

CICD ASSIGNMENT-2

NAME- RAJAT PANWAR

ROLL NO- R171218080

SAP ID- 500069414

SUBMITTED TO-

MR. HITESH KUMAR SHARMA

CODE- Code of our project

```
package cicd;

import java.util.*;

interface sales
{
    int sale();
}

class publication implements sales
{
    Scanner sc=new Scanner(System.in);
    publication()
    {
    }
    void show()
    {
        boolean t=true;
        while(t==true)
        {
            System.out.println("Select among (1)book , (2)tape , (3)Exit");
            int option=sc.nextInt();
            if(option == 1)
            {
                book obj=new book();
                System.out.println("1>add books ,2>display books");
                int opt=sc.nextInt();
                if(opt==1)
                {
                    obj.jok();
                }
                else if(opt==2)
                {
                    obj.display();
                }
                else
                {
                    System.out.println("Wrong input");
                }
            }

            else if(option == 2)
            {
                tape obj2=new tape();
                System.out.println("1>add tape ,2>Display tape");
                int ko=sc.nextInt();
                if(ko==1)
                {
                    obj2.moi();
                }
                else if(ko==2)
                {
                    obj2.display();
                }
                else
                {
                }
            }
        }
    }
}
```

```

        System.out.println("Wrong input");
    }
    }
    else if(option==3)
    {
        t=false;
    }
    else
    {
        System.out.println("Wrong input");
    }
    }
}
public int sale()
{
    System.out.println("Enter the sales ");
    int sal=sc.nextInt();
    return sal;
}
String title()
{
    String tit=sc.nextLine();

    return tit;
}
float price()
{
    float pric=sc.nextFloat();
    return pric;
}
}
class book extends publication
{
    Scanner in=new Scanner(System.in);
    String[] titl= new String[10];
    int[] count=new int[10];
    float[] pri=new float[10];
    int[] pag= new int[10];
    int number=1;
    book()
    {
    }
    void jok()
    {
        number++;
        for(int i=0;i<number-1;i++)
        {
            System.out.println("Enter the title of the book");
            titl[i]=title();
            System.out.println("Enter the price of the book");
            pri[i]=price();
            System.out.println("Enter the page of book");
            pag[i]=page();
        }
    }
    int page()

```

```

        return pagno;
    }
    void display()
    {
        for(int j=0;j<number;j++)
        {
            System.out.println("title="+titl[j]+ "price =" +pri[j]+" page="+pag[j]);
        }
    }
}
class tape extends publication
{
    Scanner s=new Scanner(System.in);
    String[] titt=new String[10];
    float[] pre=new float[10];
    float[] time=new float[10];
    int num=1;
    tape()
    {
    }
    void moi()
    {
        num++;
        for(int k=0;k<num-1;k++)
        {
            System.out.println("Enter the title of tape");
            titt[k]=title();
            System.out.println("Enter the price of tape");
            pre[k]=price();
            System.out.println("Enter the play time");
            time[k]=playing();
        }
    }
    float playing()
    {
        float tim=s.nextFloat();
        return tim;
    }
    void display()
    {
        for(int y=0;y<num;y++)
        {
            System.out.println("title="+titt[y]+ "price =" +pre[y]+" playing time="+time[y]);
        }
    }
}
}
public class test
{
    public static void main(String[] args)
    {
        System.out.println("\t\t Welcome to the store");
        publication ma=new publication();

        ma.show();
    }
}

```

```

    int sale();
}
class publication implements sales
{
    Scanner sc=new Scanner(System.in);
    publication()
    {
    }
    void show()
    {
        boolean t=true;
        while(t==true)
        {
            System.out.println("Select among (1)book , (2)tape , (3)Exit");
            int option=sc.nextInt();
            if(option == 1)
            {
                book obj=new book();
                System.out.println("\t>add books ,2>display books");
                int opt=sc.nextInt();
                if(opt==1)
                {
                    obj.jok();
                }
                else if(opt==2)
                {
                    obj.display();
                }
                else
                {
                    System.out.println("Wrong input");
                }
            }
            else if(option == 2)
            {
                tape obj2=new tape();
                System.out.println("\t>add tape ,2>Display tape");
                int ko=sc.nextInt();
                if(ko==1)
                {
                    obj2.moi();
                }
                else if(ko==2)
                {
                    obj2.display();
                }
                else
                {
                    System.out.println("Wrong input");
                }
            }
            else if(option==3)
            {
                t=false;
            }
        }
    }
}

```

```
test.java
1 package ccc;
2
3 import java.util.Scanner;
4
5 interface sales
6 {
7     int sale();
8 }
9 class publication implements sales
10 {
11     Scanner sc=new Scanner(System.in);
12     publication()
13     {
14     }
15     void show()
16     {
17         boolean t=true;
18         while(t==true)
19         {
20
21         }
22     }
23 }

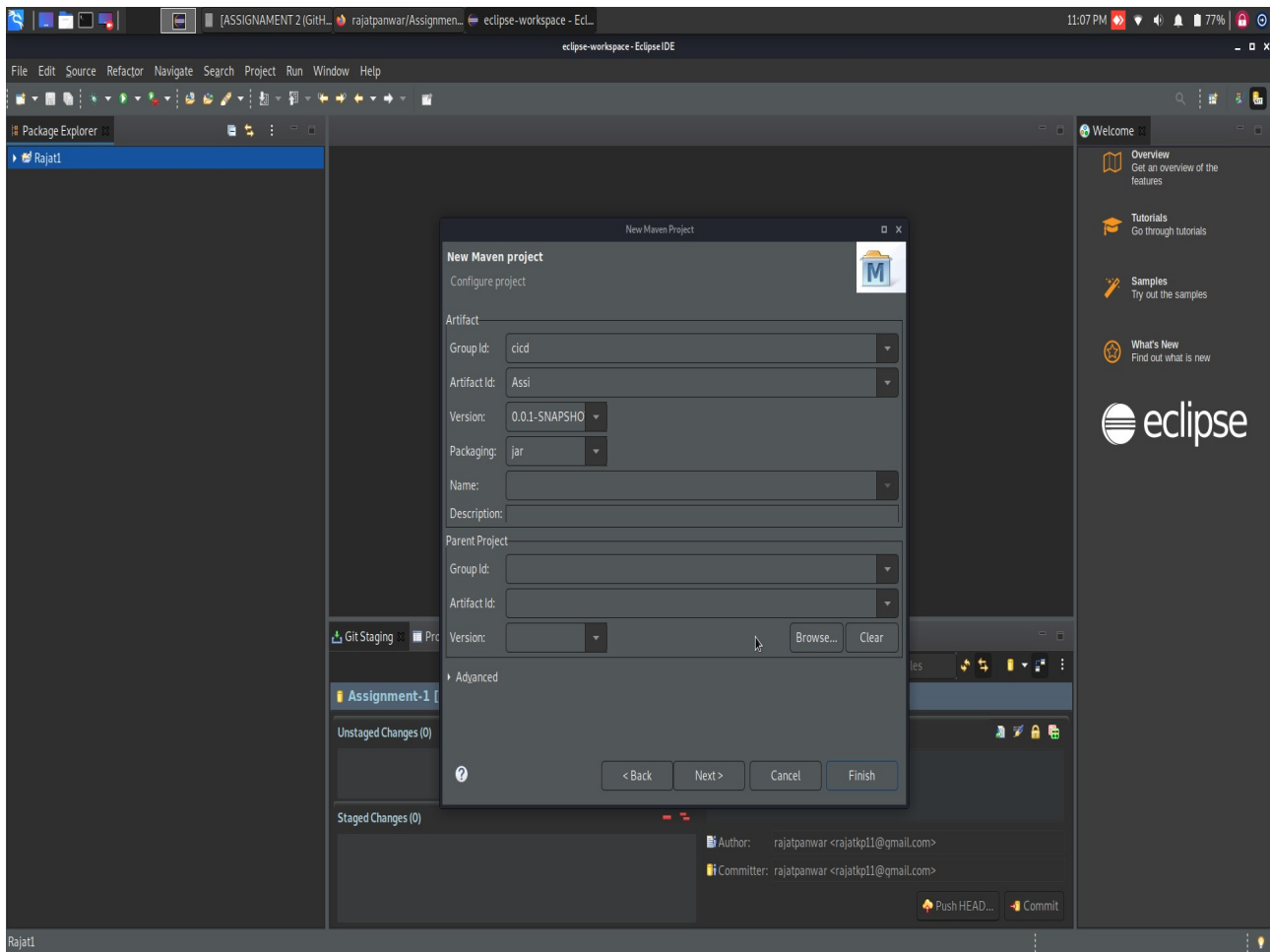
Console
<terminated> Assignment (1) [Maven Build] /usr/lib/jvm/jdk-11.0.9/bin/java (17-Nov-2020, 12:27:11 AM – 12:27:15 AM)
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]
[INFO] Scanning for projects...
[INFO] -----< upes:Assignment >-----
[INFO] Building Assignment 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Assignment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Assignment ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Assignment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Assignment ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Assignment ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.765 s
[INFO] Finished at: 2020-11-17T00:27:15+05:30
[INFO] -----
```

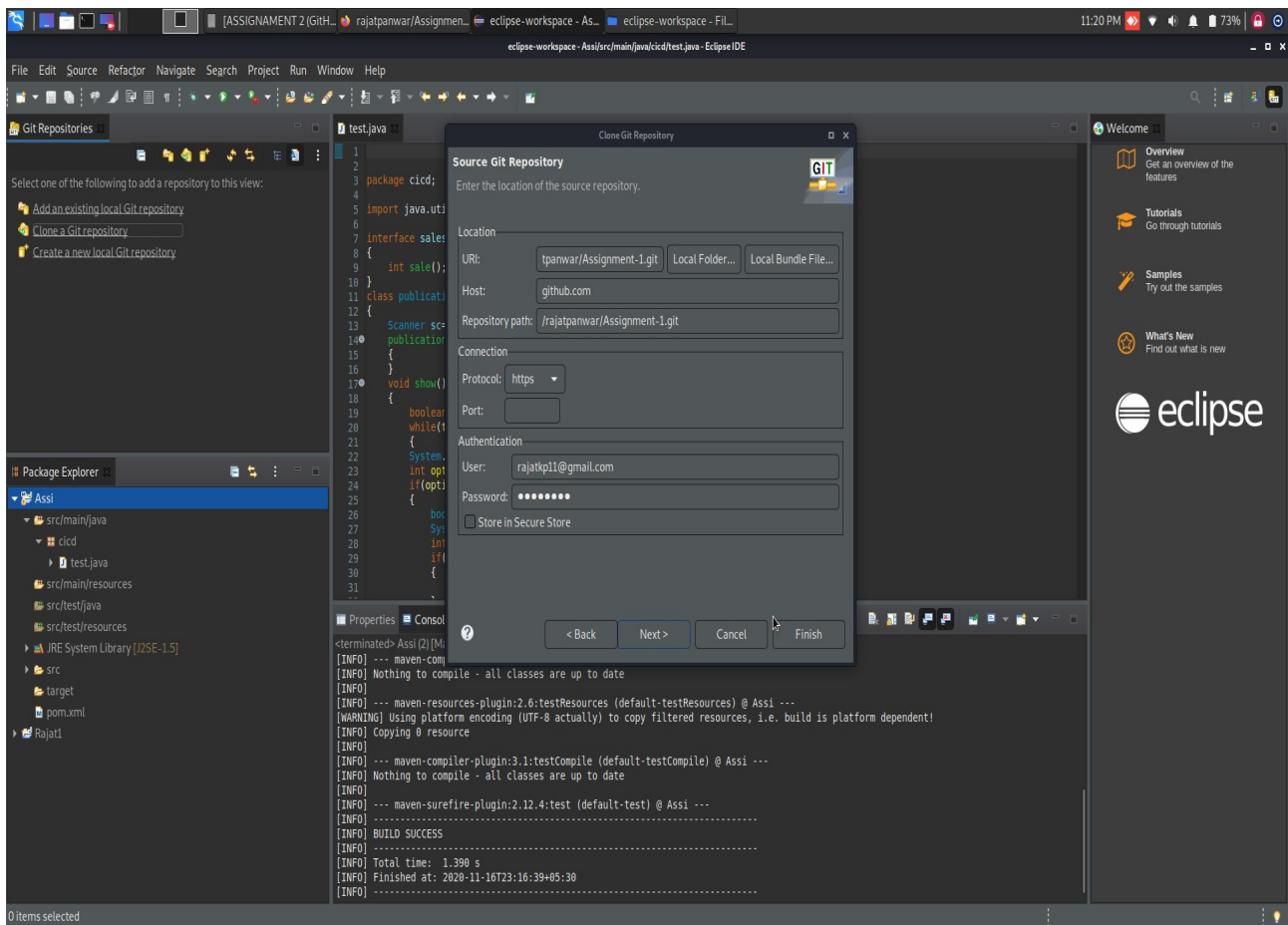
code run sucessfully with the help of “test” goal

STEP-2 :- integrte eclipse with github to send the code on github

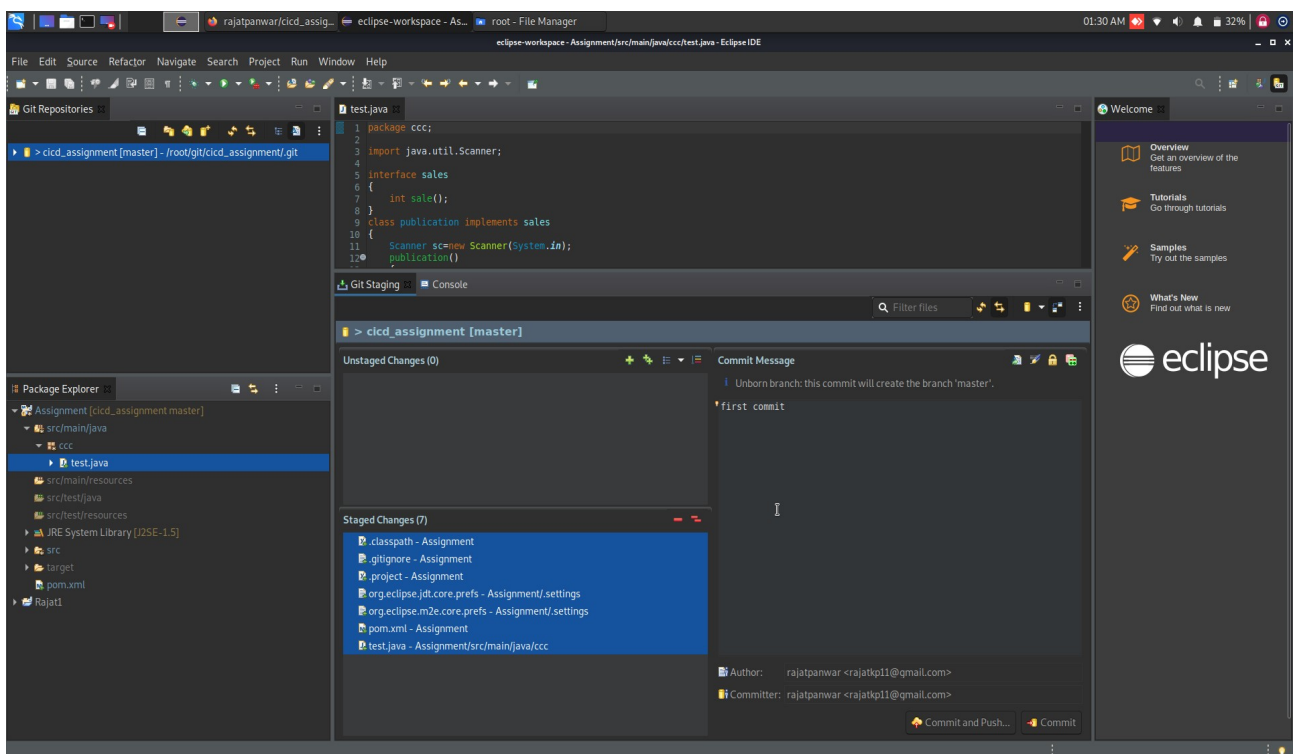
open your eclipse after that create a maven project and write code and check that code with the help of “test” goal after that

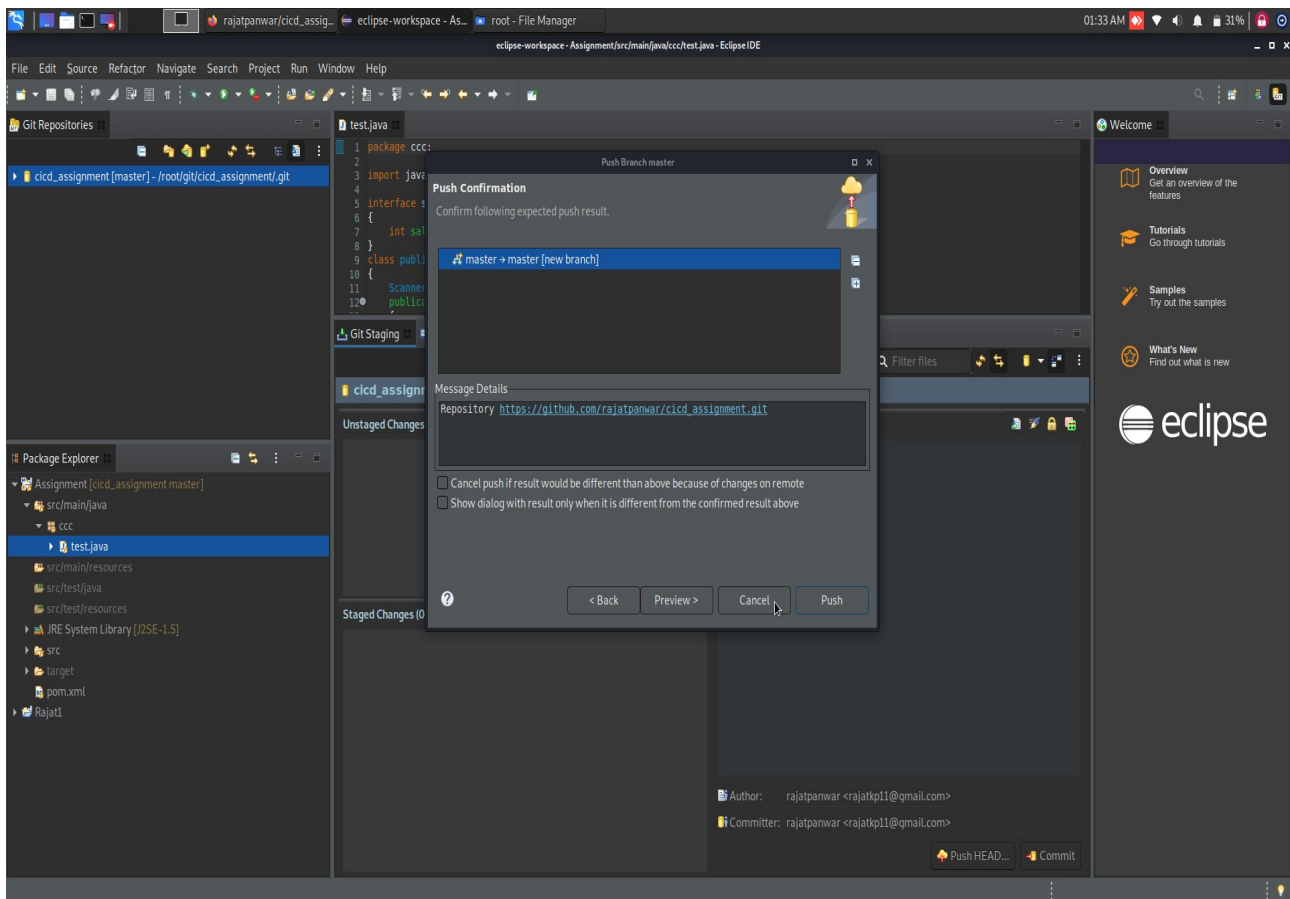
go open perspective--> git and fill your github url and password and username.



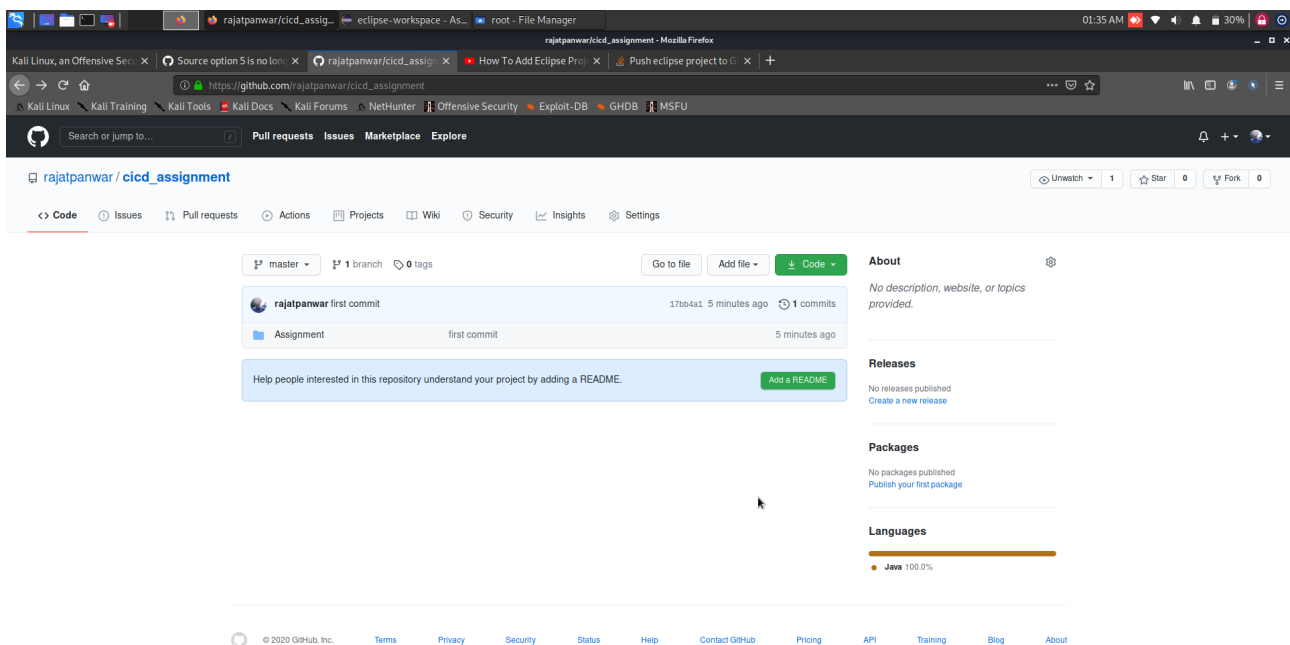


go **TEAM->COMMIT->** after that do unstages to stage changes after that click on “commit and push”

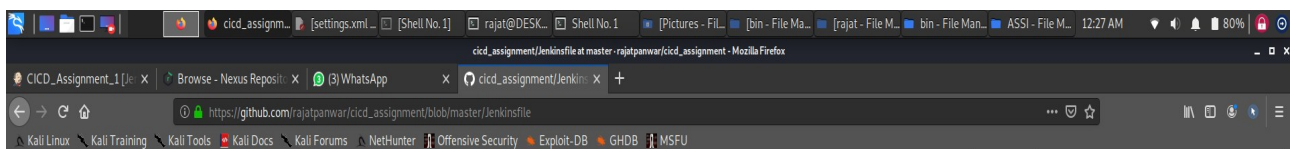
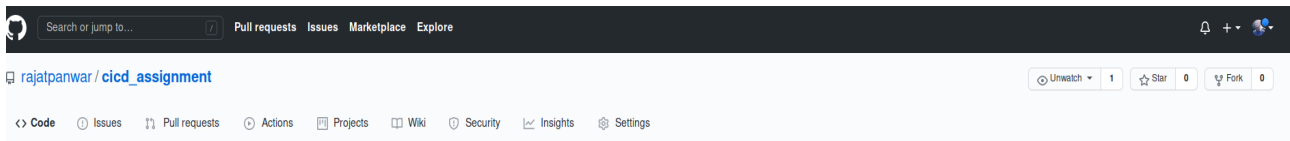




push your project successfully on github-



after that cut the pom.xml file from inside the Assignment and paste that file outside the Assignment and create the “**Jenkinsfile**” and write code



```
rajatpanwar update file Latest commit 93d67b4 4 minutes ago History
A: 1 contributor

31 lines (20 sloc) | 707 Bytes
Raw Blame

1 pipeline{
2   agent any
3
4   stages{
5
6     stage('clean')
7     { steps{
8       sh 'mvn clean'
9     }
10
11   }
12
13   stage('test')
14   { steps{
15     sh 'mvn test'
16   }
17
18   }
19
20   stage('deploy')
21   { steps{
22     sh 'mvn deploy'
23   }
24
25   }
26
27   }
28
29 }
30
31 }
```

```

pipeline{
agent any

stages{

stage('clean')
{ steps{
sh 'mvn clean'
}

}

stage('test')
{ steps{
sh 'mvn test'
}

}

stage('deploy')
{ steps{
sh 'mvn deploy'
}

}

}
}

```

after that modify the pom.xml file and adding some tag related to nexus server:- **give your nexus repository URL in url tag**


rajatpanwar update file

Latest commit 1064063 yesterday
History

1 contributor

19 lines (17 sloc)
738 Bytes

Raw
Blame

```

1 <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 https://maven.apache.org/maven-v4_0_0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>upes</groupId>
4   <artifactId>Assignment</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <properties>
7     <maven.compiler.source>1.6</maven.compiler.source>
8     <maven.compiler.target>1.6</maven.compiler.target>
9   </properties>
10
11   <distributionManagement>
12     <snapshotRepository>
13       <id>CICD_Assignment_1</id>
14       <name>CICD_Assignment_1</name>
15       <url>http://localhost:8081/repository/CICD_Assignment_1</url>
16     </snapshotRepository>
17   </distributionManagement>
18
19 </project>

```

STEP-3 start your jenkins and download plugin of nexus
after that create new pipeline project In my case my project is
“CICD_ASSSIGNMENT_1” and do some setting

The screenshot shows the Jenkins Pipeline configuration interface. The 'Pipeline' tab is selected. The 'Definition' is set to 'Pipeline script from SCM'. The 'SCM' is set to 'Git'. The 'Repositories' section shows a repository URL 'https://github.com/rajatpanwar/cicd_assignment.git' and 'Credentials' set to '- none -'. The 'Branches to build' section shows a branch specifier '*/master'. The 'Repository browser' is set to '(Auto)'. The 'Script Path' is 'Jenkinsfile'. The 'Lightweight checkout' checkbox is checked. There are help icons (question marks) next to several fields.

go managejenkins->configuration->sonatype nexus and do necessary changes to connect jenkins with nexus.

The screenshot shows the Jenkins configuration page for Sonatype Nexus. The breadcrumb navigation is 'Jenkins > configuration'. The 'Sonatype Nexus' section is expanded, showing 'Nexus Repository Manager Servers'. A new server is being added with the following details: Display Name 'nexusserver2', Server ID 'nexusserver2', Server URL 'http://localhost:8081/', and Credentials 'admin/*****'. There are 'Add' buttons for 'Global Pipeline Libraries' and 'Nexus Repository Manager Servers'.

and after that click on “**Build now**” see the console output and created pipeline

The screenshot shows the Jenkins web interface for a pipeline named 'Pipeline CICD_Assignment_1'. On the left, there is a sidebar with navigation links: Back to Dashboard, Status, Changes, Build Now, Delete Pipeline, Configure, Full Stage View, Open Blue Ocean, Rename, and Pipeline Syntax. Below these is the 'Build History' section, which lists recent builds with their IDs, timestamps, and status. The main content area displays the 'Stage View' for the pipeline. It shows a table of stages: 'Declarative: Checkout SCM', 'clean', 'test', and 'deploy'. Each stage has an 'Average stage times' bar and a 'Recent Changes' link. Below the stage view, there are 'Permalinks' for various build events. On the right side of the stage view, there are buttons for 'add description' and 'Disable Project'.

Stage	Declarative: Checkout SCM	clean	test	deploy
Average stage times	2s	2s	4s	6s
Recent Changes	2s	2s	4s	6s

Permalinks:

- Last build (#12), 2 min 35 sec ago
- Last stable build (#12), 2 min 35 sec ago
- Last successful build (#12), 2 min 35 sec ago
- Last unsuccessful build (#0), 1 day 0 hr ago
- Last completed build (#12), 2 min 35 sec ago

Console output:-

The screenshot shows the Jenkins web interface for the console output of the pipeline 'Pipeline CICD_Assignment_1'. The left sidebar contains navigation links: Back to Project, Status, Changes, Console Output, View as plain text, Edit Build Information, Open Blue Ocean, Thread Dump, Pause/resume, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area displays the 'Console Output' for the pipeline. The output shows the pipeline starting with a 'Checkout' stage, followed by a 'clean' stage, and then a 'test' stage. The output includes the following commands and their results:

```
Started by user Rajat panwar
Obtained Jenkinsfile from git https://github.com/rajatpanwar/cicd_assignment.git
Running in Durability level: MAX_SURVIVABILITY
(Pipeline) Start of Pipeline
(Pipeline) node
Running on Jenkins in /var/lib/jenkins/workspace/CICD_Assignment_1
(Pipeline) {
(Pipeline) stage
(Pipeline) { (Declarative: Checkout SCM)
(Pipeline) checkout
The recommended git tool is: git
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/rajatpanwar/cicd_assignment.git # timeout=10
Fetching upstream changes from https://github.com/rajatpanwar/cicd_assignment.git
> git --version # timeout=10
> git --version # 'git version 2.28.0'
> git fetch --tags --force --progress -- https://github.com/rajatpanwar/cicd_assignment.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 93d67b47c86cd42536bda2d6c783801634ca2b1 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 93d67b47c86cd42536bda2d6c783801634ca2b1 # timeout=10
Commit message: "update file"
> git rev-list --no-walk 10640630c9bb8f1ac638c81b3d746cba54c2279 # timeout=10
(Pipeline) }
(Pipeline) // stage
(Pipeline) withEnv
(Pipeline) {
(Pipeline) stage
(Pipeline) { (clean)
(Pipeline) sh
+ mvn clean
[[1;34mINFO[m] Scanning for projects...
[[1;34mINFO[m]
[[1;34mINFO[m] [m-----< [0;36mupes:Assignment[0;1m >-----[m
[[1;34mINFO[m] [mBuilding Assignment 0.0.1-SNAPSHOT[m
[[1;34mINFO[m] [m-----[ jar ]-----[m
...

```



after that do some changes in maven open “**setting.xml**” and add server tag and give your nexus repo name as a id

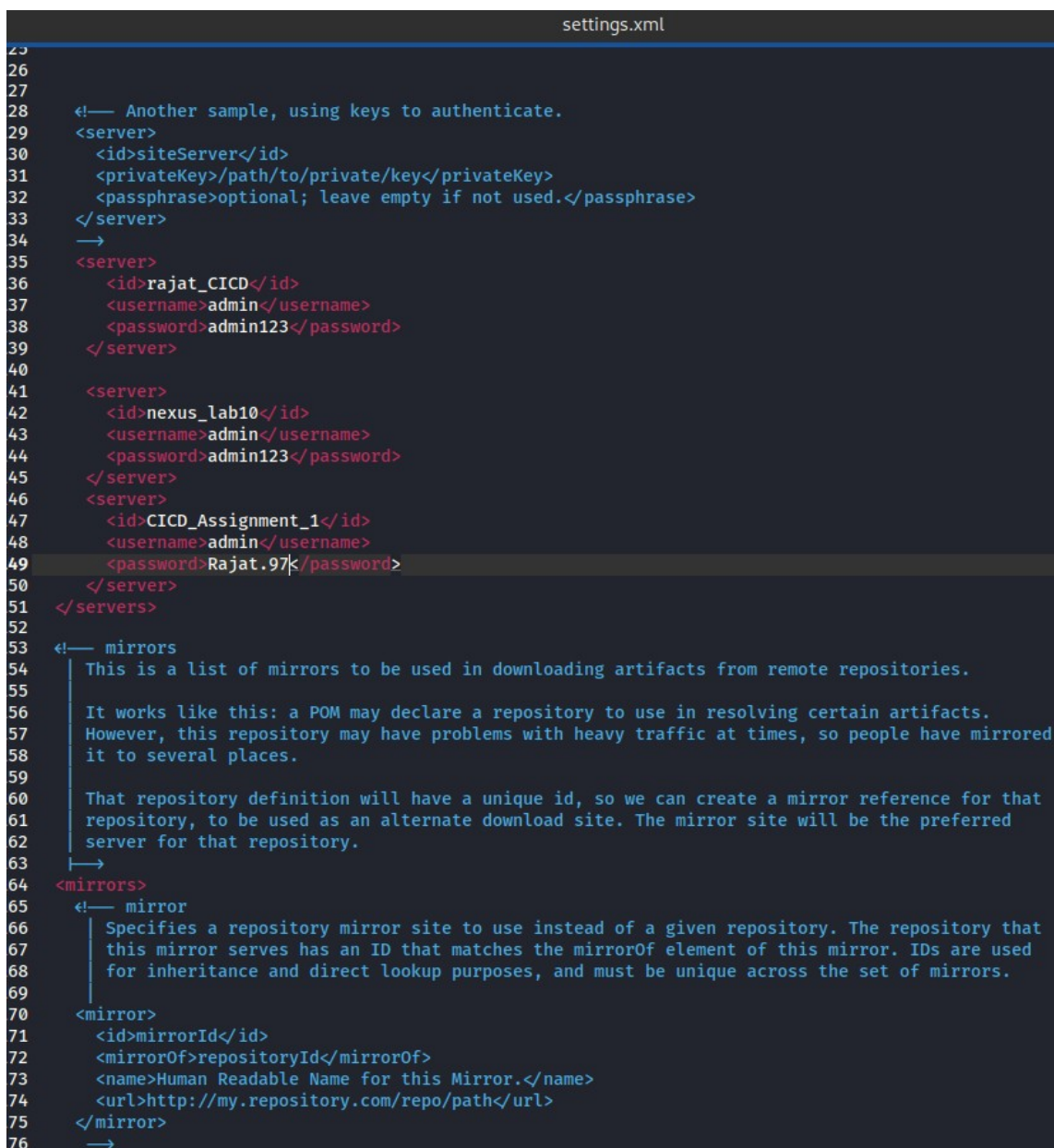
2. username} NEXUS

3. password} NEXUS

path of setting.xml file in KALI LINUX

go to otherlocation-->usr-->share-->maven-->conf-->setting.xml

```
<server>
  <id>"repository name"</id>
  <username>"nexus username"</username>
  <password>"give your password of nexus"</password>
</server>
```



```
settings.xml
25
26
27
28  <!-- Another sample, using keys to authenticate.
29  <server>
30    <id>siteServer</id>
31    <privateKey>/path/to/private/key</privateKey>
32    <passphrase>optional; leave empty if not used.</passphrase>
33  </server>
34  →
35  <server>
36    <id>rajat_CICD</id>
37    <username>admin</username>
38    <password>admin123</password>
39  </server>
40
41  <server>
42    <id>nexus_lab10</id>
43    <username>admin</username>
44    <password>admin123</password>
45  </server>
46  <server>
47    <id>CICD_Assignment_1</id>
48    <username>admin</username>
49    <password>Rajat.97</password>
50  </server>
51 </servers>
52
53  <!-- mirrors
54  This is a list of mirrors to be used in downloading artifacts from remote repositories.
55
56  It works like this: a POM may declare a repository to use in resolving certain artifacts.
57  However, this repository may have problems with heavy traffic at times, so people have mirrored
58  it to several places.
59
60  That repository definition will have a unique id, so we can create a mirror reference for that
61  repository, to be used as an alternate download site. The mirror site will be the preferred
62  server for that repository.
63  →
64  <mirrors>
65    <!-- mirror
66    Specifies a repository mirror site to use instead of a given repository. The repository that
67    this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used
68    for inheritance and direct lookup purposes, and must be unique across the set of mirrors.
69
70    <mirror>
71      <id>mirrorId</id>
72      <mirrorOf>repositoryId</mirrorOf>
73      <name>Human Readable Name for this Mirror.</name>
74      <url>http://my.repository.com/repo/path</url>
75    </mirror>
76    →
```

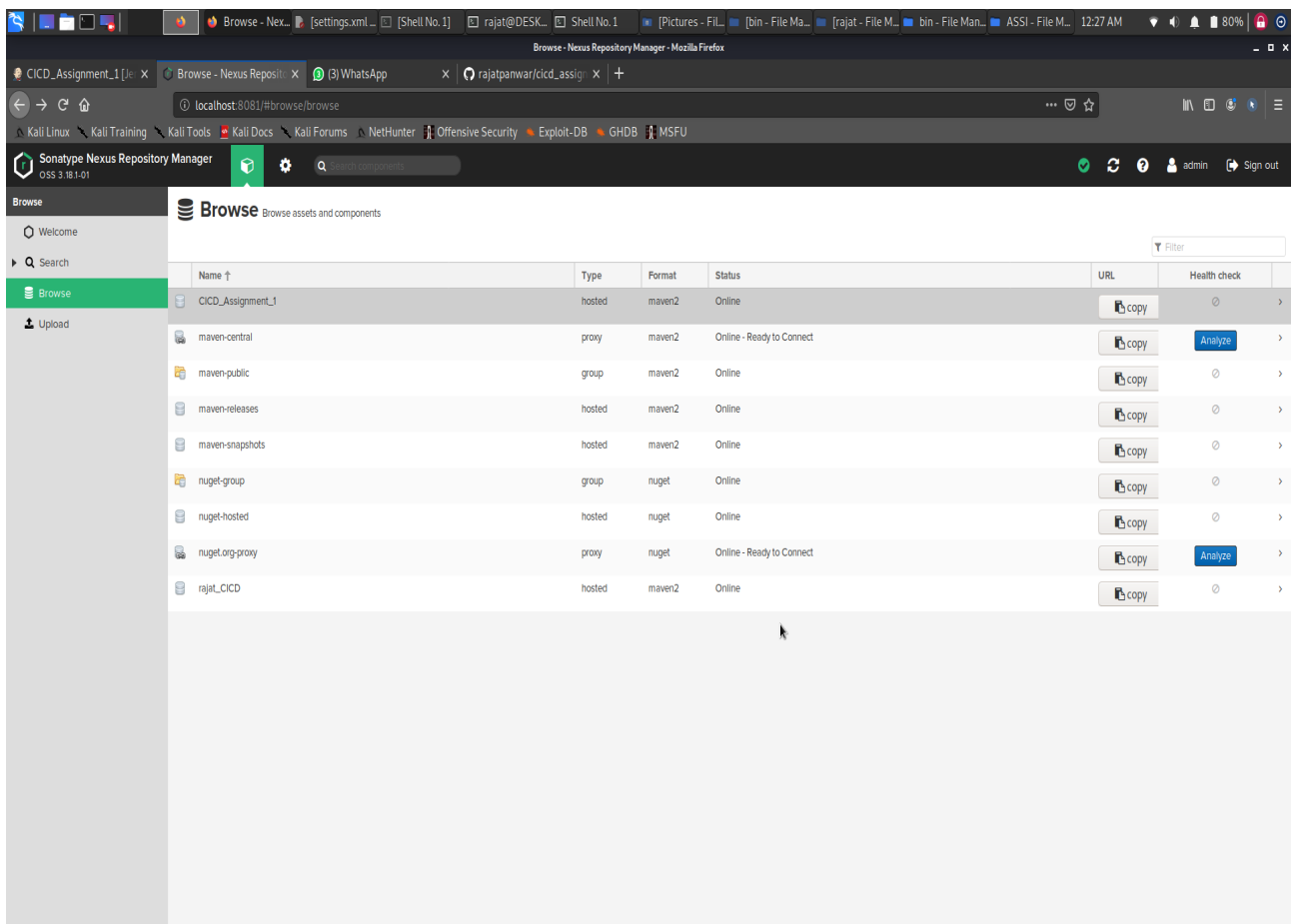
after that start your nexus “localhost.8081” with the command of “./nexus start”.

And create a new repository IN my case my repo name is “CICD_ASSIGNMENT_1”

step for create a repo in nexus:-

1. Go to setting and after that click on create repository.
2. after that clcik on maven2 (hosted).
3. Give repo name
4. Version policy--> SNAPSHOT
- 5.Deployment policy--> ALLOW REDOPLOY

after that click on create repository
and copy this URL and paste that URL in pom.xml file of <url>””</url> tag



and after that upload all files on nexus

