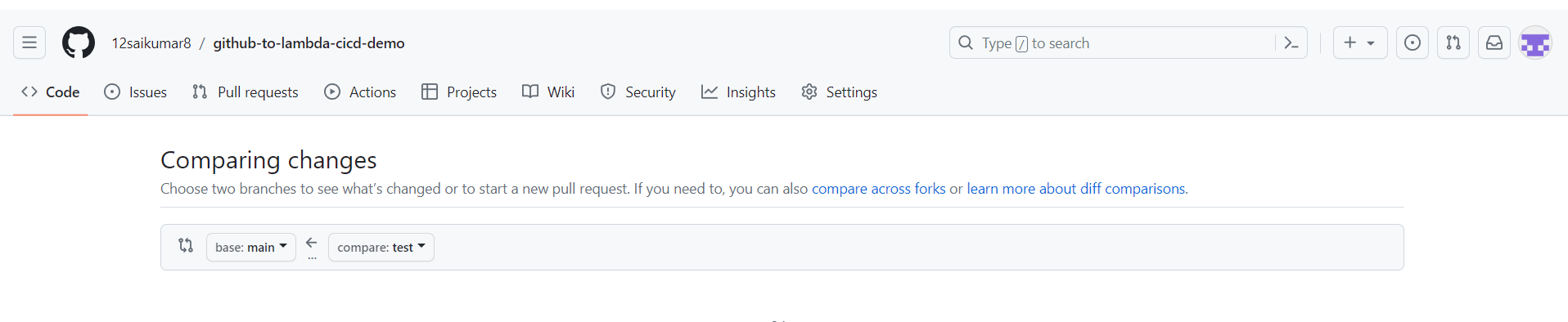


1. Now we can see test branch in github



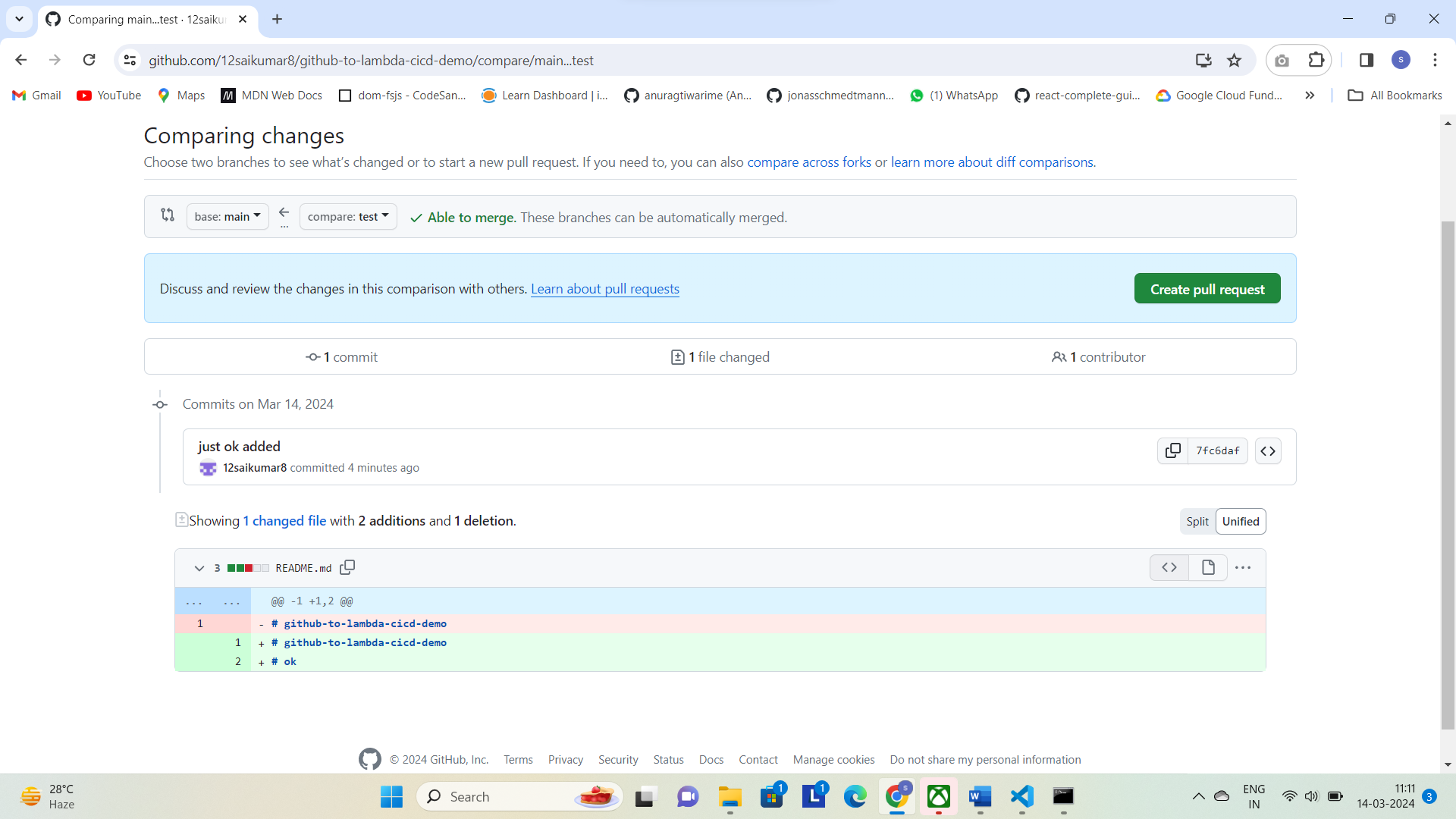
Here above buildspec.yml file is showing but don’t consider that file here

Next we need to implement CICD in main branch so click on compare &Pull request (create PR)



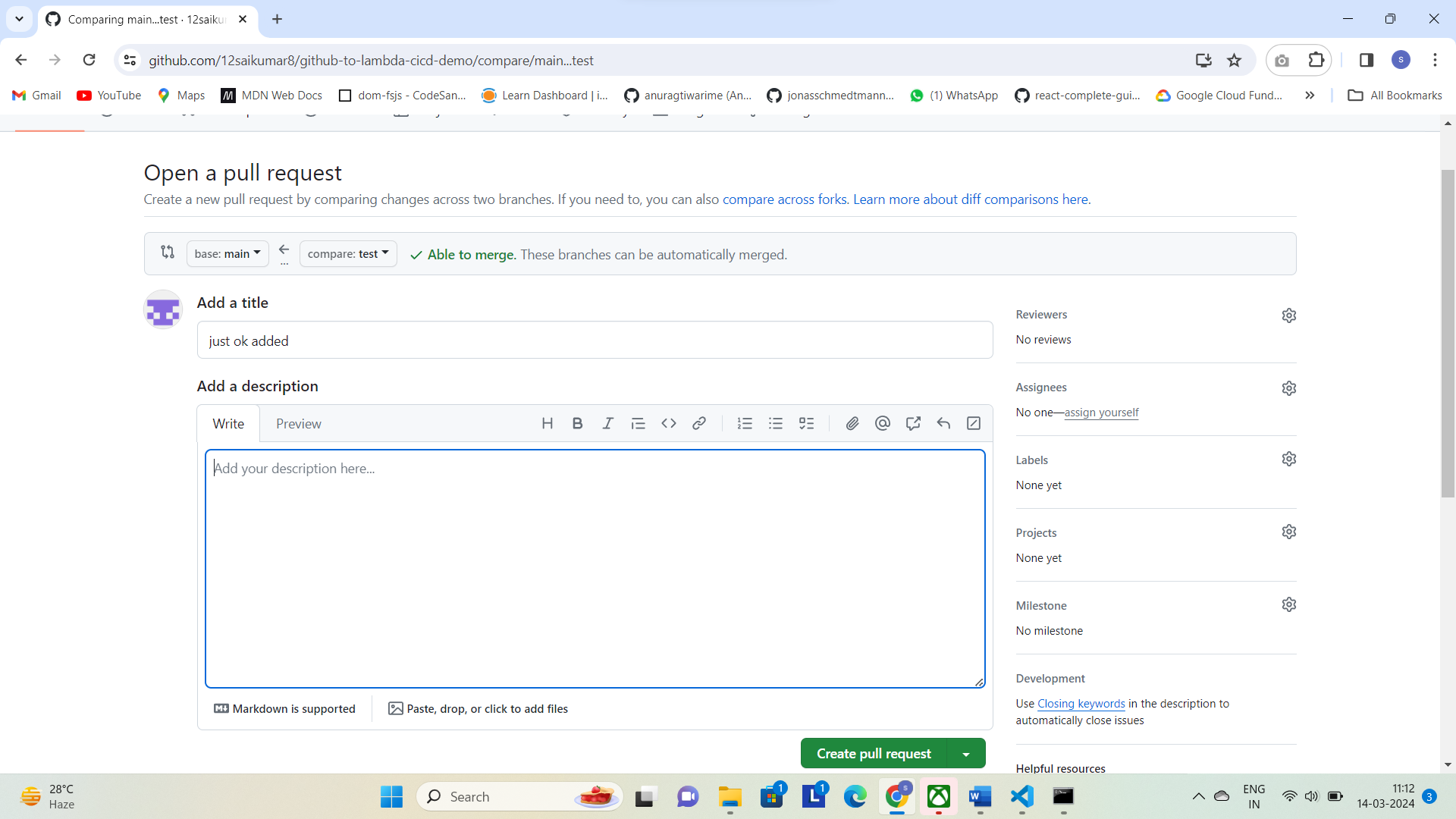
Here we need to here select base as main branch and the branch we need to compare i.e test

Here it will if any conflicts are there

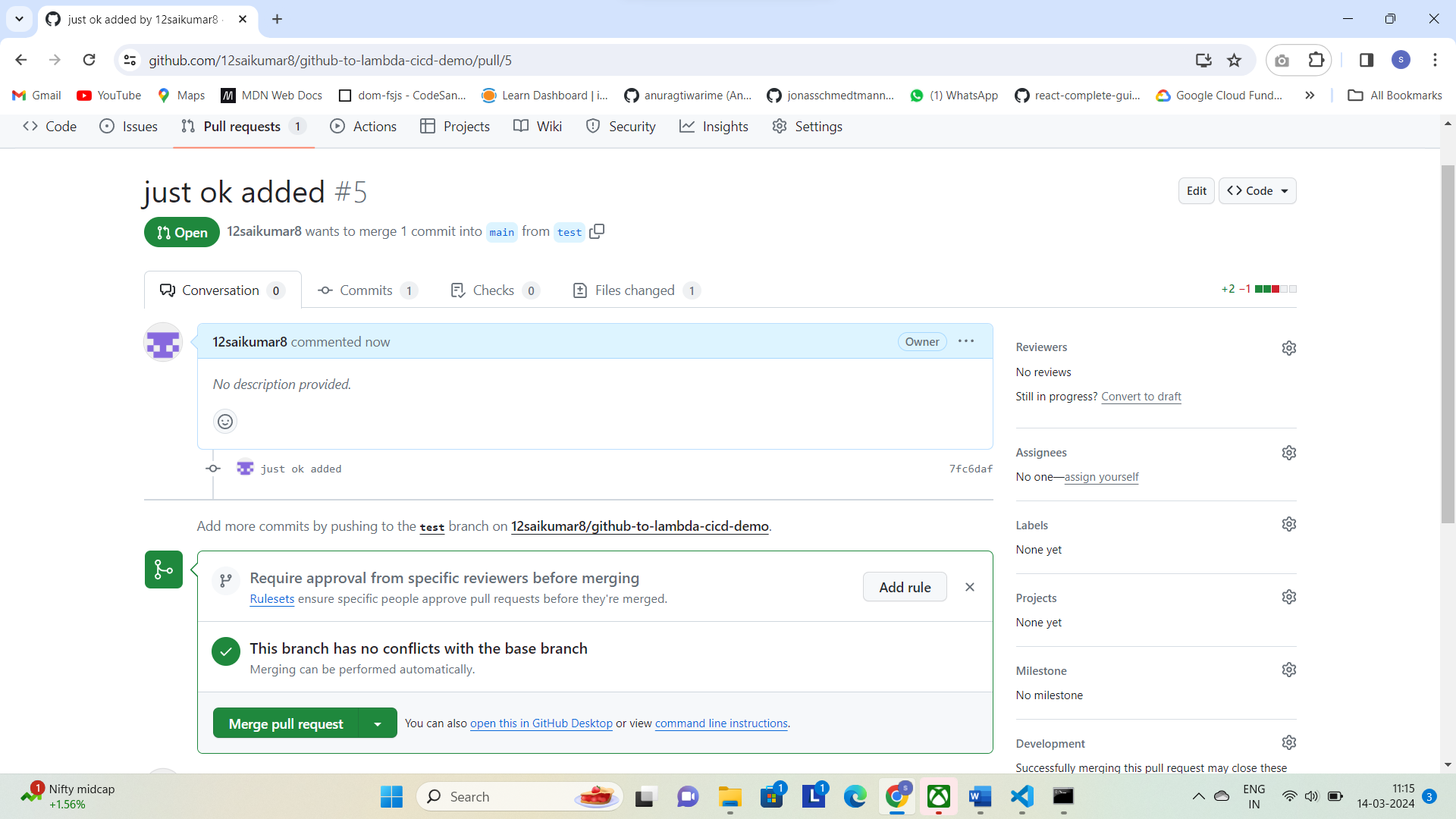


Here we can see all code in Main and code in test branch. And + for code added and – for code deleted.

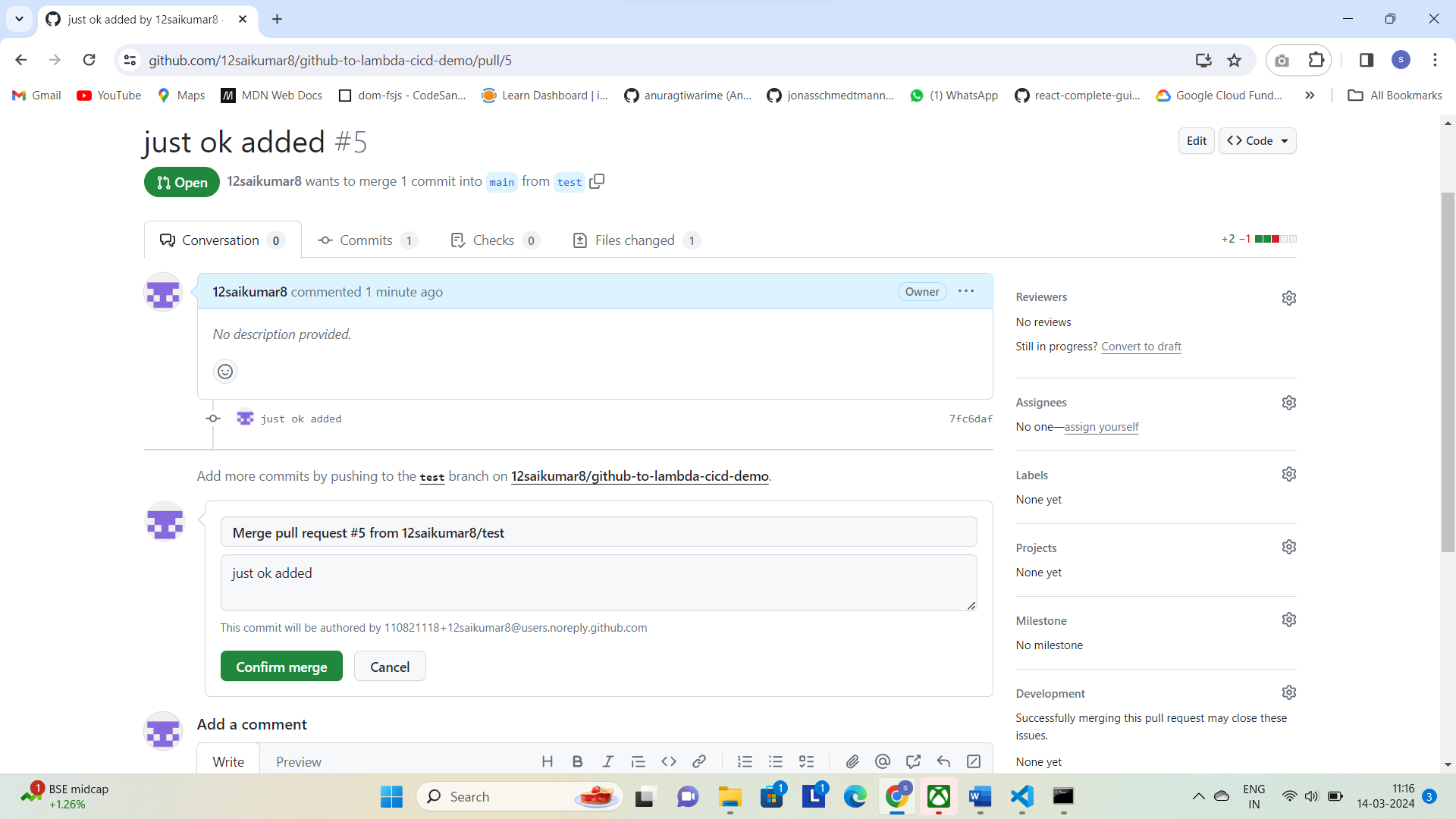
Next when we click on create PR we can add any description regarding what we made we can add description as per project or in some companies it will have Jira Tickets so we can mention those Jira tickets in description.



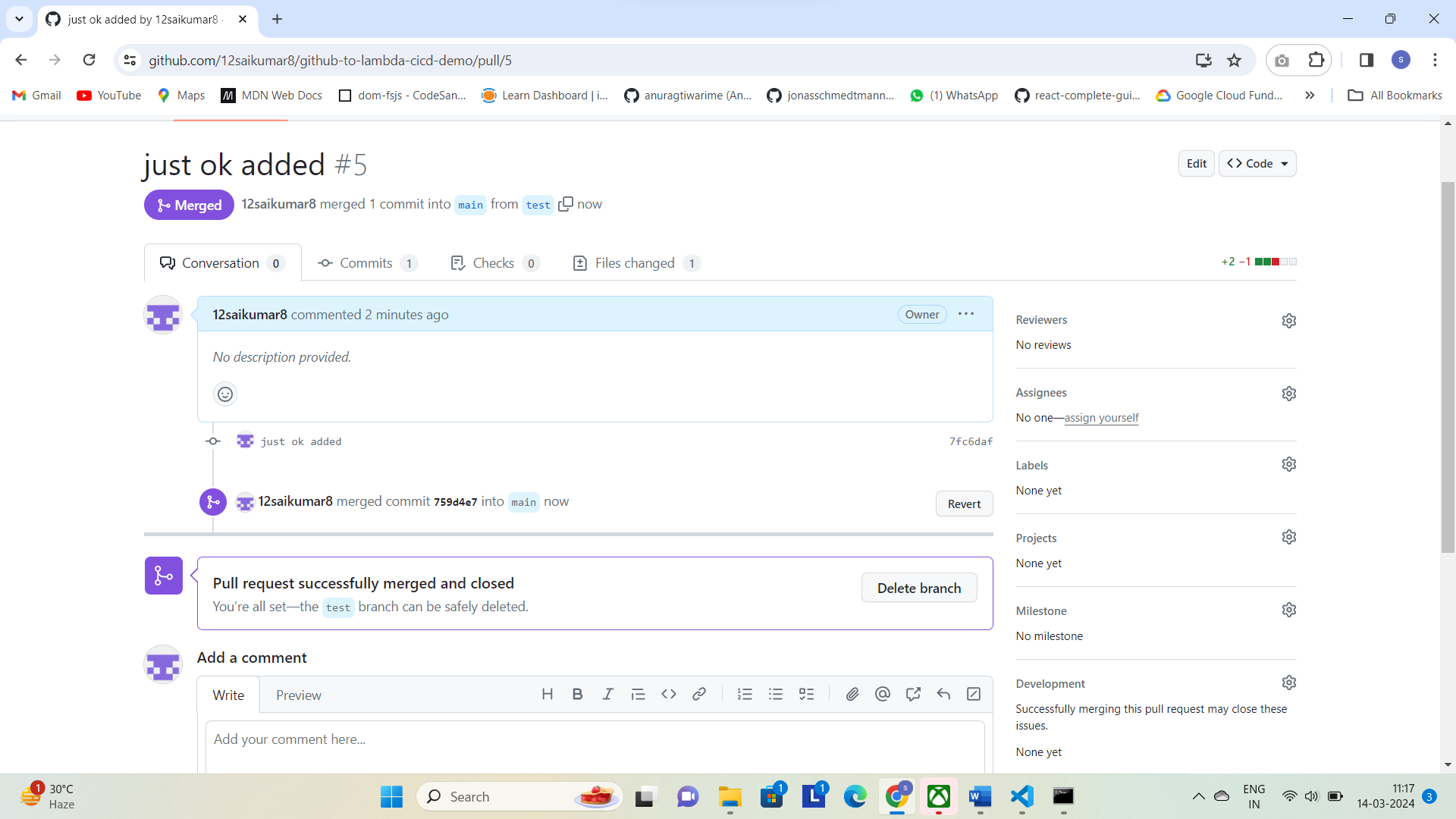
Next screenshot is below



Click on mergepull request and next screenshot is below



Click on confirm merge next screenshot below that it is merged.



We can see all changes merged to main Branch.

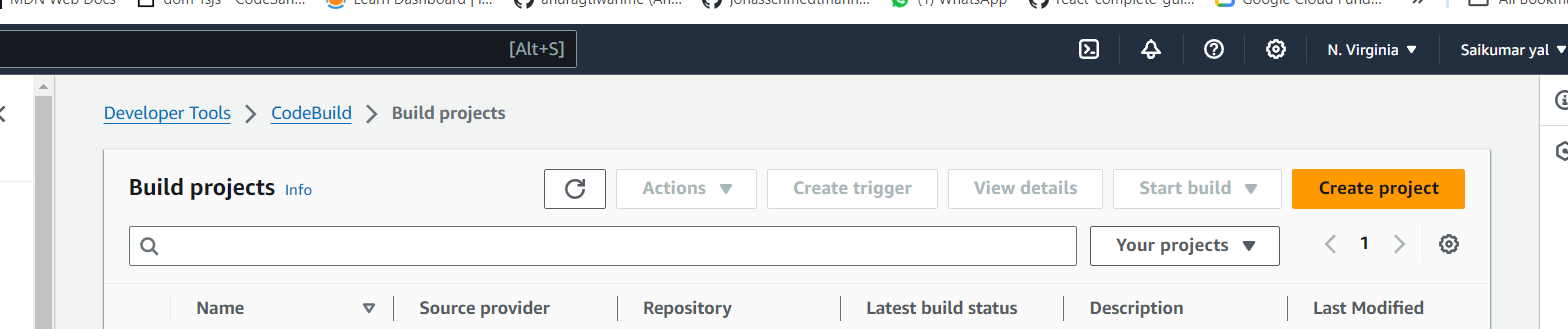


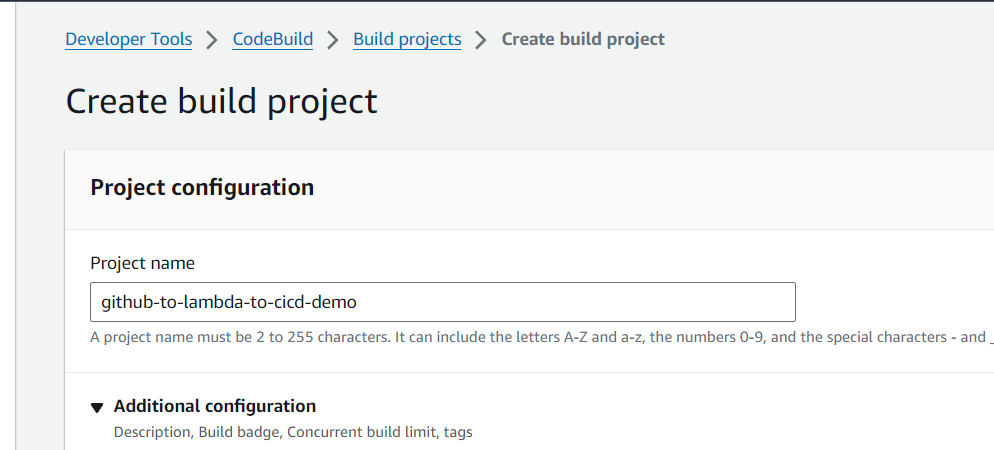
In companies we will not have access to merge from test to main branch unless we added as admin.

Steps Create Pull Reuest(PR) 🡪 next click on merge PR 🡪 Click Confirm 🡪 Merged.

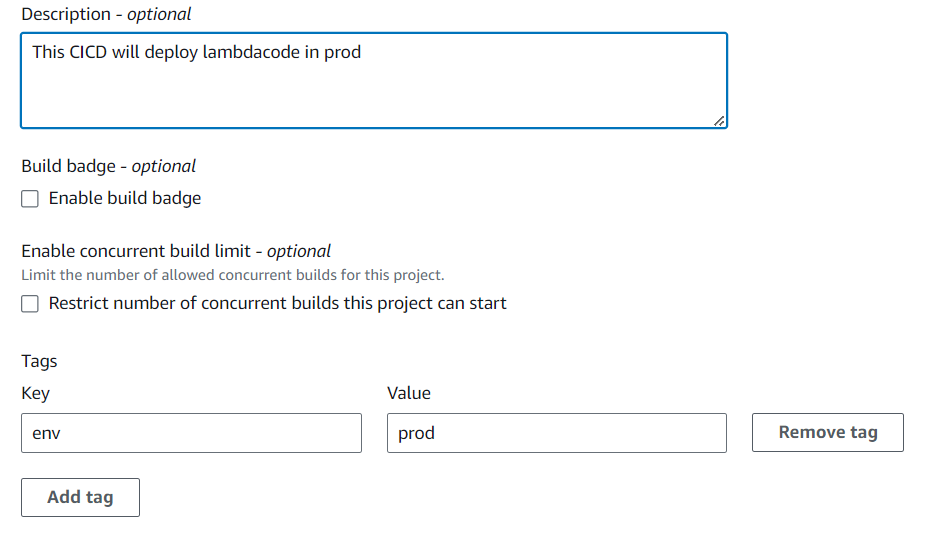
**Creating New Code Build**

1.Click create new project



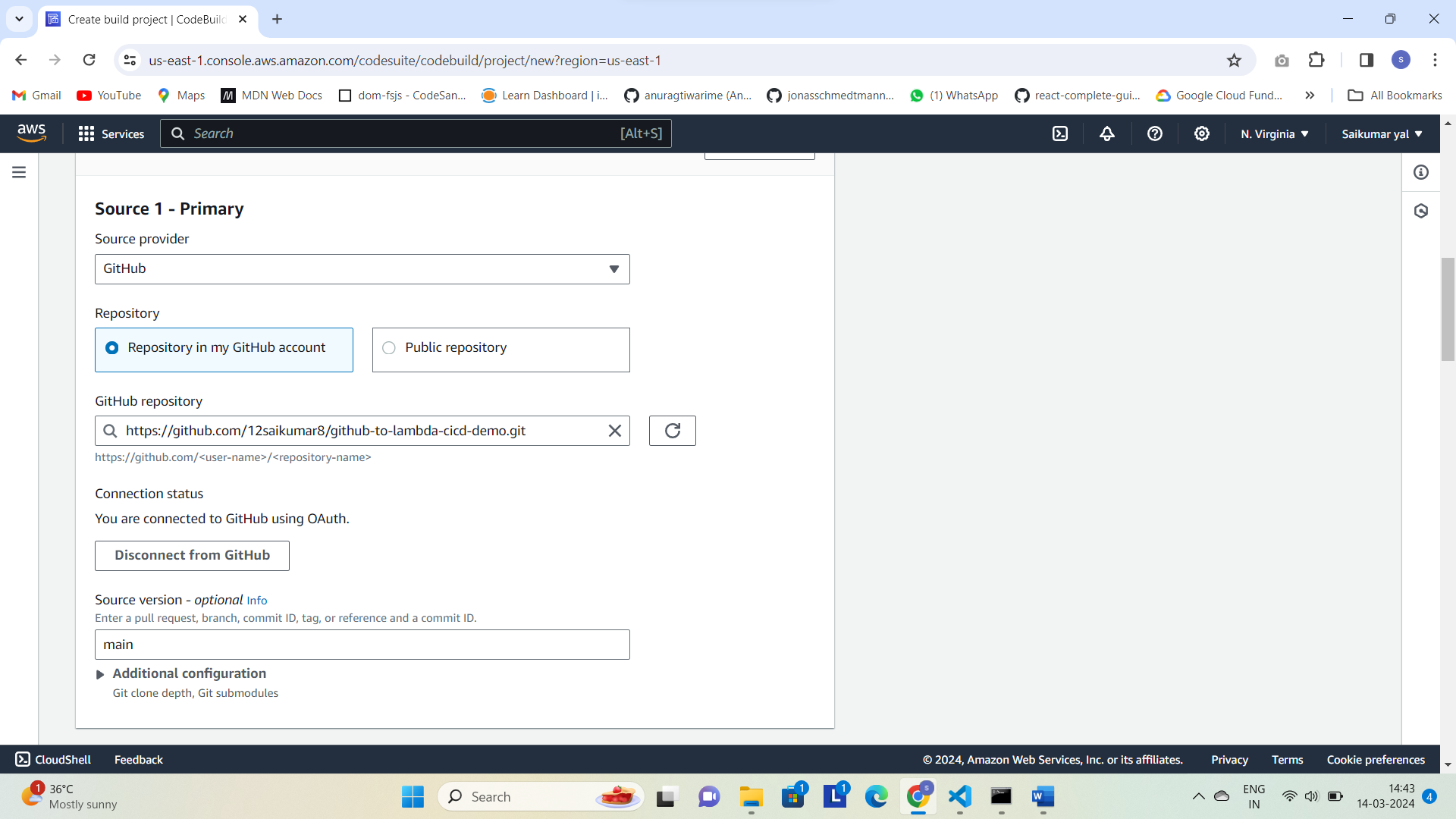


In Additional configuration add key and value optional and description u can add something



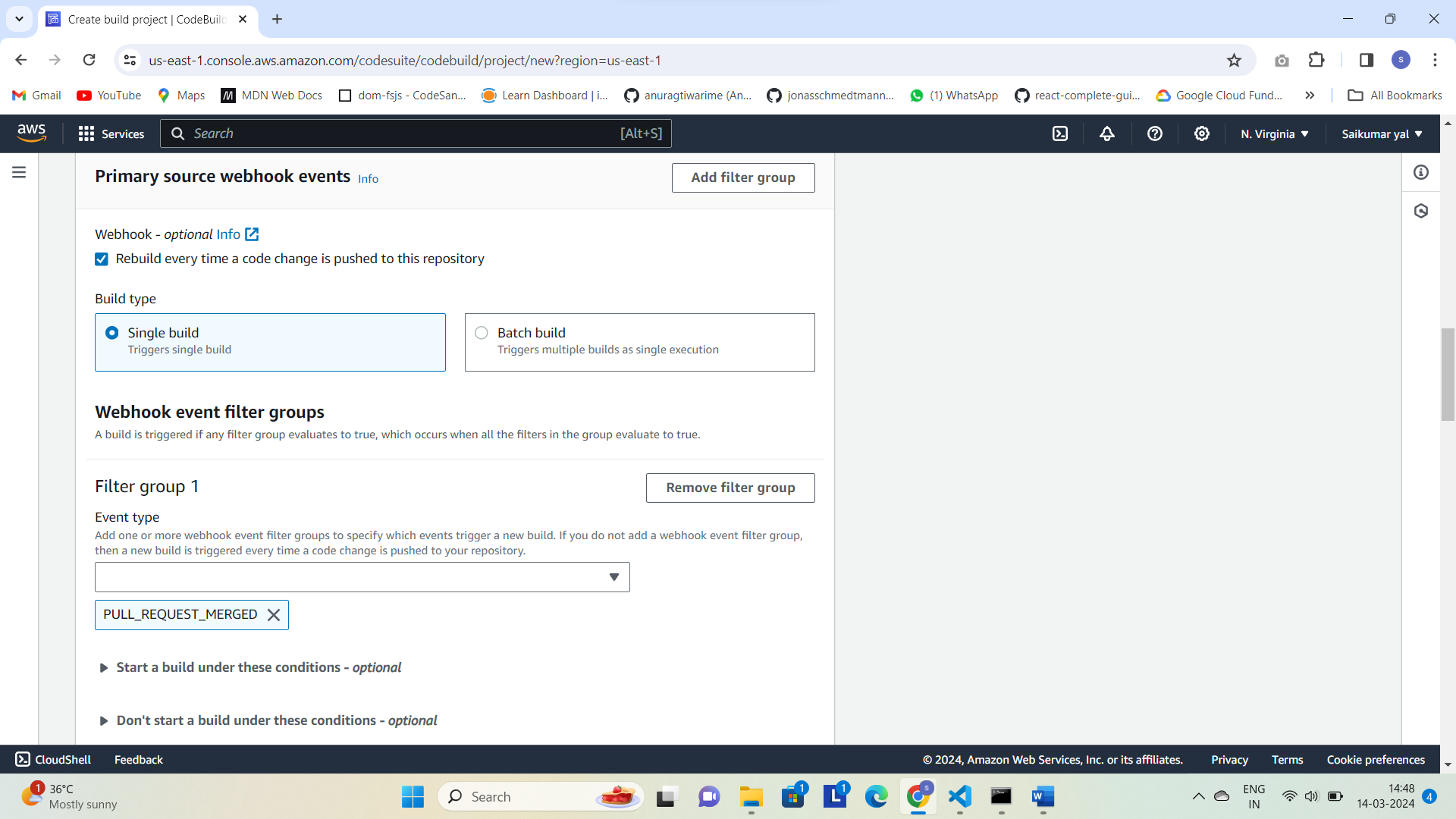
In source we can add whatever here we added GitHub

We can select our git repo where folder is created.



And source we have added Main .means we are saying as soon as changes are commited to main this code build should trigger them.

Next tick webhook events and filter we used pull\_request\_merged but other options also there.



Next create a lambda function ex (github-to-lambda-cicd)

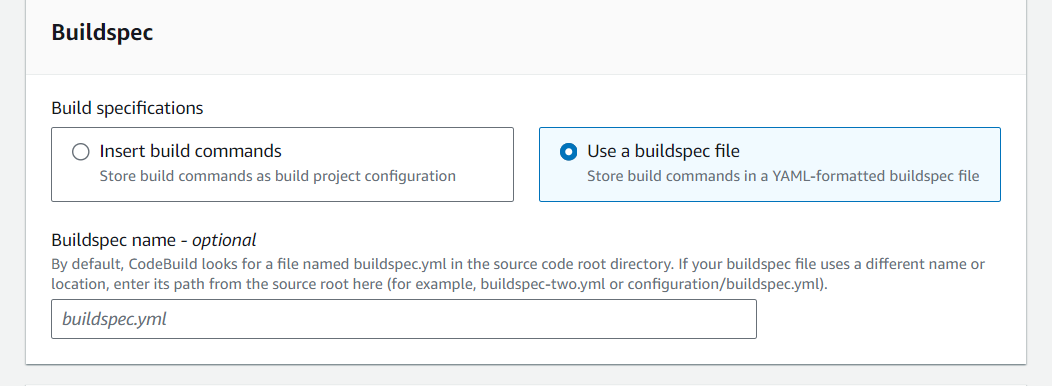
Select Envirorment for codebuild that all python modules or dependencies can run like as per lambda\_function (ex: python 3.11,etc..)

and select whatever python envirorment we are using in lambda function and makesure to setup that envirorment in codebuild so that envirorment in codebuild cannot break

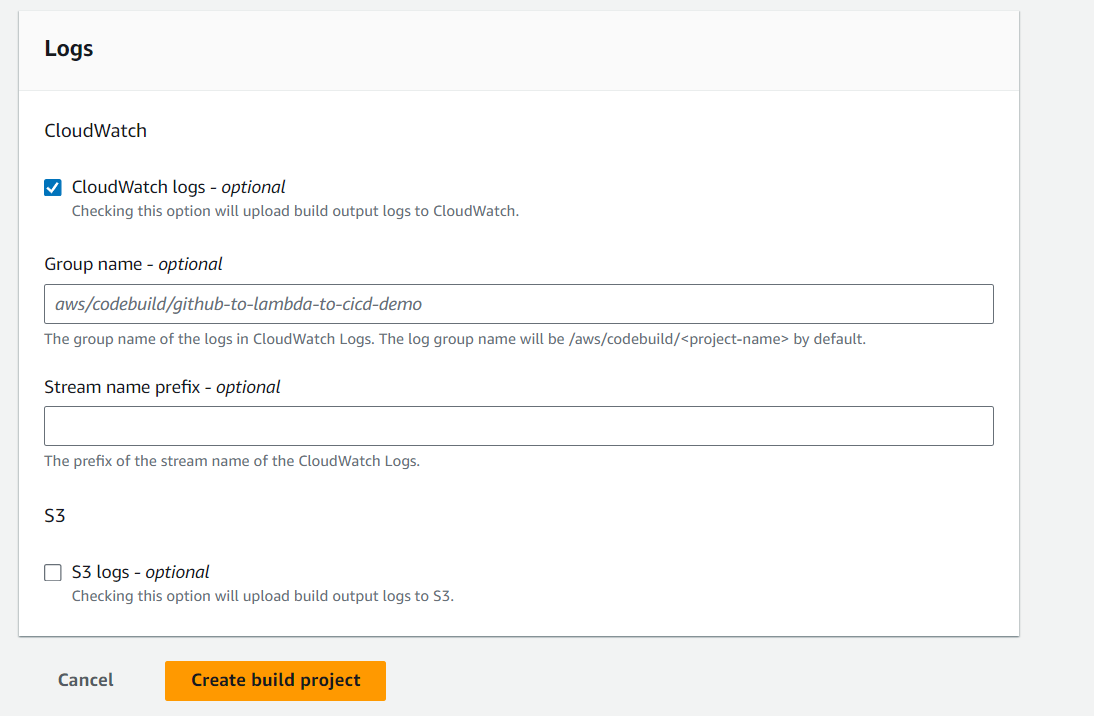


For image and image version select and depend on documentation

* Next we insert all buidspec commands
* We can keep all build commands in a file buildspec and click on buildspec so it will find that file automatically and use it

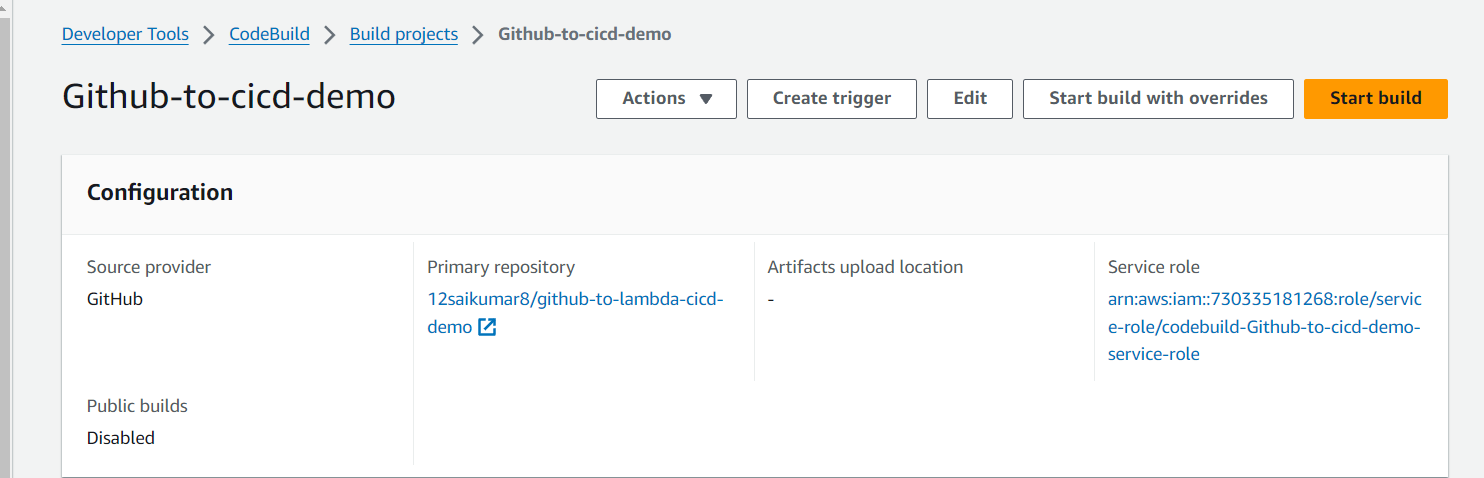


And next add cloudwatch logs tick it and create

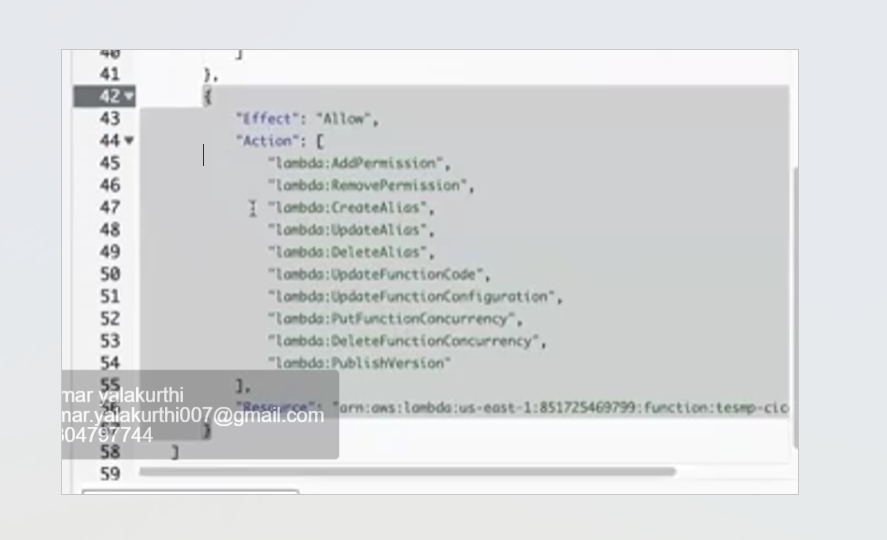


After creating open service IAM Role and bcz if we add any buildspec.yml to add lambda\_function .

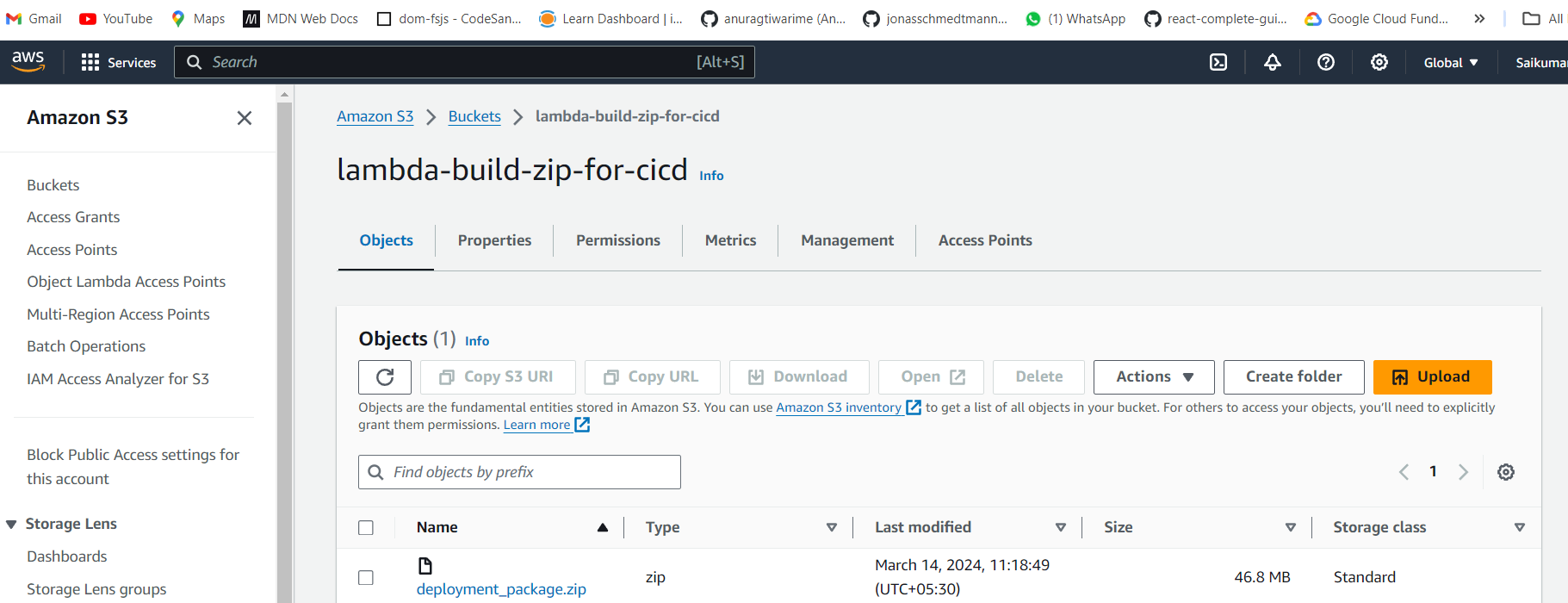
* So we need to give permissions related to lambda

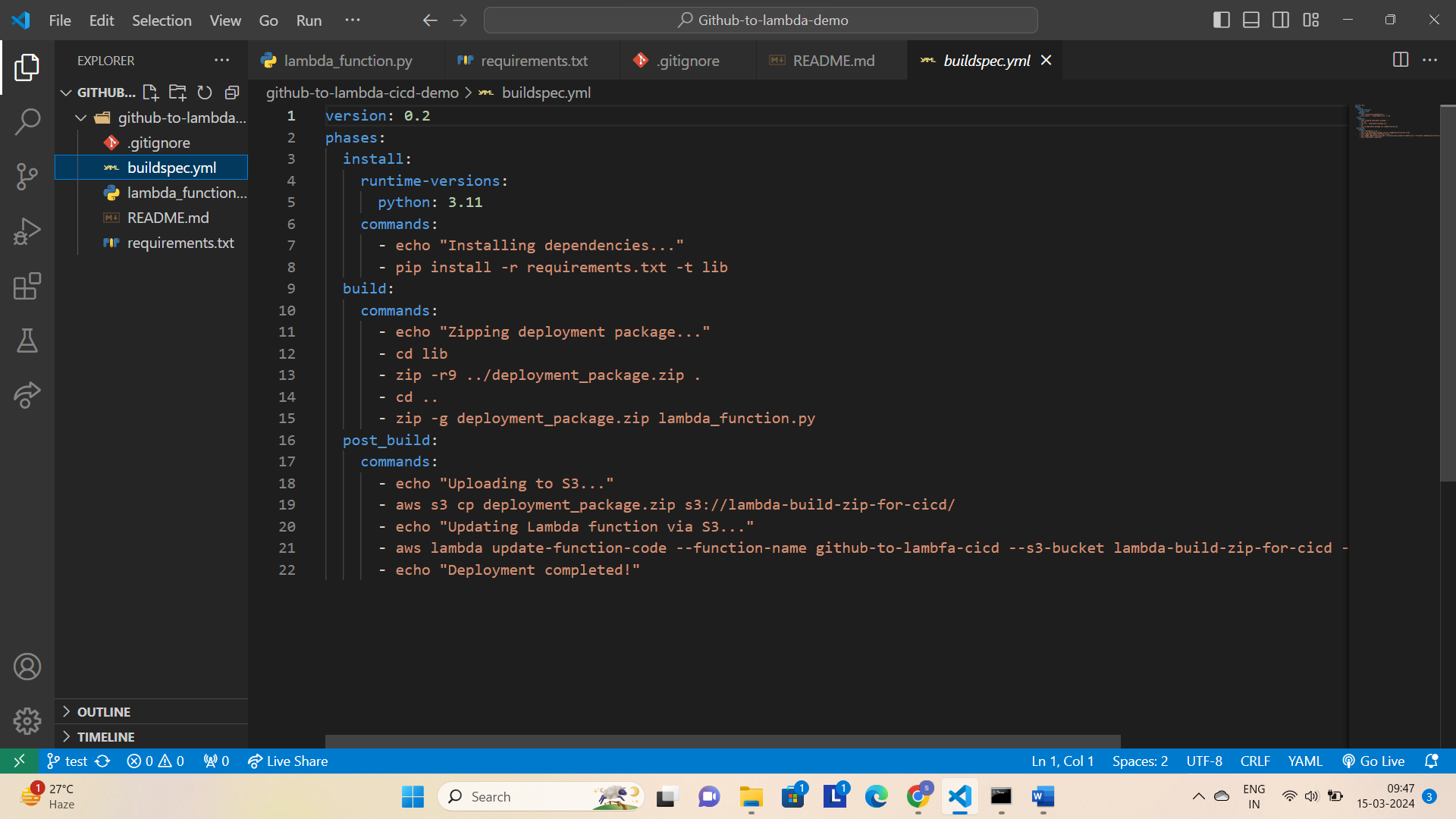


So we add permissions related to lambda\_function . By edit policy and add for lambda permissions like lambda actions allowing

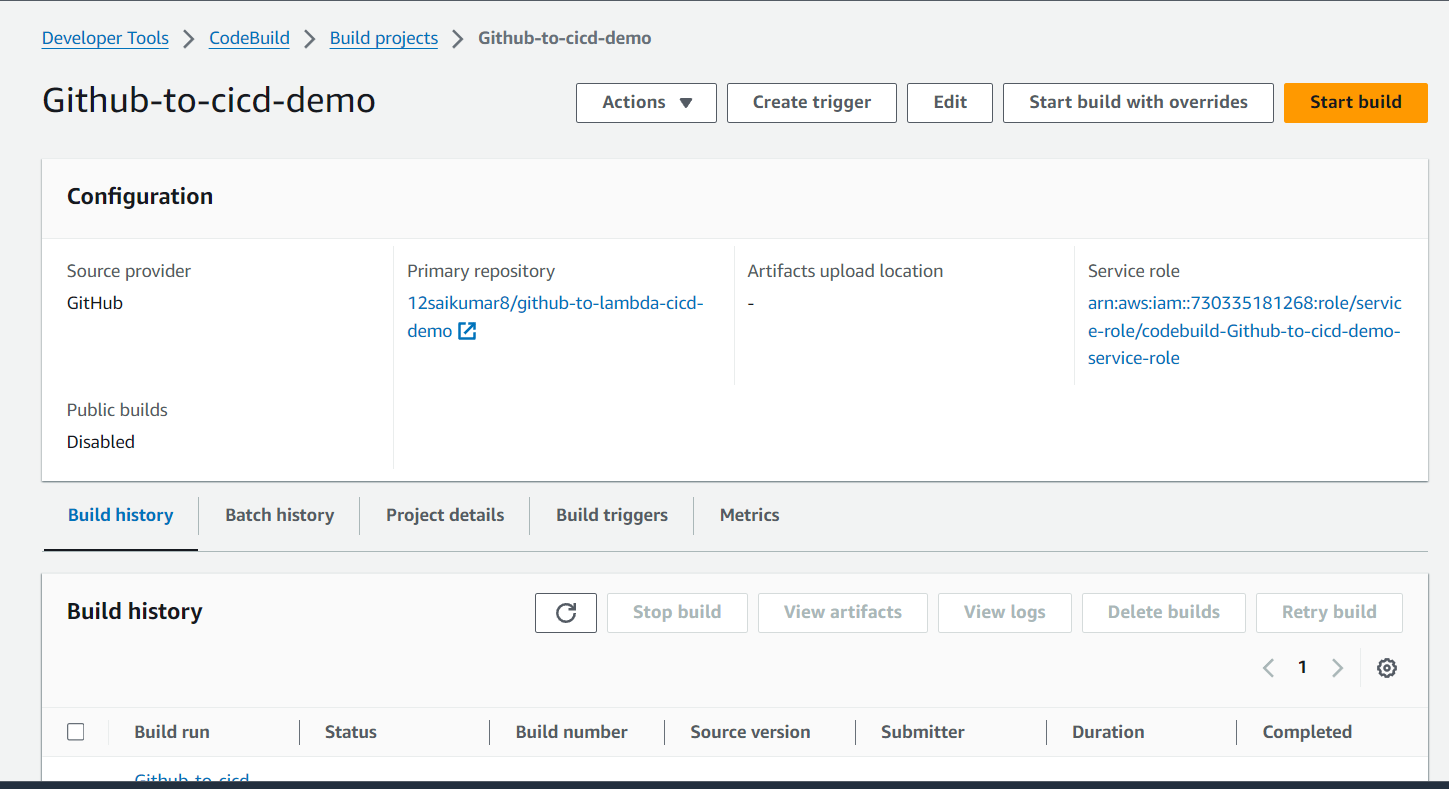


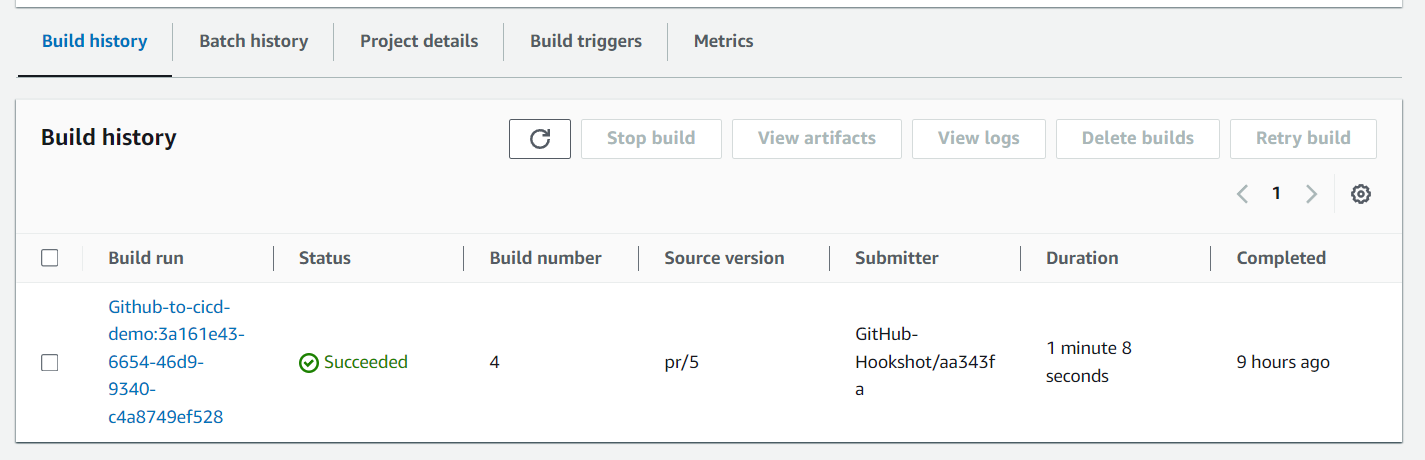
Next will create s3 bucket ex :lambda\_builds-zip



* Next add buildspec.yml file in vscode and next permissions i.e add s3 full access In the lambda\_function
* 
* After that do
* In cmd git status
* Git add.
* Here we are saying to add buildspec file which is created in vscode directory to git
* Git commit -m “message”
* Git push origin test

Next see codebuild for cicd is created and we see its is failed or success

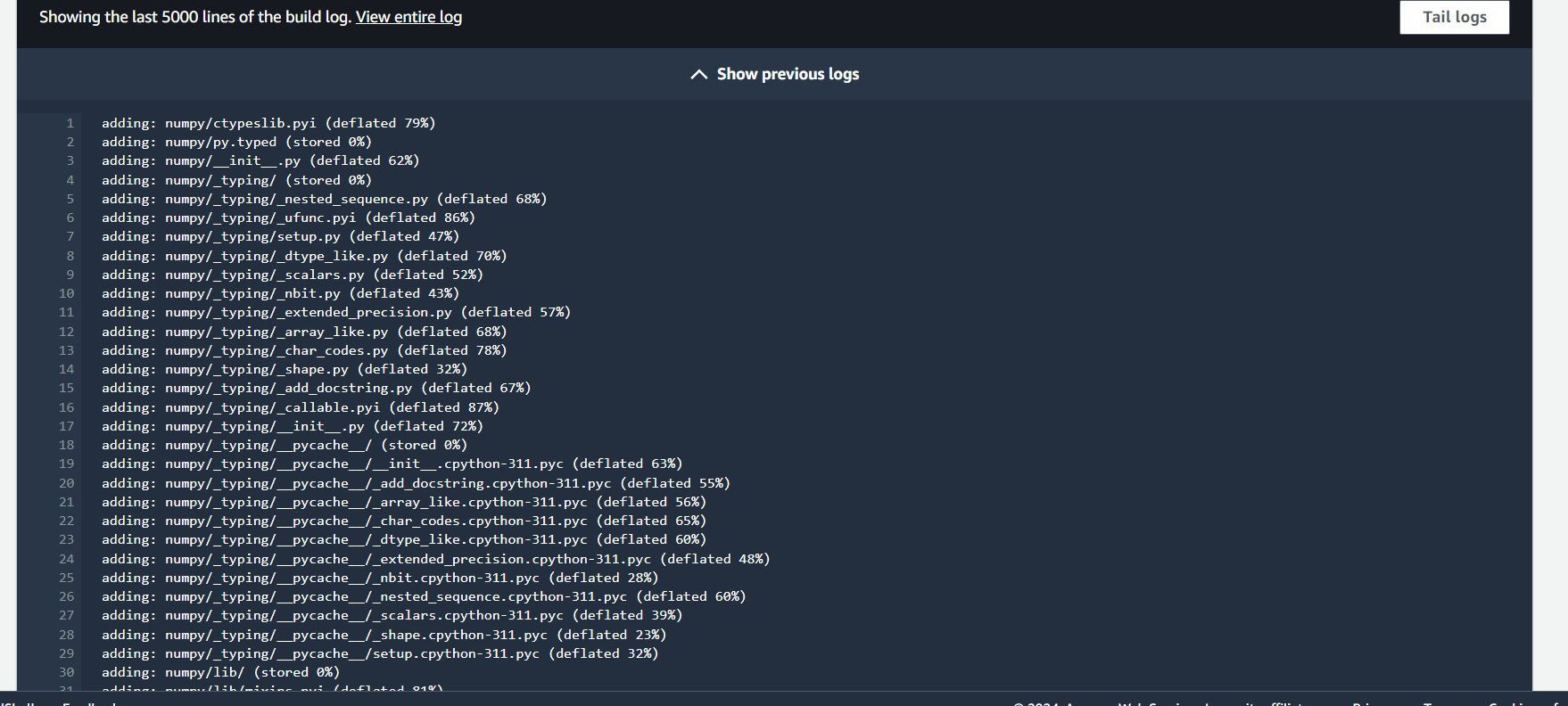


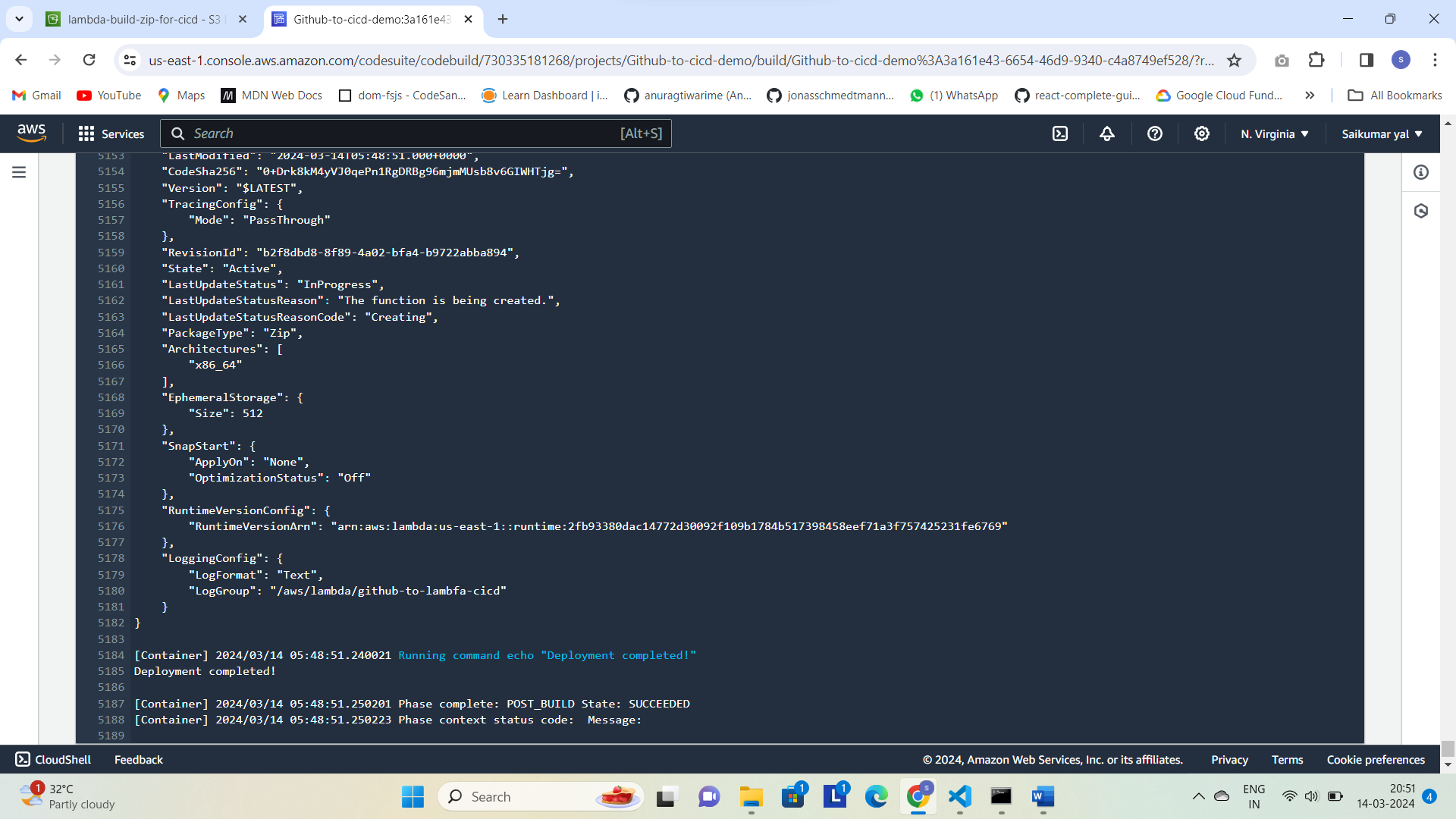


Click on the thing shown below Build run

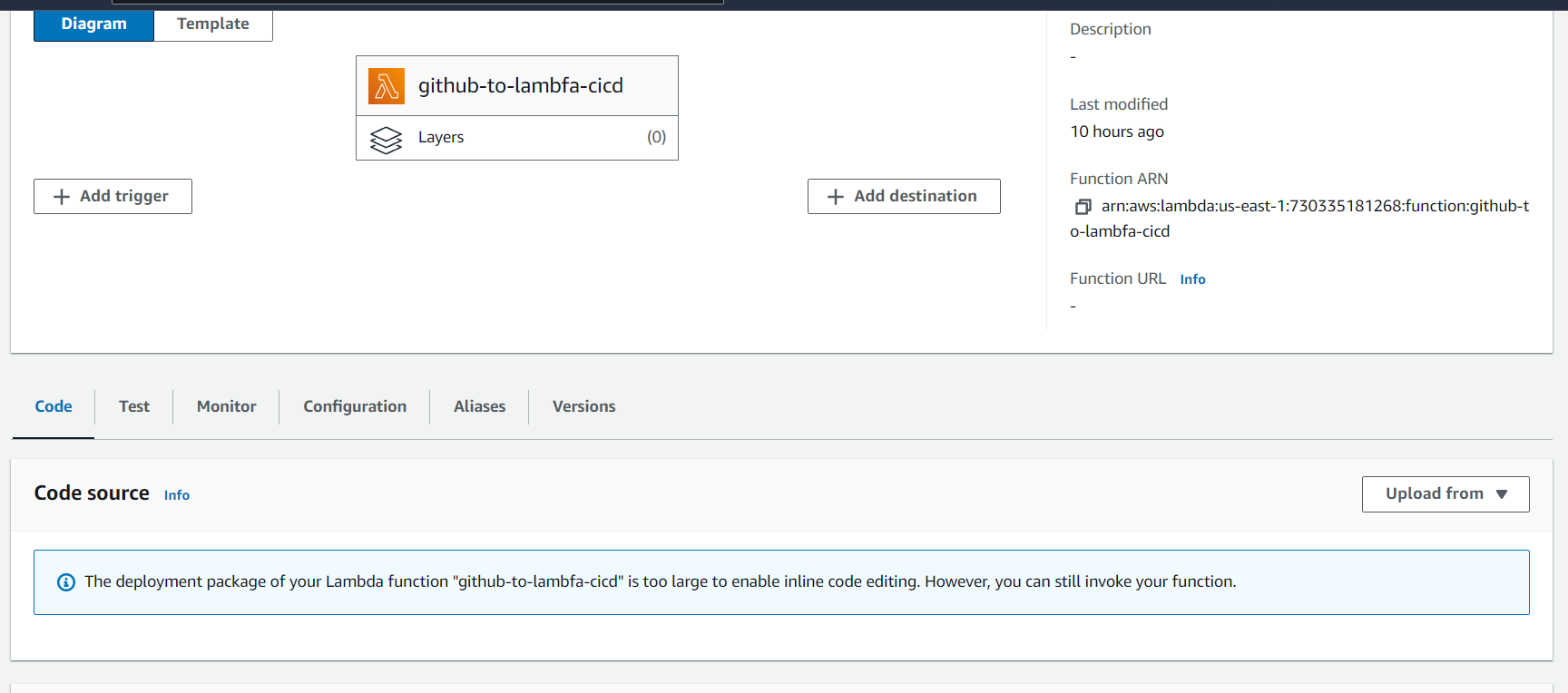
And we see build logs for all things running

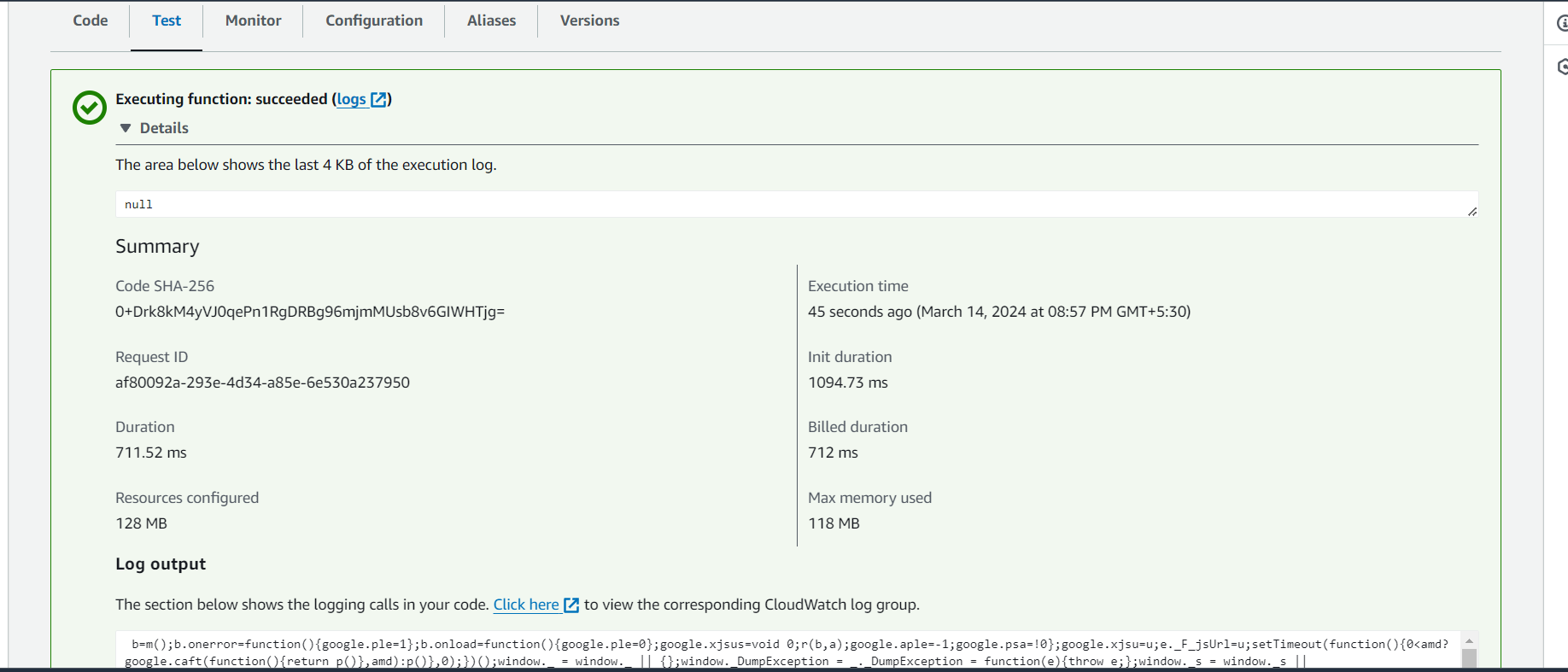


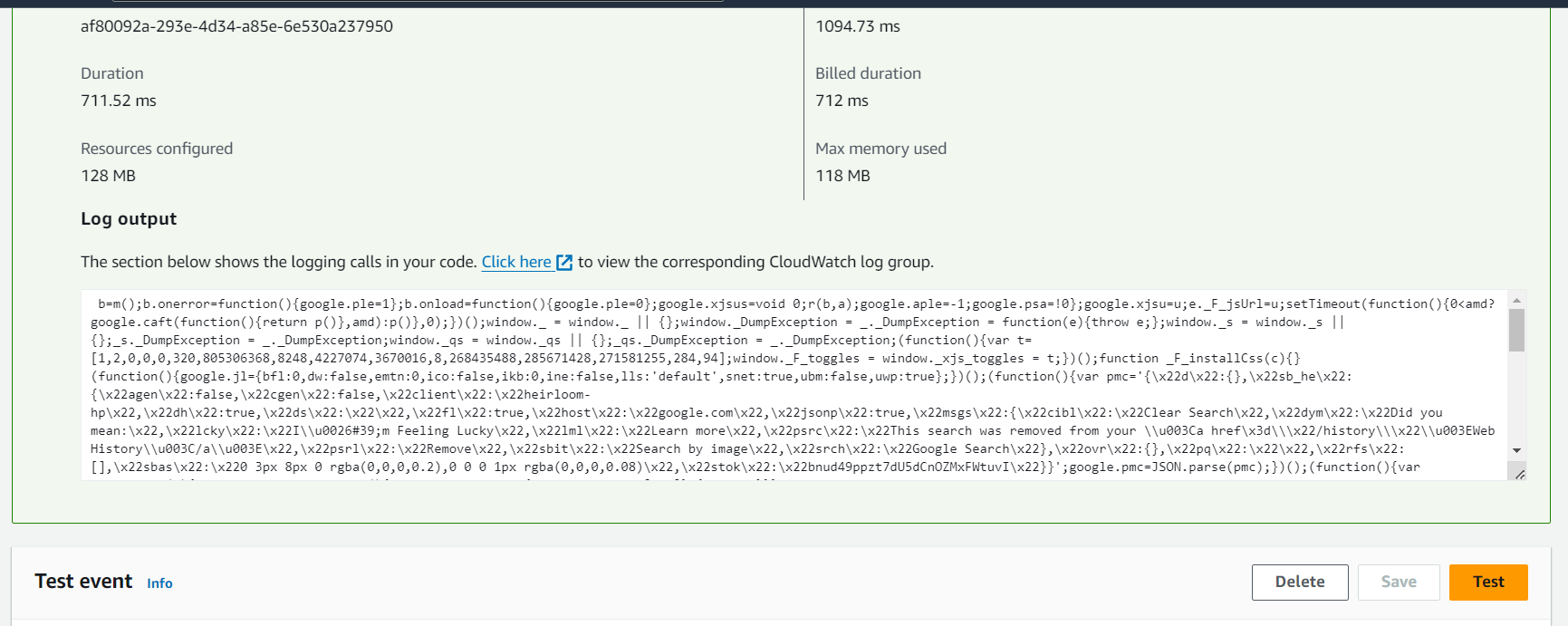




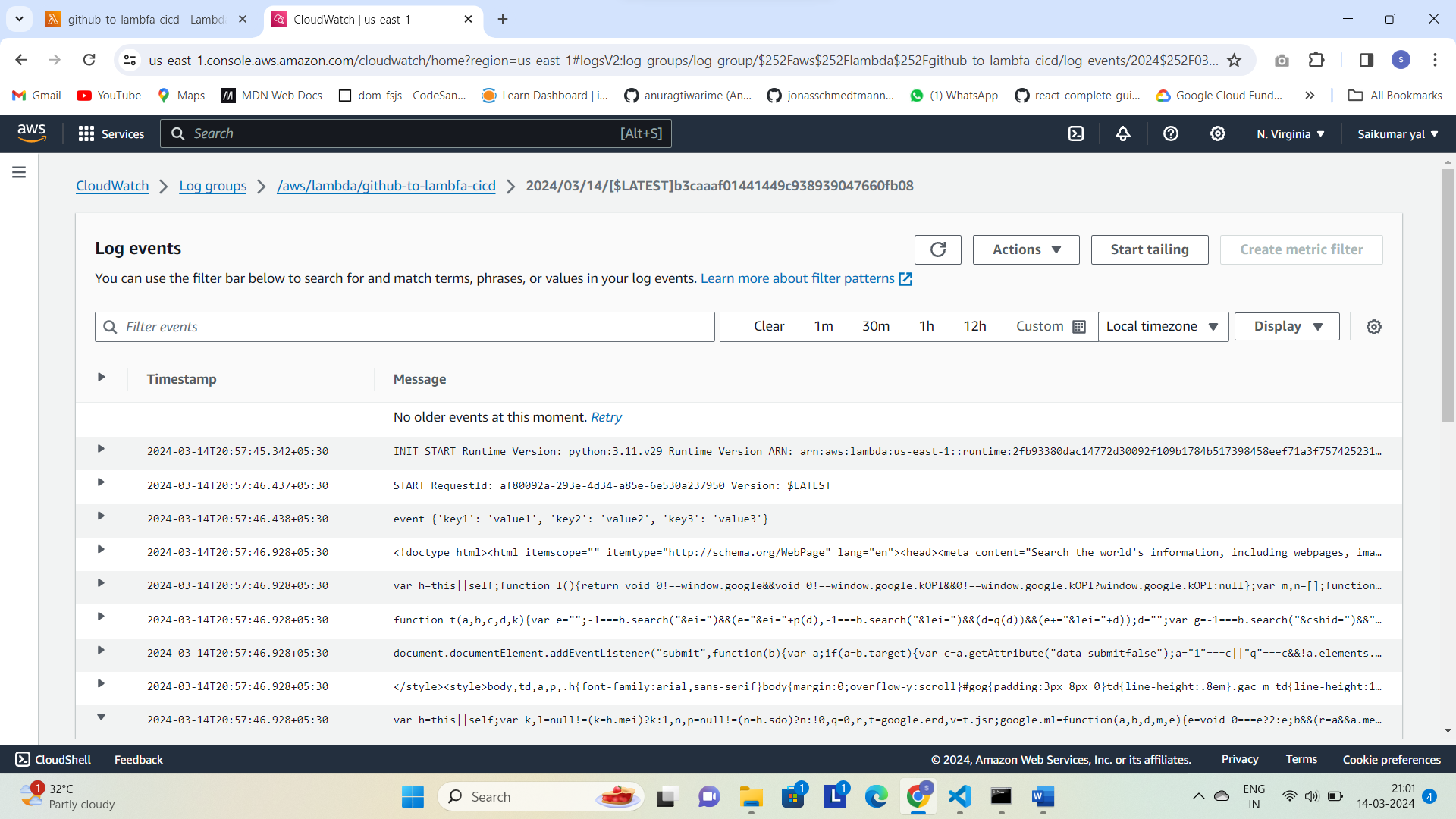
Next go to lambda function and test it by creating event and we can see output

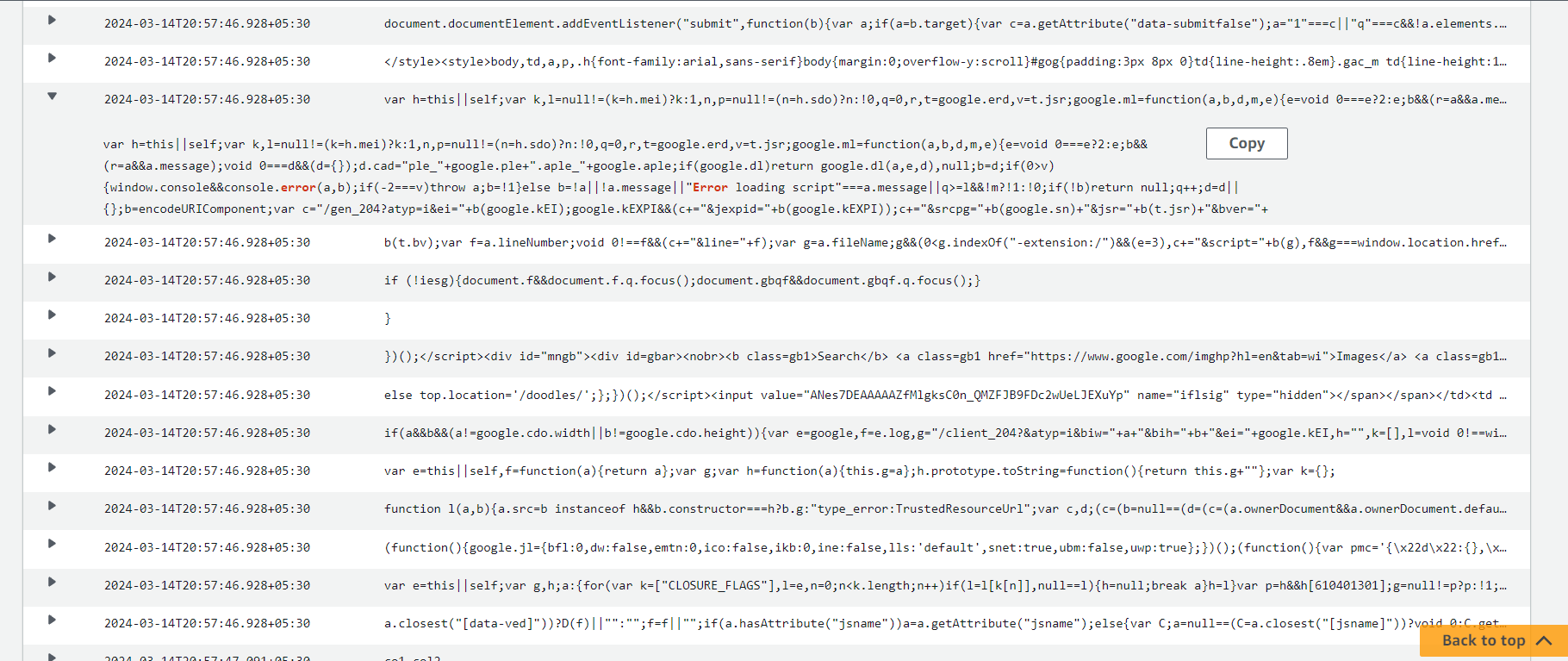






And we can see Cloudwatch logs for all output







In this we had seen vscode to github and to cicd