

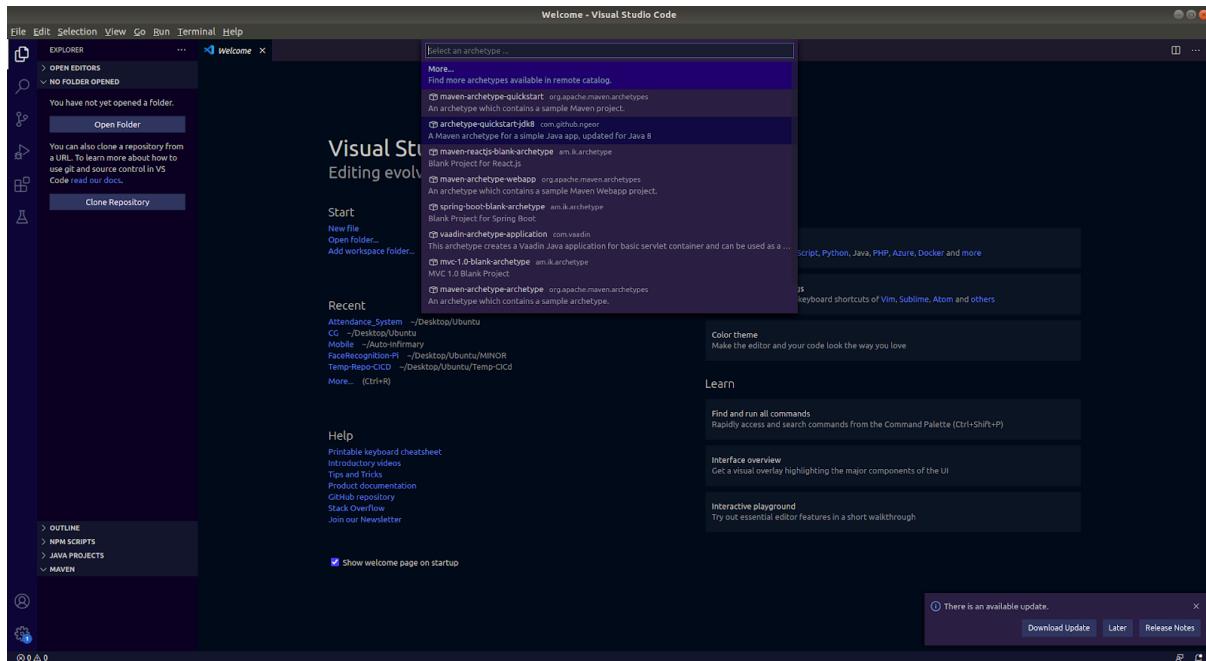
CICD

Assignment 2

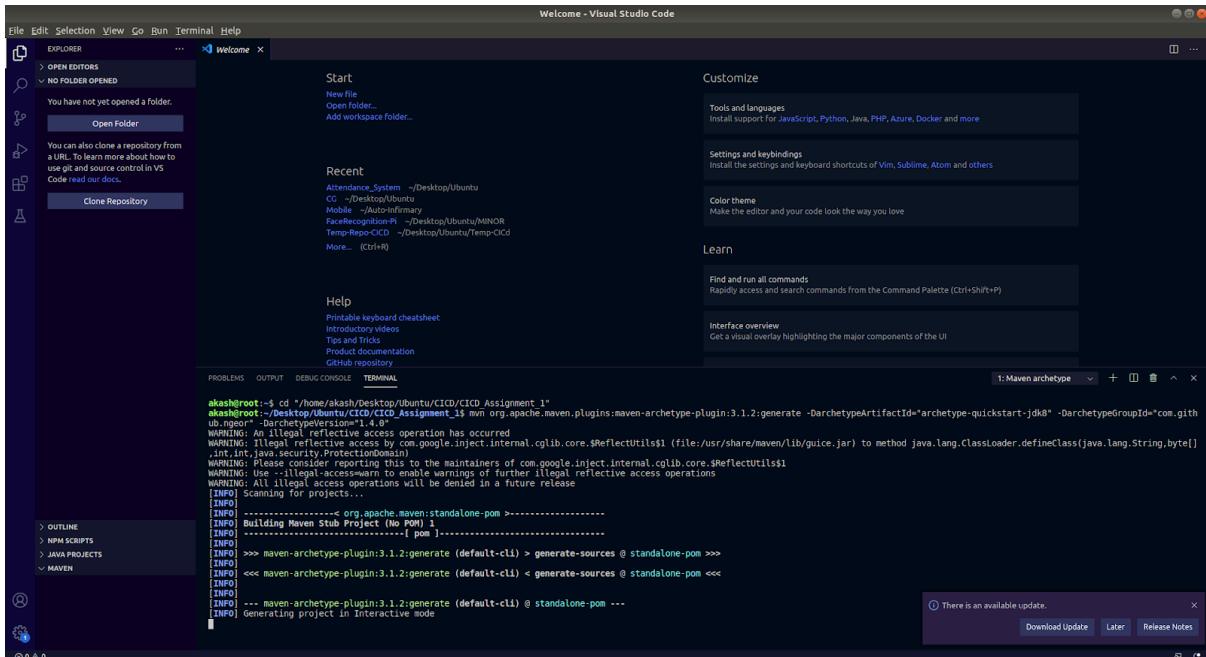
Aim: To create a pipeline using Jenkins File to build a maven project and deploy to the NEXUS server on every commit.

Tools Used: VS Code, Maven, Git, Jenkins, Nexus.

Step 1: Create a Maven Project.



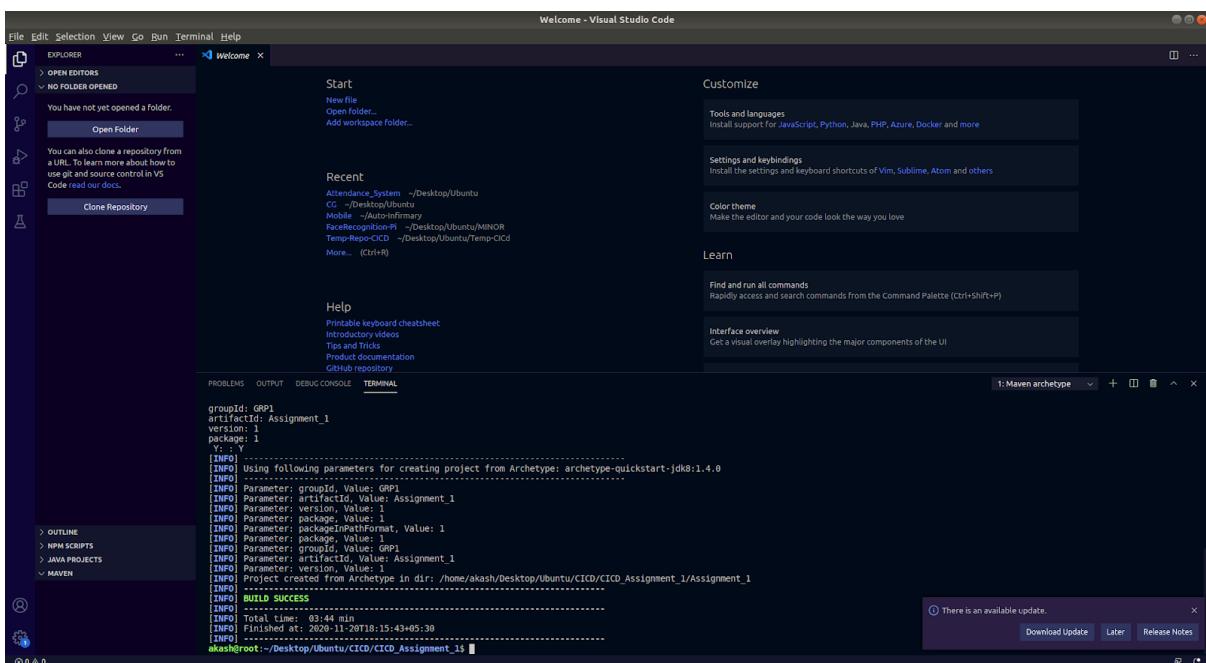
Step 2: Add Project ID, Artifact ID etc.



The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the following Maven command being run:

```
akash@root:~$ cd "/home/akash/Desktop/Ubuntu/CICD/CICD_Assignment_1"
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1$ mvn org.apache.maven.plugins:maven-archetype-plugin:3.1.2:generate -DarchetypeArtifactId="archetype-quickstart-jdk8" -DarchetypeGroupId="com.github.ngoeart" -DarchetypeVersion="1.4.0"
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[])
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[])
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.ReflectUtils$1
WARNING: com.google.inject.internal.cglib.core.ReflectUtils$1 uses or overrides an enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO] [INFO] ::::::::::::::: org.apache.maven:standalone-pom ::::::::::::::
[INFO] Building Maven Stub Project (NO POM) 1
[INFO] [INFO] :::::::::::::: pom ::::::::::::::
[INFO] >>> maven-archetype-plugin:3.1.2:generate (default-cli) > generate-sources @ standalone-pom >>
[INFO] <<< maven-archetype-plugin:3.1.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO] [INFO] :::::::::::::: maven-archetype-plugin:3.1.2:generate (default-cli) @ standalone-pom ::::::::::::::
[INFO] Generating project in Interactive mode
```

A notification bar at the bottom right indicates "There is an available update." with options "Download Update", "Later", and "Release Notes".

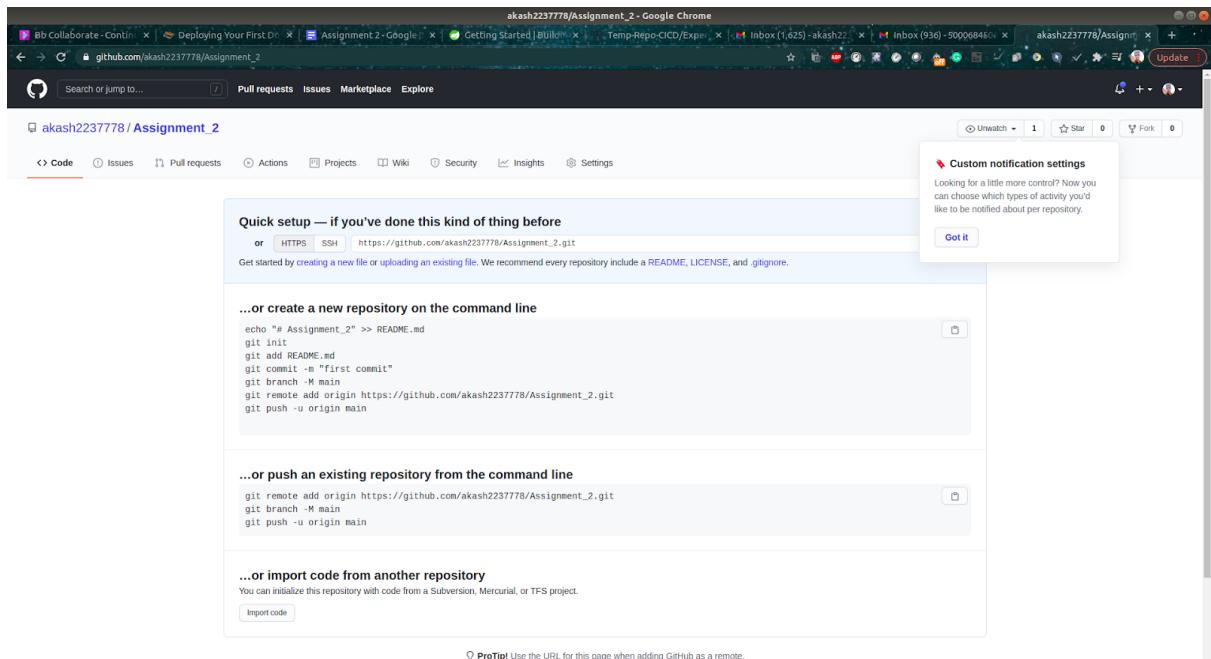


The screenshot shows the Visual Studio Code interface with the Explorer tab active. The left sidebar shows a Java project structure with an outline of files and folders. The terminal window at the bottom shows the command that was run to generate the project:

```
groupID: GRP1
artifactID: Assignment_1
version: 1.0-SNAPSHOT
package: com.akash.yt
Y: : Y
[INFO] ::::::::::::::: org.apache.maven:standalone-pom ::::::::::::::
[INFO] Building Maven Stub Project (NO POM) 1
[INFO] [INFO] :::::::::::::: pom ::::::::::::::
[INFO] Parameter: groupId, Value: GRP1
[INFO] Parameter: artifactId, Value: Assignment_1
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.akash.yt
[INFO] Parameter: packageInPathFormat, Value: 1
[INFO] Parameter: repositorySystemId, Value: https://repo.maven.apache.org/maven2/
[INFO] Parameter: repositoryId, Value: central
[INFO] Parameter: repositoryUrl, Value: https://repo.maven.apache.org/maven2/
[INFO] Parameter: groupid, Value: GRP1
[INFO] Parameter: artifactId, Value: Assignment_1
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/akash/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
[INFO] :::::::::::::: pom ::::::::::::::
[INFO] BUILD SUCCESS
[INFO] Total time: 03:44 min
[INFO] Finished at: 2020-11-20T18:15:43+05:30
[INFO] :::::::::::::: pom ::::::::::::::
```

A notification bar at the bottom right indicates "There is an available update." with options "Download Update", "Later", and "Release Notes".

Step 3: Create a Repository on GitHub.



Step 4: Initialize git in project and add remote.

```
akash@root: ~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
File Edit View Search Terminal Help
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git init
Initialized empty Git repository in /home/akash/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1/.git/
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git remote add origin https://github.com/akash2237778/Assignment_2.git
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ ls
pom.xml  src
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git add .
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

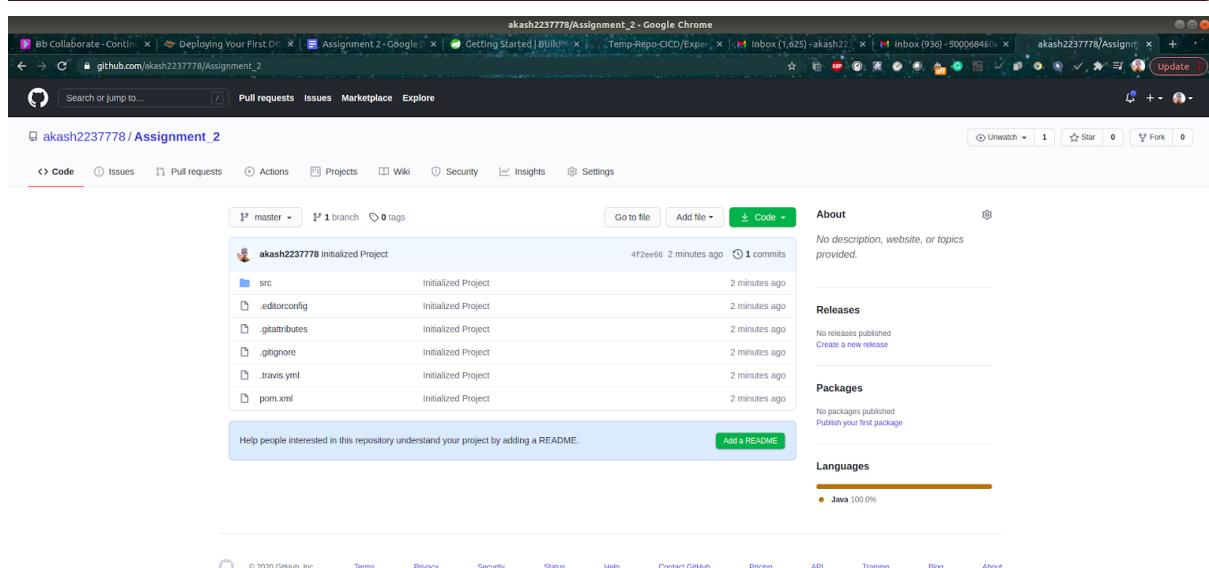
    new file:  .editorconfig
    new file:  .gitattributes
    new file:  .gitignore
    new file:  .travis.yml
    new file:  pom.xml
    new file:  src/main/java/1/App.java
    new file:  src/test/java/1/AppTest.java

akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$
```

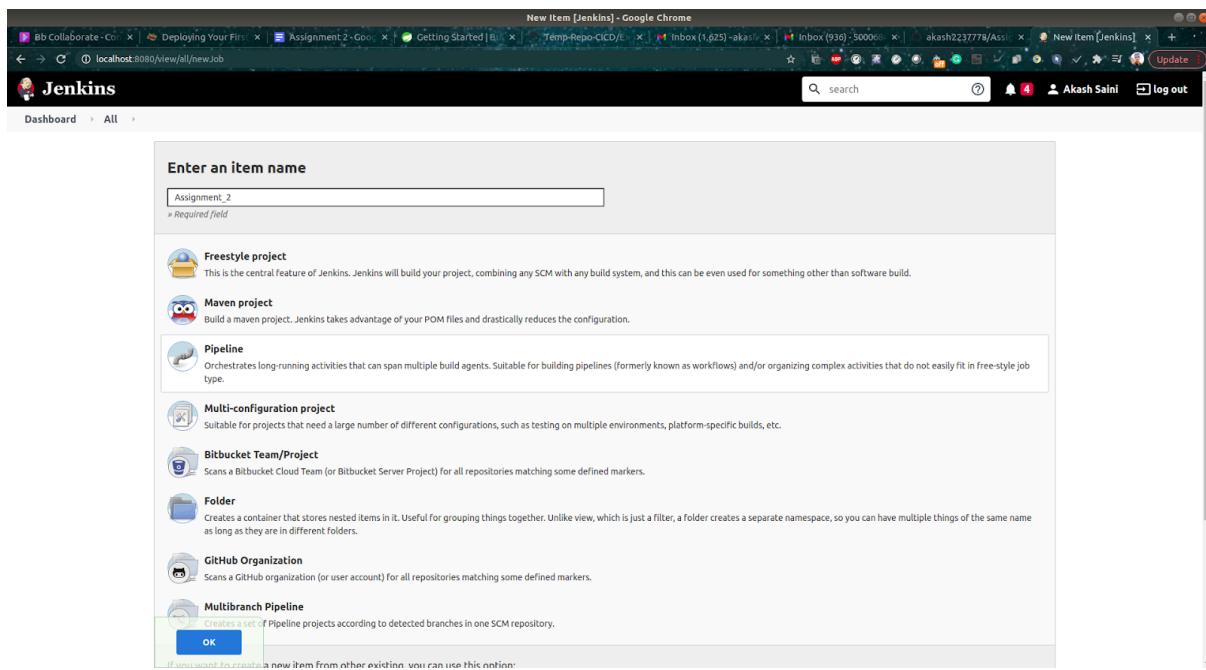
Step 5: Commit and push changes.

```
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
File Edit View Search Terminal Help

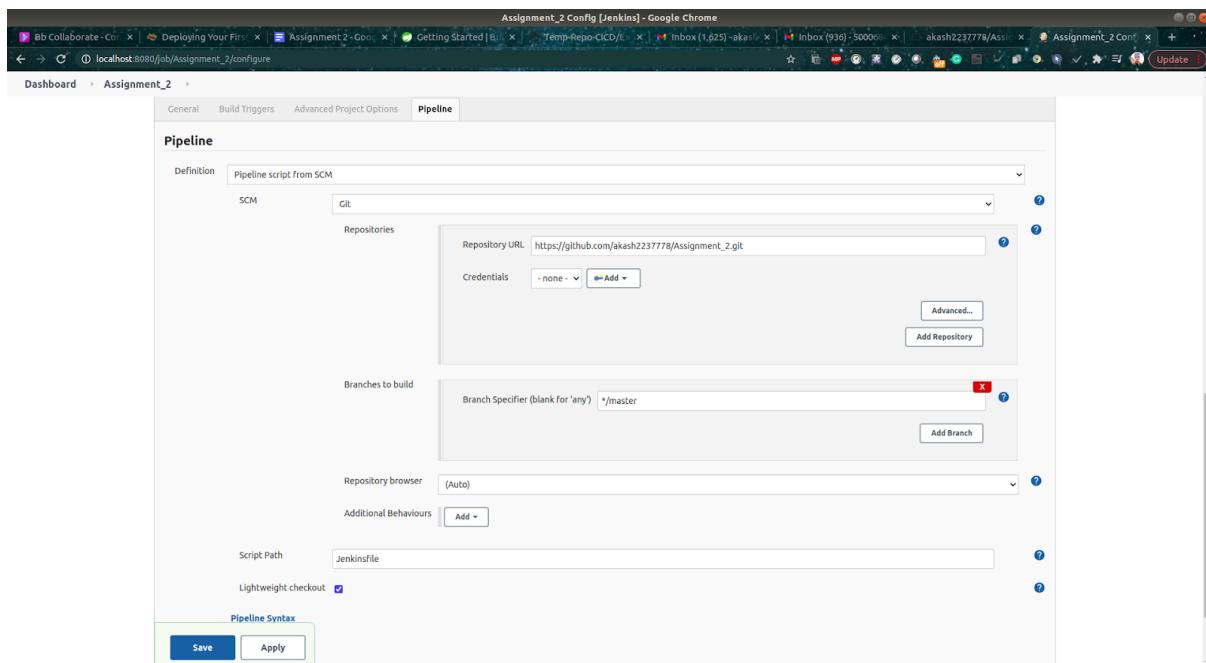
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git commit -m "Initialized Project"
[master (root-commit) 4f2ee66] Initialized Project
 7 files changed, 268 insertions(+)
 create mode 100644 .editorconfig
 create mode 100644 .gitattributes
 create mode 100644 .gitignore
 create mode 100644 .travis.yml
 create mode 100644 pom.xml
 create mode 100644 src/main/java/1/App.java
 create mode 100644 src/test/java/1/AppTest.java
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git push origin master
git: 'credential-wincred' is not a git command. See 'git --help'.
Username for 'https://github.com': akash2237778
Password for 'https://akash2237778@github.com':
git: 'credential-wincred' is not a git command. See 'git --help'.
Counting objects: 16, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (16/16), 2.93 KiB | 999.00 KiB/s, done.
Total 16 (delta 0), reused 0 (delta 0)
To https://github.com/akash2237778/Assignment_2.git
 * [new branch]      master -> master
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$
```



Step 6: Create a Jenkins project. Enter Name-> Pipeline -> OK



Step 7: Add SCM details and location of Jenkinsfile.



Step 8: Create Jenkins File.

```
akash@root: ~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
File Edit View Search Terminal Help
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ vim Jenkinsfile
```

```
akash@root: ~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
File Edit View Search Terminal Help
pipeline{
    agent any
    stages{
        stage('Build'){
            steps{
                echo 'Build Stage'
            }
        }
        stage('Clean'){
            steps{
                sh 'mvn clean'
            }
        }
    }
}
```

Step 9: Commit Push

```
akash@root: ~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1
File Edit View Search Terminal Help
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ vim Jenkinsfile
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Jenkinsfile

nothing added to commit but untracked files present (use "git add" to track)
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git add .
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git commit -m 'Added Jenkinsfile'
[master 4d55e43] Added Jenkinsfile
 1 file changed, 15 insertions(+)
 create mode 100644 Jenkinsfile
akash@root:~/Desktop/Ubuntu/CICD/CICD_Assignment_1/Assignment_1$ git push origin master
git: 'credential-wincred' is not a git command. See 'git --help'.
Username for 'https://github.com': akash2237778
Password for 'https://akash2237778@github.com':
git: 'credential-wincred' is not a git command. See 'git --help'.
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

Step 10: Build will automatically started on Jenkins.

The screenshot shows the Jenkins interface for the 'Pipeline Assignment_2' project. On the left, there's a sidebar with various management options like Status, Changes, Build Now, and Delete Pipeline. The main area displays the 'Stage View' for the pipeline, which consists of three stages: 'Declarative: Checkout SCM', 'Build', and 'Clean'. Each stage has a duration of 6s, 3s, and 16s respectively. Below the stage view, there's a 'Recent Changes' section showing two recent commits from 'Nov 20'. At the bottom, there's a 'Permalinks' section listing several build links. The top of the page shows a navigation bar with multiple tabs and a user profile for 'Akash Saini'.

Step 11: Create an Artifact Repository on Nexus.

The image shows two screenshots of the Sonatype Nexus Repository Manager interface. The top screenshot displays the 'Repositories' list under the 'Administration' menu. The bottom screenshot shows the 'Create Repository' dialog for creating a new Maven 2 hosted repository named 'Assignment_2'.

Screenshot 1: Repositories List

The left sidebar shows the navigation menu:

- Administration
- Repository
- Blob Stores
- Repositories** (highlighted)
- Content Selectors
- Cleanup Policies
- Routing Rules
- Security
- Privileges
- Roles
- Users
- Anonymous
- LDAP
- Realms
- SSL Certificates
- IQ Server
- Support
- System

The main area shows a list of available recipes:

- apt (hosted)
- apt (proxy)
- bower (group)
- bower (hosted)
- bower (proxy)
- docker (group)
- docker (hosted)
- docker proxy
- gitfs (hosted)
- go (group)
- go (proxy)
- maven2 (group)
- maven2 (hosted)
- maven2 (proxy)
- npm (group)
- npm (hosted)
- npm (proxy)
- nugget (group)
- nugget (hosted)
- nugget (proxy)
- pypi (group)
- pypi (hosted)
- pypi (proxy)
- raw (group)
- raw (hosted)
- raw (proxy)
- rubygems (group)
- rubygems (hosted)
- rubygems (proxy)
- yum (group)
- yum (hosted)
- yum (proxy)

Screenshot 2: Create Repository Dialog

The dialog is titled 'Create Repository: maven2 (hosted)'. The fields are as follows:

- Name: Assignment_2
- Online: checked
- Maven 2
 - Version policy: Release
 - Layout policy: Strict
- Storage
 - Blob store: default
- Hosted
 - Deployment policy: Disable redeploy
- Cleanup Policy
 - Available cleanup policies: None

At the bottom are 'Create repository' and 'Cancel' buttons.

Step 12: Add Nexus Repository conf in pom.xml.

The screenshot shows the Visual Studio Code interface with the pom.xml file open. The code editor displays the XML configuration for the Maven project, specifically focusing on the distributionManagement section which defines a snapshot repository for the Assignment 2 module. The configuration includes the group ID, artifact ID, version, and URL for the repository.

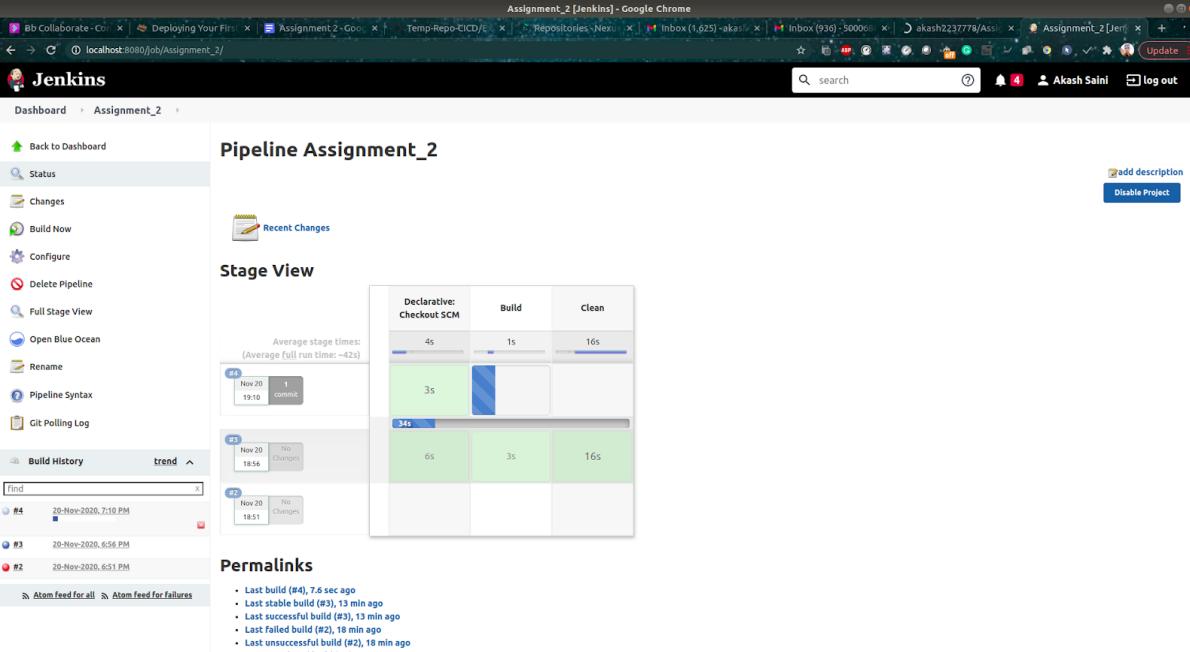
```
<distributionManagement>
    <snapshotRepository>
        <id>Assignment 2</id>
        <name>Assignment 2</name>
        <url>http://localhost:8081/repository/Assignment 2</url>
    </snapshotRepository>
</distributionManagement>
```

Step 13: Add Deploy stage to Jenkins File and push it to GitHub.

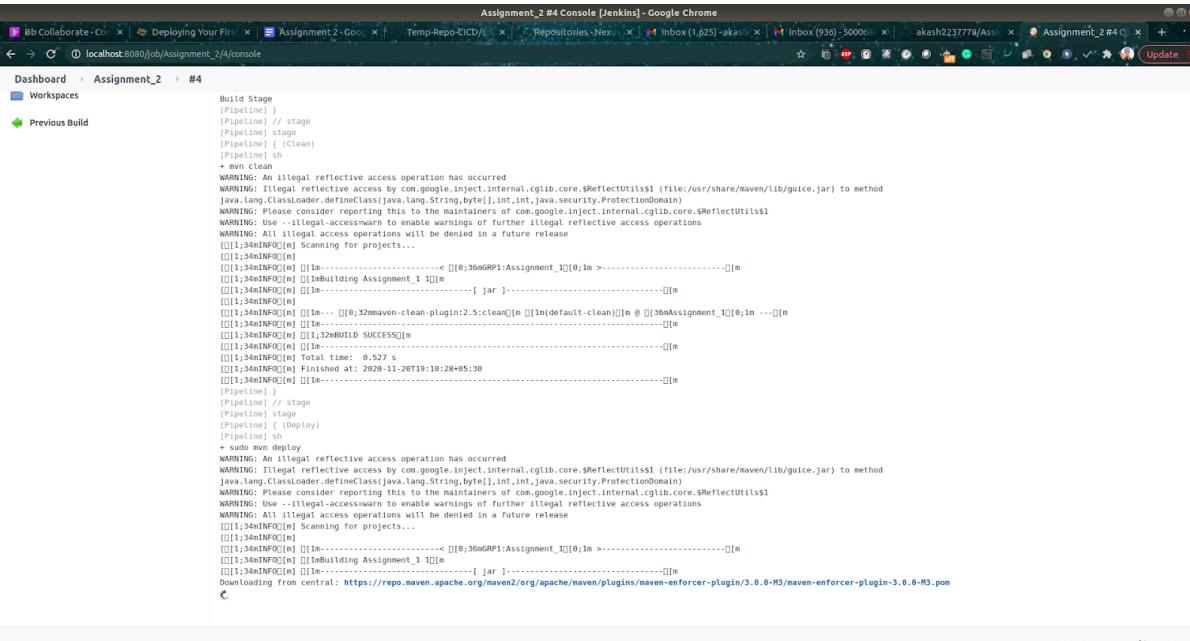
The screenshot shows the Visual Studio Code interface with the Jenkinsfile open. The Jenkins pipeline script is displayed, showing the addition of a 'Deploy' stage. This stage uses the 'sh' step to run the 'sudo mvn deploy' command, indicating the deployment of the build artifacts to a remote repository.

```
stage('Deploy'){
    steps{
        sh 'sudo mvn deploy'
    }
}
```

Step 14: Build will automatically triggered.



The screenshot shows the Jenkins Pipeline Assignment_2 dashboard. On the left, there's a sidebar with various options like Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Open Blue Ocean, Rename, Pipeline Syntax, and Git Polling Log. Below that is a Build History section with three builds listed: #4 (20-Nov-2020, 7:10 PM), #3 (20-Nov-2020, 6:56 PM), and #2 (20-Nov-2020, 6:51 PM). A Permalinks section lists recent builds. The main area is titled "Stage View" and displays a timeline of stages: Declarative: Checkout SCM (4s), Build (1s), and Clean (16s). The total average stage time is 42s. Below the stage view is a "Recent Changes" section showing two commits from Nov 20: "No changes" at 18:56 and "No changes" at 18:51.



The screenshot shows the Jenkins Assignment_2 #4 Console. The logs display the build process for build #4. It starts with a mvn clean command, followed by a warning about illegal reflective access operations. Then it moves to the build stage, where it runs mvn -Dmaven-clean-plugin:2.5:clean and successfully completes the build. The total time for the build is 0.527 s. Finally, it performs a sudo mvn deploy command, which also triggers a warning about illegal reflective access operations. The logs end with a download message from the central repository.

```
mvn clean
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by con.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of con.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use -illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[0;34m[INFO][n] Scanning for projects...
[0;34m[INFO][n] [1m-----< @0:36mGRP1:Assignment_1[@:1m >-----]
[0;34m[INFO][n] [1m-----[ jar ]-----]
[0;34m[INFO][n] [1m-----@0:32m[maven-clean-plugin:2.5:clean@m @ @36mAssignment_1@:1m <-----@n
[0;34m[INFO][n] [1m-----@0:32mBUILD SUCCESS@n-----@n
[0;34m[INFO][n] [1m-----@0:32mBUILD SUCCESS@n-----@n
[0;34m[INFO][n] Total time: 0.527 s
[0;34m[INFO][n] Finished at: 2020-11-28T19:10:28+05:30
[0;34m[INFO][n] [1m-----[ jar ]-----]
[0;34m[INFO][n] [1m-----[ Pipeline ]-----]
[0;34m[INFO][n] [1m-----[ Pipeline ] // stage
[0;34m[INFO][n] [1m-----[ Pipeline ] stage
[0;34m[INFO][n] [1m-----[ Pipeline ] { (Deploy)
[0;34m[INFO][n] [1m-----[ Pipeline ] }
+ sudo mvn deploy
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by con.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of con.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use -illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[0;34m[INFO][n] Scanning for projects...
[0;34m[INFO][n] [1m-----< @0:36mGRP1:Assignment_1[@:1m >-----]
[0;34m[INFO][n] [1m-----[ jar ]-----]
[0;34m[INFO][n] [1m-----@0:32m[maven-enforcer-plugin:3.0.0-M3:maven-enforcer-plugin@:1m <-----@n
[0;34m[INFO][n] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-plugin/3.0.0-M3/maven-enforcer-plugin-3.0.0-M3.pom
[0;34m[INFO][n]
```

Step 15: Package deployed at Nexus Repository.

