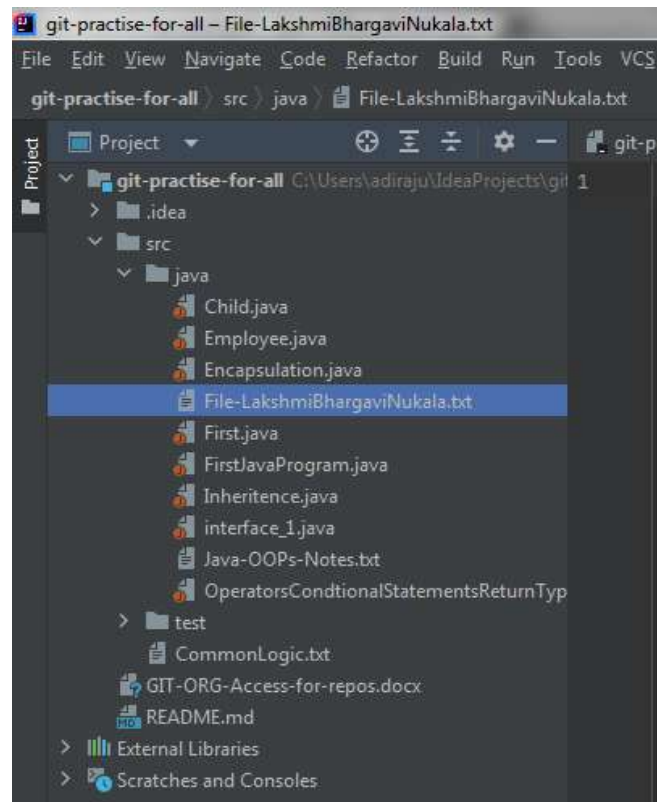


Name: Lakshmi Bhargavi Nukala
Email: bhargavi.aqua04@gmail.com
Reg No: 222037
Hall Ticket No: 22063CDO117

GIT:

Task-1

Add a new file with your name in IntelliJ



Add and Commit it to master branch

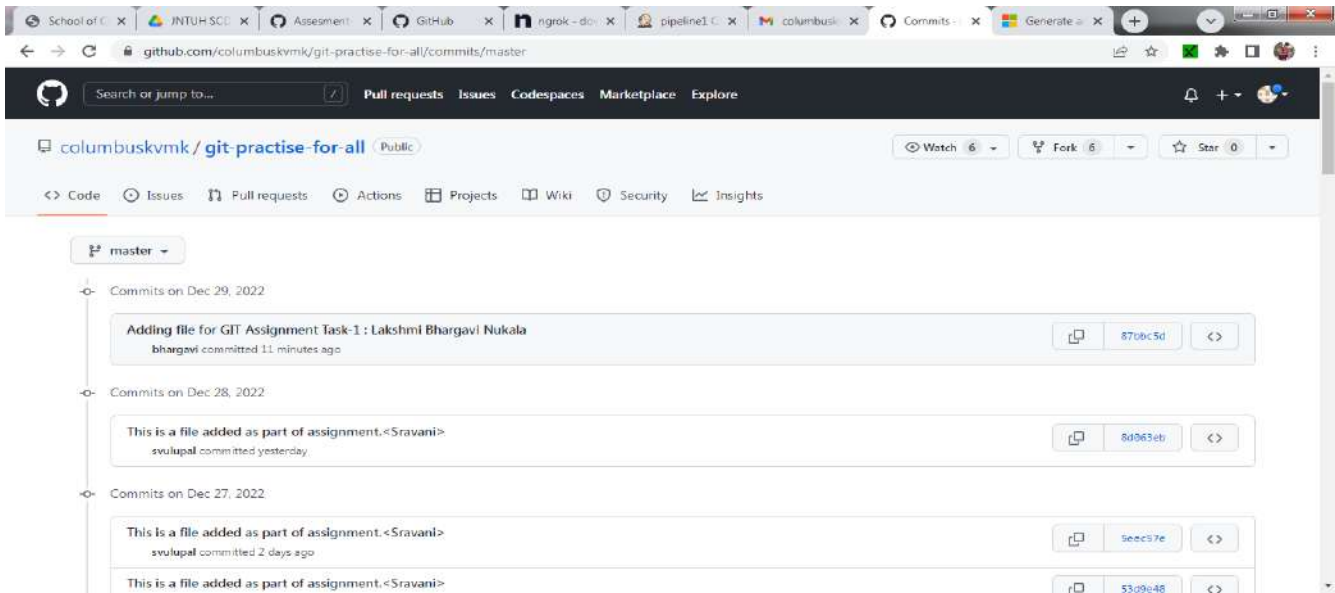
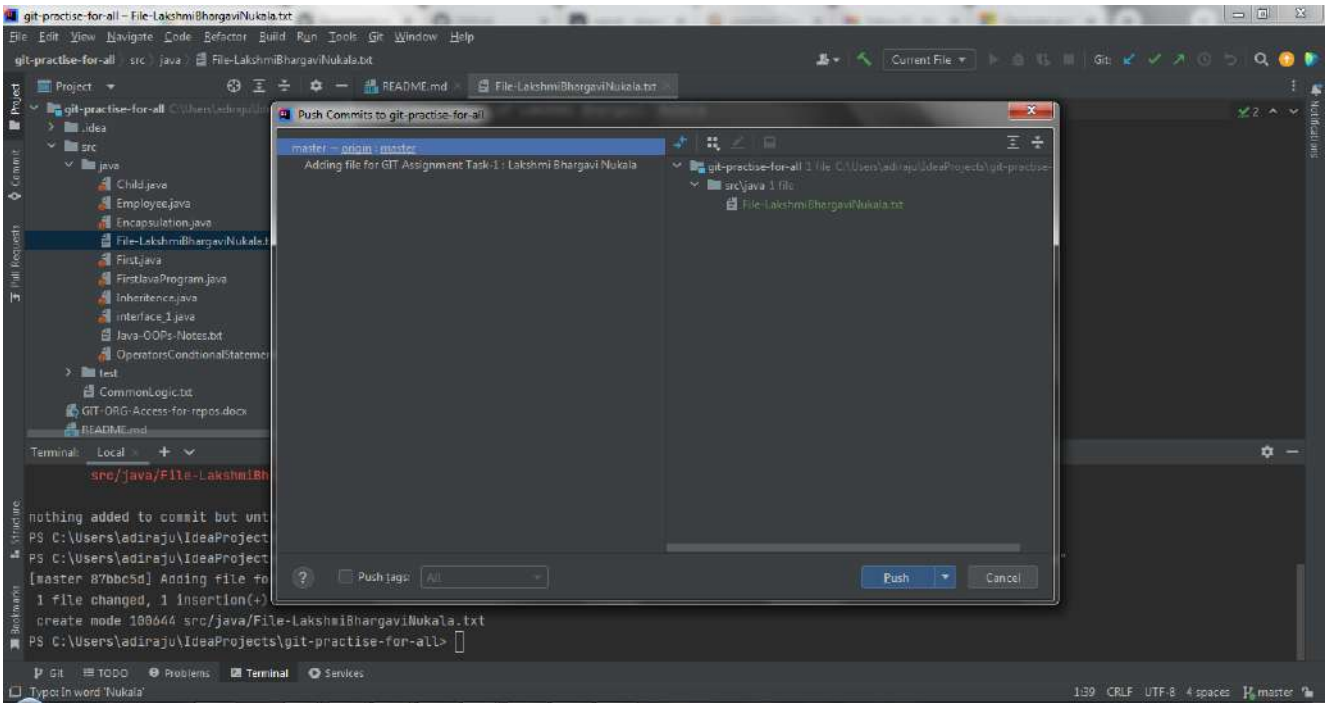
```
Terminal: Local x + v
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/java/File-LakshmiBhargaviNukala.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\adira\IdeaProjects\git-practise-for-all> git add --all
PS C:\Users\adira\IdeaProjects\git-practise-for-all>
```

```
Terminal: Local +
src/java/File-LakshmiBhargaviNukala.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\adiraaju\IdeaProjects\git-practise-for-all> git add --all
PS C:\Users\adiraaju\IdeaProjects\git-practise-for-all> git commit -m "Adding file for GIT Assignment Task-1 : Lakshmi Bhargavi Nukala"
[master 870bc5d] Adding file for GIT Assignment Task-1 : Lakshmi Bhargavi Nukala
1 file changed, 1 insertion(+)
create mode 100644 src/java/File-LakshmiBhargaviNukala.txt
PS C:\Users\adiraaju\IdeaProjects\git-practise-for-all>
```



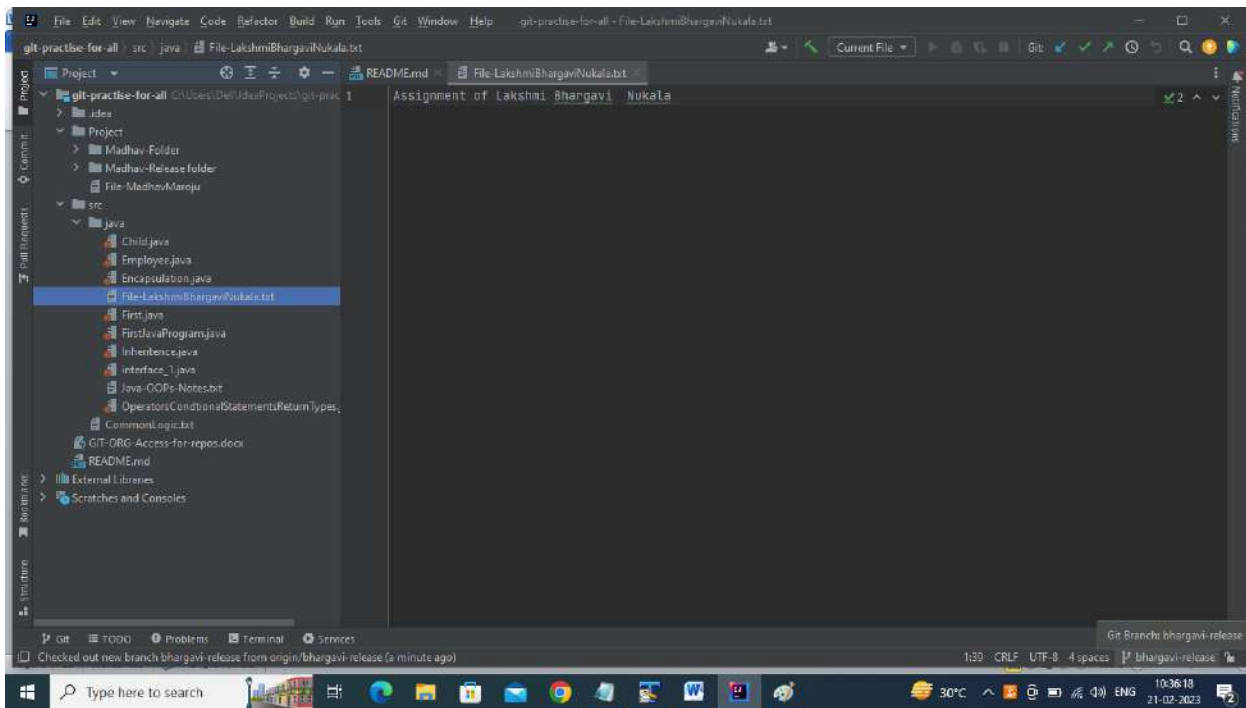
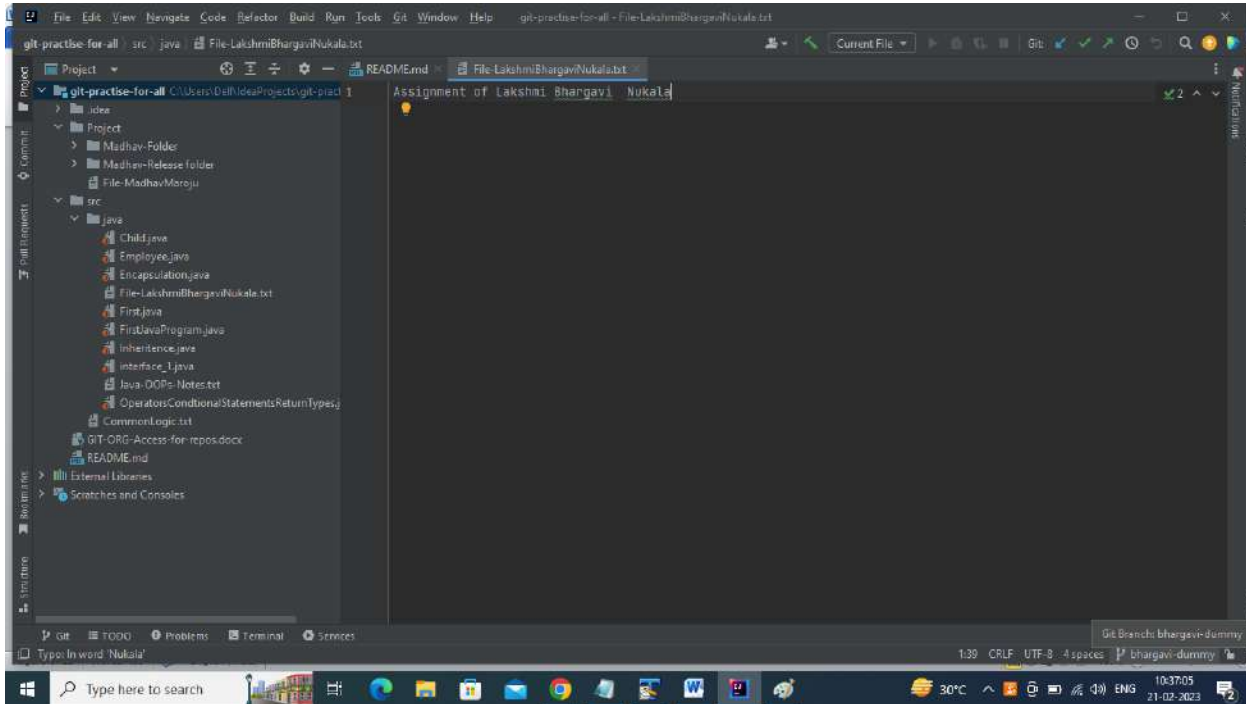
GIT Task-2:

Created two branches in GIT

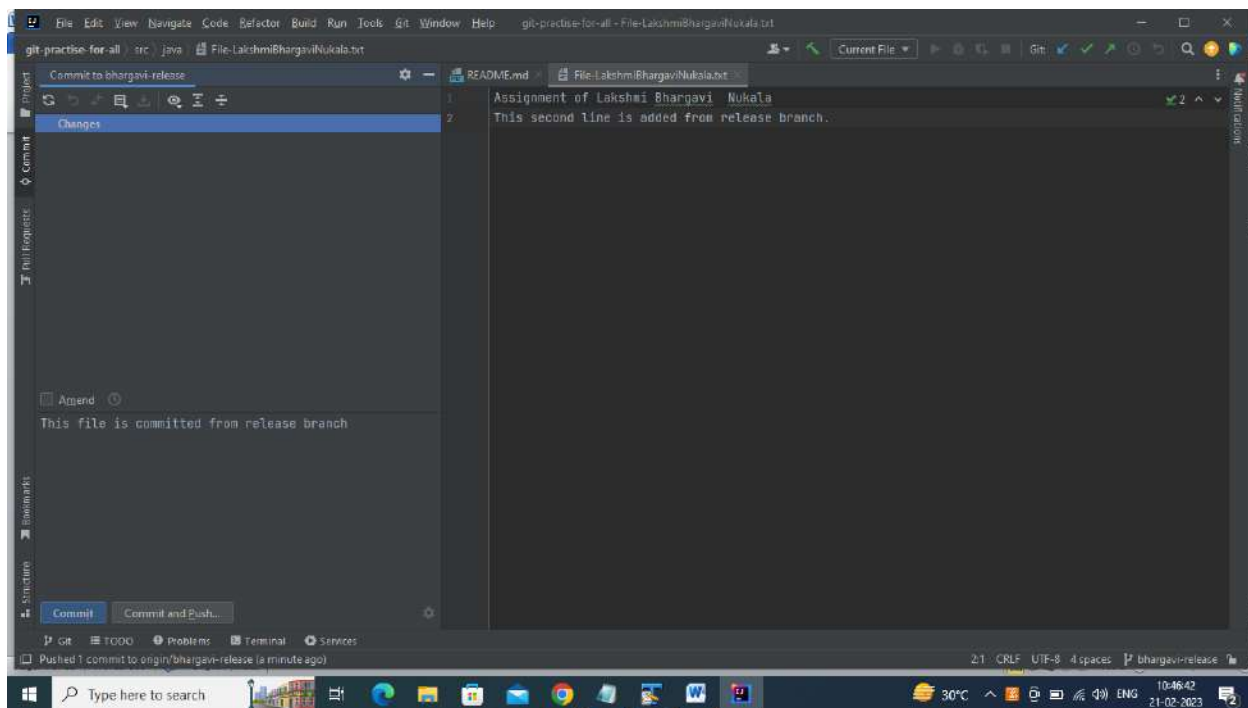
bhargavi-dummy

bhargavi-release

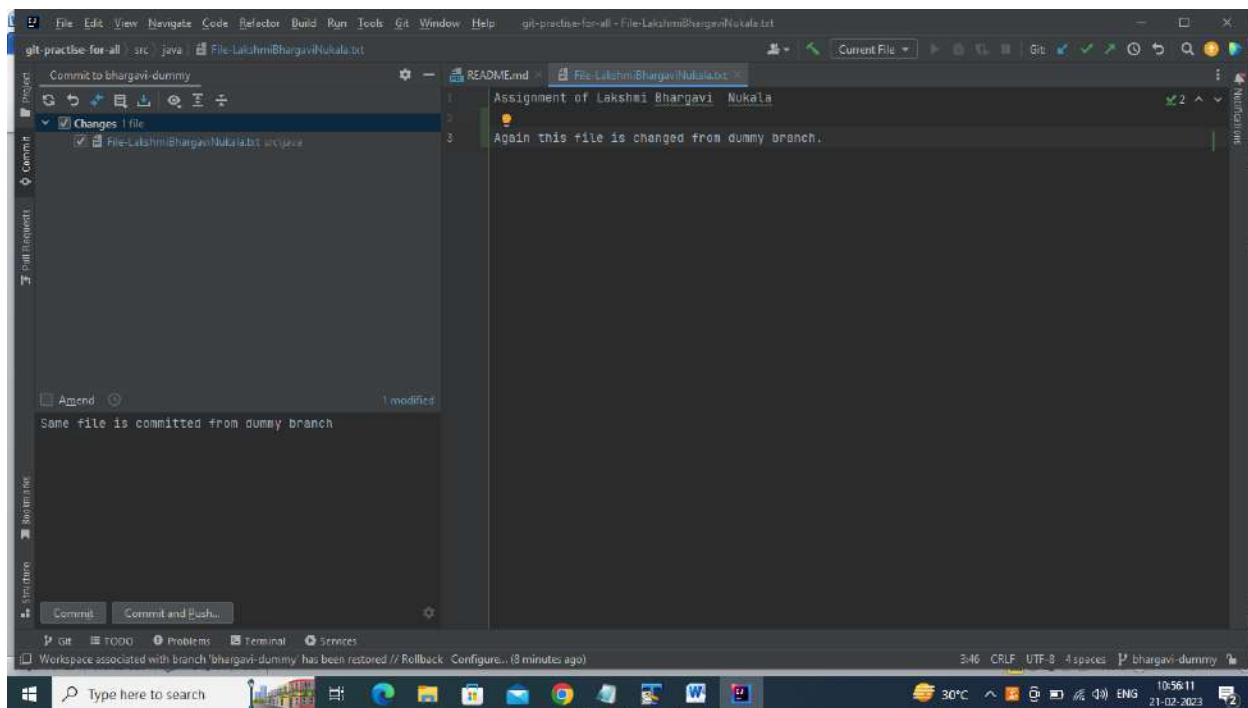
In IntelliJ, check out the code from the branches using GIT Fetch and GIT Pull options.



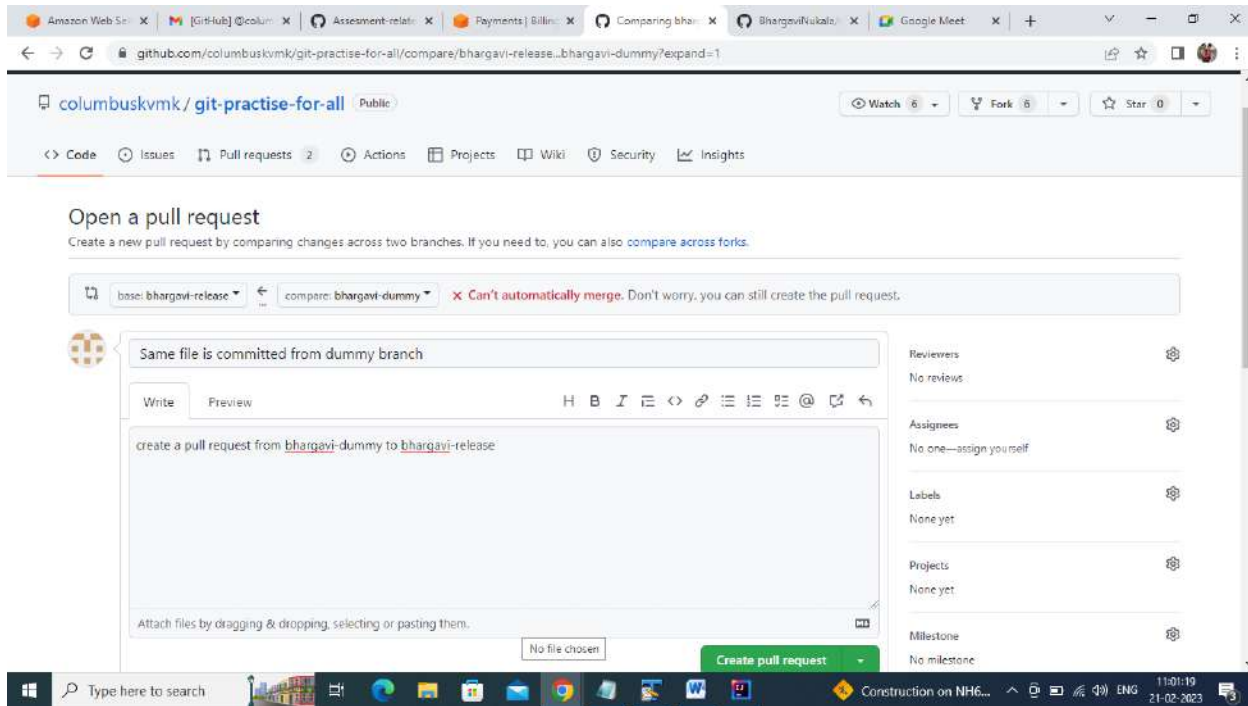
Change the file File-LakshmiBhargaviNukala.txt in bhargavi-release branch, commit and push.



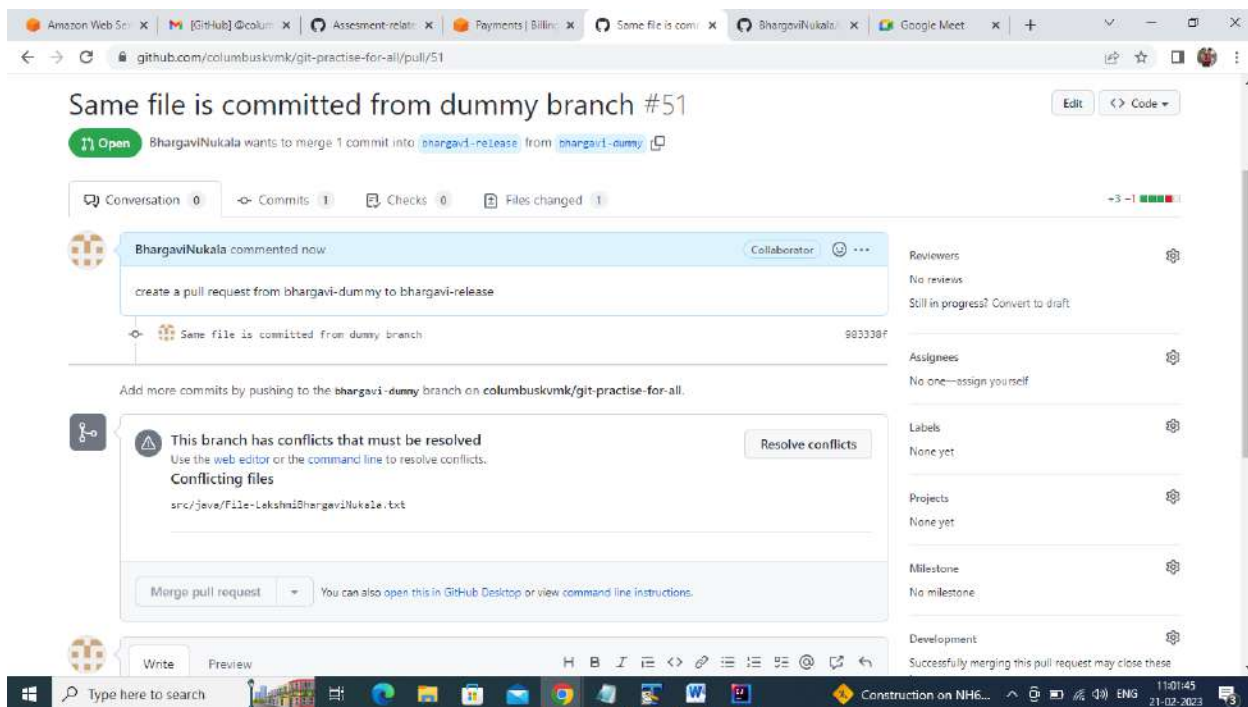
Now switch the branch to bhargavi-dummy and without pulling the code, change the same file and commit and push.

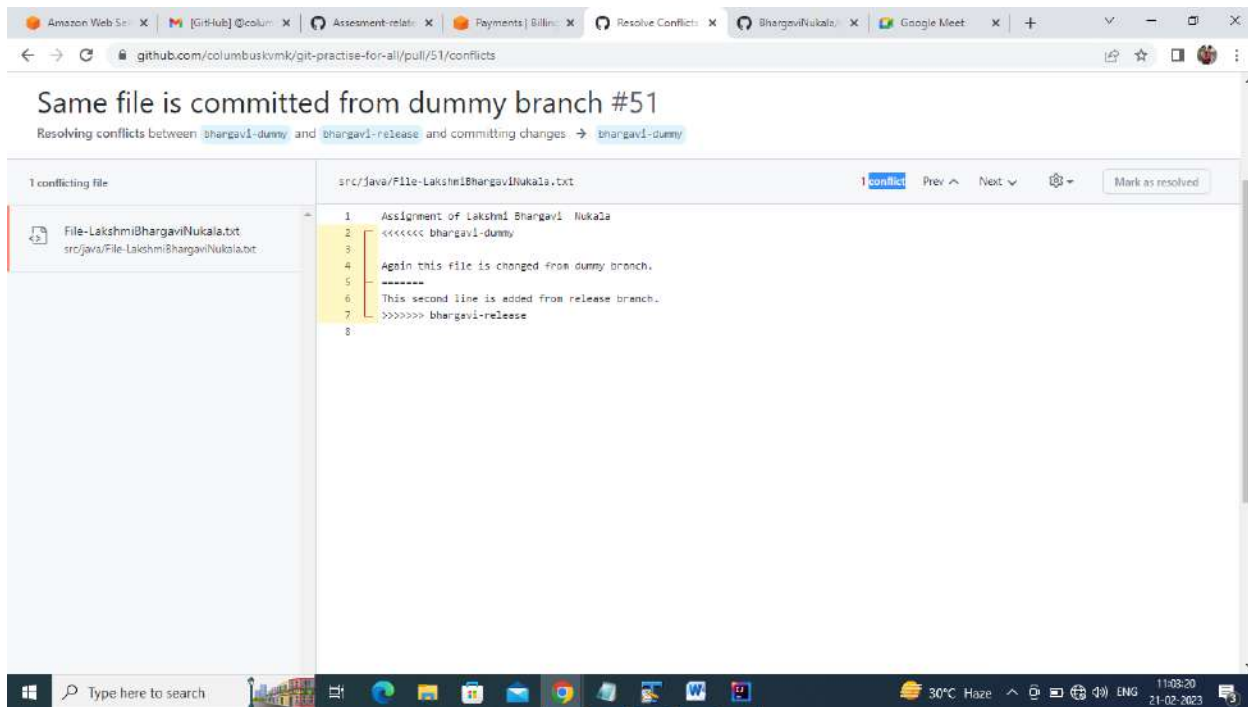


Create a Pull Request (PR) from bhargavi-dummy branch to bhargavi-release branch.

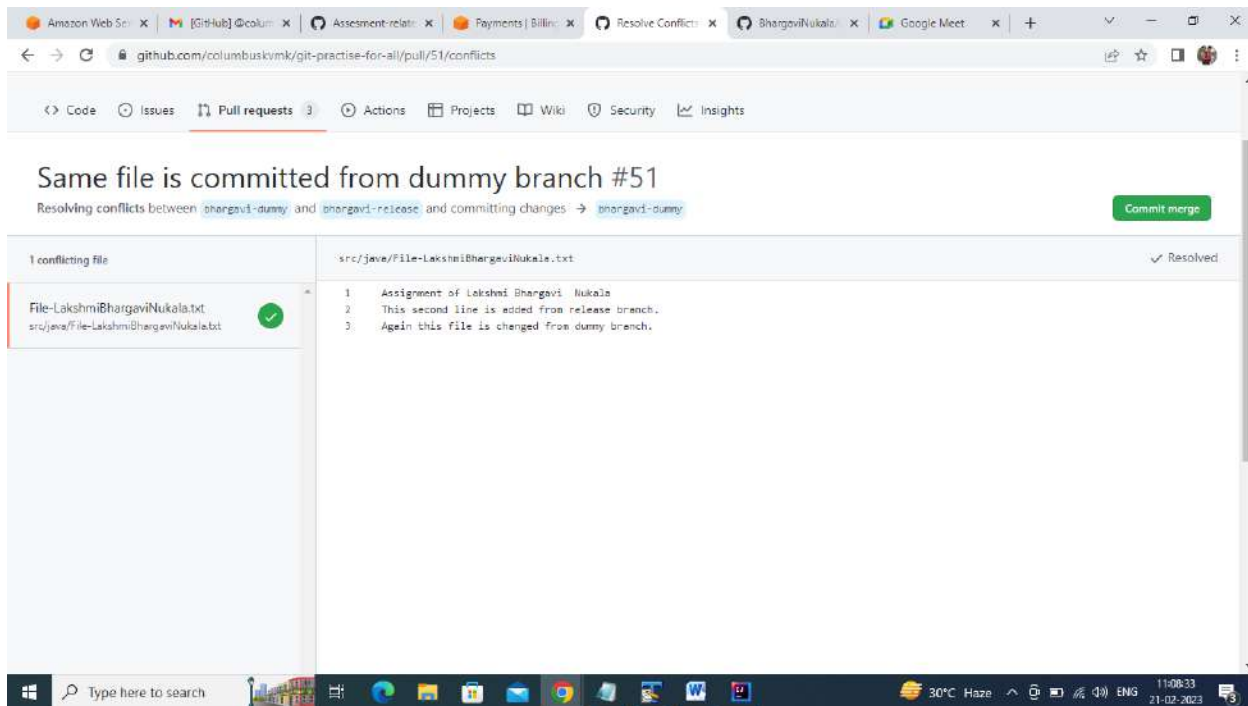


A merge request to resolve conflicts is displayed.

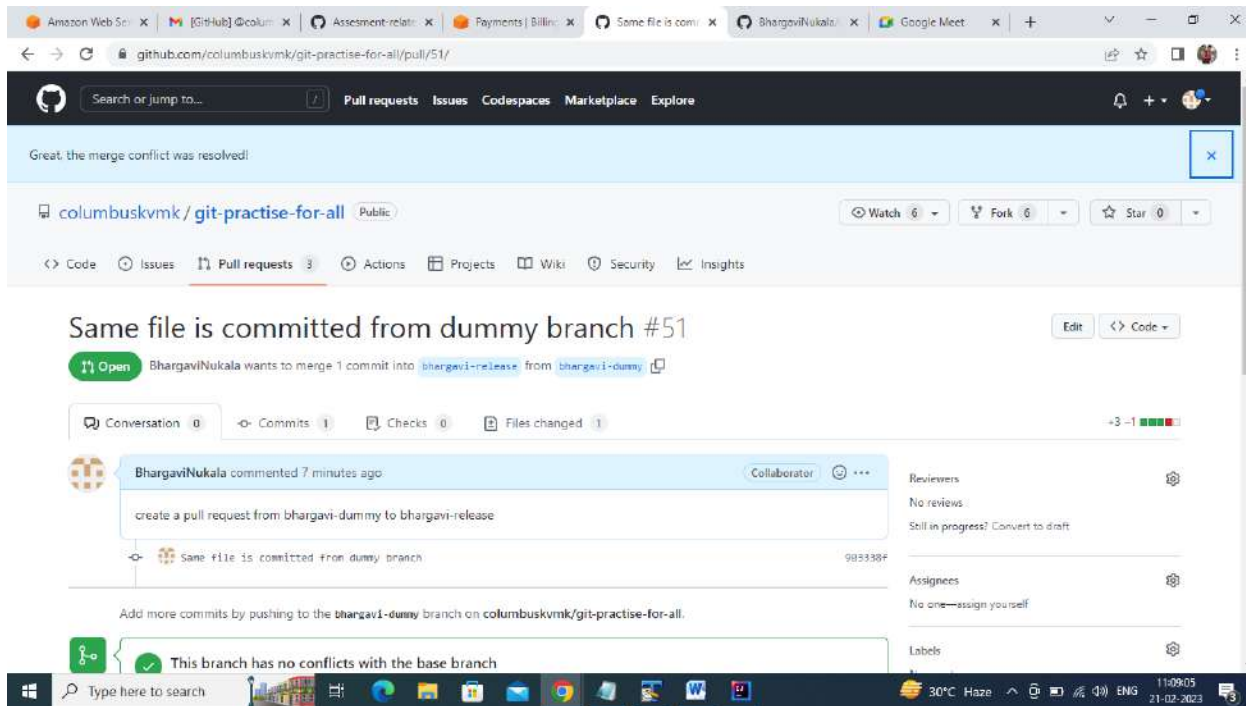




Resolve conflicts and commit merge

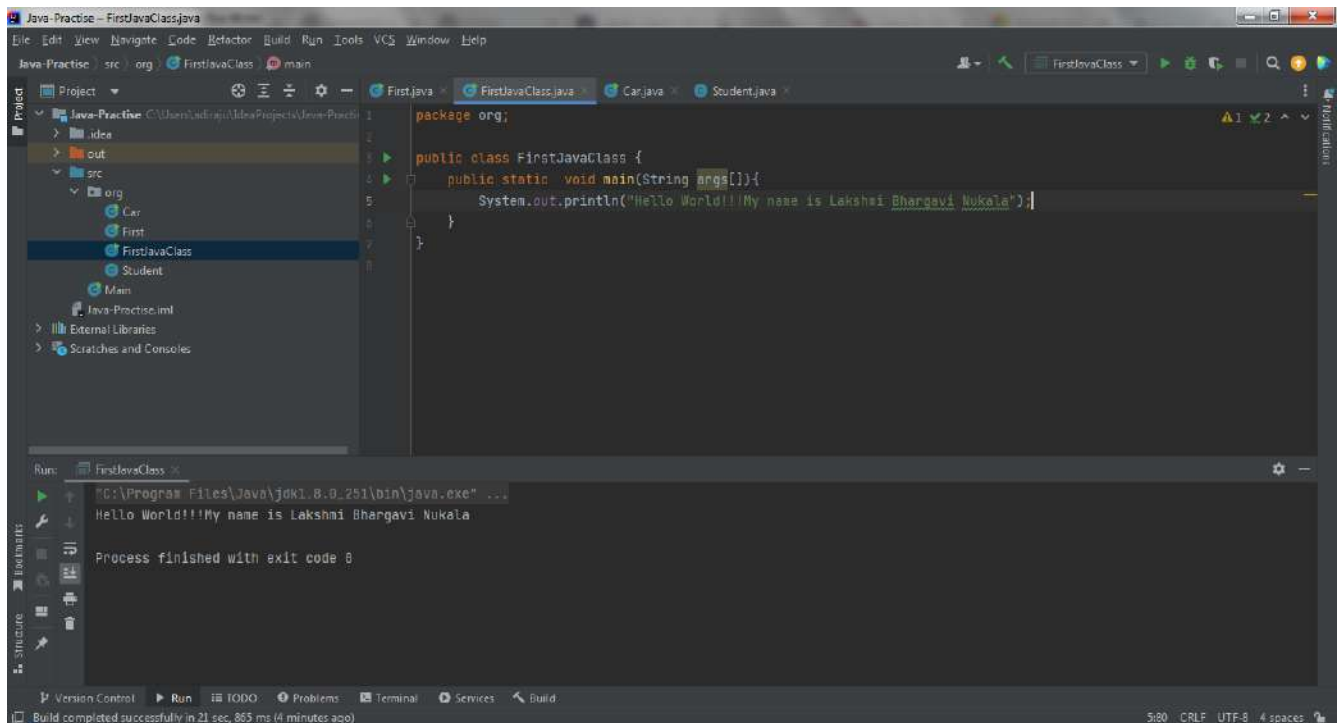


Merge Conflicts are resolved.



Java Task 1:

Run a simple Java application that prints our name



Task-2:

Byte code is in the .class file created and captured it

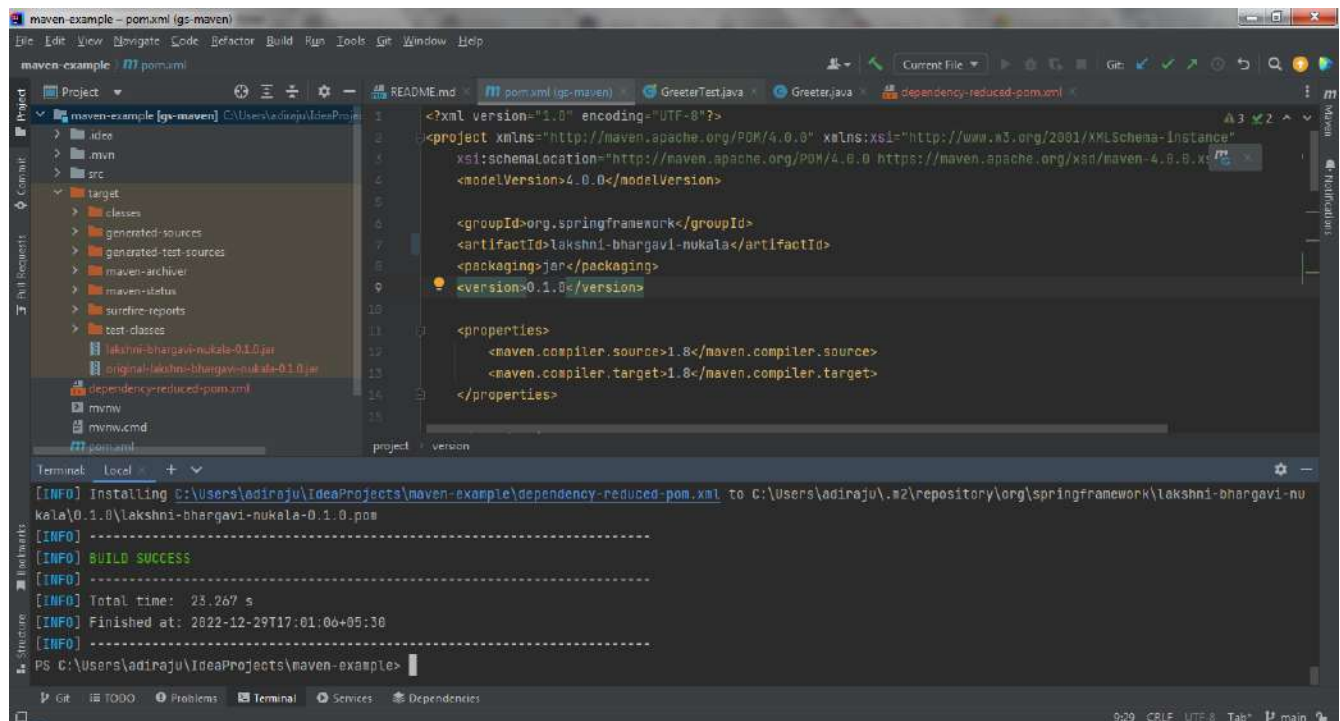
```
//  
// Source code recreated from a .class file by IntelliJ IDEA  
// (powered by FernFlower decompiler)  
//  
package org;  
public class FirstJavaClass {  
    public FirstJavaClass() {  
    }  
    public static void main(String[] args) {  
        System.out.println("Hello World!!!My name is Lakshmi Bhargavi Nukala");  
    }  
}
```

Maven:

Create a maven project with artifact id as your name and generate jar/war file

In pom.xml file, change the artifact-id to your name and run the command
mvn clean install

```
<groupId>org.springframework</groupId>  
<artifactId>lakshni-bhargavi-nukala</artifactId>  
<packaging>jar</packaging>  
<version>0.1.0</version>
```

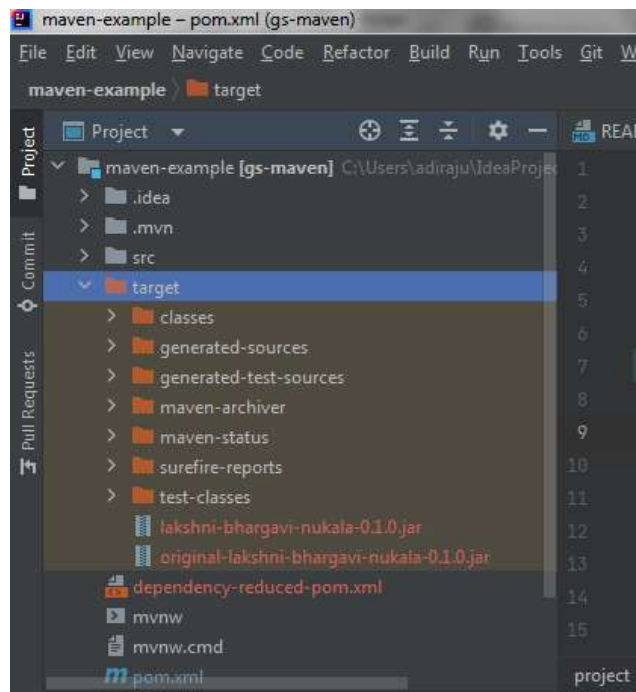


The screenshot shows an IDE window titled 'maven-example - pom.xml (gs-maven)'. The main editor displays the contents of the pom.xml file, which includes the following XML structure:

```
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-Instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
    <groupId>org.springframework</groupId>  
    <artifactId>lakshni-bhargavi-nukala</artifactId>  
    <packaging>jar</packaging>  
    <version>0.1.0</version>  
    <properties>  
        <maven.compiler.source>1.8</maven.compiler.source>  
        <maven.compiler.target>1.8</maven.compiler.target>  
    </properties>  
</project>
```

The left sidebar shows the project structure, including folders like 'src' and 'target', and files like 'pom.xml' and 'mvnw.cmd'. The bottom terminal window shows the output of the 'mvn clean install' command, indicating a successful build:

```
[INFO] Installing C:\Users\adiraaju\IdeaProjects\maven-example\dependency-reduced-pom.xml to C:\Users\adiraaju\.m2\repository\org\springframework\lakshni-bhargavi-nukala\0.1.0\lakshni-bhargavi-nukala-0.1.0.pom  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 23.267 s  
[INFO] Finished at: 2022-12-29T17:01:06+05:30  
[INFO] -----  
PS C:\Users\adiraaju\IdeaProjects\maven-example>
```

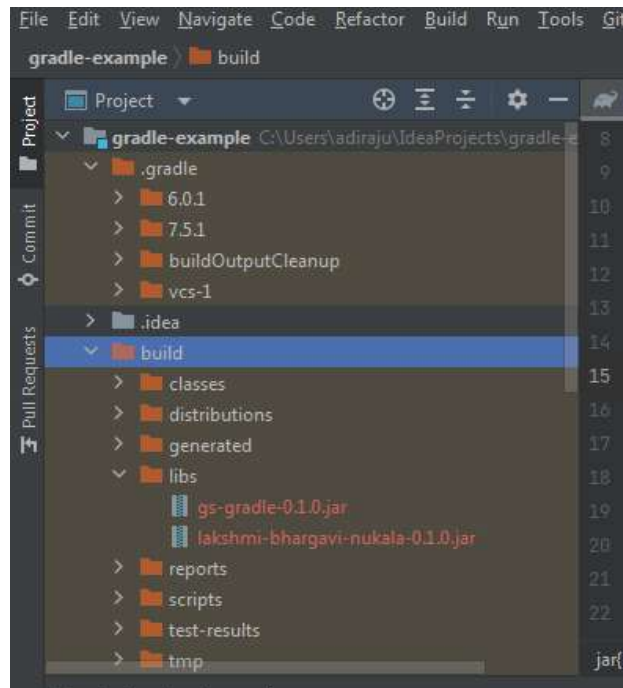
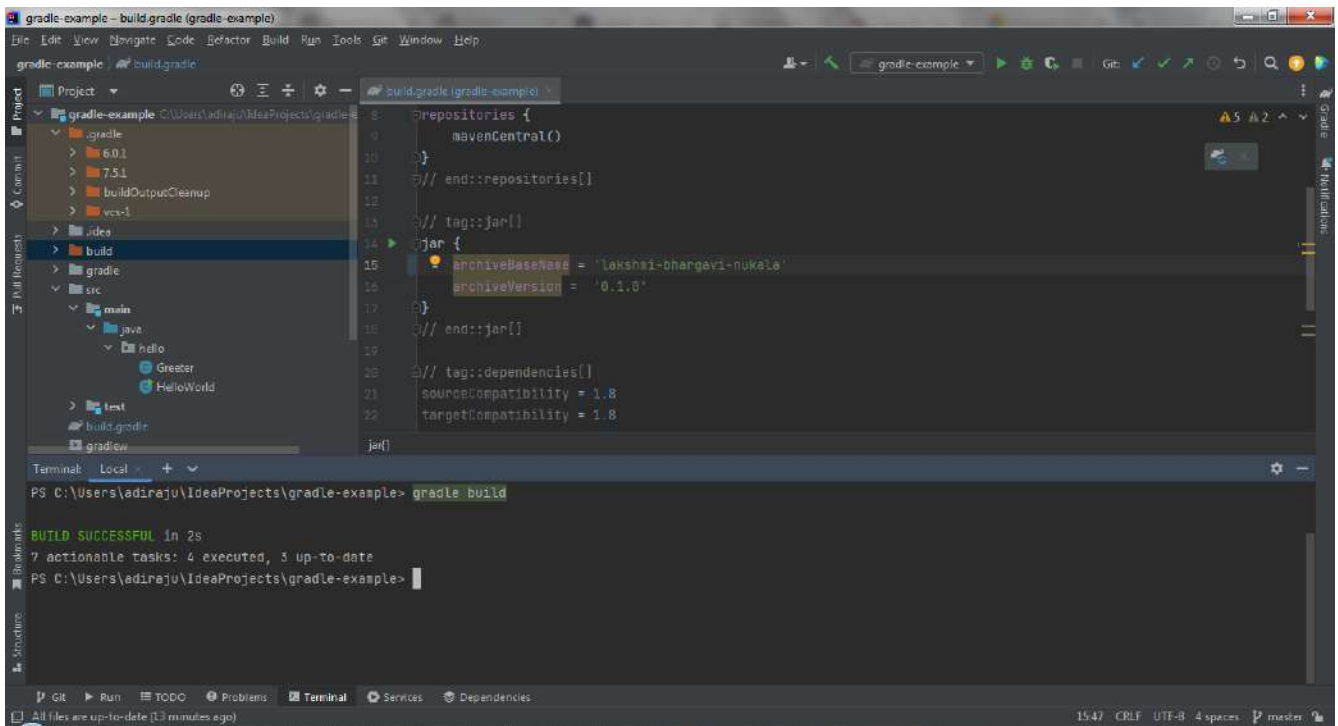



Gradle:

Create a gradle project with artifact-id as your name and generate jar/war artifact.
In build.gradle file, change the artifact-id name

```
// tag::jar[]  
jar {  
    archiveBaseName = 'lakshmi-bhargavi-nukala'  
    archiveVersion = '0.1.0'  
}  
// end::jar[]
```

Build gradle project using the command
gradle build



SpringBoot microservices: Task 1: Run the SpringBoot in the repo provided

```
package com.test.demoartifact;

import ...

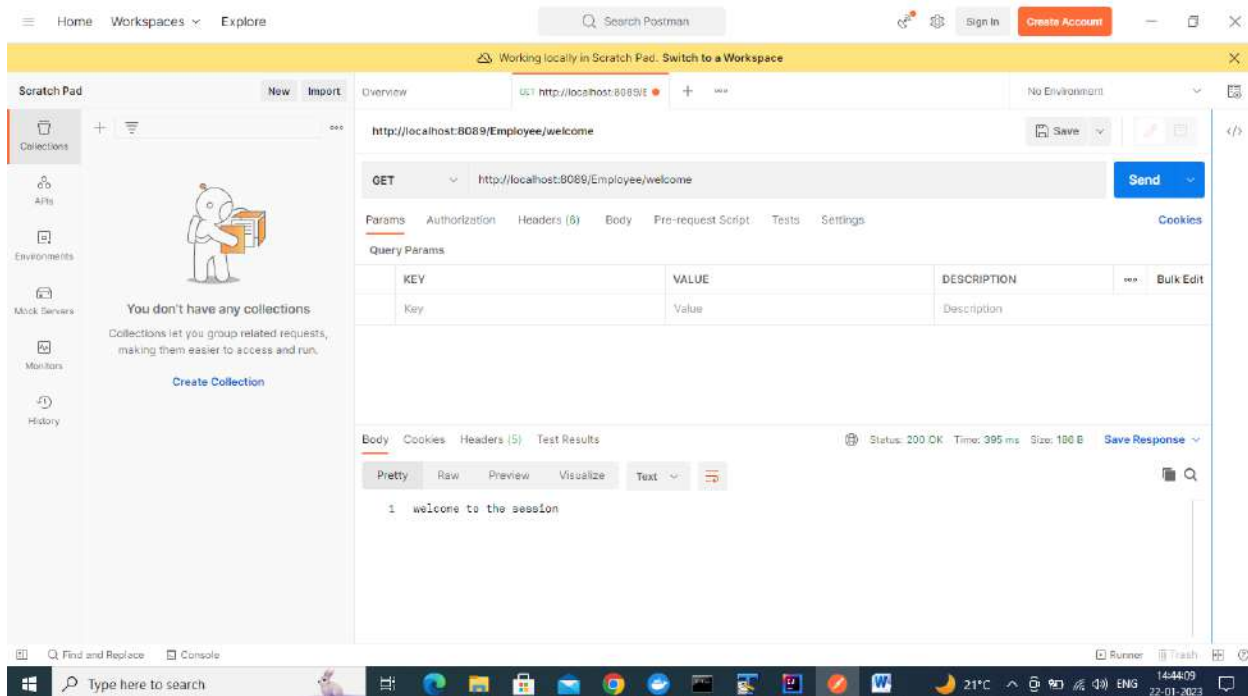
@SpringBootApplication
public class DemoArtifactApplication {

    ...
}
```

```
2023-01-22 14:16:13.663 INFO 13932 --- [main] o.t.d.DemoArtifactApplication : Str
2023-01-22 14:16:13.662 INFO 13932 --- [main] o.t.d.DemoArtifactApplication : TH
2023-01-22 14:16:17.729 INFO 13932 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : To
2023-01-22 14:16:17.754 INFO 13932 --- [main] o.apache.catalina.core.StandardService : St
2023-01-22 14:16:17.755 INFO 13932 --- [main] org.apache.catalina.core.StandardEngine : St
2023-01-22 14:16:18.211 INFO 13932 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : In
2023-01-22 14:16:18.212 INFO 13932 --- [main] w.s.c.ServletWebServerApplicationContext : Ro
2023-01-22 14:16:19.568 INFO 13932 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : To
2023-01-22 14:16:19.609 INFO 13932 --- [main] o.t.d.DemoArtifactApplication : Str
2023-01-22 14:35:32.141 INFO 13932 --- [nio-8089-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : In
2023-01-22 14:35:32.142 INFO 13932 --- [nio-8089-exec-2] o.s.web.servlet.DispatcherServlet : In
2023-01-22 14:35:32.163 INFO 13932 --- [nio-8089-exec-2] o.s.web.servlet.DispatcherServlet : Co
dev-url
```

Task 2: Verify the following Urls on Postman

<http://localhost:8089/Employee/welcome>



<http://localhost:8089/Employee/getAllEmployees>

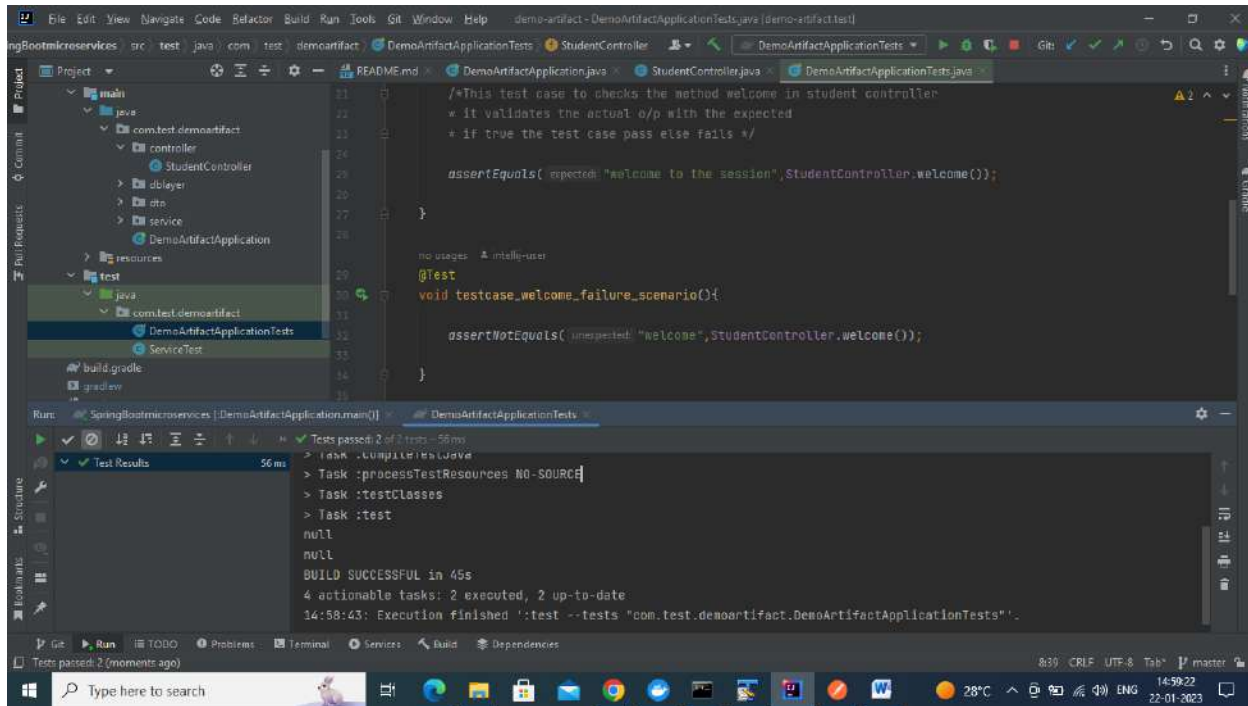
The screenshot shows Postman with a GET request to `http://localhost:8089/Employee/getAllEmployees`. The response status is 200 OK, and the body is a JSON array of 9 employee objects. The response is displayed in the 'Pretty' view.

```
1 [{"Employee":{"employee_name":"emp1", "employee_id":"1", "employee_salary":0}, {"Employee":{"employee_name":"emp2", "employee_id":"13", "employee_salary":0}, {"Employee":{"employee_name":"emp3", "employee_id":"6", "employee_salary":60000}, {"Employee":{"employee_name":"emp4", "employee_id":"6", "employee_salary":60000}, {"Employee":{"employee_name":"emp7", "employee_id":"7", "employee_salary":70000}, {"Employee":{"employee_name":"emp8", "employee_id":"8", "employee_salary":80000}, {"Employee":{"employee_name":"emp9", "employee_id":"9", "employee_salary":90000}]}
```

<http://localhost:8089/Employee/getEmployeeDetails> -- POST method , passed empid=13 in body

The screenshot shows Postman with a POST request to `http://localhost:8089/Employee/getEmployeeDetails`. The request body is `13`. The response status is 200 OK, and the body is `Employee Name : emp2 Employee Id :13 employee salary : 0`. The response is displayed in the 'Text' view.

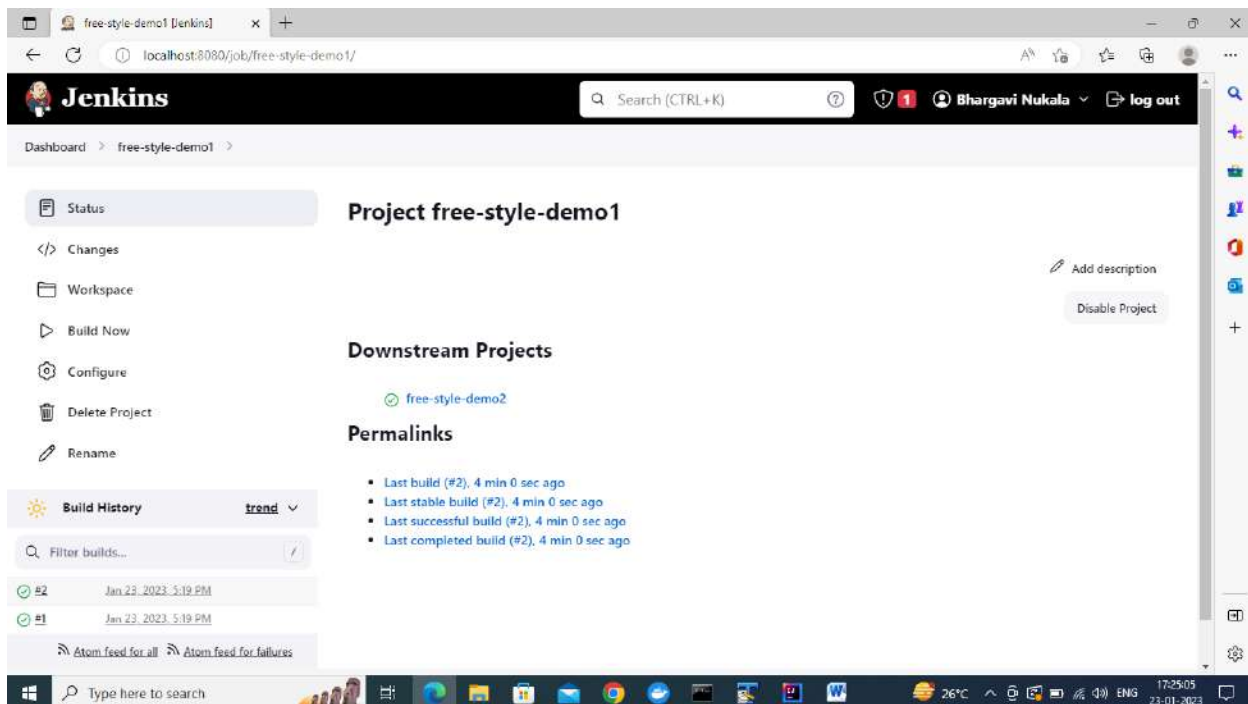
Task 3: Run any test-cases



Jenkins project:

Task 1:

In jenkins create 3 free style projects,
free-style-demo1
free-style-demo2
free-style-demo3



free-style-demo2 [jenkins] x +

localhost:8080/job/free-style-demo2/

Jenkins Search (CTRL+K) Bhargavi Nukala log out

Dashboard > free-style-demo2 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend

Filter builds...

#1 Jan 23, 2023, 5:23 PM

Atom feed for all Atom feed for failures

Project free-style-demo2

Add description

Disable Project

Upstream Projects

- free-style-demo1

Downstream Projects

- free-style-demo3

Permalinks

- Last build (#1), 1 min 44 sec ago
- Last stable build (#1), 1 min 44 sec ago
- Last successful build (#1), 1 min 44 sec ago
- Last completed build (#1), 1 min 44 sec ago

Type here to search 26°C 17:25:19 23-01-2023

free-style-demo3 [jenkins] x +

localhost:8080/job/free-style-demo3/

Jenkins Search (CTRL+K) Bhargavi Nukala log out

Dashboard > free-style-demo3 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend

Filter builds...

#2 Jan 23, 2023, 5:23 PM

#2 Jan 23, 2023, 5:22 PM

#1 Jan 23, 2023, 5:22 PM

Project free-style-demo3

Add description

Disable Project

Upstream Projects

- free-style-demo2

Permalinks

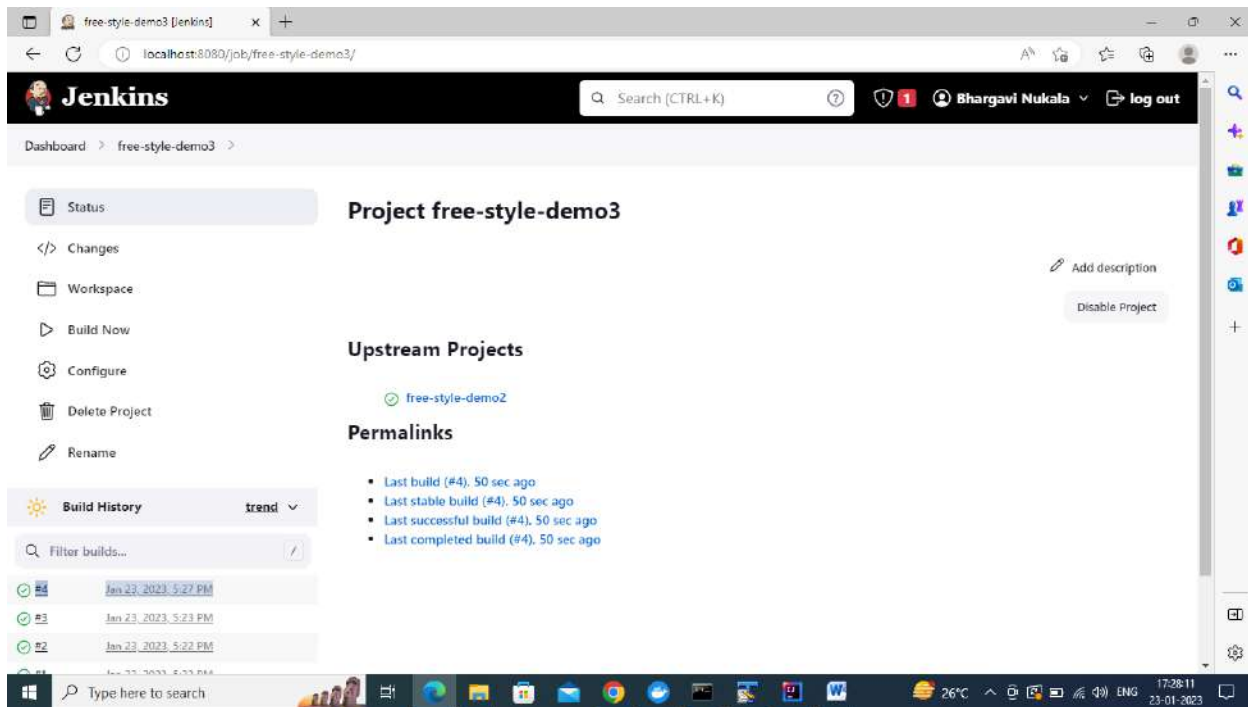
- Last build (#3), 1 min 49 sec ago
- Last stable build (#3), 1 min 49 sec ago
- Last successful build (#3), 1 min 49 sec ago
- Last completed build (#3), 1 min 49 sec ago

Type here to search 26°C 17:25:30 23-01-2023

Once free-style-1 is built then free-style-demo2 and later free-style-demo3 should build.

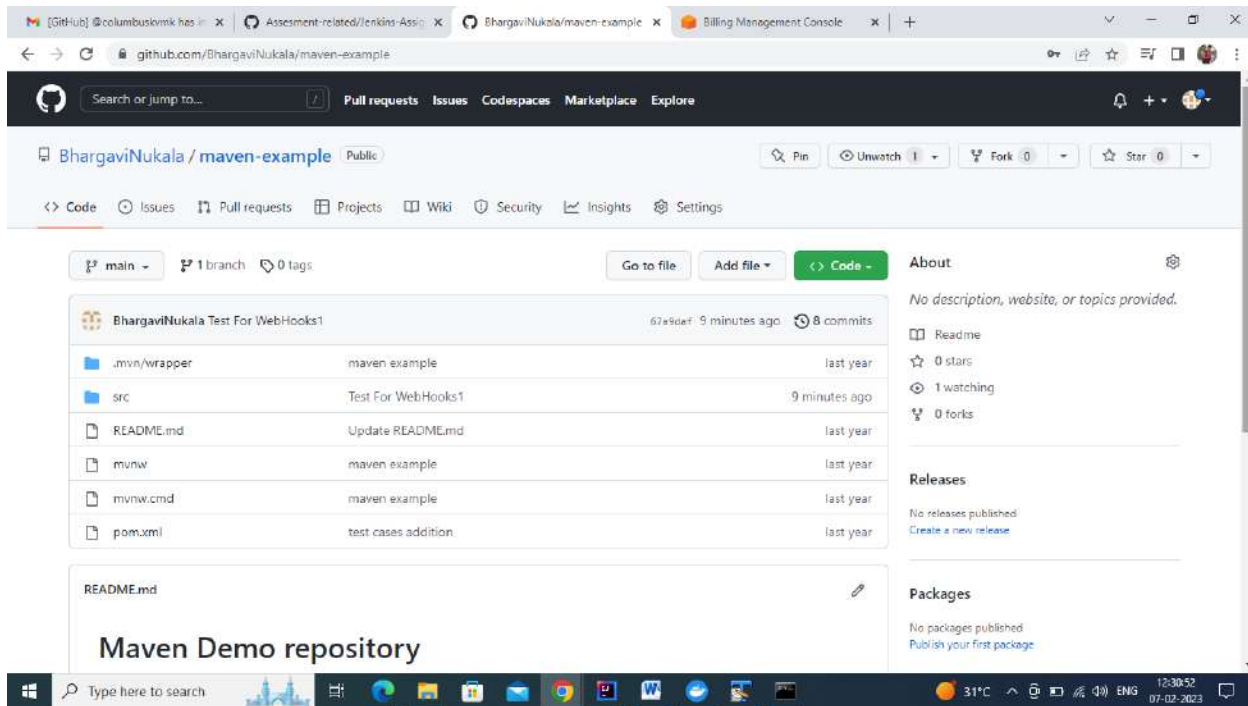
The screenshot shows the Jenkins dashboard for the 'free-style-demo1' project. The interface includes a sidebar with navigation options: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area displays the project name 'Project free-style-demo1' with options to 'Add description' and 'Disable Project'. Below this, there are sections for 'Downstream Projects' (listing 'free-style-demo2') and 'Permalinks' (listing build statistics for build #3, all 24 seconds ago). A 'Build History' section on the left shows three successful builds from Jan 23, 2023, at 5:19 PM.

The screenshot shows the Jenkins dashboard for the 'free-style-demo2' project. The sidebar navigation is identical to the first screenshot. The main content area displays 'Project free-style-demo2' with 'Add description' and 'Disable Project' options. The 'Upstream Projects' section lists 'free-style-demo1'. The 'Downstream Projects' section lists 'free-style-damo3'. The 'Permalinks' section lists build statistics for build #2, all 42 seconds ago. The 'Build History' section on the left shows two successful builds from Jan 23, 2023, at 5:27 PM and 5:23 PM.



Task 2:

Create a new repo in your login and checkin the maven project.



Configure a Maven project from GIT in Jenkins. Configure credentials of Git as username=GIT username and password=GIT token, ID is some random name.

maven-proj Config [Jenkins] x +

localhost:8080/job/maven-proj/configure

Dashboard > maven-proj > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Repositories

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: BhargaviNukala

Treat username as secret

Password: [Redacted]

ID: [Redacted]

Save Apply

Type here to search

17:53:46 23-01-2023

maven-example Config [Jenkins] x + Viewing logs

localhost:8080/job/maven-example/configure

Dashboard > maven-example > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Repositories

Repository URL: https://github.com/BhargaviNukala/maven-example.git

Credentials: BhargaviNukala/*****

+ Add

Advanced

Add Repository

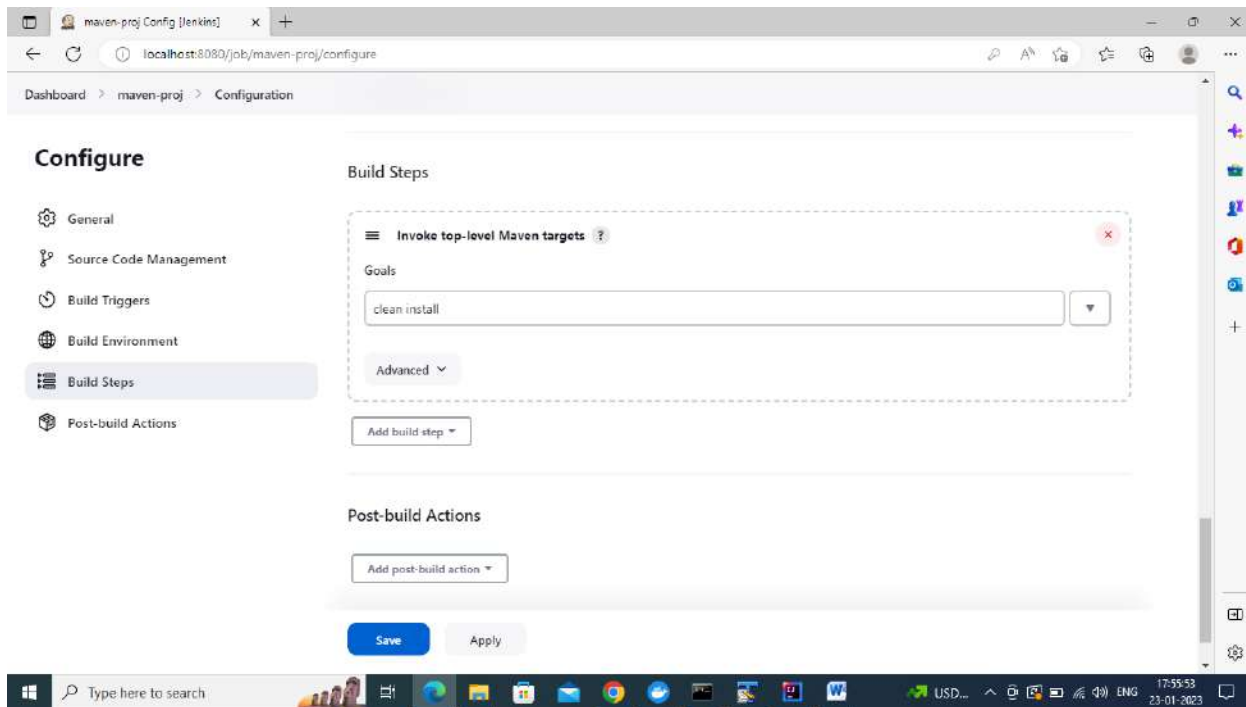
Branches to build

Branch Specifier (blank for 'any')

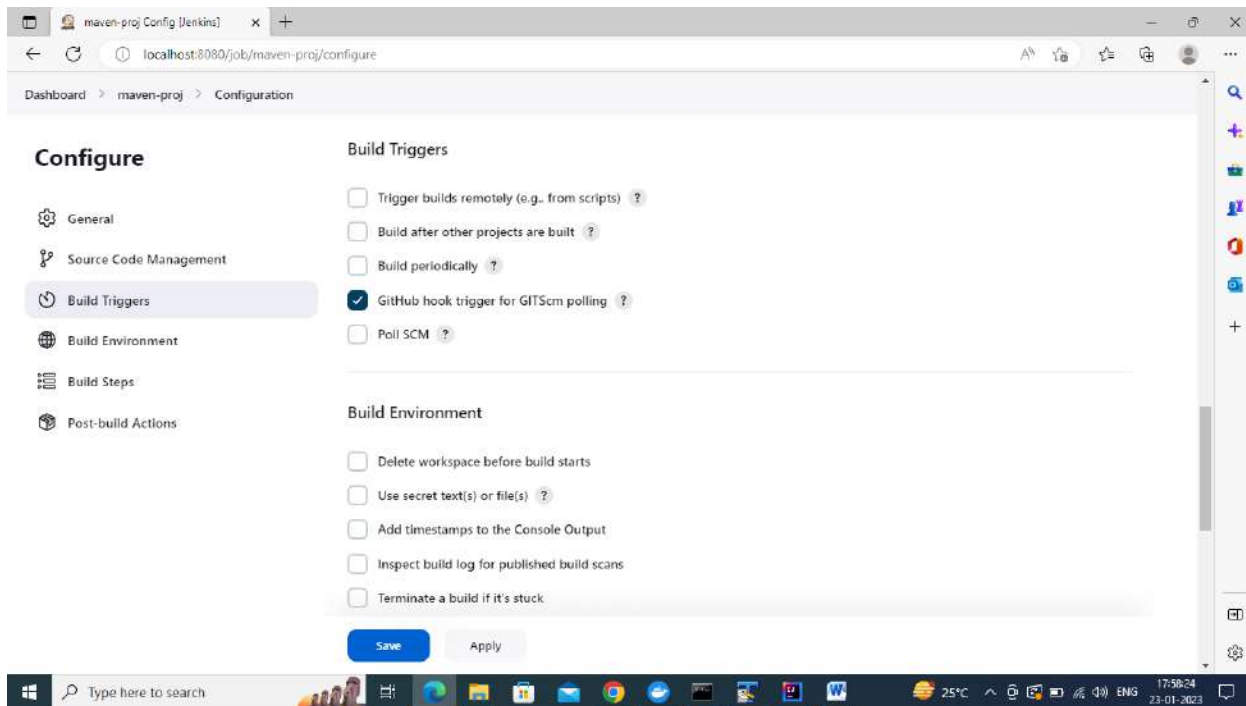
Save Apply

Type here to search

29°C 11:37:00 07-02-2023



Select Github hook trigger option under Build Triggers



Download ngrok from <https://ngrok.com/download>

Extract to a folder.

From the cmd, go to the extracted folder and run the command
ngrok http 8080

Copy the Forwarding entry in Command Prompt

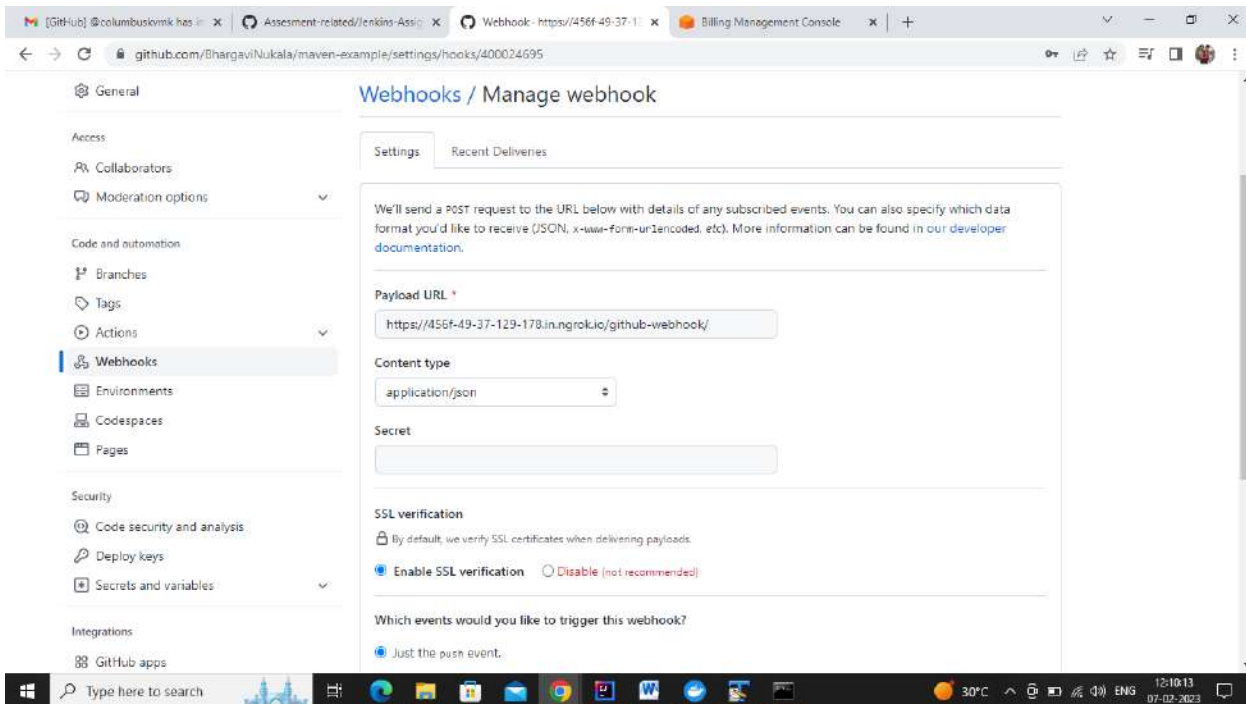
```
Select Command Prompt - ngrok http 8080
ngrok
(CTRL+C to quit)

We added a plan for ngrok hobbyists @ https://ngrok.com/pricing

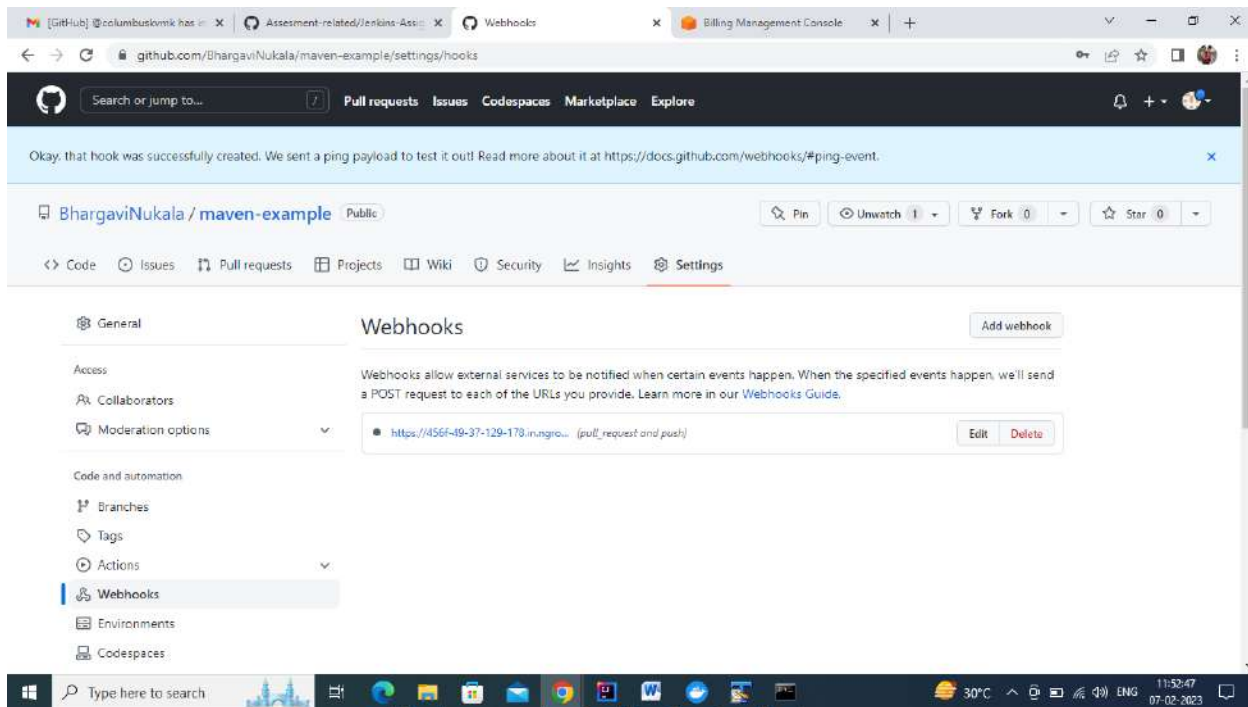
Session Status      online
Session Expires     1 hour, 58 minutes
Terms of Service     https://ngrok.com/tos
Version              3.1.1
Region               India (In)
Latency              25ms
Web Interface        http://127.0.0.1:4041
Forwarding            https://456f-49-37-129-178.in.ngrok.io -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p98
                    0      0      0.00  0.00  0.00  0.00
```

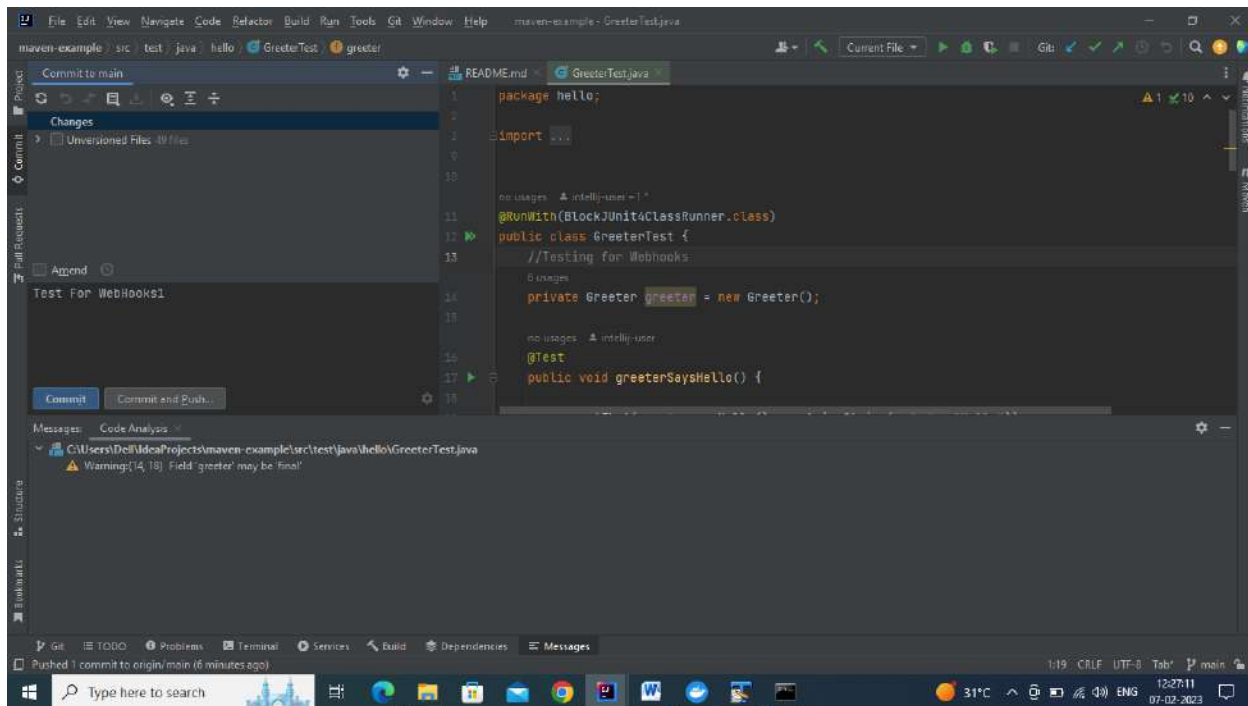
In Git Hub Maven project, under Settings tab ->Click on WebHook and add a new WebHook. Payload URL => URL copied from ngrok/github-webhook



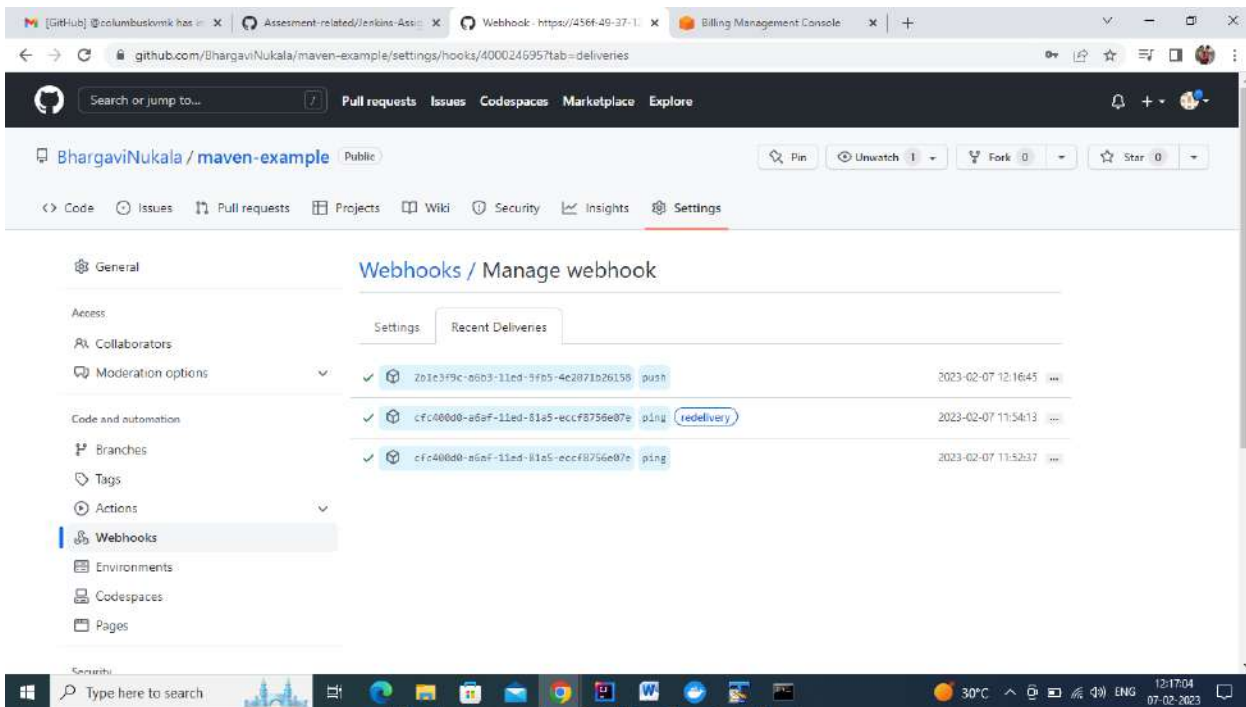
A new WebHook is added



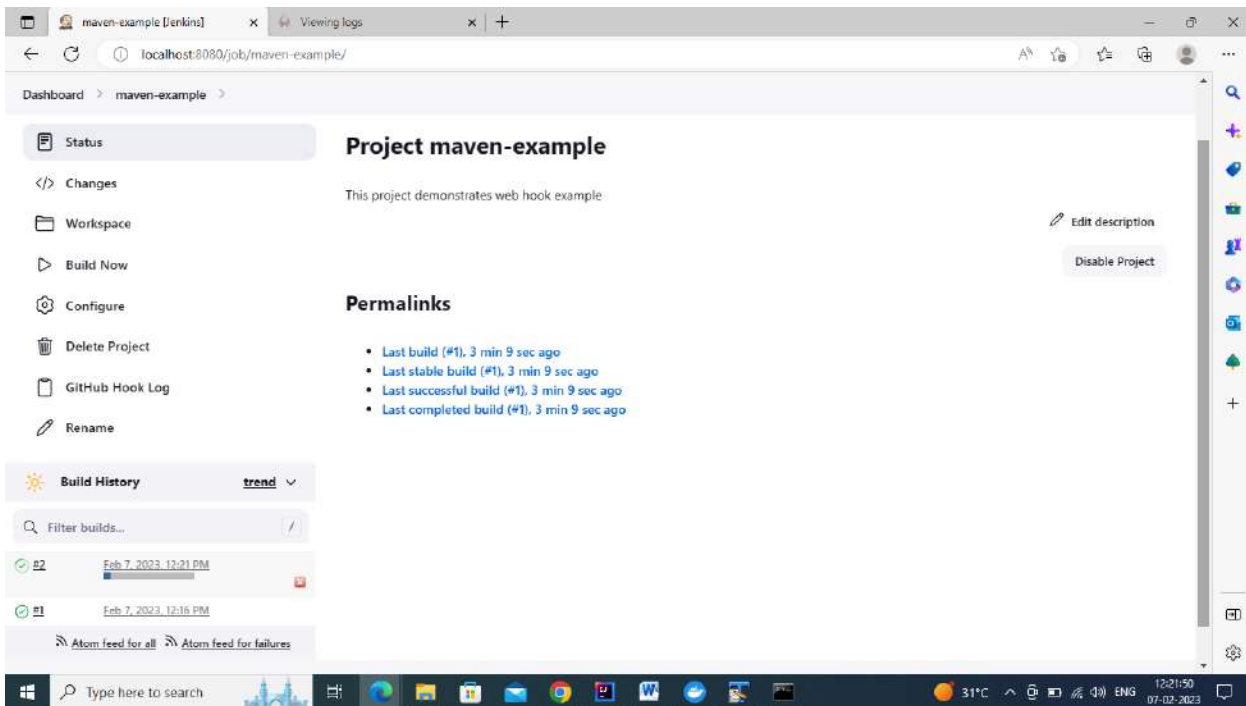
Make a change in the project and commit and push the changes to the repo

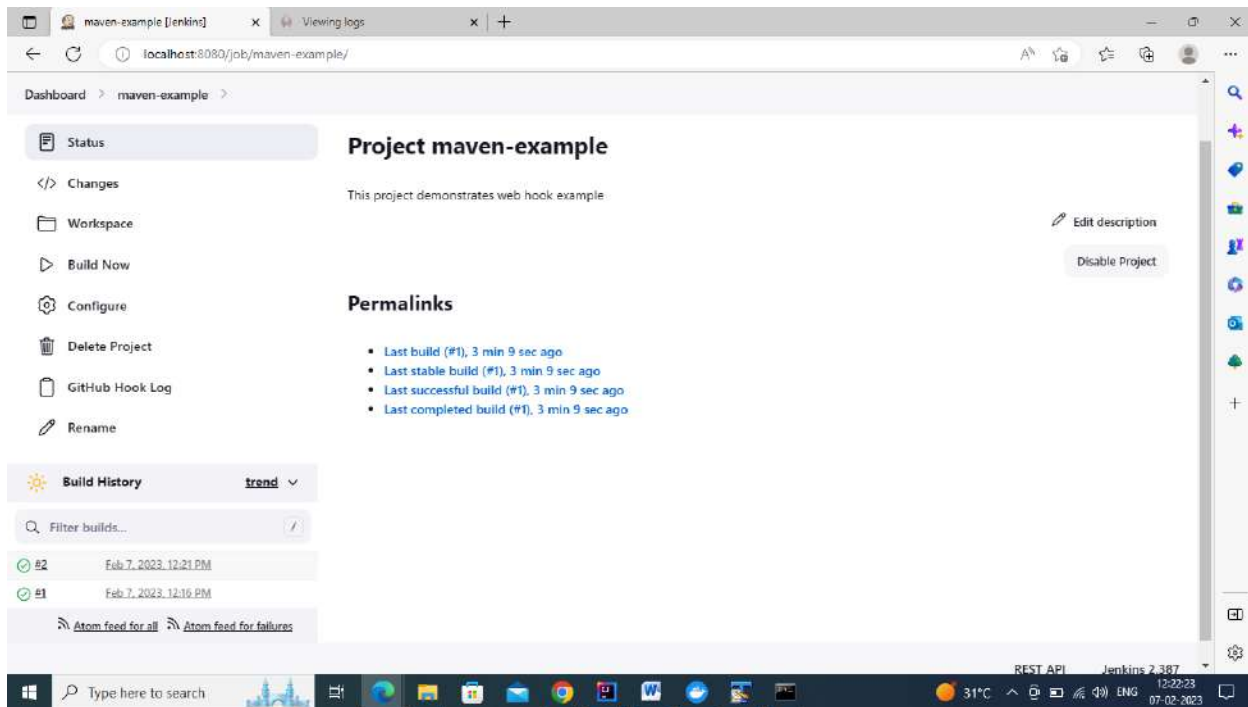


If we check in the GitHub WebHooks, a new entry for the push can be seen.



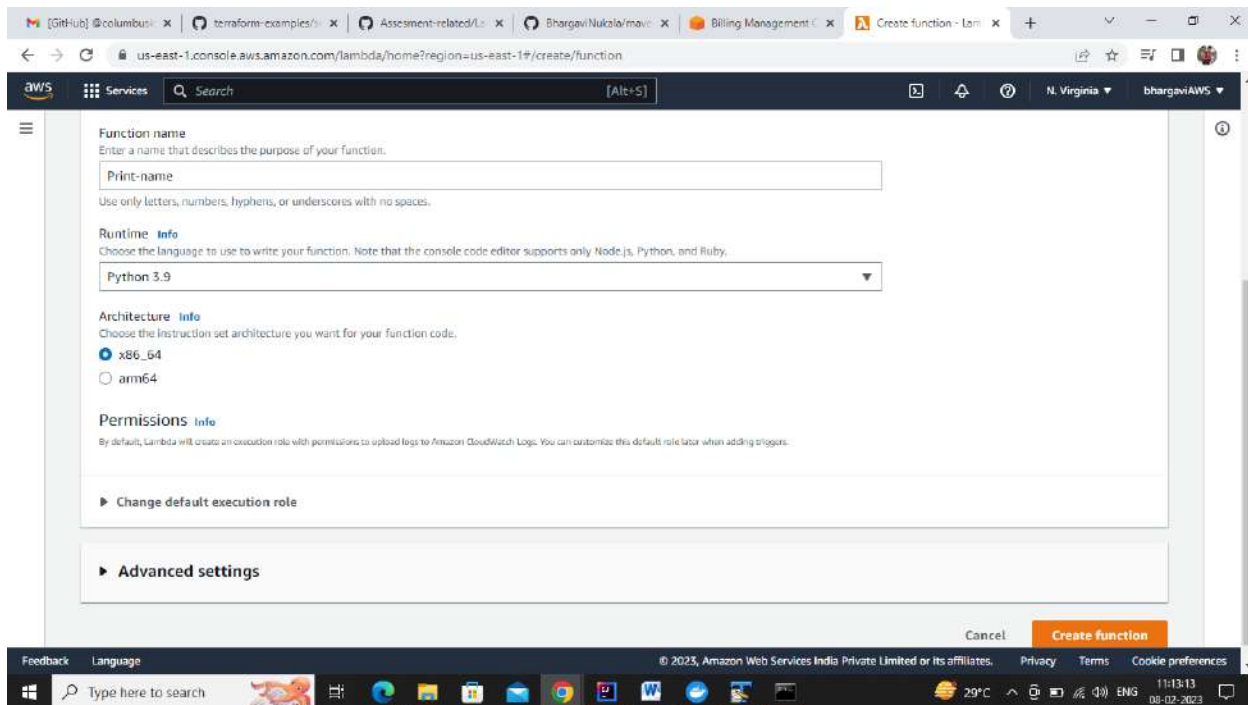
And at the same time, we can see that a new build is triggered in Jenkins.



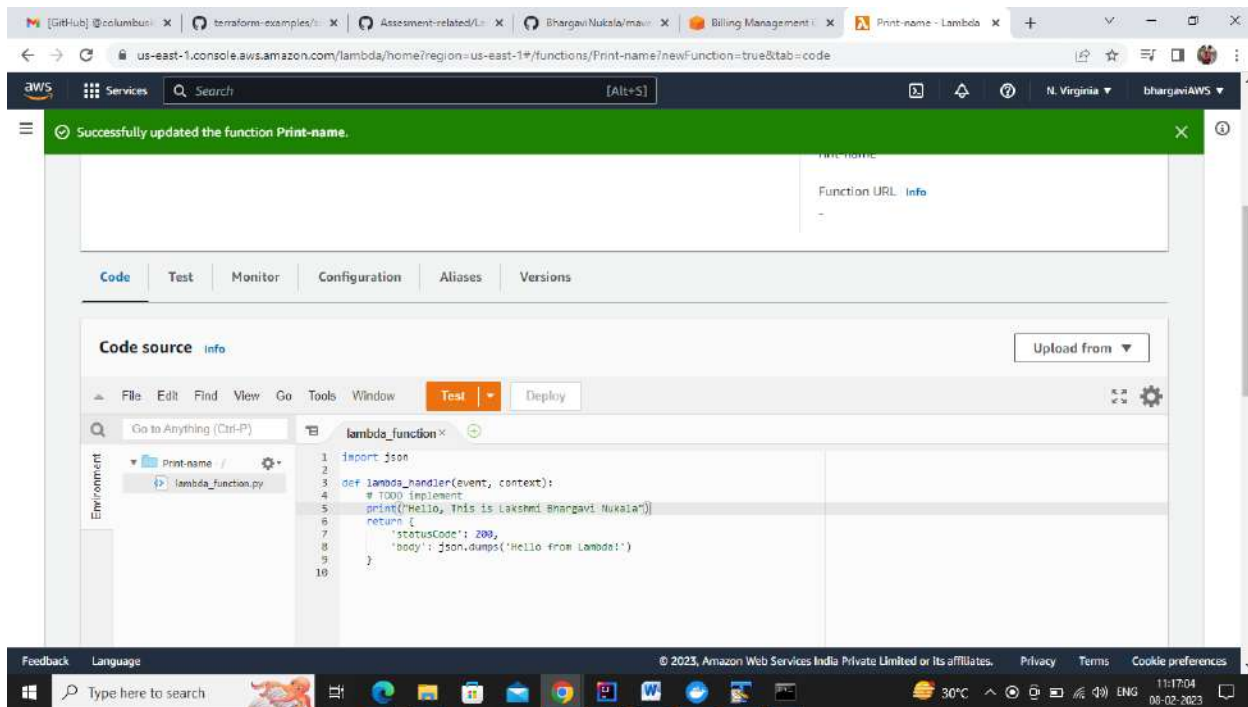


Lambda:

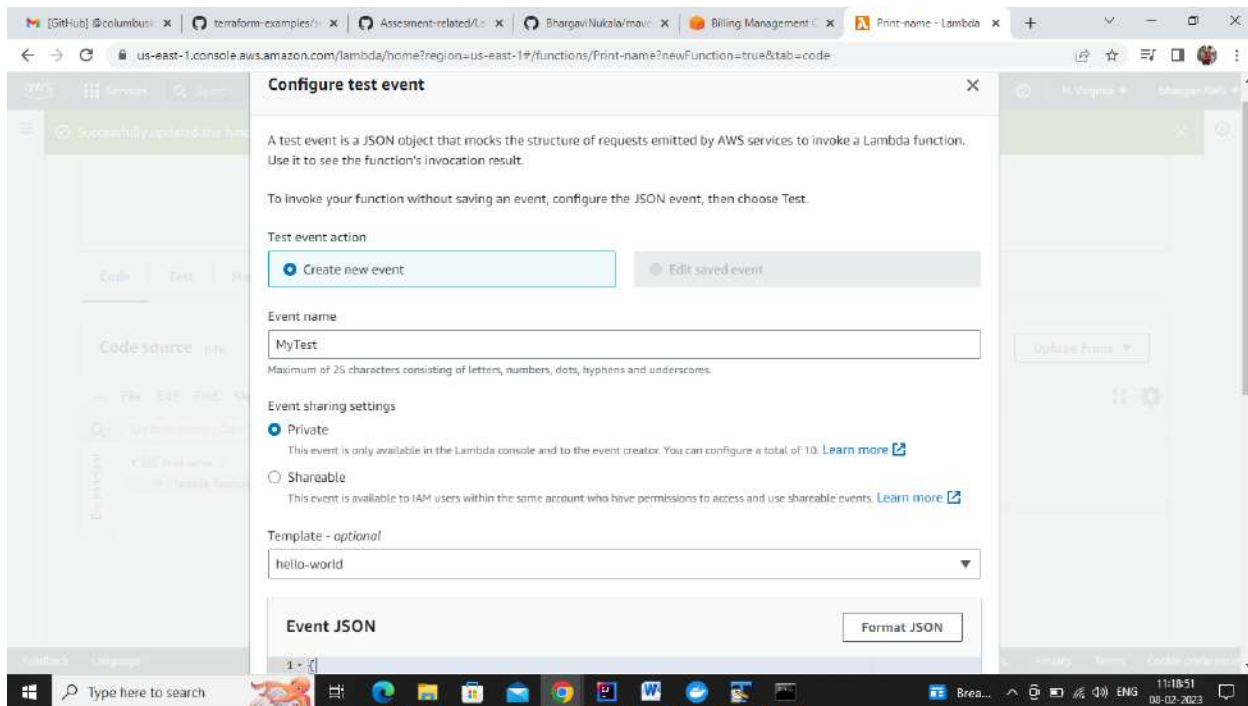
Task -1: Create a Lambda function that prints your name.



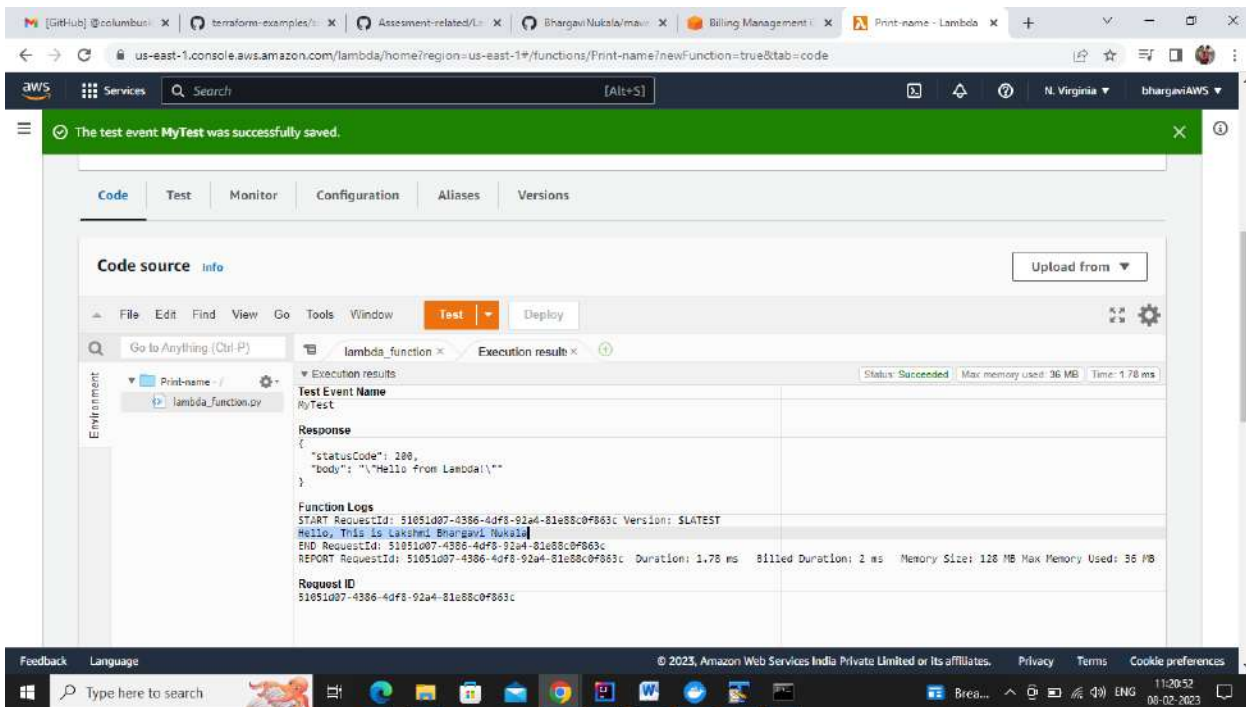
Add a line printing your name in the code. Then Deploy the code.



Now click on Test to configure a test event.

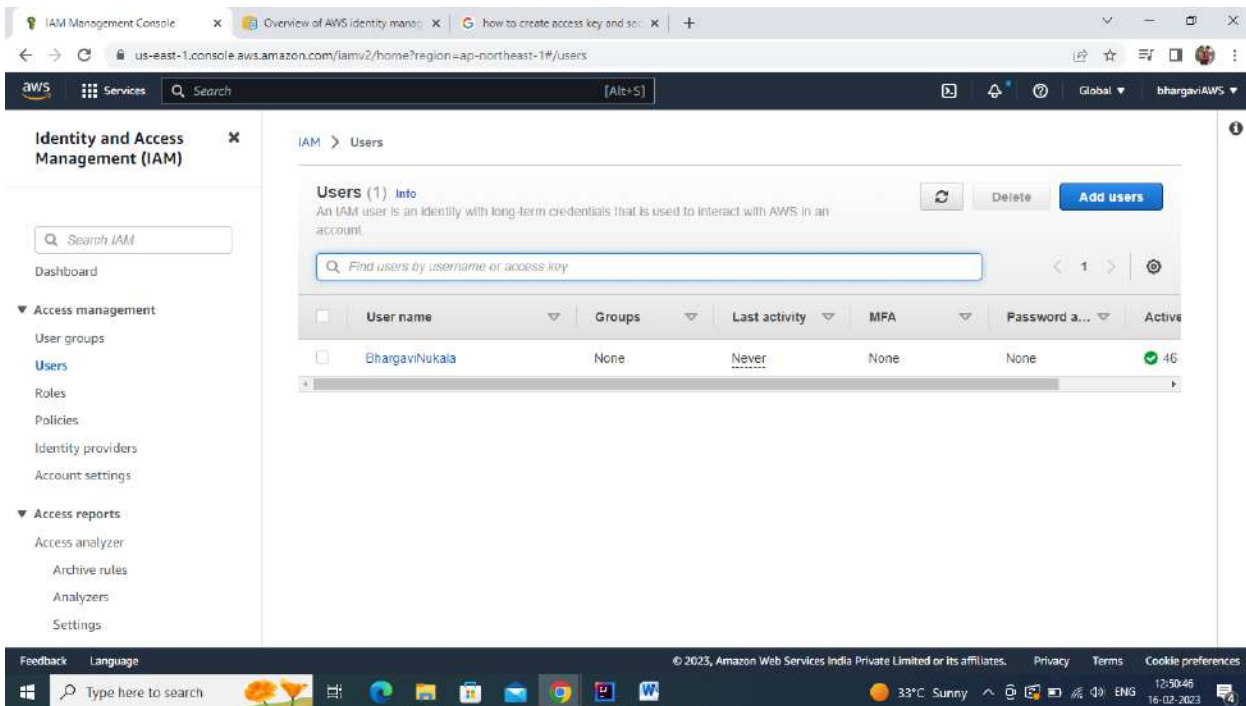


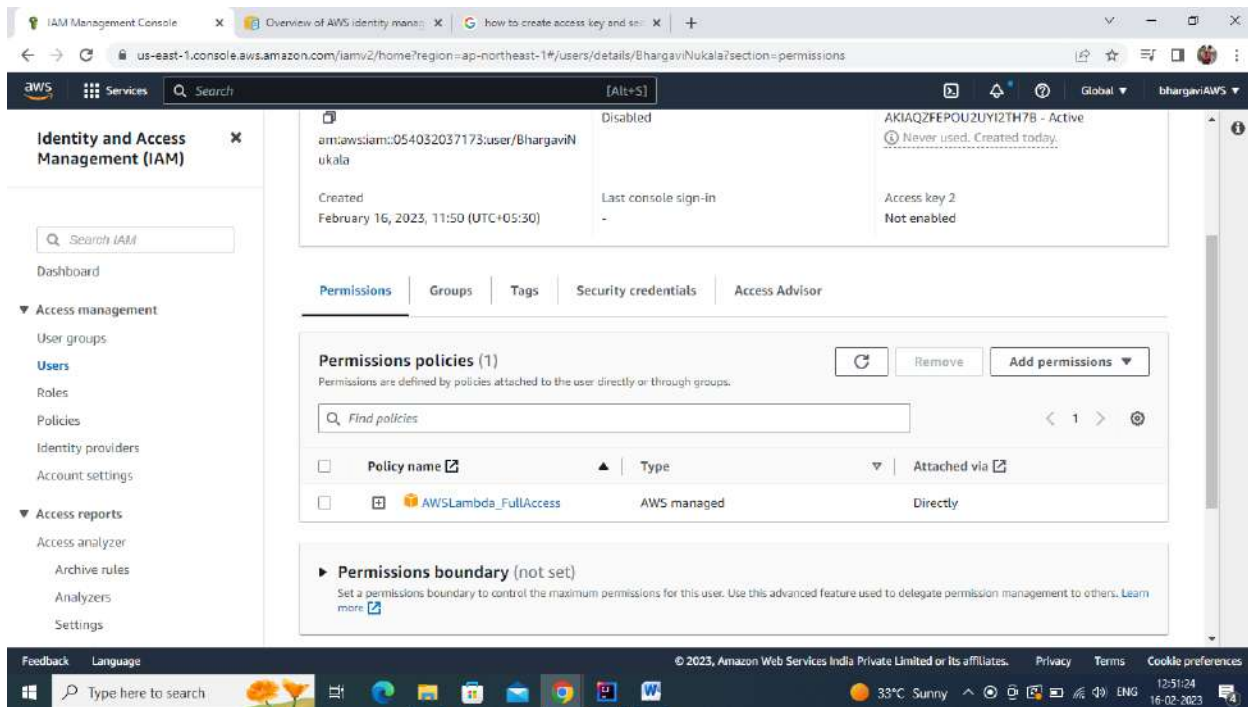
Then Click on Test to run the event and print your name.



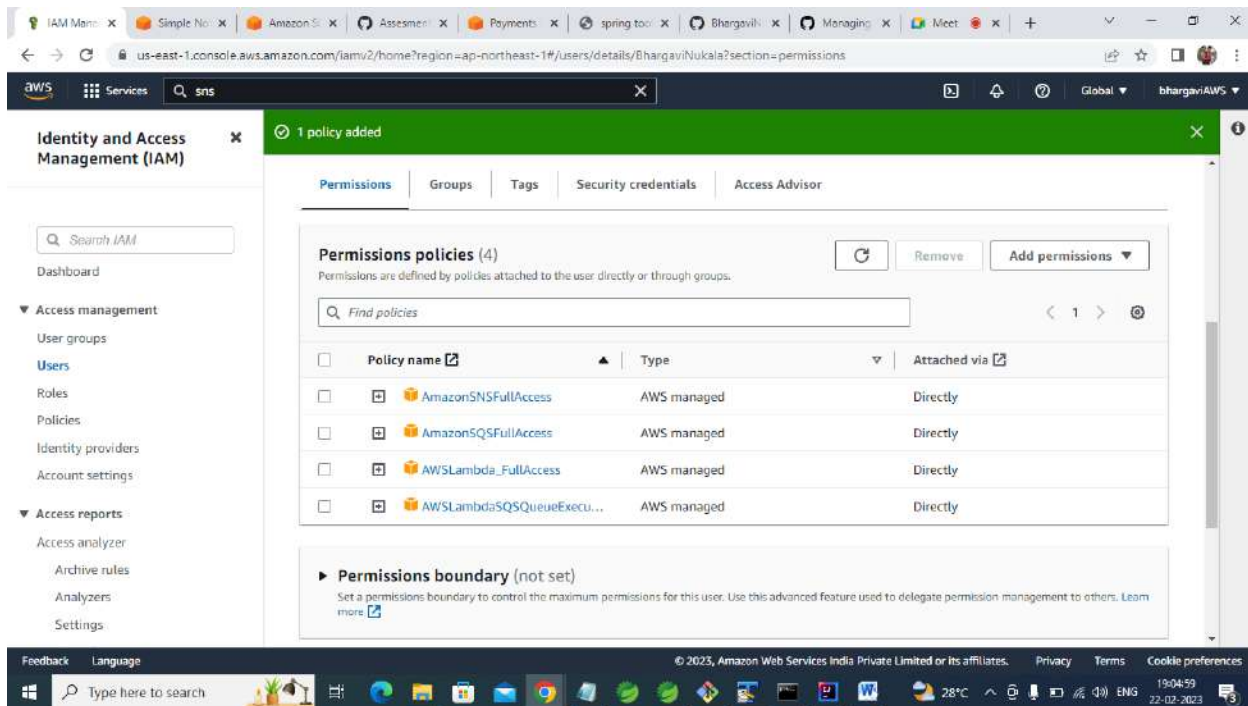
Task-2:

Create an IAM user BhargaviNukala in AWS and give full access to Lambda permission to this user.





Also give additional permissions for SQS and SNS in Permissions tab

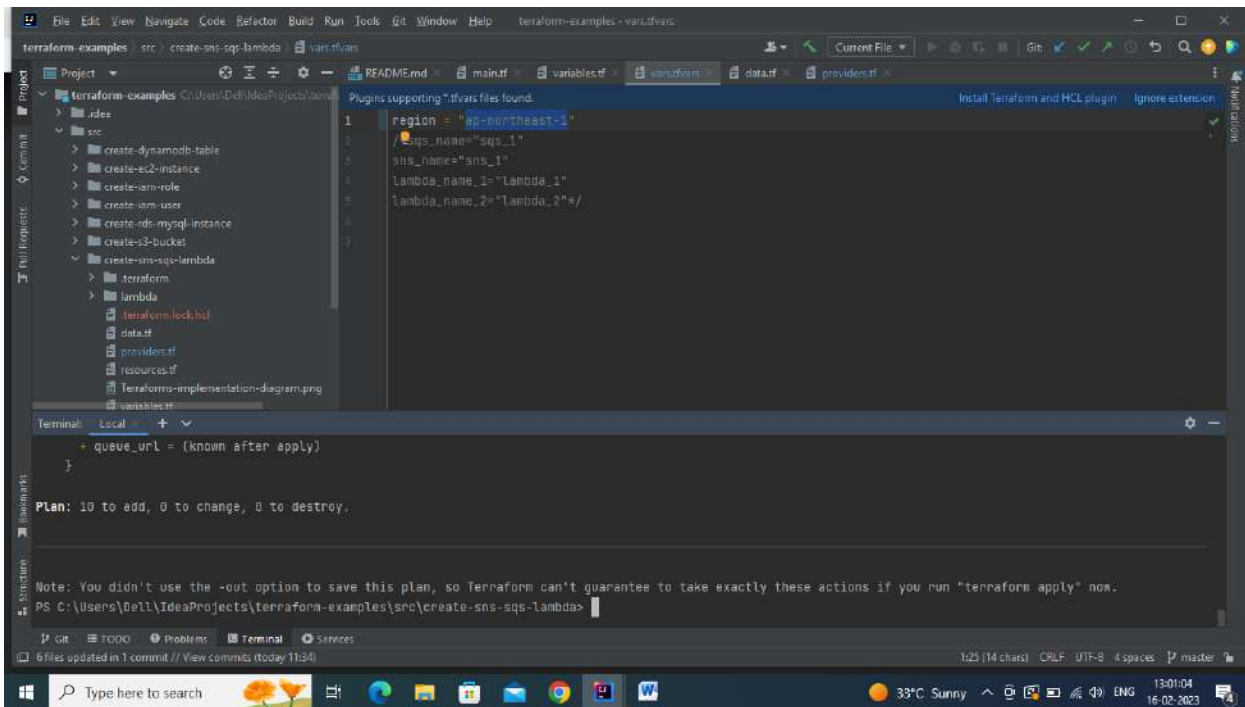


Check out the project terraform-examples into IntelliJ and set accesskey and secret access code of your AWS IAM user in providers.tf and region in vars.tf

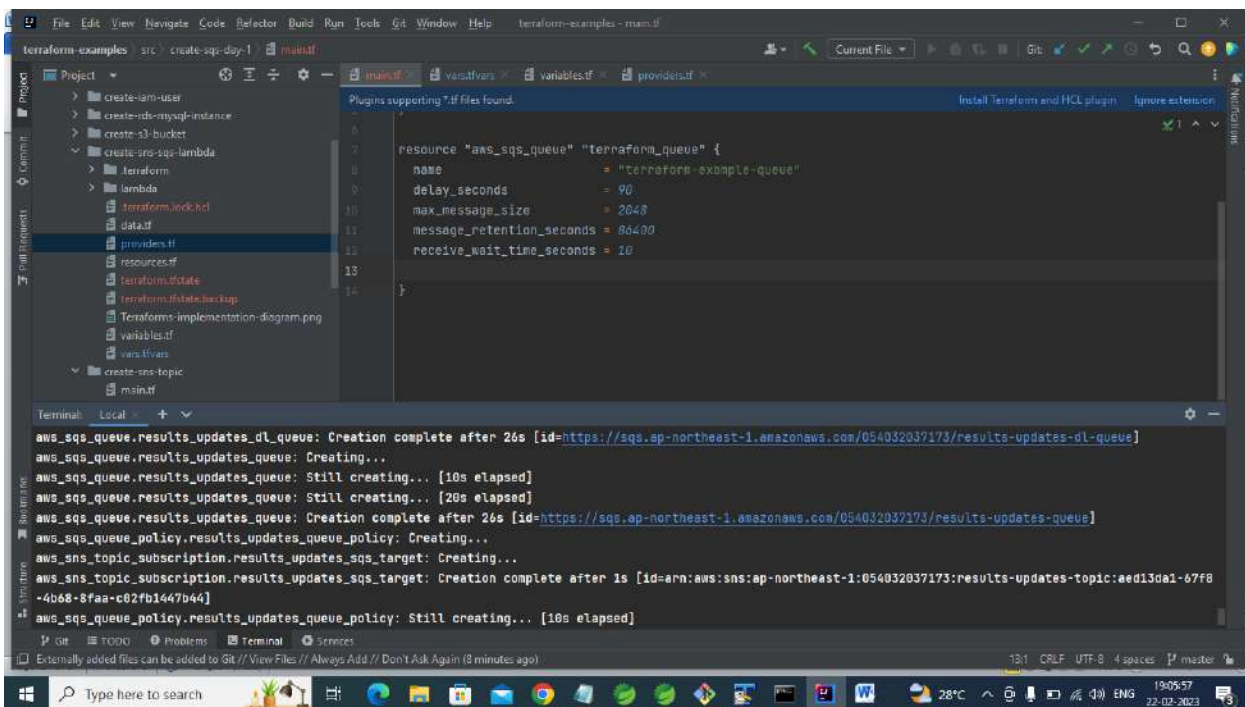
```
1 provider "aws" {
2
3     access_key = "AKIAQZ7EPG0U0YI2TH7B"
4     secret_key = "foWw/11TM1DXugPrj3j2Jy6jHR1+v4jdxRA2eu"
5     region = var.region
6 }
7
8 resource "aws_sns_topic" "results_updates" {
9     name = "results-updates-topic"
10 }
11
12 resource "aws_sqs_queue" "results_updates_dl_queue" {
13     name = "results-updates-dl-queue"
14 }
15
16 resource "aws_sqs_queue" "results_updates_queue" {
17     name = "results-updates-queue"
18     redrive_policy = "{ \"deadLetterTargetArn\": \"${aws_sqs_queue.results_updates_dl_queue.arn}\", \"maxReceiveCount\": 3, \"visibilityTimeoutSeconds\": 300 }"
19
20     tags = {
21         Environment = "dev"
22     }
23 }
24
25 resource "aws_sns_topic_subscription" "results_updates_sqs_target" {
```

```
1 region = "ap-northeast-1"
2 /sqs_name="sqs_1"
3 sns_name="sns_1"
4 lambda_name_1="lambda_1"
5 lambda_name_2="lambda_2"/
```

Run “terraform init” and “terraform plan” in the folder



Now run “terraform apply” command. It will ask to enter region and approve by entering yes. Then it will start creating Queue and Topic.



We can see that a Topic and Queue has been created in AWS

Browser tabs: IAM Man... Simple No... Amazon S... Assessm... Payments... spring too... Bhargavi... Managin... Meet... Tokyo | bhargaviAWS

URL: ap-northeast-1.console.aws.amazon.com/sns/v3/home?region=ap-northeast-1#/topics

Amazon SNS > Topics

Topics (1) [Edit] [Delete] [Publish message] [Create topic]

Search

Name	Type	ARN
results-updates-topic	Standard	arn:aws:sns:ap-northeast-1:054032057...

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 19:07:39 22-02-2023

Browser tabs: IAM Man... Simple No... Amazon S... Assessm... Payments... spring too... Bhargavi... Managin... Meet... Tokyo | bhargaviAWS

URL: ap-northeast-1.console.aws.amazon.com/sqs/v2/home?region=ap-northeast-1#/queues

Amazon SQS > Queues

Queues (3) [Refresh] [Edit] [Delete] [Send and receive messages] [Actions] [Create queue]

Search queues by prefix

Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
results-updates-dl-queue	Standard	22 Feb 2023, 19:00:49 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	-
results-updates-queue	Standard	22 Feb 2023, 19:01:16 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	-
terraform-example-queue	Standard	22 Feb 2023, 18:58:27 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	-

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 19:07:51 22-02-2023