Code commit-> store the code in private git repository

Code build -> build the project (creates jar/zip/docker image) and push to respective place suppose jar file its store in s3 bucket , if it docker image it push that image in to ECR

**AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy.**

Code deploy -> deploy the code in respective target place(EBS/ECS)

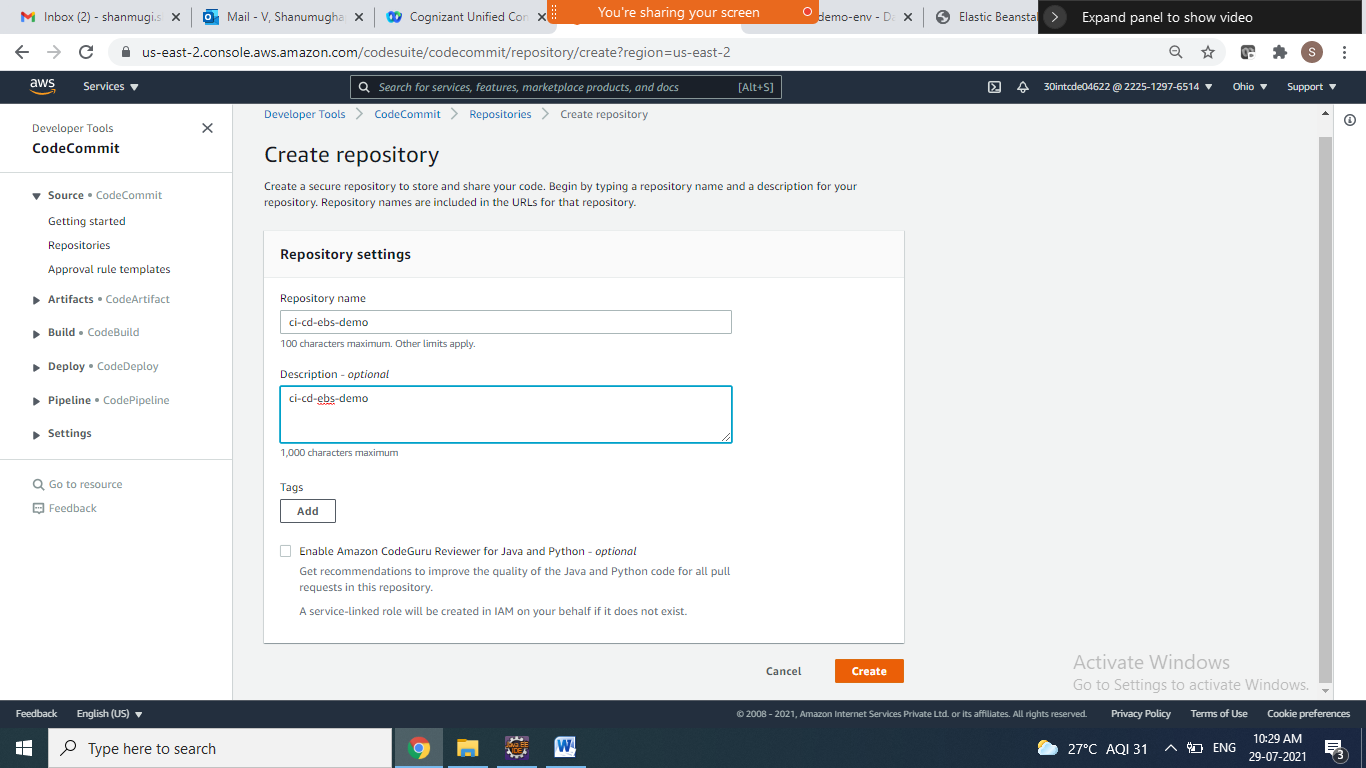
Code pipeline -> Continues delivery =it automates all the above process

**CI-CD-EBS**

1. Create project with buildspec.yml file
2. Create jar file
3. Login aws=> create EBS environment and upload your project and test it , it’s working

**CODE –COMMIT**

1. download code-commit credentials from IAM user
2. Create repository with project name and copy http clone url
   1. Give repository name and click create button



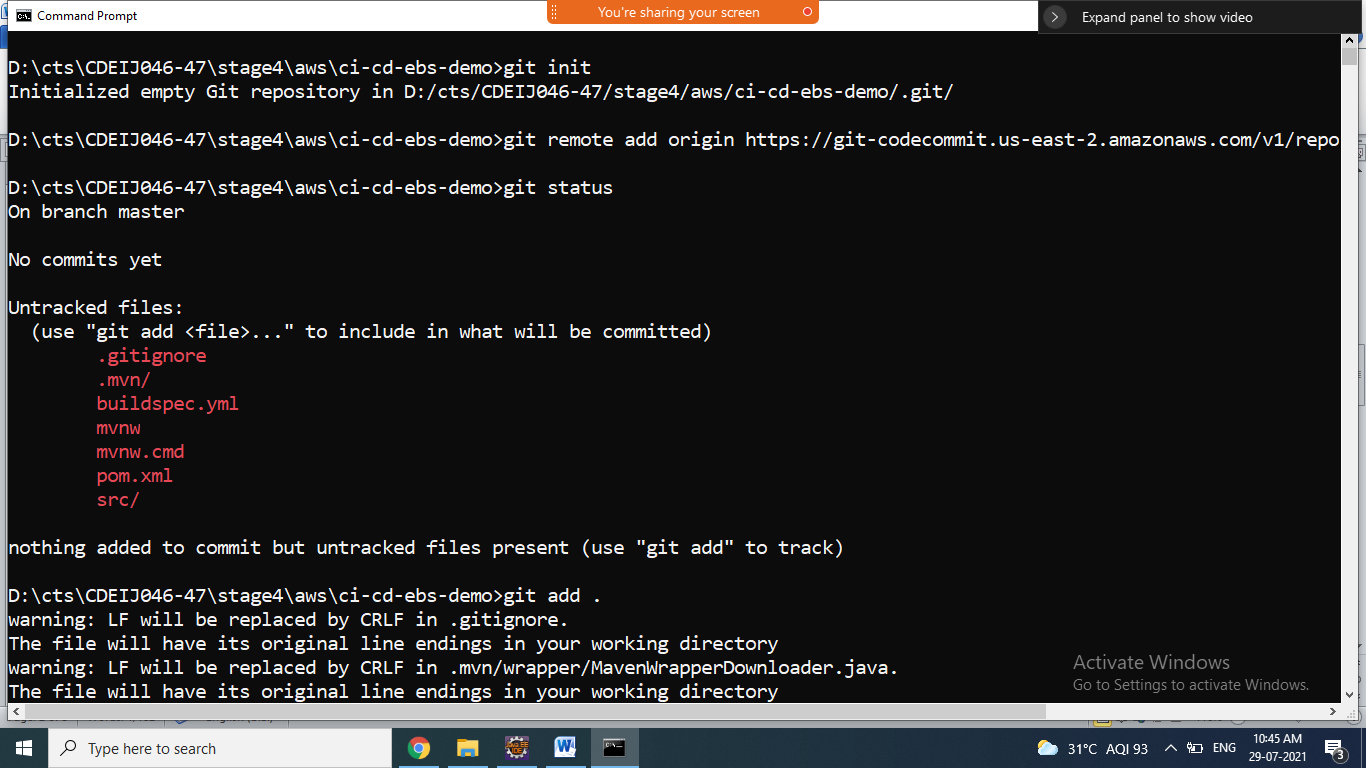
1. Go go cmd project folder and type below git commands

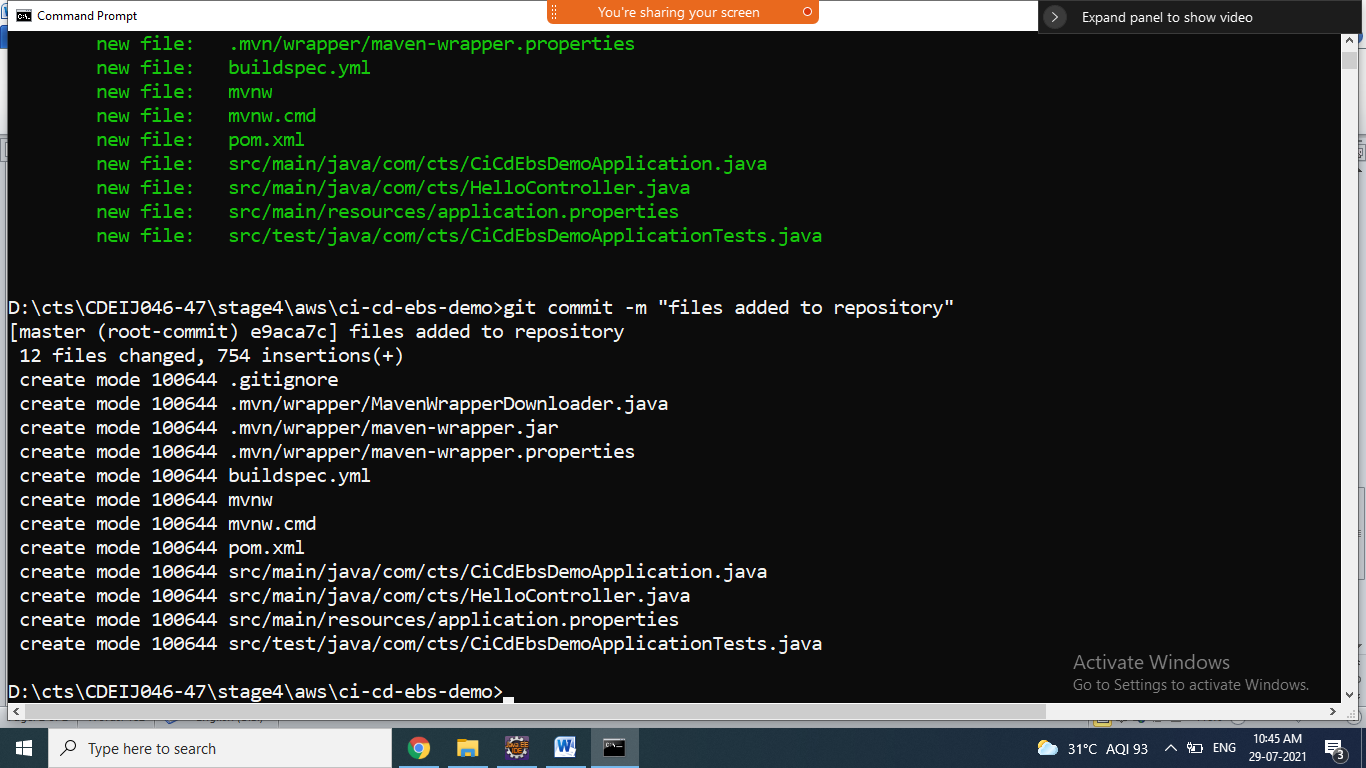
* git init
* git remote add origin <https://git-codecommit.us-east-2.amazonaws.com/v1/repos/ci-cd-ebs-demo>
* git status
* git add .
* git commit –m “files added to repository”
* git status
* git push origin master

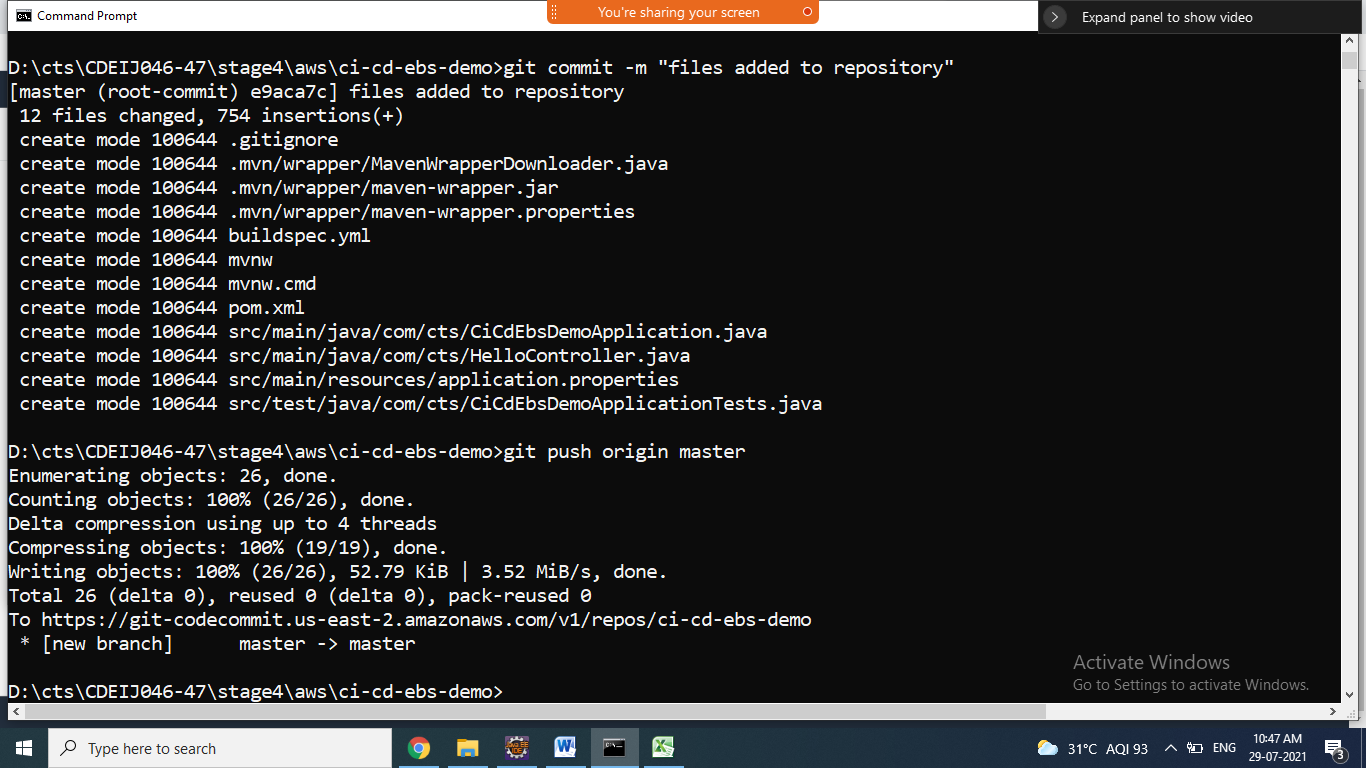
it ask username and password

provide codecommit username and password

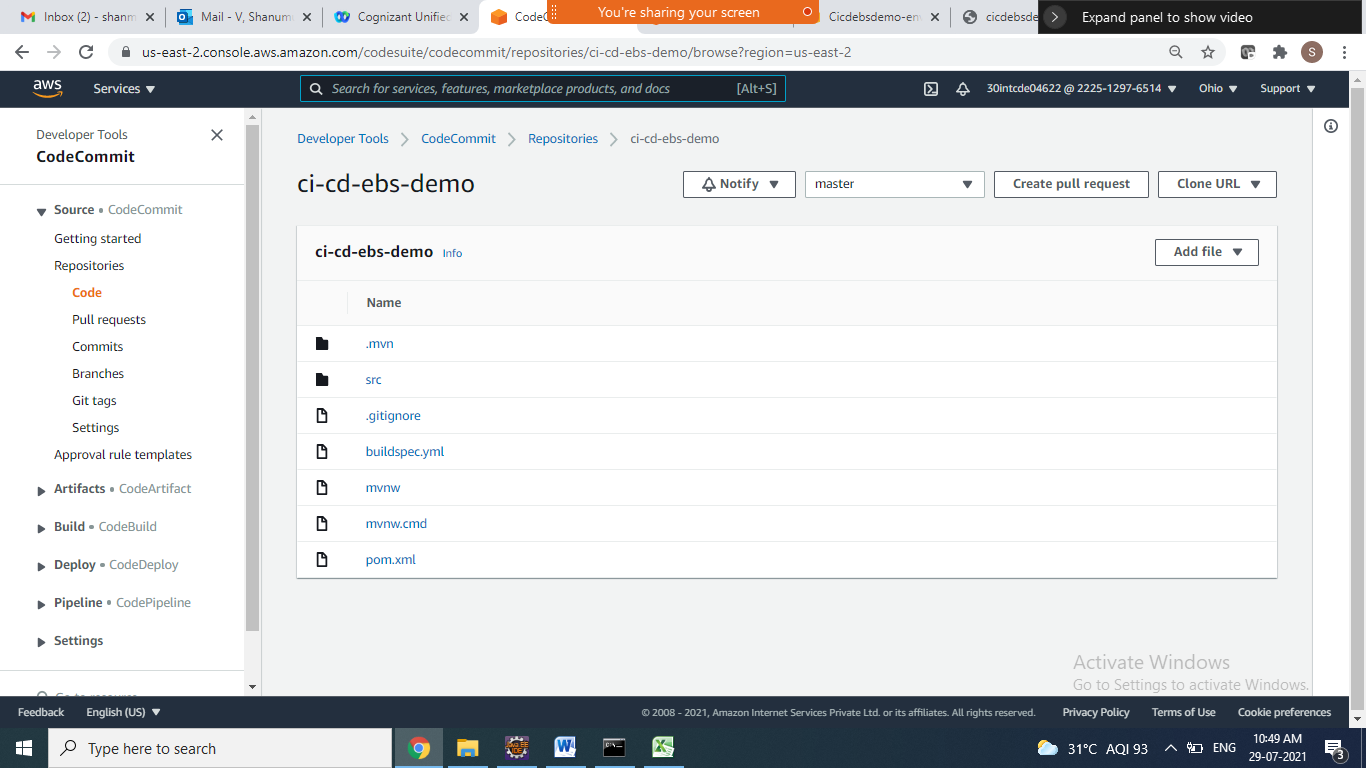
then it successfully pushed your project into code commit.







1. Now check all files copied into code commit repository



**CODE-BUILD:**

1. **Aws – code build => getting started**
2. **Click on create project button**

***Project Configuration section***

Project Name : cicd-ebs-build

***Source section***

Source provider: AWS codeCommit

Repository: select ur repository ci-cd-ebs-demo

Branch : choose master

*Environment Section:*

Environment image: choose Managed image

Operating system: choose Amazon Linux2

Runtime(s): standard

Image : choose latest version images

Image version :choosed automatically

Environment type : linux

Service role: choose New service role

Role name: created automatically

codebuild-cicd-ebs-build-service-role

Additional configuration : leave default

*Buildspec section*

Build specifications : select Use a buildspec file

Buildspec name – optional

Suppose file is inside repo/project folder means : give path and file name in the text box

Leave all these 3 things default

***Batch Configuration***

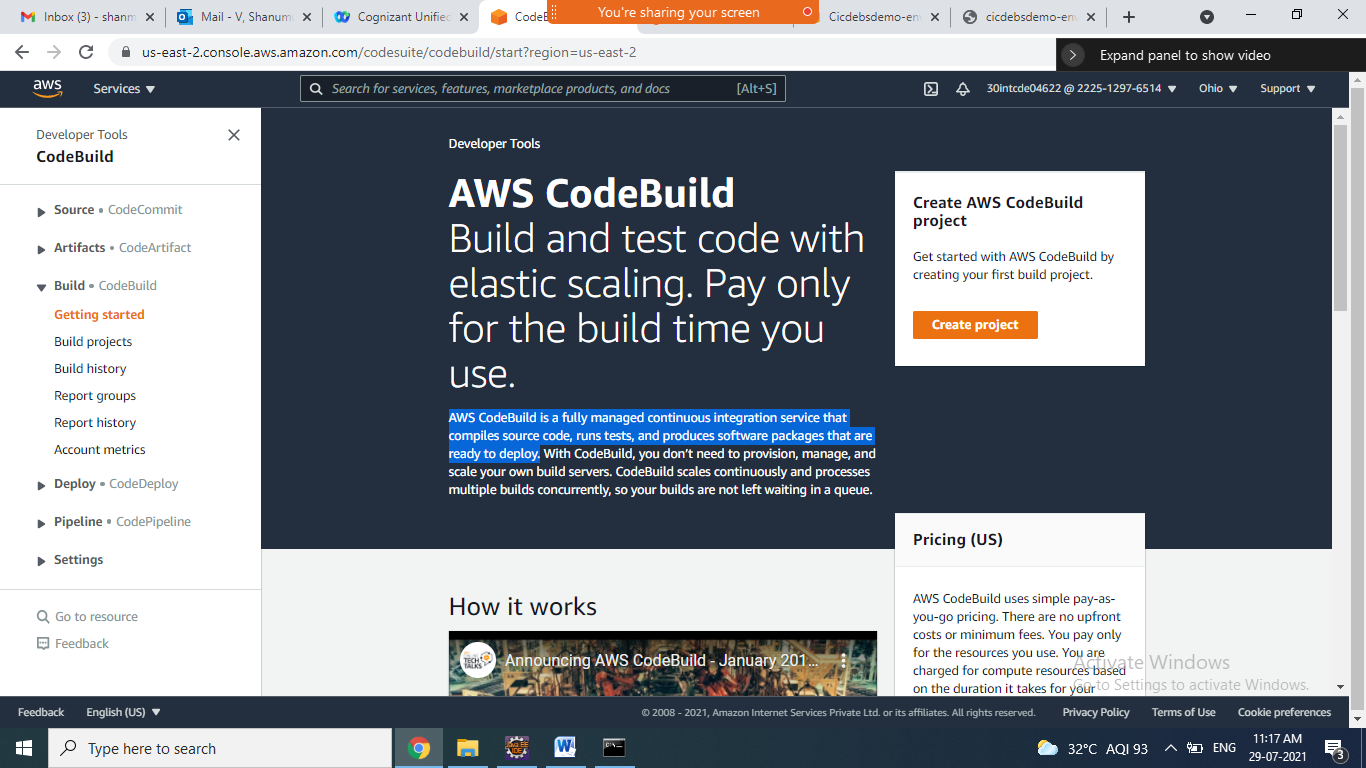
***Artifacts => choose no artifacts***

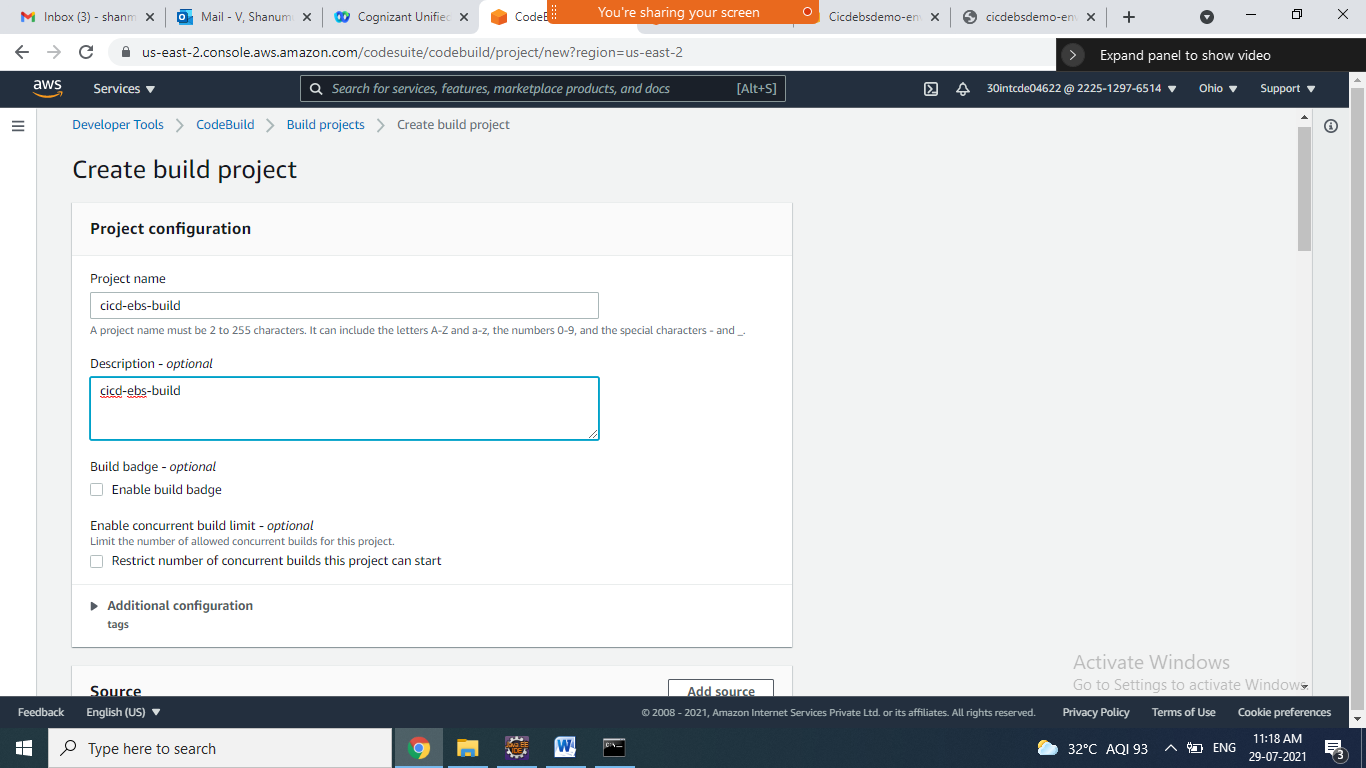
***Logs***

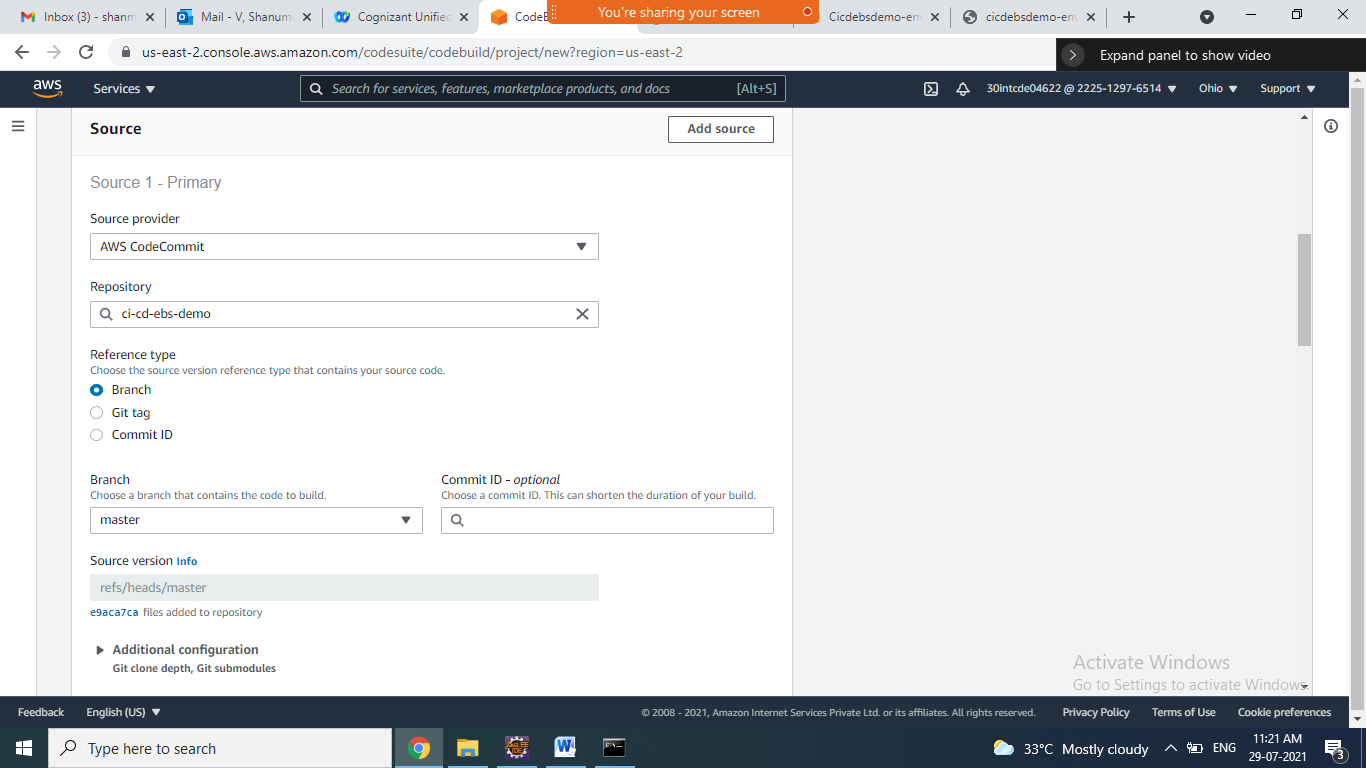
1. **Click on the Create build project button**

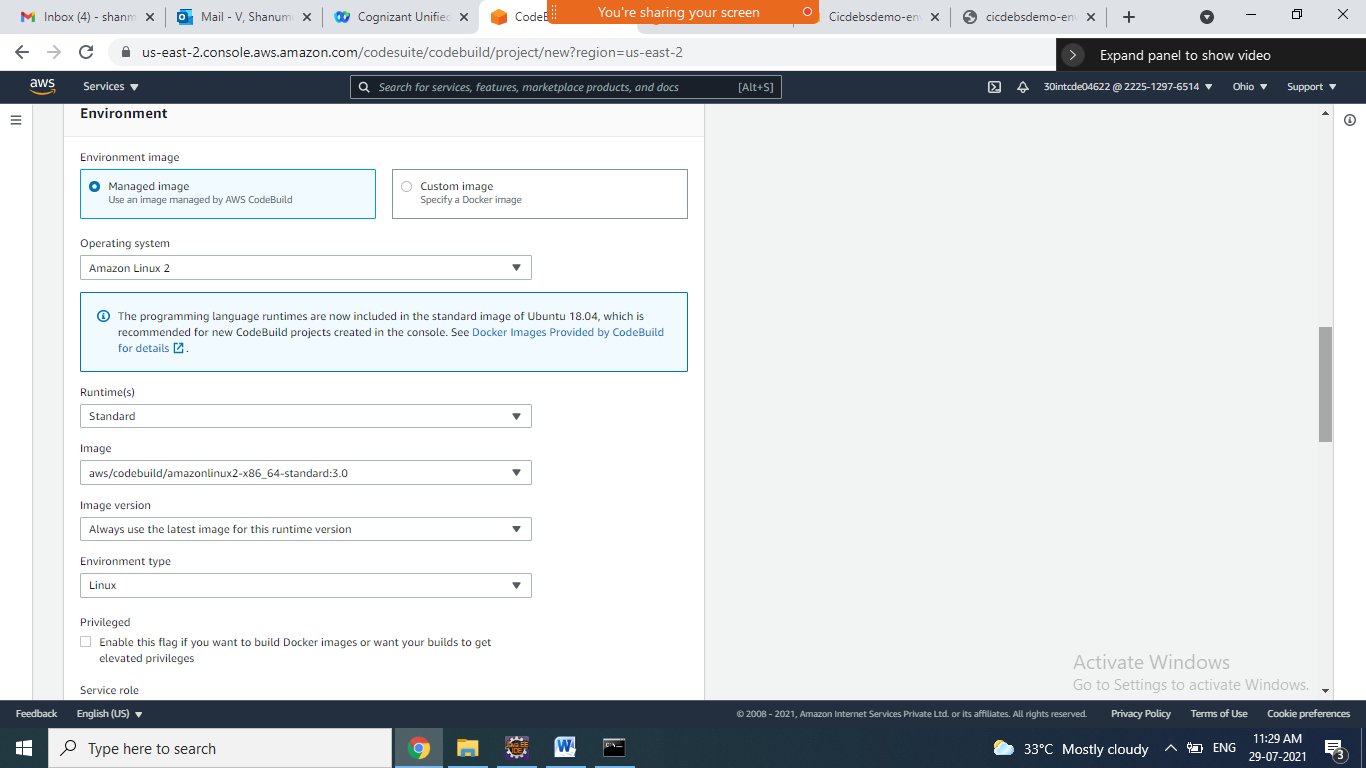
**Successfully build created**

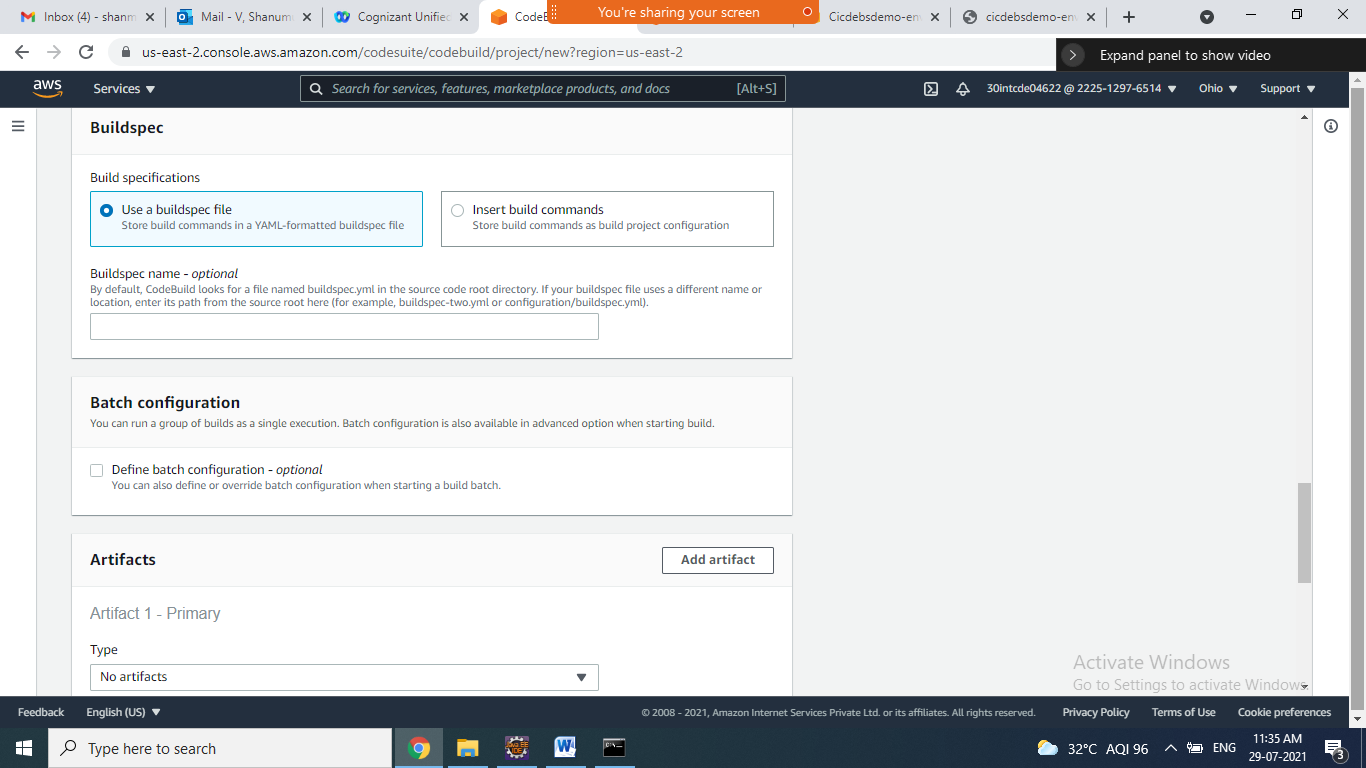
1. **Click on start build button**
2. **See the build logs and build phase successful**

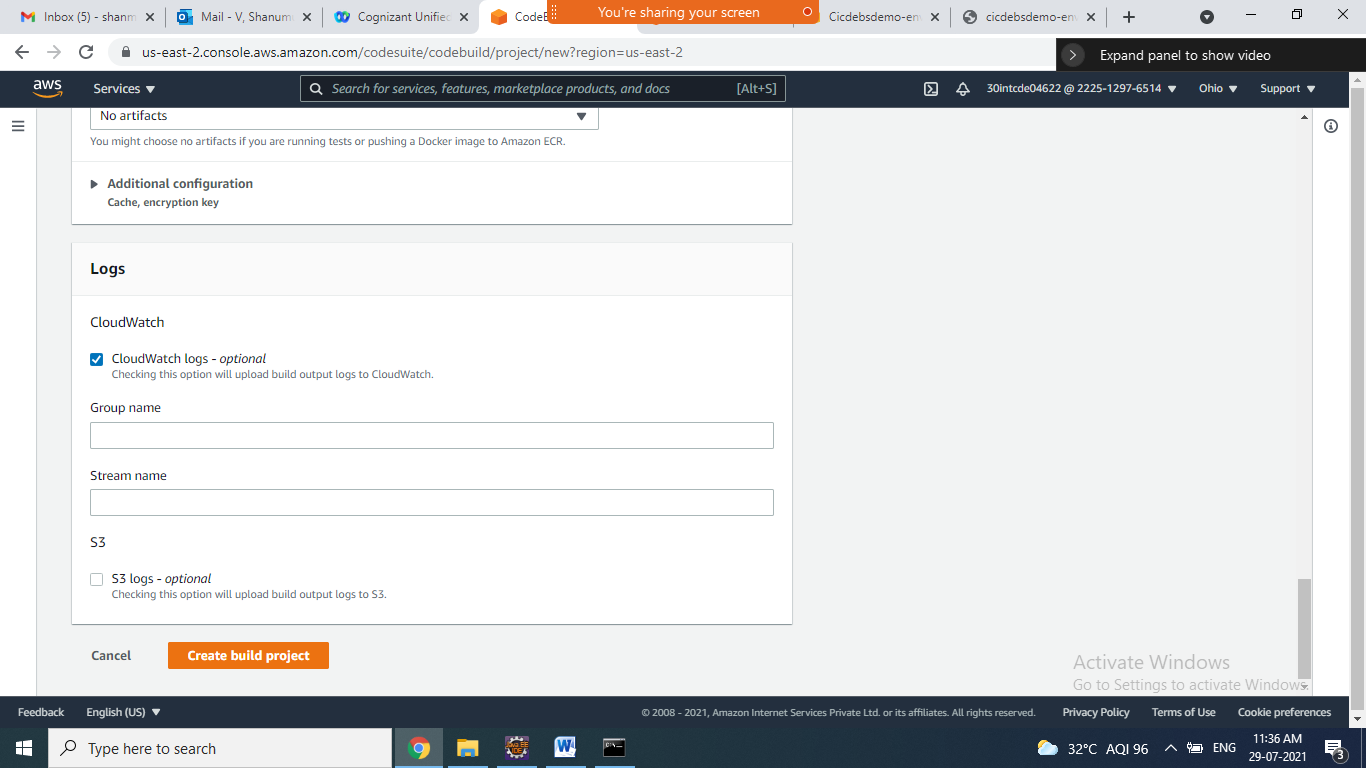


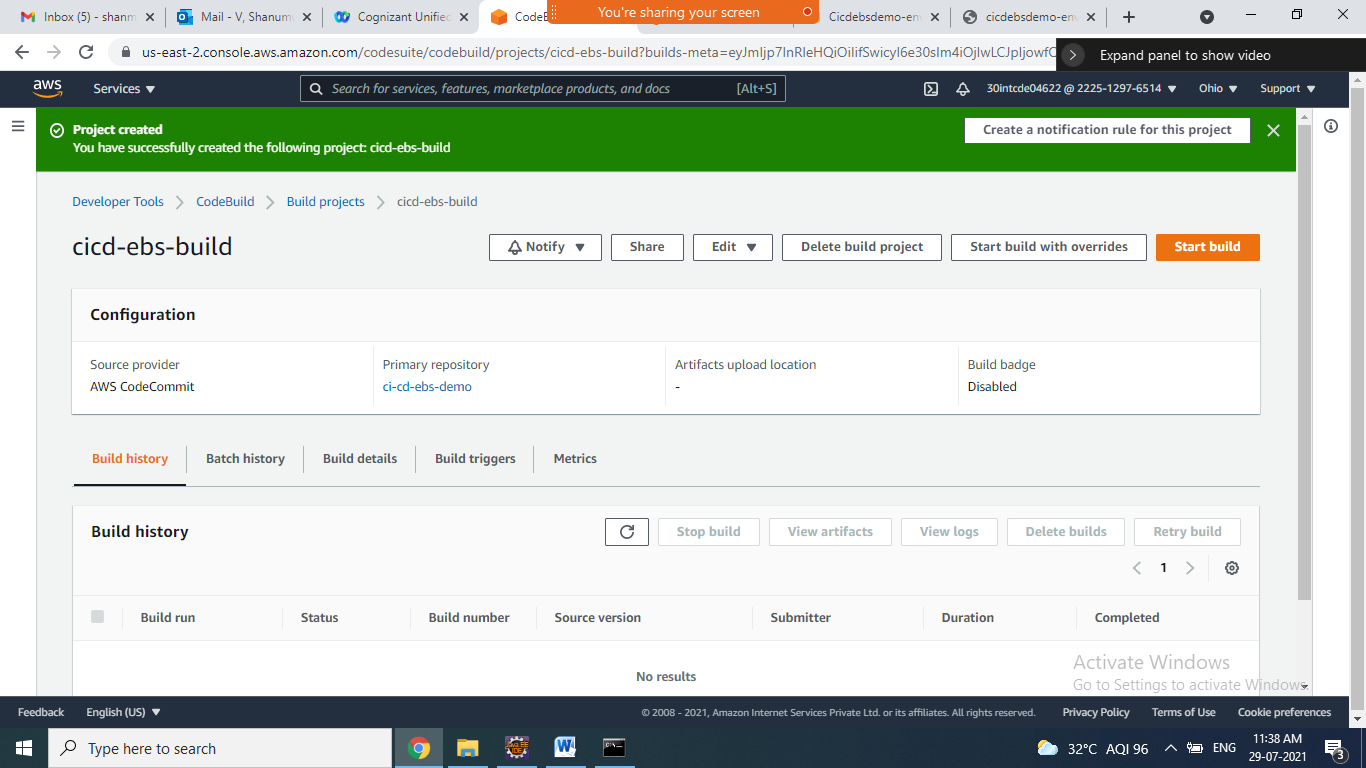


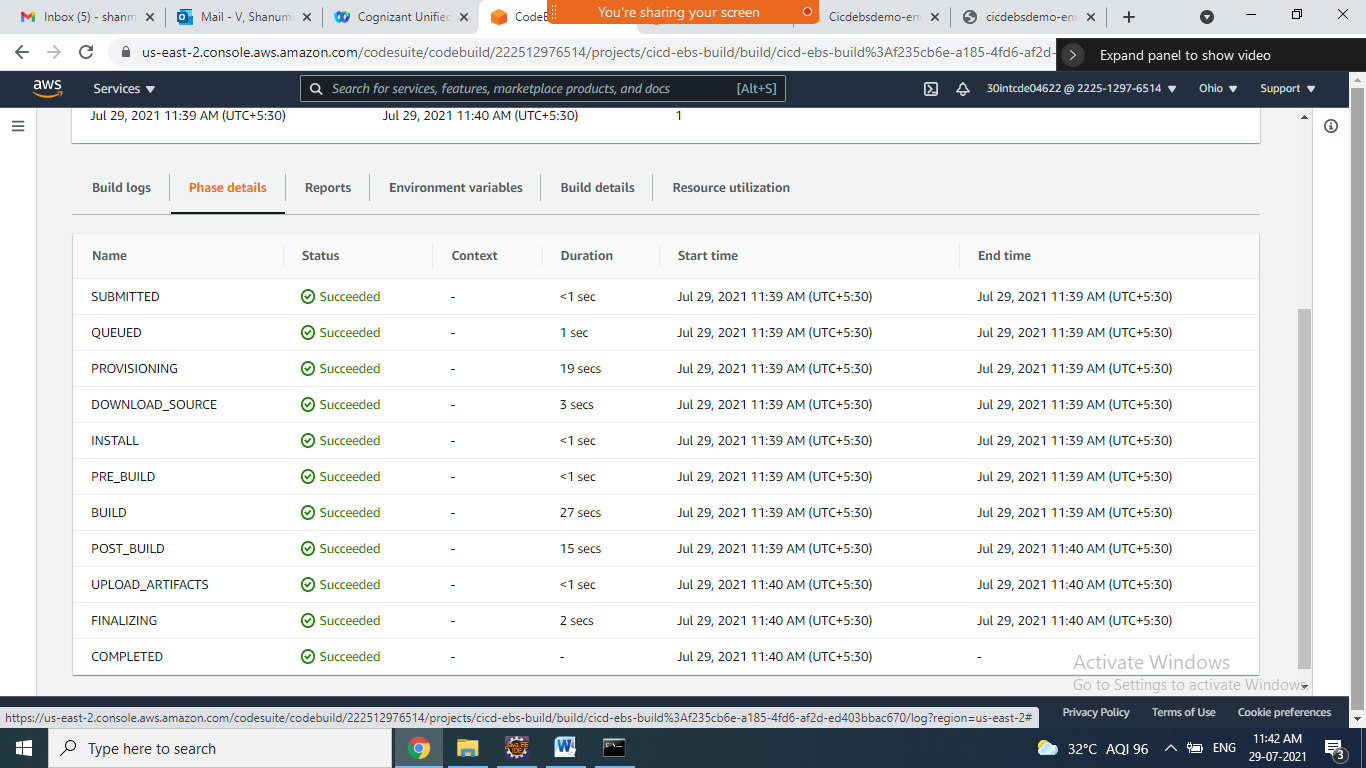


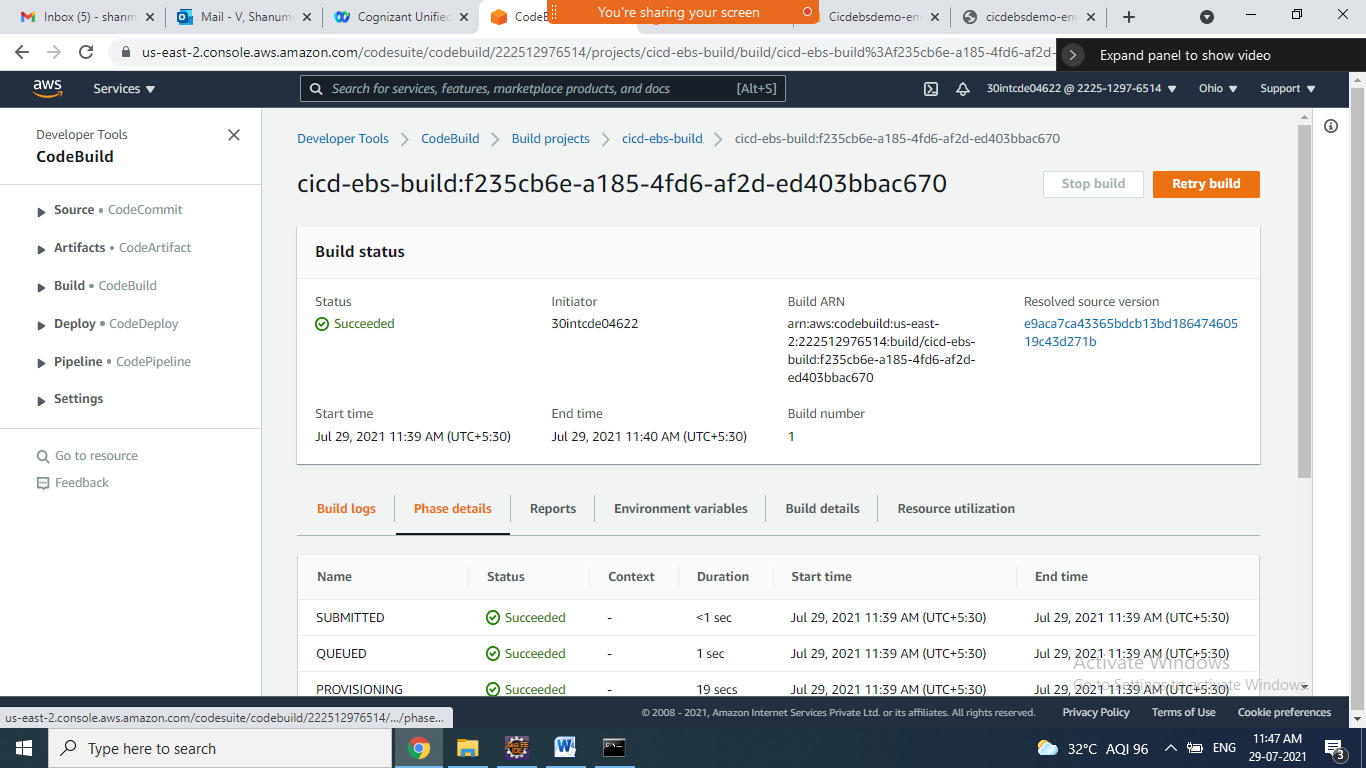












**CODE –PIPE**

1. **Goto code pipe => click on getting started => click on create pipeline button**
2. **Step 1**

**Choose pipeline settings**

Pipeline name : cicd-ebs-pipe

Service role: choose New service role

**Click on Next button**

1. **Step 2**

**Add source stage**

Source provider: AWS CodeCommit

Repository name : choose ur repository

Branch name: choose master

Change detection options :Choose Amazon CloudWatch Events (recommended)

Output artifact format: Choose CodePipeline default

**Click on Next button**

1. **Step 3**

**Add build stage**

Build provider : choose AWS Code Build

Project name: choose ur build project ie cicd-ebs-build

Build type : select Single build

**Click on Next button**

1. **Step 4**

**Add deploy stage**

Deploy provider : Choose AWS Elastic Beanstalk

Region : US East (Ohio)

Application name: Choose ur applicationname which u created EBS deployment => cicdebsdemo

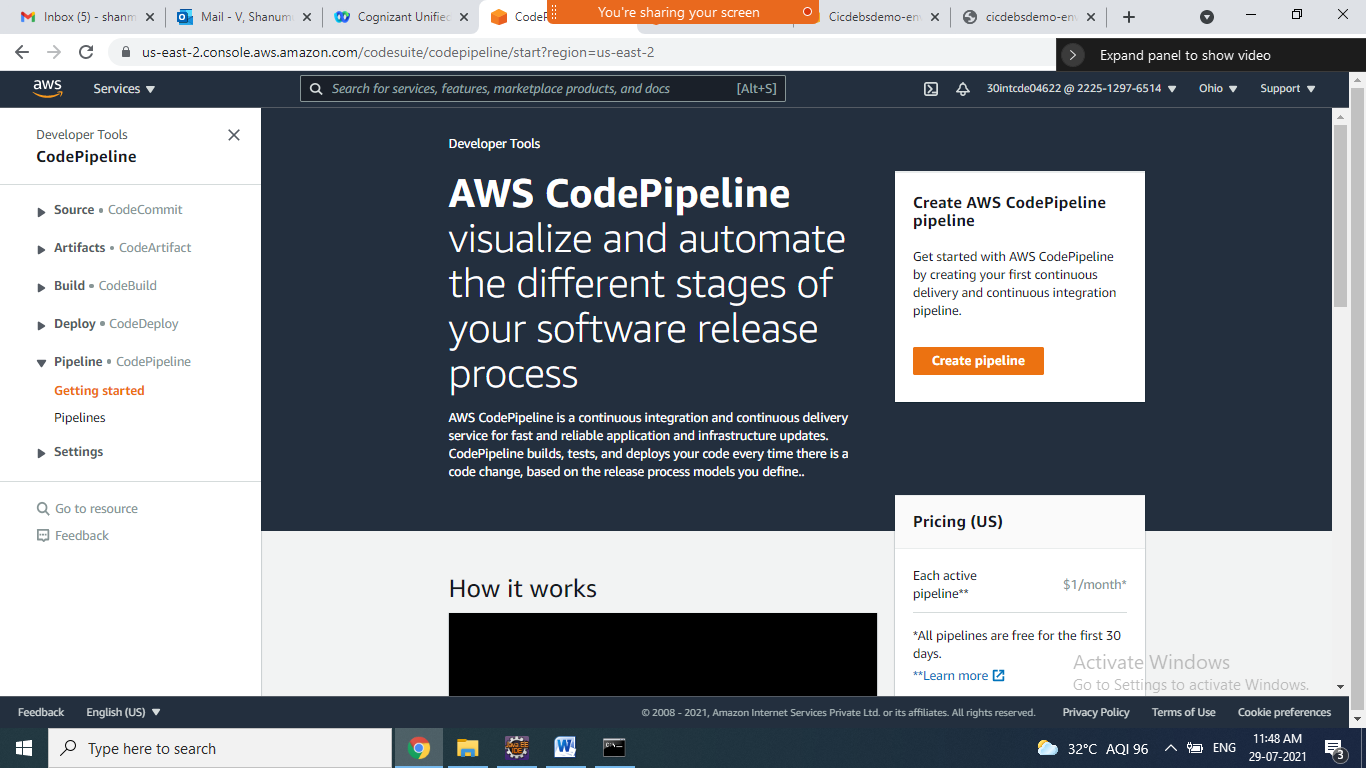
Environment name: choose ur env name which u created EBS deployment => cicdebsdemo-env

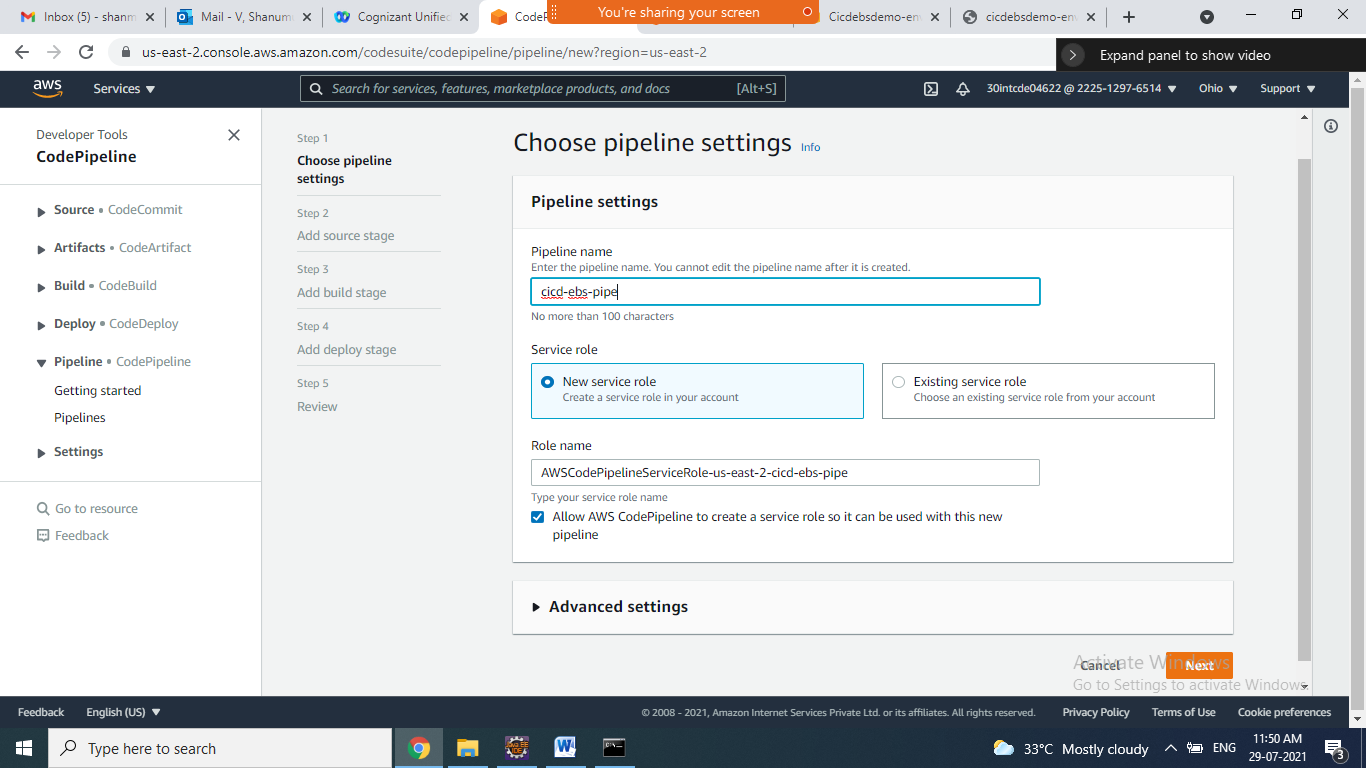
1. **Step 5**

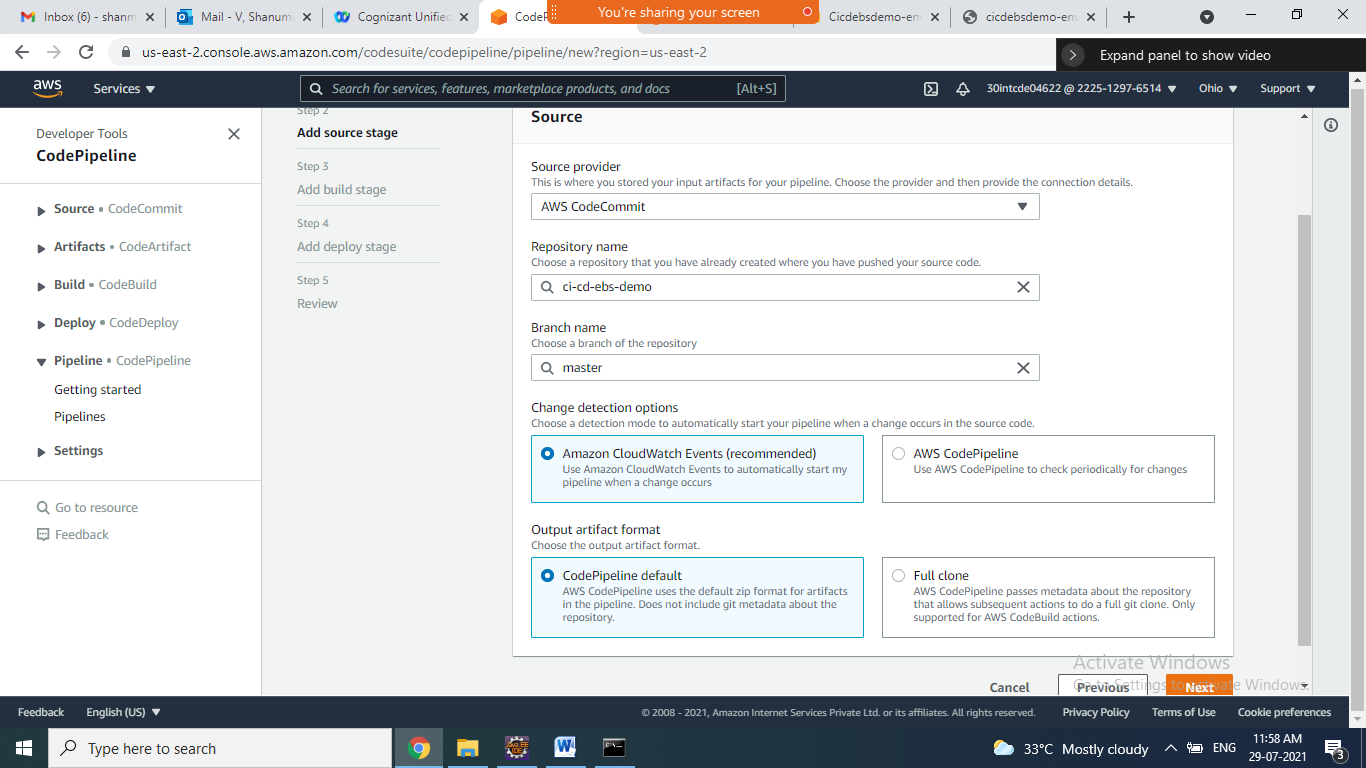
**Review all steps are correct click on create pipeline button**

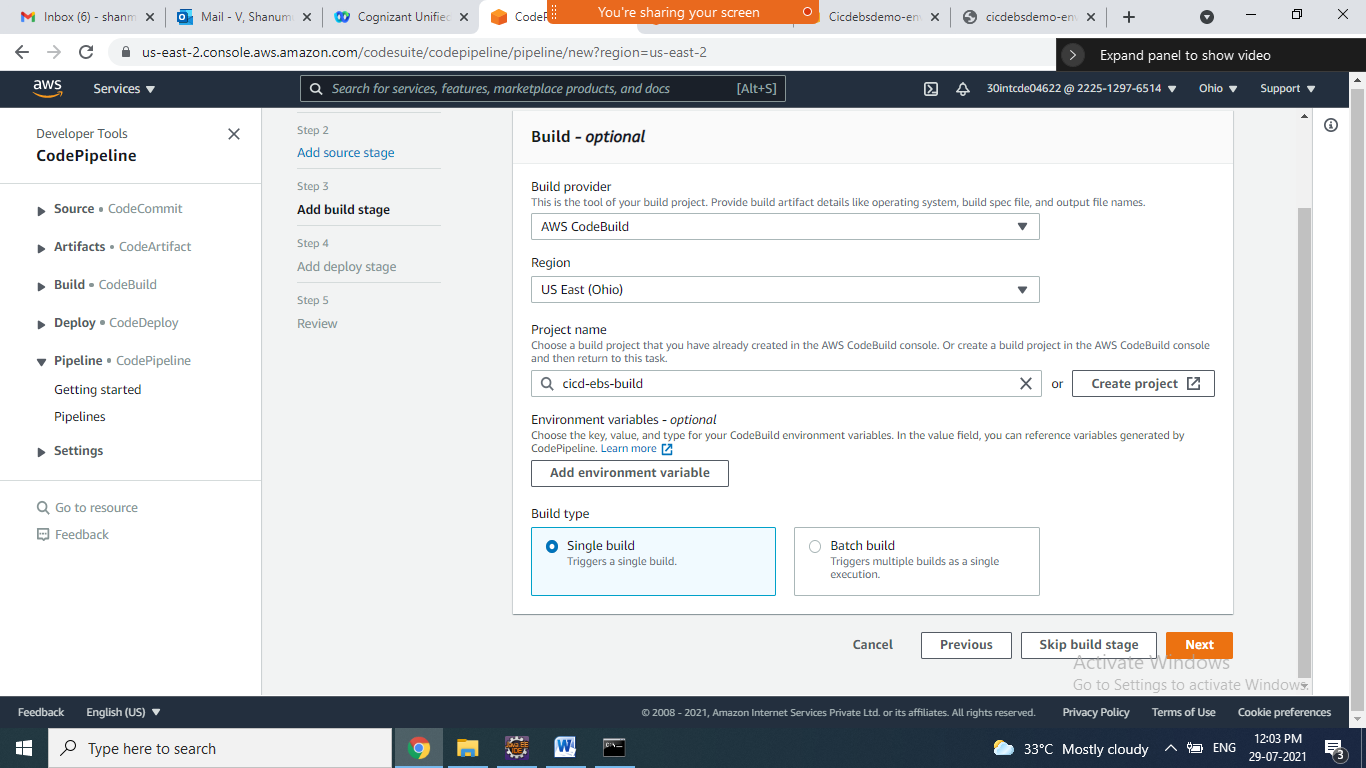
**Successfully pipeline get started source=>build =>deploy**

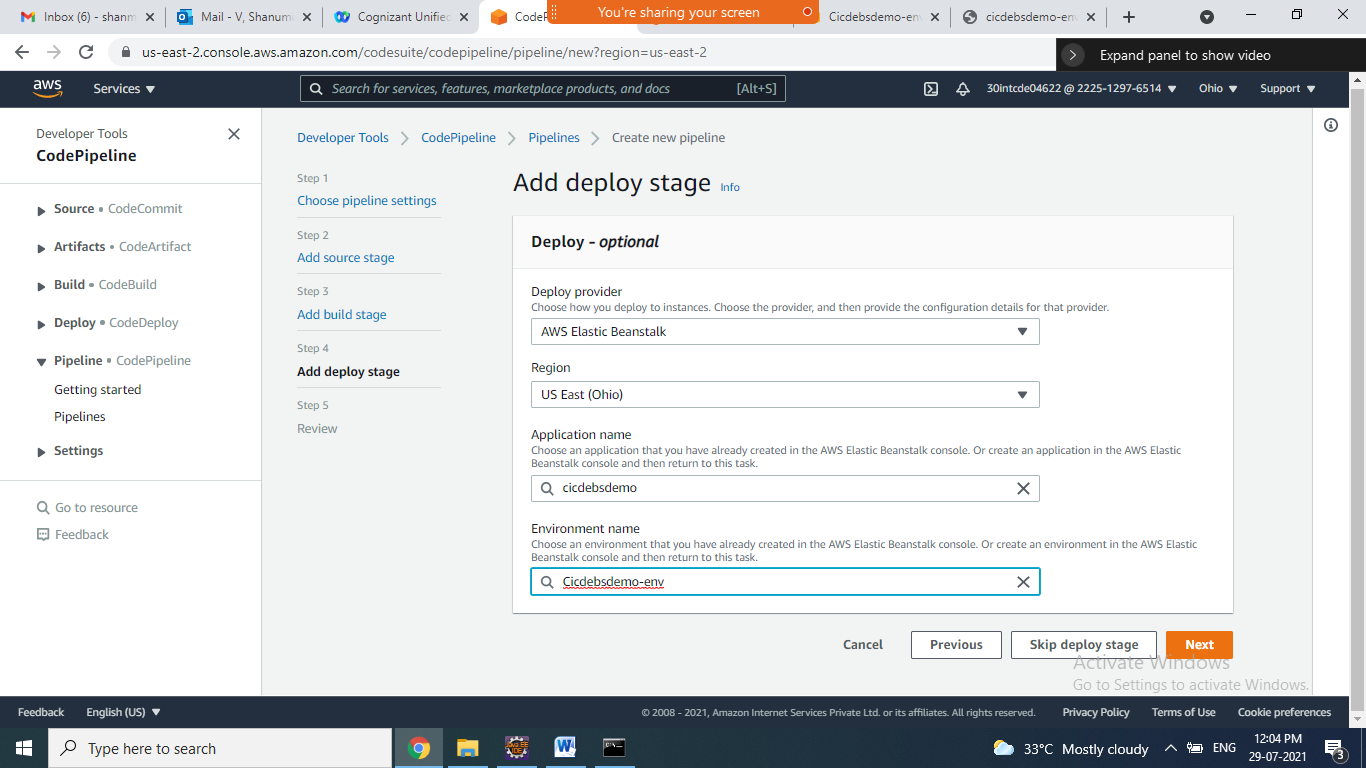
**All green color => go to browser and check it**

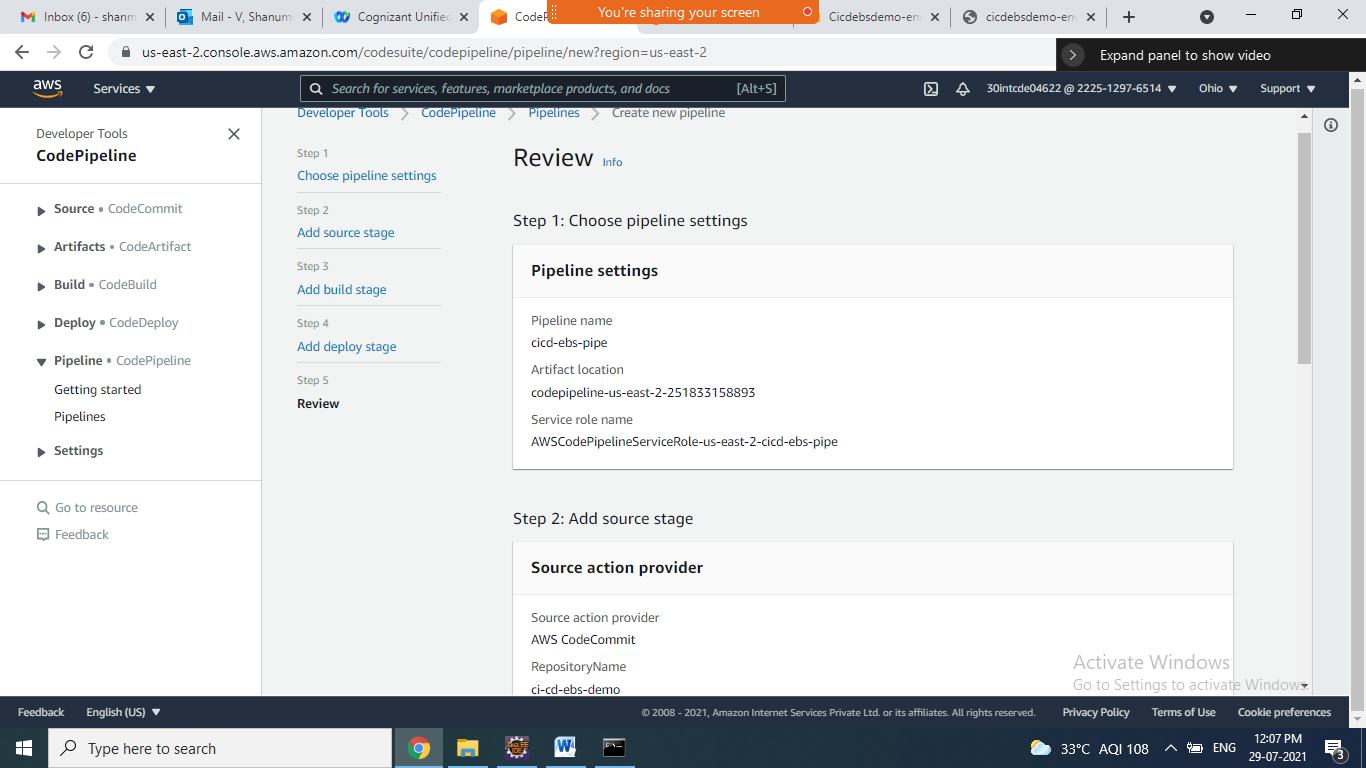


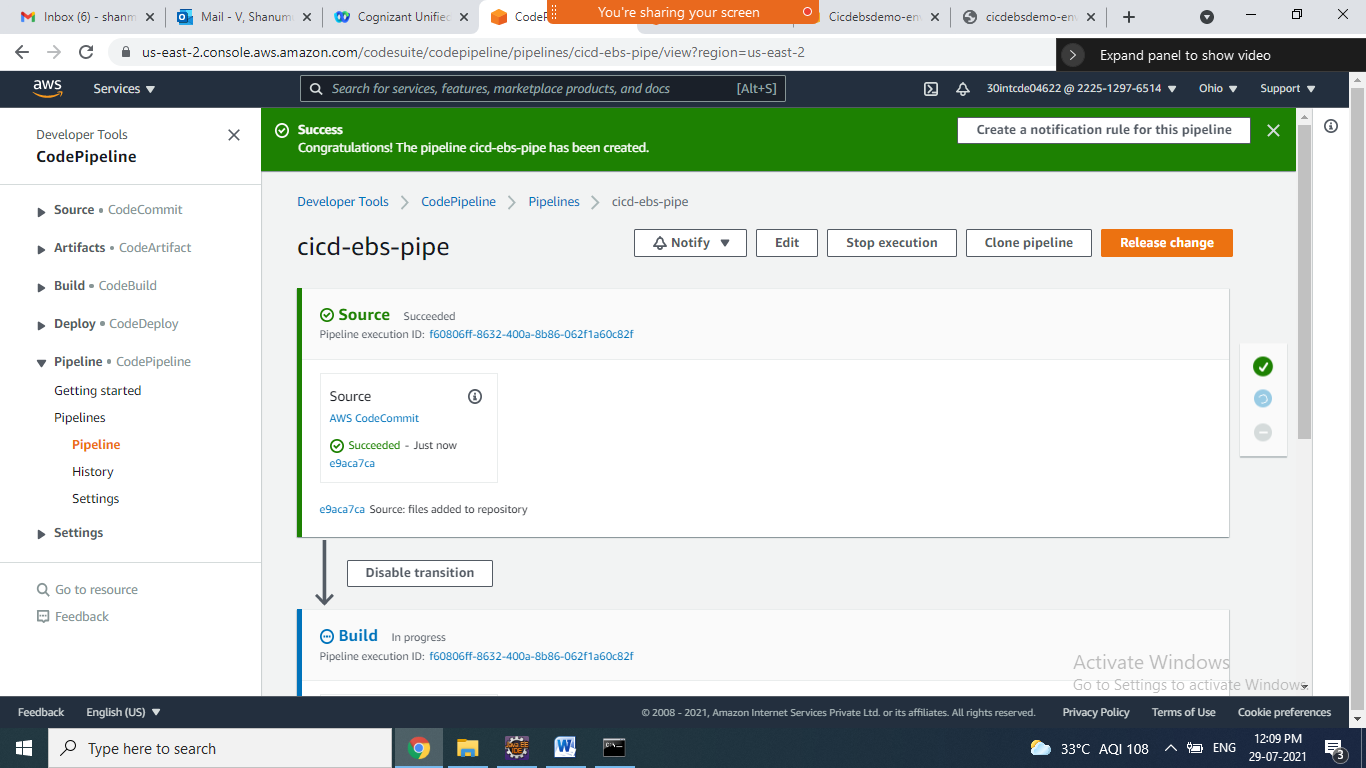


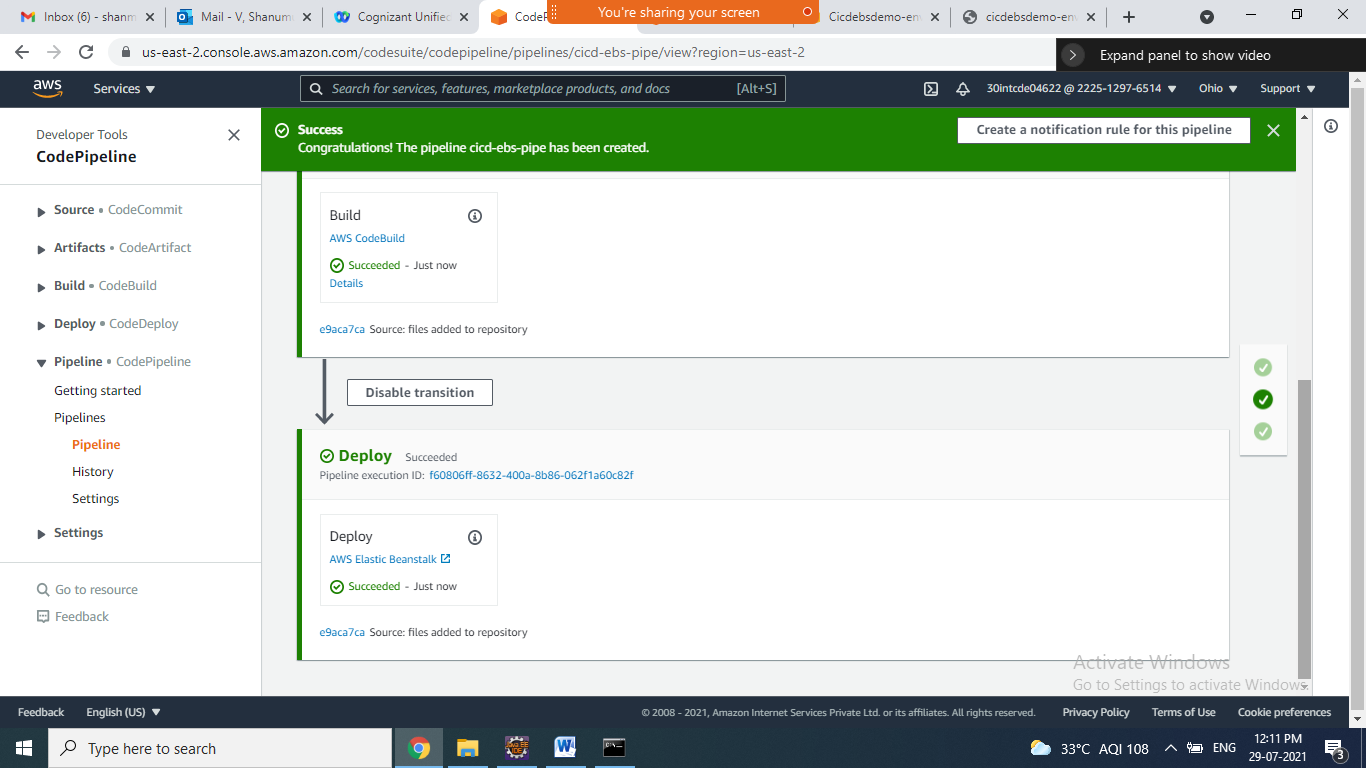


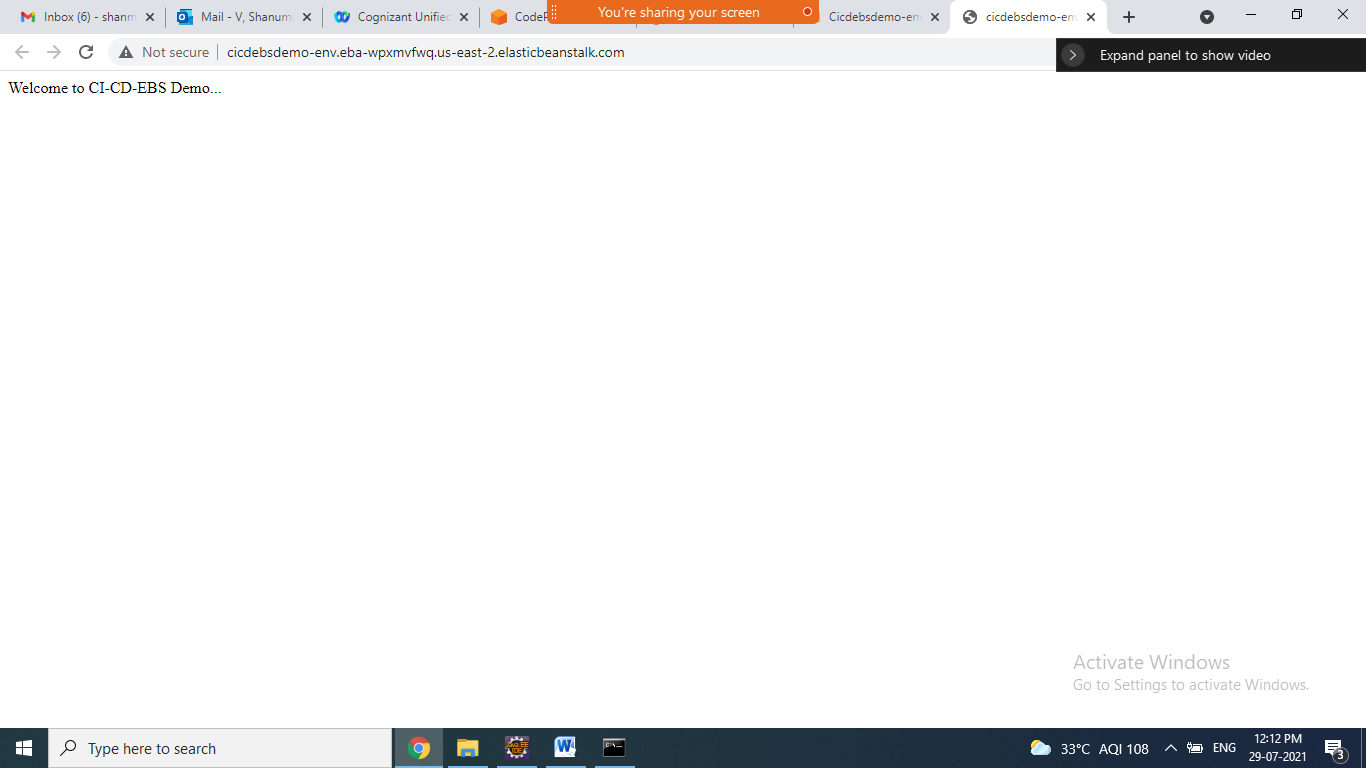












**Goto Eclipse modify ur code and test**

**Create one method in controller and test it**

**After adding goto cmd type below commands**

@GetMapping("/hi")

public String sayHi1() {

return "Hi from CI-CD-EBS Demo...";

}

