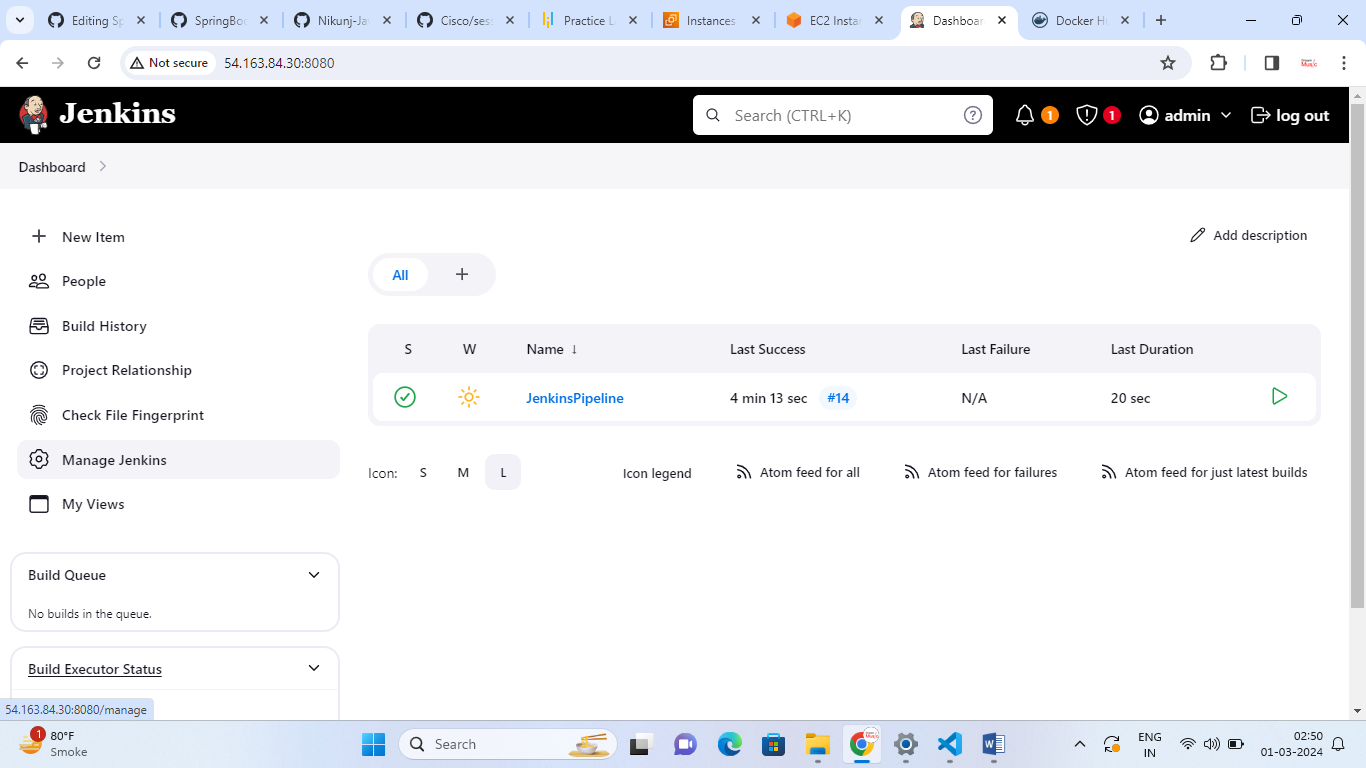
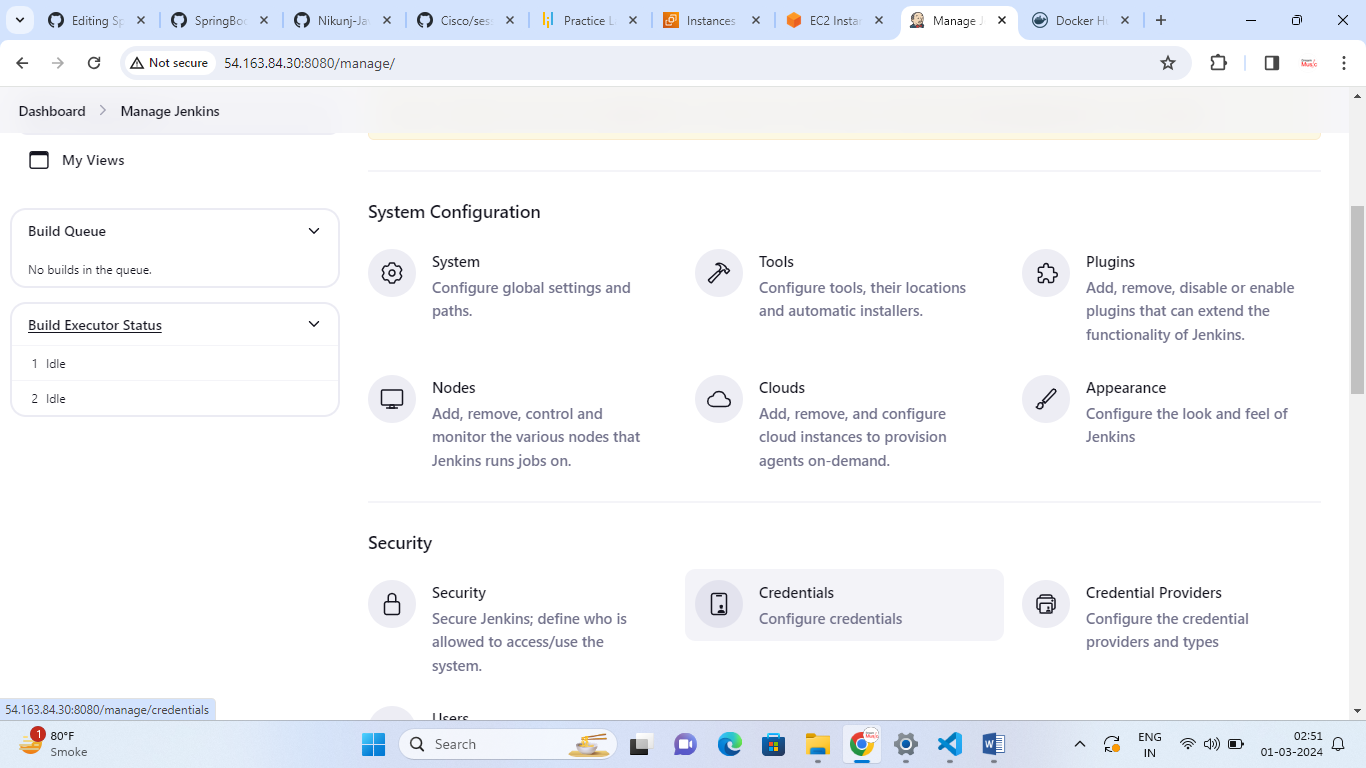
If you want to Push Your Code to Docker Hub follow This Steps

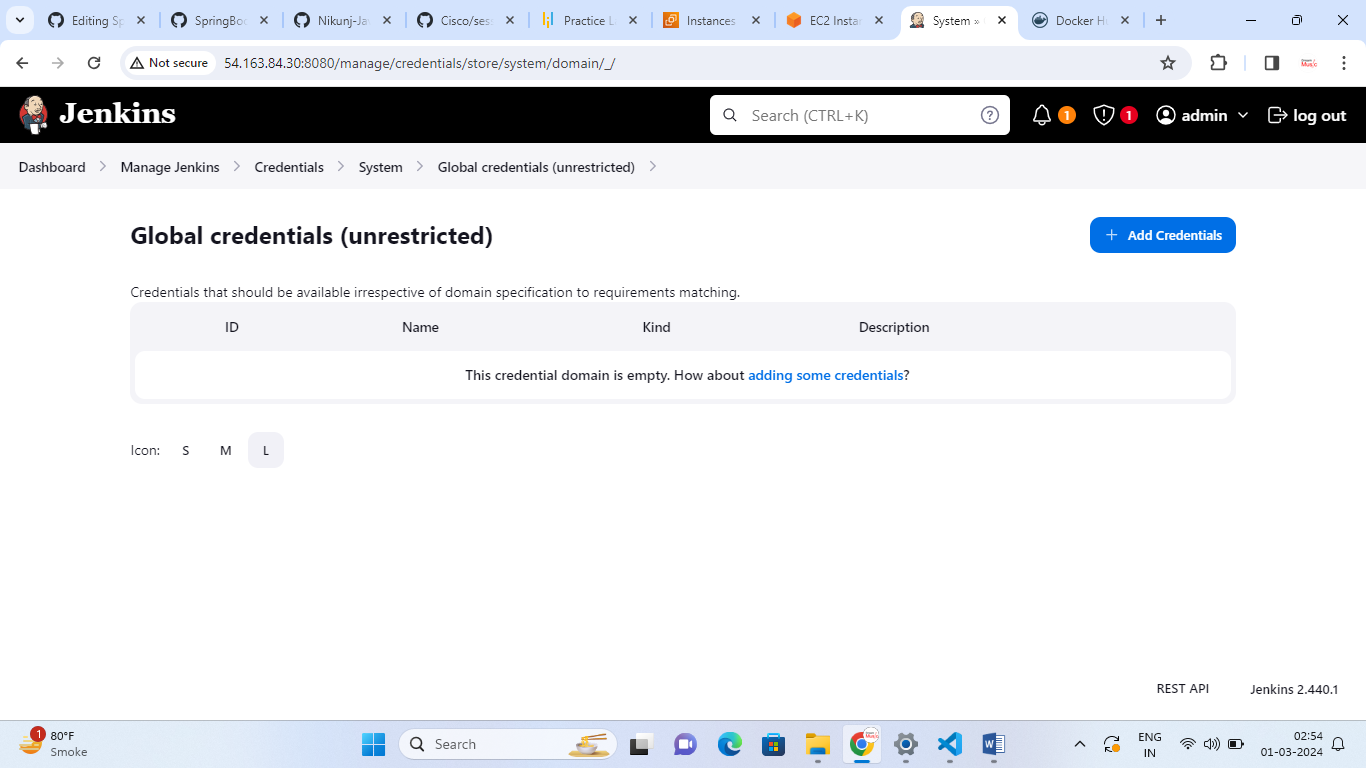
1. GOTO> MANAGE JENKINS>



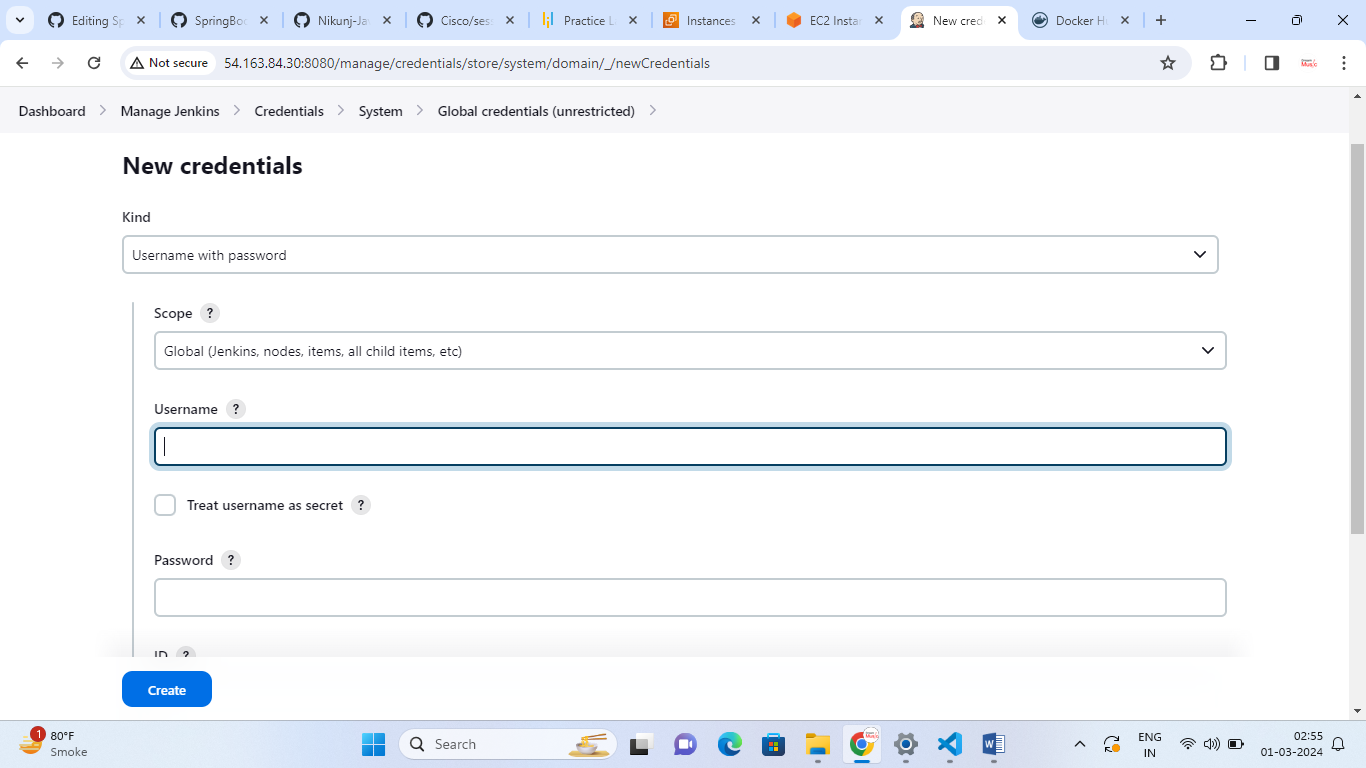
1. GOTO> CREDENTIALS



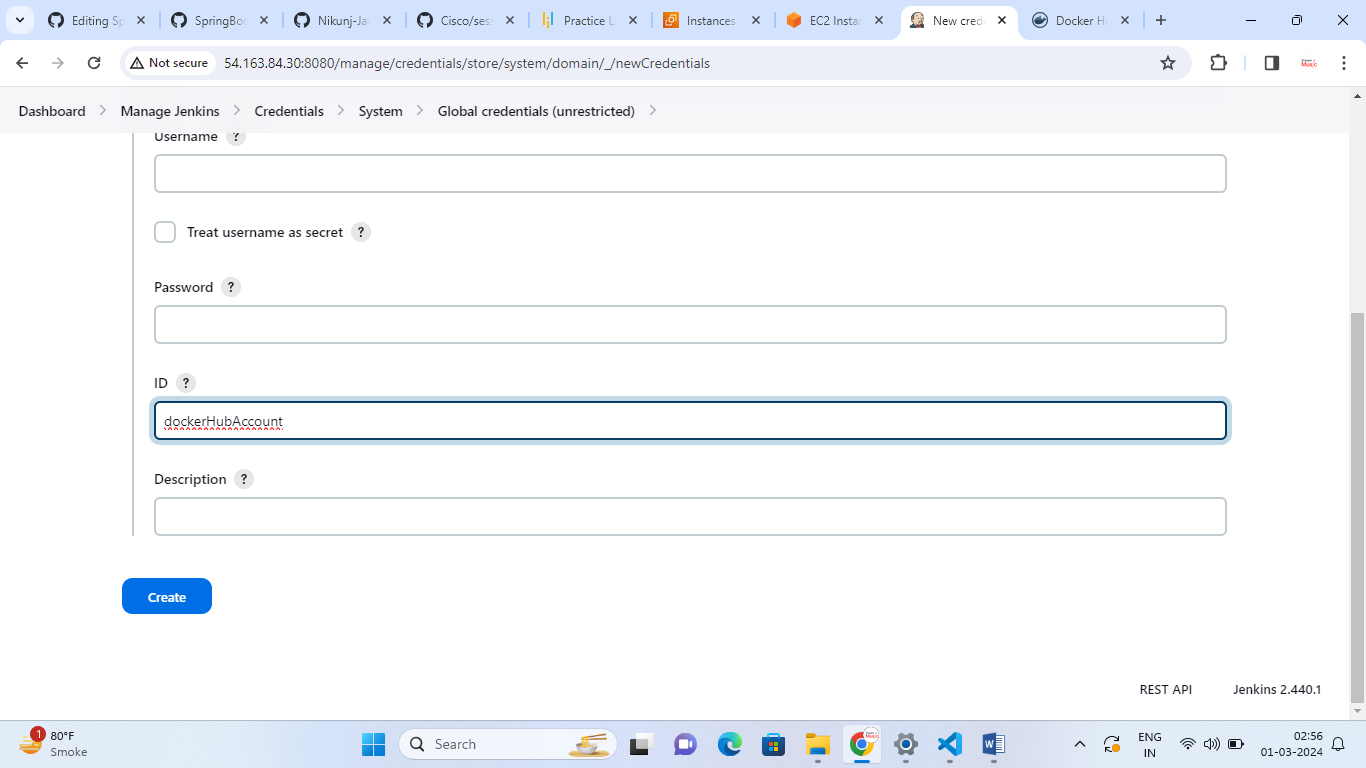
1. CLICK ON ADD CREDENTIALS



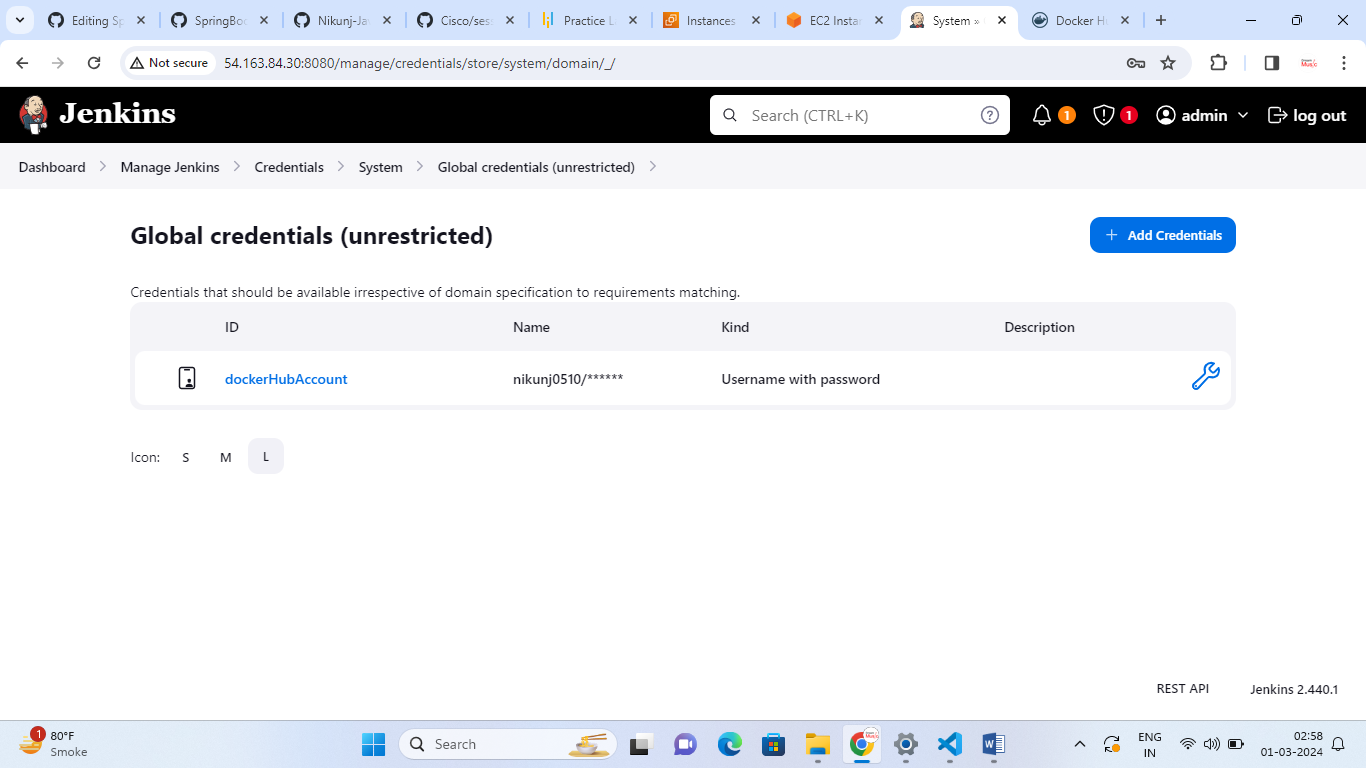
1. FILL THE FORM



Provide your DOCKERHUB Username and Password



* Give the id name as : dockerHubAccount
* If you provide different it will not work
* Then click on Create



Now You will be able to see your Credentials for Docker Hub Account

* Now Update Your Jenkins File
* Add Build Stage as Below

stage('Push to Docker Registry'){

        withCredentials([usernamePassword(credentialsId: 'dockerHubAccount', usernameVariable: 'dockerUser', passwordVariable: 'dockerPassword')]) {

            sh "docker login -u $dockerUser -p $dockerPassword"

            sh "docker tag $imageName:${env.BUILD\_NUMBER} $dockerUser/$imageName:${env.BUILD\_NUMBER}"

            sh "docker push $dockerUser/$imageName:${env.BUILD\_NUMBER}"

            echo "Image push complete"

        }

Jenkins File

def containerName="spring\_container"

def imageName="spring\_image"

def dockerHubUser="nikunj0510"

def tag="latest"

node {

  stage('Checkout Source Code') {

    checkout scm

  }

   stage("Compilation") {

    sh "mvn clean install -DskipTests"

   }

    stage('Image Build'){

        sh "docker build -t $imageName:${env.BUILD\_NUMBER} --pull --no-cache ."

        echo "Image build complete"

    }

    stage ('Run Application') {

        try {

          // Stop existing Container

           //sh 'docker rm $containerName-${env.BUILD\_NUMBER} -f'

          // Start database container here

          sh "docker run -d --name $containerName -p 8082:8082 $imageName:${env.BUILD\_NUMBER}"

         // sudo docker run -d --name mycontainer-${env.BUILD\_NUMBER} -p 8082:8082 spring\_container:5

        }

       catch (error) {

        } finally {

          // Stop and remove database container here

        }

    }

    stage('Push to Docker Registry'){

        withCredentials([usernamePassword(credentialsId: 'dockerHubAccount', usernameVariable: 'dockerUser', passwordVariable: 'dockerPassword')]) {

            sh "docker login -u $dockerUser -p $dockerPassword"

            sh "docker tag $imageName:${env.BUILD\_NUMBER} $dockerUser/$imageName:${env.BUILD\_NUMBER}"

            sh "docker push $dockerUser/$imageName:${env.BUILD\_NUMBER}"

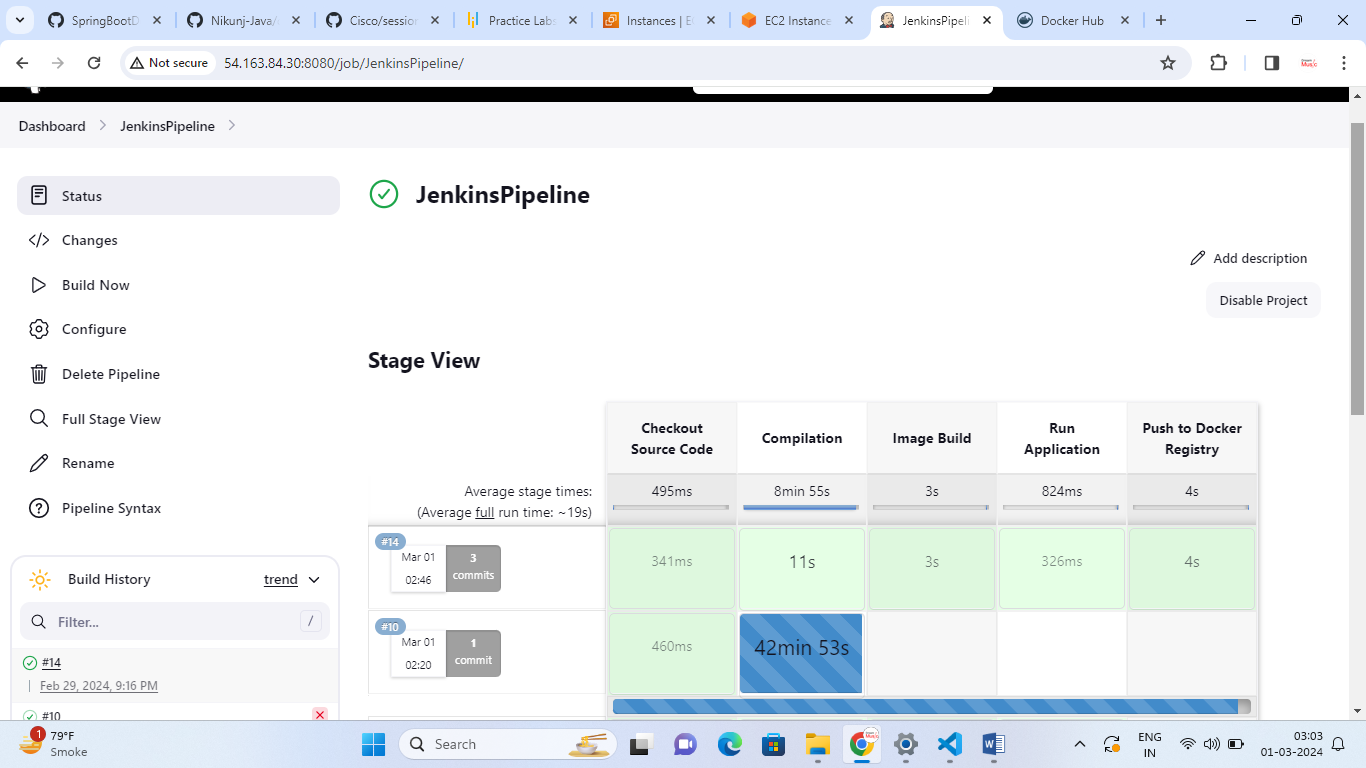
            echo "Image push complete"

        }

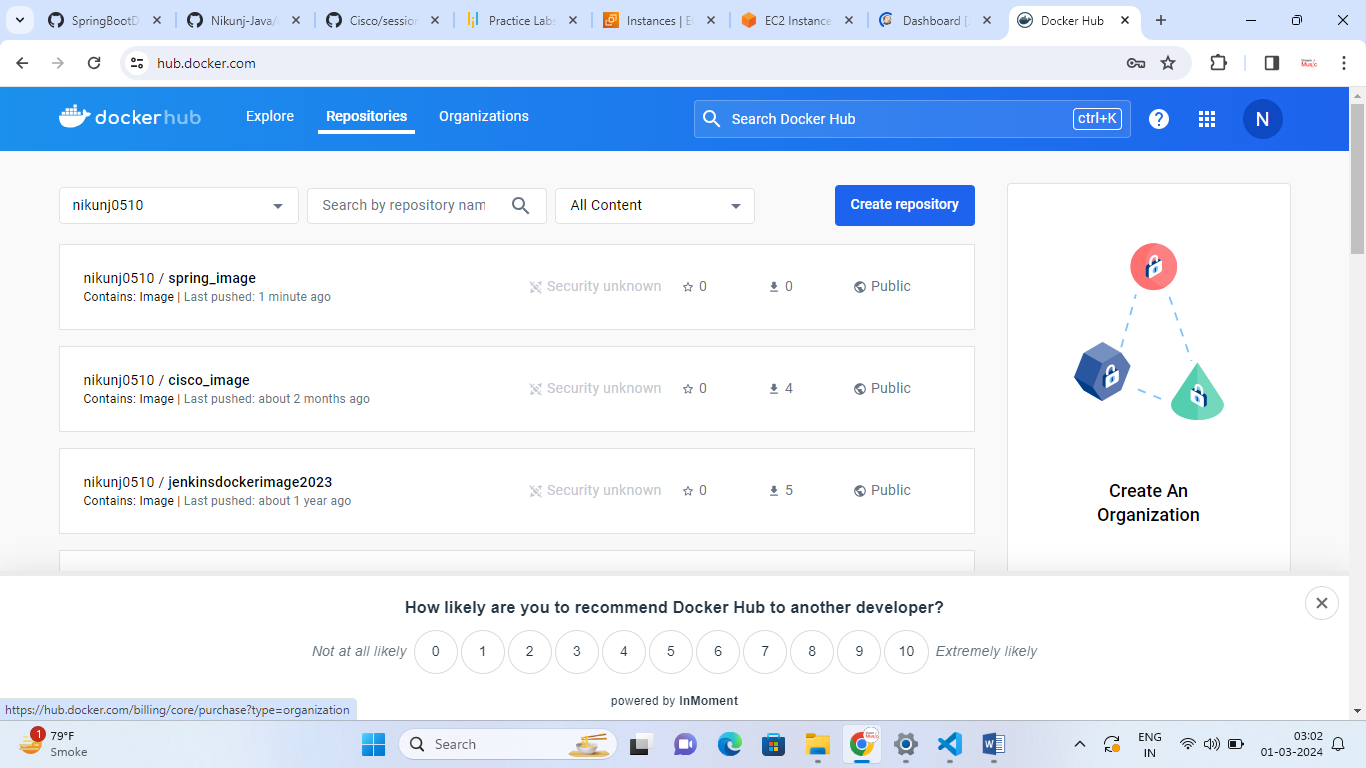
    }

}

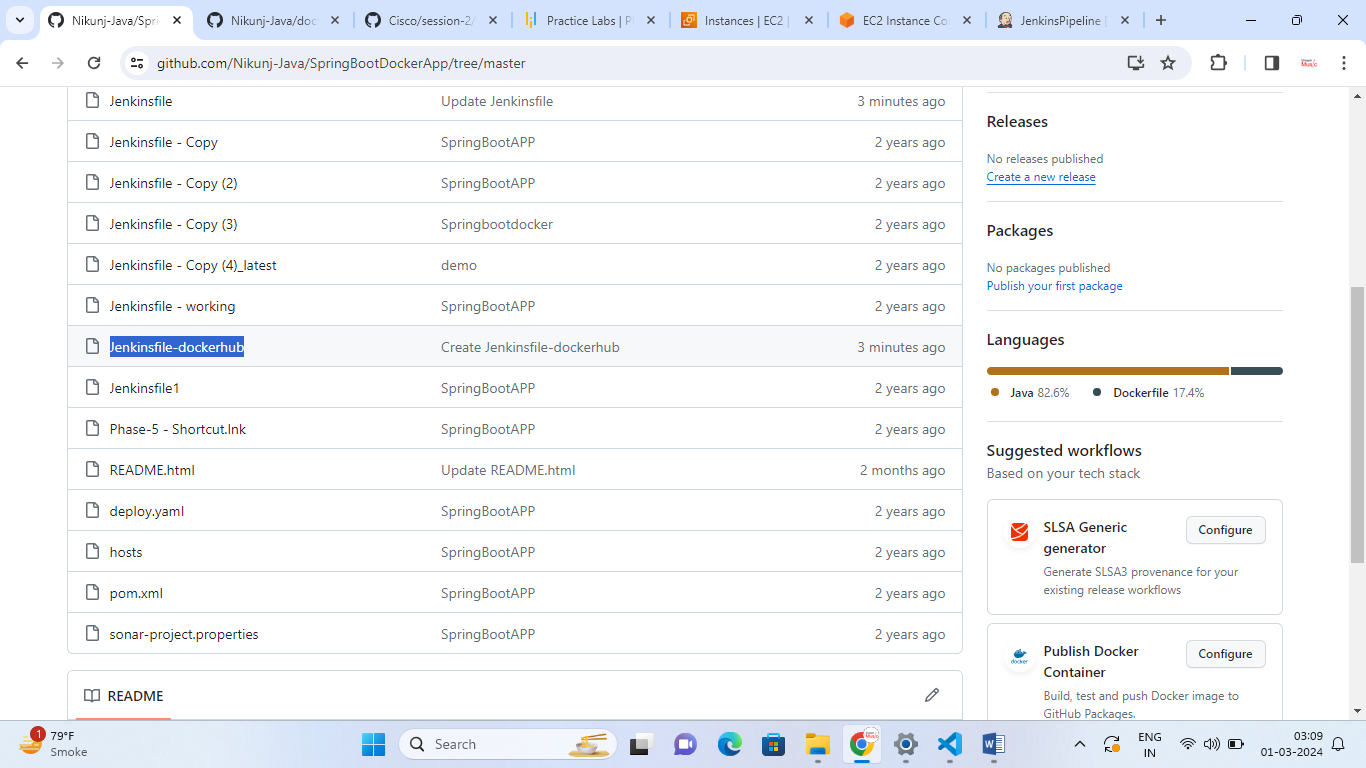
Build The Project Again



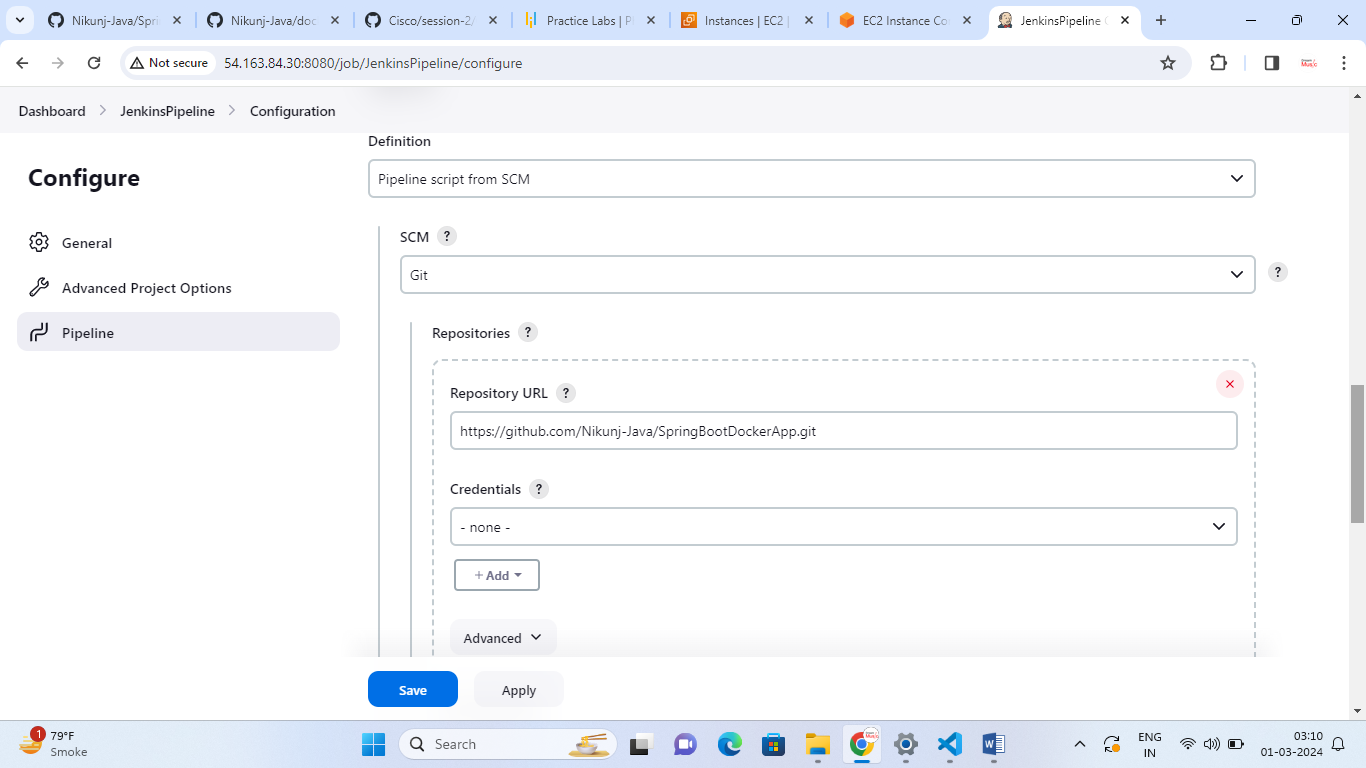
Check your Docker Hub Account Dashboard



If u do not wish to make changes on your Jenkins file you can use My SpringbootdockerApps: Jenkins File



Goto>yourpipeline>configure



I have now given different Jenkins file as

* Jenkinsfile-dockerhub

