



# **CONTINUOUS INTEGRATION** **&** **CONTINUOUS DELIVERY**

## **LAB REPORT**

<b>SUBMITTED BY</b>	<b>SUBMITTED TO</b>
ABHIJEET PRASAD 500069170 R171218003	DR. HITESH KUMAR SHARMA
CSE - DEVOPS	DEPT. OF CYBERNETICS (SoCS, UPES)

# TABLE OF CONTENTS

---

S. NO.	TOPIC	PAGE NO.
1.	Installation of Jenkins and executing a simple job.	
2.	Jenkins integration with Github	
3.	Jenkins integration with Github and Maven	
4.	Static code analysis using SonarQube	
5.	Jenkins integration with SonarQube	
6.	Creating a pipeline with Jenkinsfile	
7.	Creating a pipeline with BlueOcean plugin	
8.	Implementing Master/Slave architecture in Jenkins	
9.	Uploading artifacts on Nexus server using CLI	
10.	Jenkins integration with Nexus	

## OFFICIAL DOCUMENTATIONS

---

S. NO.	TOOL/PLUGIN	LINK
1.	JENKINS	<a href="https://www.jenkins.io/doc/">https://www.jenkins.io/doc/</a>
2.	JENKINS + GITHUB	
3.	SONARQUBE	
4.	JENKINS + BLUEOCEAN	
5.	NEXUS	

# 01.Installation of Jenkins and executing a simple Job

## 1. INSTALLATION

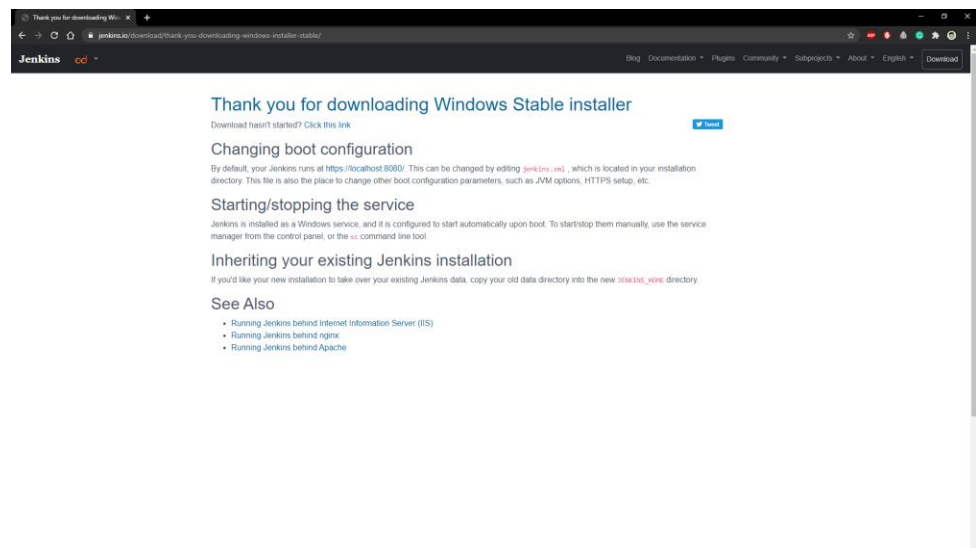
### 1.1. Windows

**To install Jenkins as a local service on your Windows machine, follow along the following steps.**

#### **Requirements: JDK**

#### **1.1.1. Download Jenkins**

Head to <https://www.jenkins.io/download/> and choose Windows as your OS, and download Jenkins Stable Installer.



#### **1.1.2. Install Jenkins**

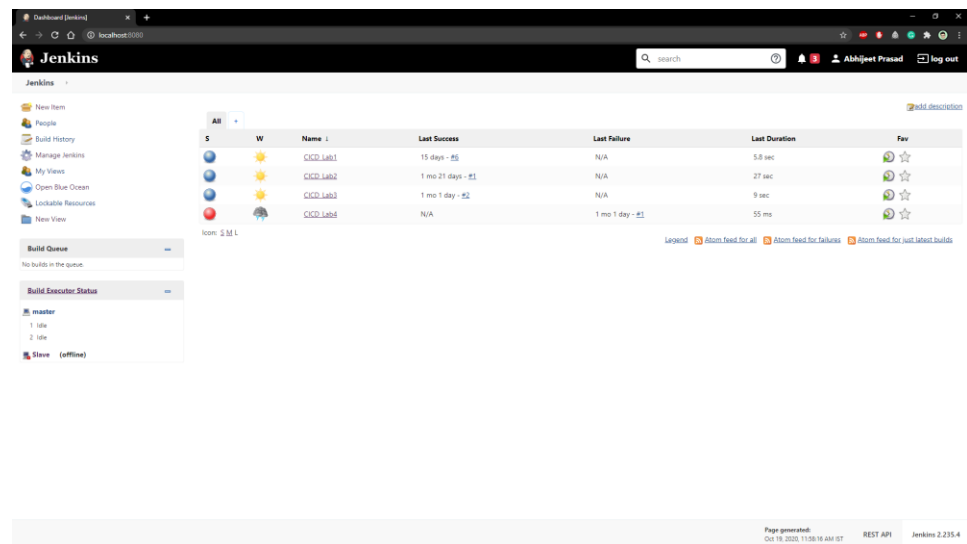
Run the downloaded installer and follow the on-screen instructions. You will see a success page once it is done. Move to the next step.

#### **1.1.3. Test run**

Head to you web browser and navigate to <http://localhost:8080/> , you will be redirected to Jenkins

login page. It'll prompt you with the location of the default password. Find the file and paste your default password.

**DONE! We have successfully installed Jenkins on our Windows Machine.**



## 1.2. Linux/Debian

**To install Jenkins service as localhost on your Debian based Linux Distribution, follow along the following steps.**

**Requirements: JDK**

### 1.2.1. Install Jenkins

Run the following commands to install Jenkins on Debian/Ubuntu Machine.

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \
/etc/apt/sources.list.d/jenkins.list'
```

```
sudo apt-get update
```

```
sudo apt-get install Jenkins
```

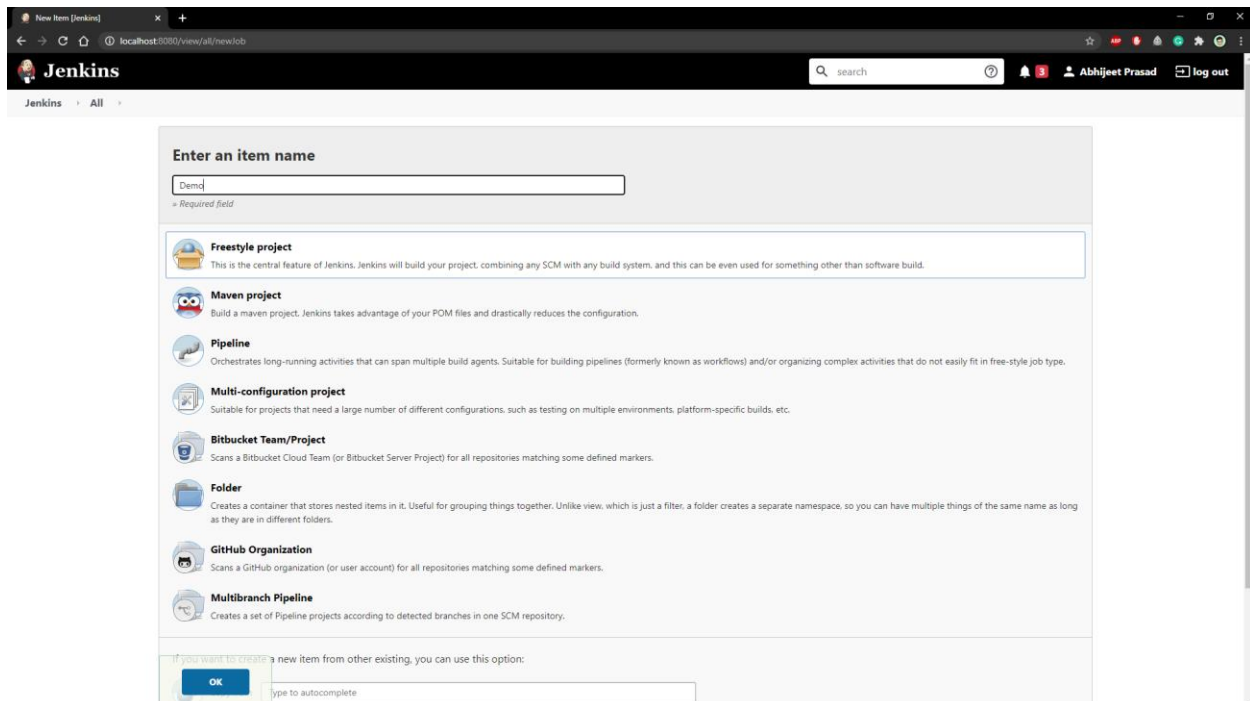
## 1.2.2. Run Jenkins

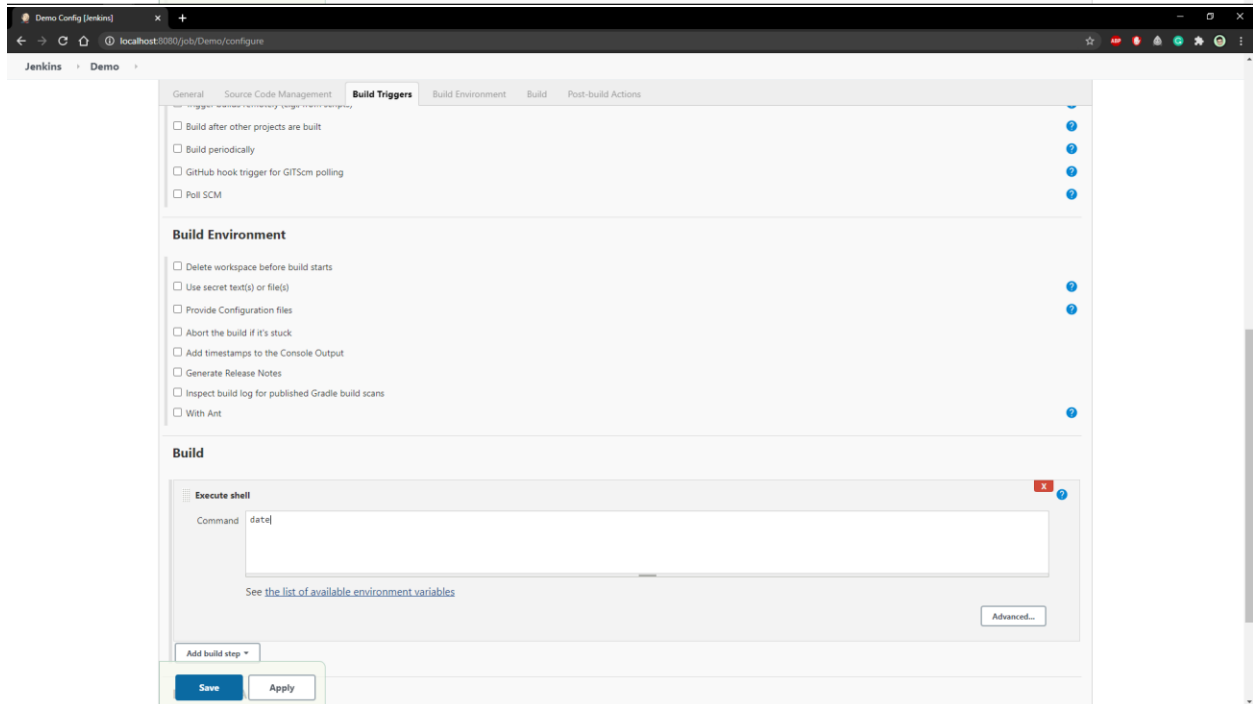
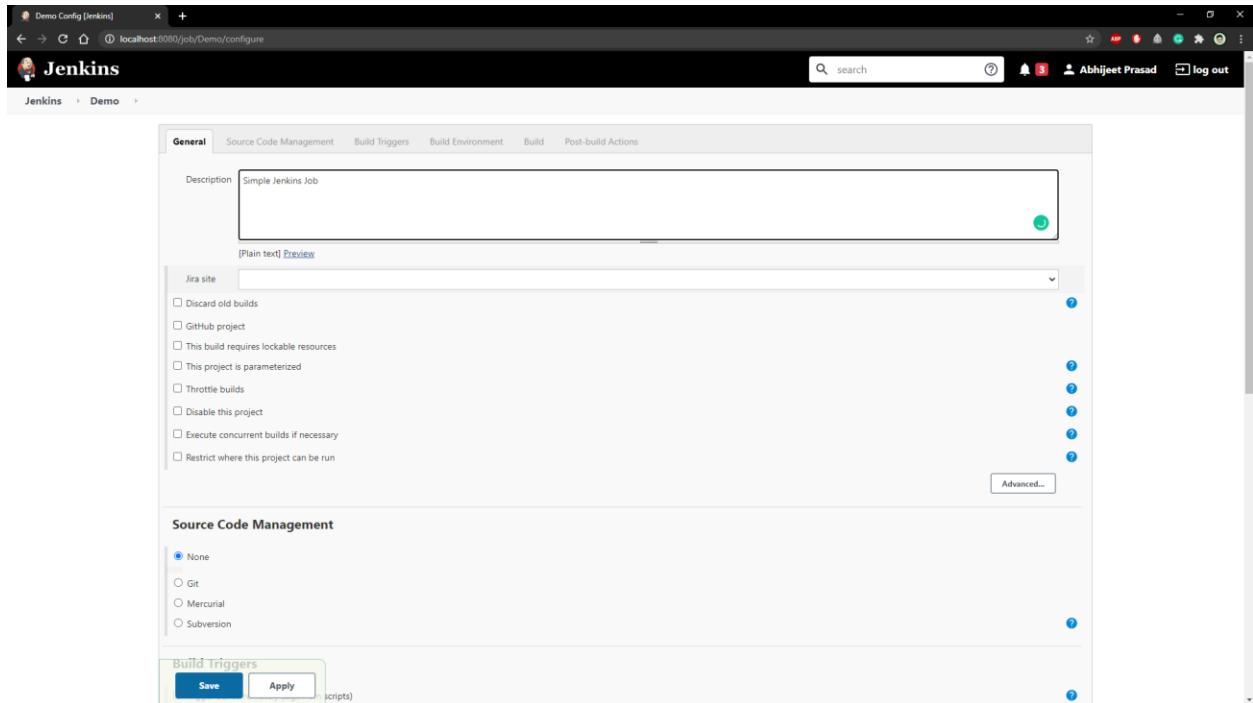
Start your Jenkins service with the following command.

```
sudo systemctl start jenkins
```

## 2. EXECUTING A SIMPLE JOB

**To execute a simple job in Jenkins, we need a Maven Project that can perform Maven Goals. We will be executing a simple build through Jenkins.**





Demo Config [Jenkins]

localhost:8080/job/Demo/configure

Jenkins

Demo

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

☐ Use secret text(s) or file(s)

☐ Provide Configuration files

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ Generate Release Notes

☐ Inspect build log for published Gradle build scans

☐ With Ant

Build

☐ Execute shell

Command

date

See [the list of available environment variables](#)

Advanced...

Add build step

Post-build Actions

Add post-build action

Save

Apply

Page generated:

Oct 23, 2020, 10:55:05 AM IST

REST API

Jenkins 2.235.4

Demo #2 Console [Jenkins]

localhost:8080/job/Demo/2/console

Jenkins

Demo

#2

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build #2

Open Blue Ocean

Previous Build

Next Build

Console Output

Started by user [Abhijeet Prasad](#)

Running as SYSTEM

Building on master in workspace C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\Demo

[Demo] \$ cmd /c call C:\Windows\TEMP\Jenkins1110091340776410111.bat

C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\Demo:date

The current date is: 23-10-2020

Enter the new date: (dd-mm-yy)

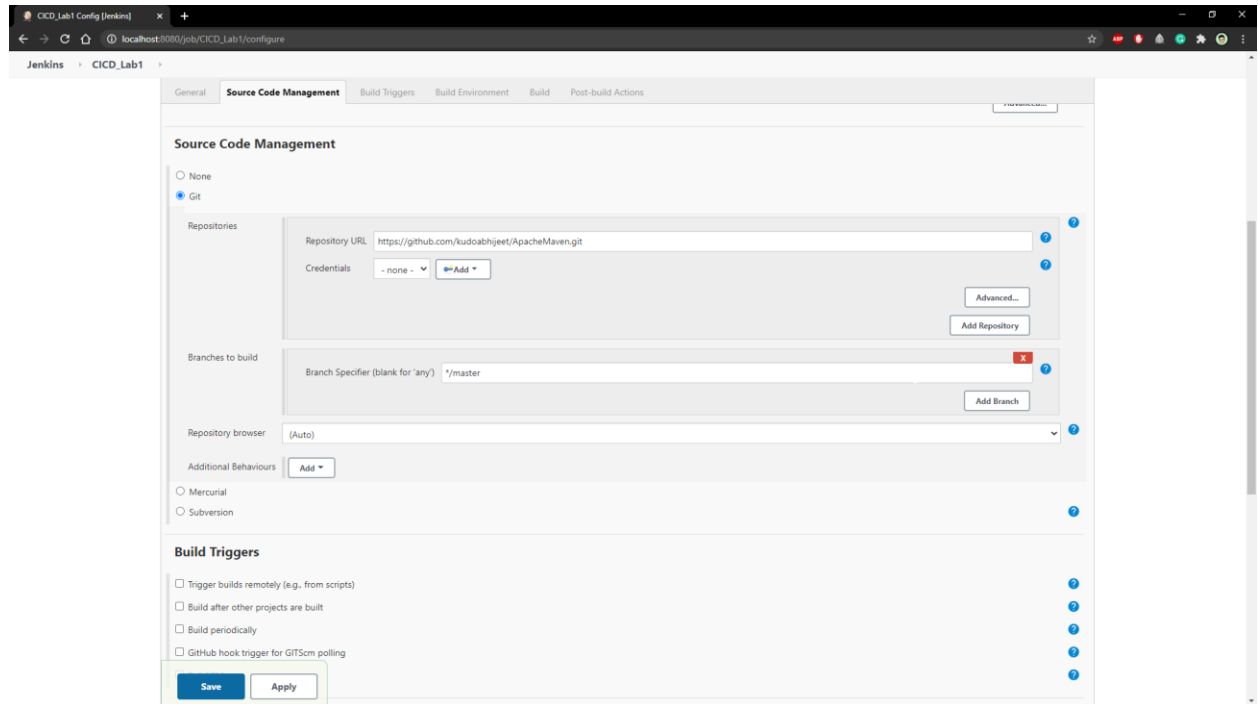
C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\Demo:exit 1

Build step 'Execute Windows batch command' marked build as failure

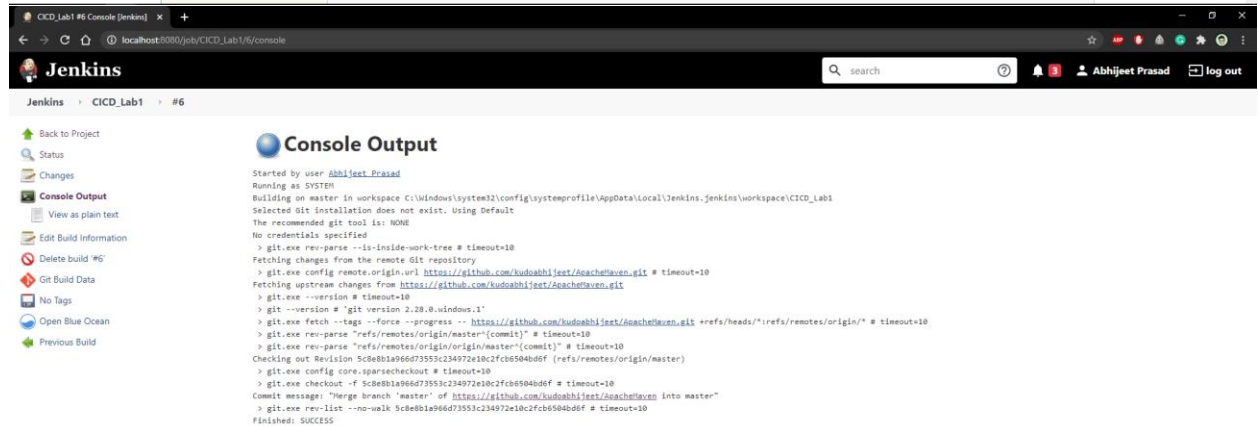
Finished: FAILURE



## 02.Jenkins integration with Github

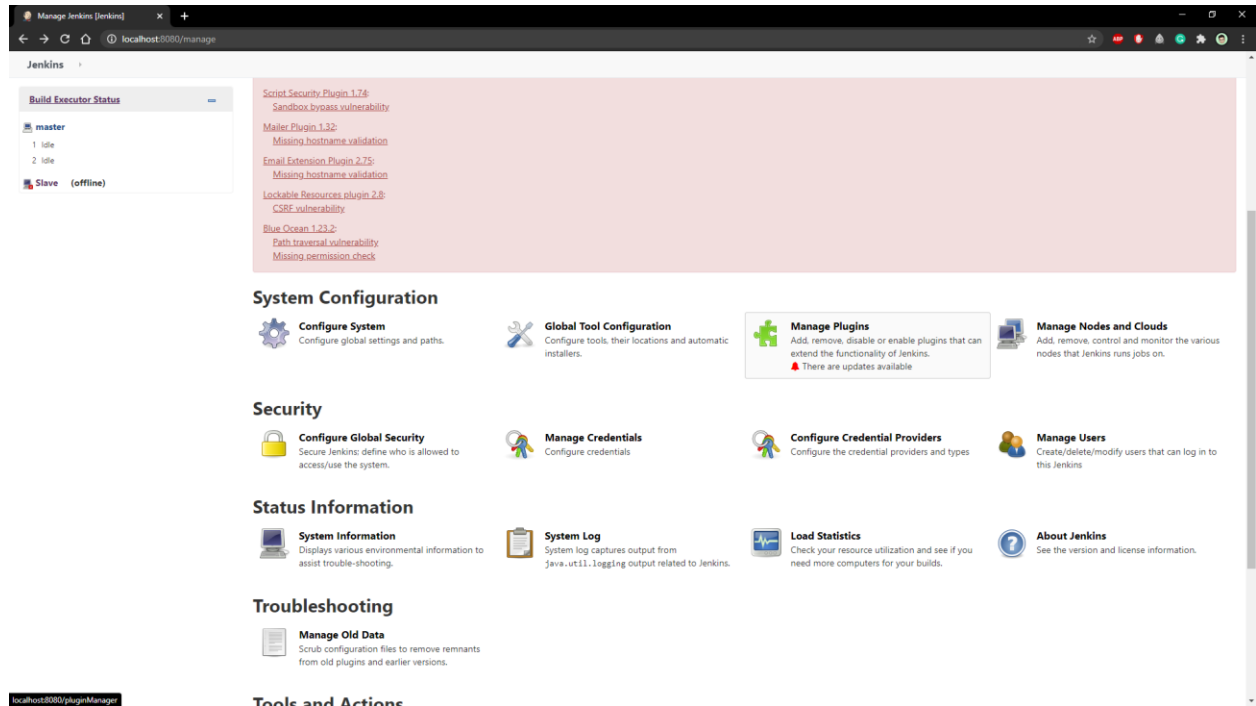


The screenshot shows the Jenkins configuration page for 'CICD\_Lab1'. The 'Source Code Management' tab is active. Under 'Source Code Management', 'Git' is selected. The 'Repositories' section shows a repository URL of 'https://github.com/kudoabhijet/ApacheMaven.git' and a dropdown for 'Credentials' set to '- none'. The 'Branches to build' section has a 'Branch Specifier (blank for \'any\')' set to '\*/master'. The 'Repository browser' is set to '(Auto)'. The 'Additional Behaviours' section has an 'Add' button. The 'Build Triggers' section has four checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', and 'GitHub hook trigger for GITScm polling', all of which are currently unchecked. At the bottom are 'Save' and 'Apply' buttons.

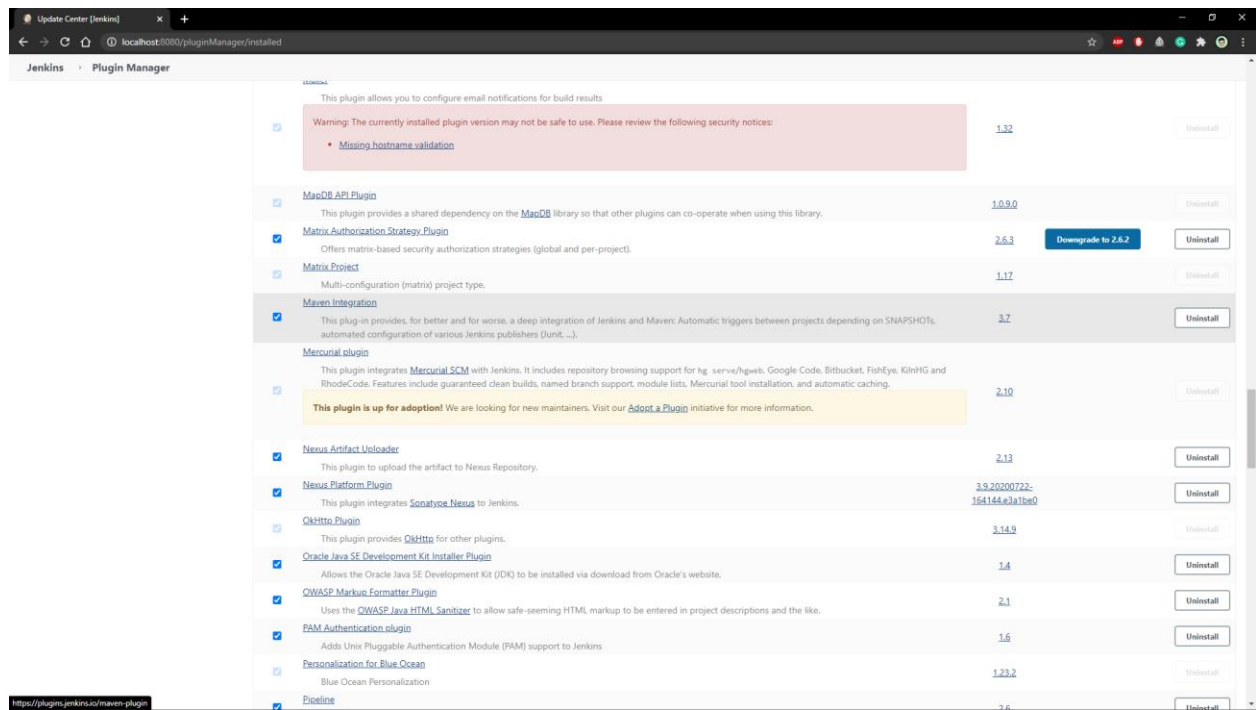


The screenshot shows the Jenkins console output for build #6 of 'CICD\_Lab1'. The output starts with 'Started by user Abhijeet Prasad' and 'Running as SYSTEM'. It shows the build directory 'C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\CICD\_Lab1'. The output then shows the execution of 'git.exe rev-parse --is-inside-work-tree # timeout=10', which returns 'true'. This is followed by 'git.exe fetch --tags --force --progress -- https://github.com/kudoabhijet/ApacheMaven.git # timeout=10', which returns 'true'. Then 'git.exe rev-parse "refs/remotes/origin/master" # timeout=10' returns 'true'. The output continues with 'git.exe rev-parse "refs/remotes/origin/master" # timeout=10', which returns 'true'. Then 'git.exe checkout -f 5c8e8b1a966d73553c234972e10c2fcb6504bd6f # timeout=10', which returns 'true'. The output ends with 'Commit message: "Merge branch \'master\' of https://github.com/kudoabhijet/ApacheMaven into master"' and 'git.exe rev-list --no-walk 5c8e8b1a966d73553c234972e10c2fcb6504bd6f # timeout=10', which returns 'true'. The final output is 'Finished: SUCCESS'.

## 03. Jenkins Integration with Github and Maven



The screenshot shows the Jenkins 'Manage' page at localhost:8080/manage. The left sidebar includes 'Build Executor Status' with a list of nodes (master, 1 idle, 2 idle) and a 'Slave (offline)' node. The main content area features a red warning banner for several plugins: Script Security Plugin 1.74 (Sandbox.bypass.vulnerability), Mailer Plugin 1.32 (Missing.hostname.validation), Email Extension Plugin 2.75 (Missing.hostname.validation), Lockable Resources plugin 2.0 (CSRF.vulnerability), and Blue Ocean 1.23.2 (Path.traversal.vulnerability, Missing.permission.check). Below the banner are sections for 'System Configuration' (Configure System, Global Tool Configuration, Manage Plugins, Manage Nodes and Clouds), 'Security' (Configure Global Security, Manage Credentials, Configure Credential Providers, Manage Users), 'Status Information' (System Information, System Log, Load Statistics, About Jenkins), and 'Troubleshooting' (Manage Old Data).



The screenshot shows the Jenkins 'Plugin Manager' page at localhost:8080/pluginManager/installed. It lists various plugins with their descriptions, versions, and actions. A red warning banner at the top states: 'Warning: The currently installed plugin version may not be safe to use. Please review the following security notices: Missing.hostname.validation'. The plugins listed include:

Plugin Name	Description	Version	Action
Script Security Plugin	This plugin allows you to configure email notifications for build results.	1.32	Uninstall
MapDB API Plugin	This plugin provides a shared dependency on the MapDB library so that other plugins can co-operate when using this library.	1.0.9.0	Uninstall
Matrix Authorization Strategy Plugin	Offers matrix-based security authorization strategies (global and per-project).	2.6.3	Downgrade to 2.6.2, Uninstall
Matrix Project	Multi-configuration (matrix) project type.	1.17	Uninstall
Maven Integration	This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (Unit,...).	3.7	Uninstall
Mercurial plugin	This plugin integrates Mercurial SCM with Jenkins. It includes repository browsing support for hg, serve/hgweb, Google Code, Bitbucket, FishEye, KintHG and RhodeCode. Features include guaranteed clean builds, named branch support, module lists, Mercurial tool installation, and automatic caching.	2.10	Uninstall
Nexus Artifact Uploader	This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	2.13	Uninstall
Nexus Platform Plugin	This plugin to upload the artifact to Nexus Repository.	3.9.20200722-164144e3a1be0	Uninstall
OWASP Markup Sanitizer	This plugin integrates Sonatype Nexus to Jenkins.	3.14.9	Uninstall
OWASP Markup Sanitizer	This plugin provides OWASP for other plugins.	1.4	Uninstall
OWASP Markup Sanitizer	Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	2.1	Uninstall
OWASP Markup Sanitizer	Uses the OWASP Java HTML Sanitizer to allow safe-seeming HTML markup to be entered in project descriptions and the like.	1.6	Uninstall
PAM Authentication plugin	Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	1.23.2	Uninstall
Personalization for Blue Ocean	Blue Ocean Personalization	2.6	Uninstall

CICD\_Lab2 Config [Jenkins]

localhost:8080/job/CICD\_Lab2/configure

Jenkins

CICD\_Lab2

General

Build Triggers

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/kudcabhijeet/EverythingCICD.git

Credentials

- none -

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\*/master

Add Branch

Repository browser

(Auto)

Additional Behaviours

Add

Script Path

Jenkinsfile

Lightweight checkout

☒

[Pipeline Syntax](#)

Save

Apply

Page generated:

Oct 23, 2020, 11:03:39 AM IST

REST API

Jenkins 2.235.4

## 04.Static Code Analysis using SonarQube

```
D:\college\CI-CD\git\Exp2\my-app>mvn clean install sonar:sonar -Dsonar.host.url=http://localhost:9000 -Dsonar.analysis.mode=publish
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.app:my-app >-----
[INFO] Building my-app 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ my-app ---
[INFO] Deleting D:\college\CI-CD\git\Exp2\my-app\target
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ my-app ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\college\CI-CD\git\Exp2\my-app\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ my-app ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\college\CI-CD\git\Exp2\my-app\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ my-app ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\college\CI-CD\git\Exp2\my-app\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ my-app ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\college\CI-CD\git\Exp2\my-app\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ my-app ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.mycompany.app.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.042 s - in com.mycompany.app.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ my-app ---
[INFO] Building jar: D:\college\CI-CD\git\Exp2\my-app\target\my-app-1.0-SNAPSHOT.jar
[INFO]
```

```
[INFO] ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=com.mycompany.app%3Amy-app
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AXTzbyKrS02Y9D3g-6_U
[INFO] Analysis total time: 22.599 s
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 37.719 s
[INFO] Finished at: 2020-10-04T17:16:21+05:30
[INFO]
[INFO] -----
```

Perspective: Overall Status

Sort by: Name

🔍 Search by project name or key

1 projects

🏠

☆ my-app Passed

Last analysis: October 4, 2020, 5:16 PM

0 A  
🐛 Bugs

0 A  
🔒 Vulnerabilities

– A  
🔍 Hotspots Reviewed

2 A  
🧐 Code Smells

0.0%  
📊 Coverage

0.0%  
🔄 Duplications

73 XS  
XML, Java

1 of 1 shown

## 05.Jenkins Integration with SonarQube

[Back to Dashboard](#)  
[Manage Jenkins](#)  
[Update Center](#)

Updates Available Installed Advanced

Install	Name	Version	Released
<input type="checkbox"/>	<b>SonarQube Scanner</b> <div>External Site/Tool Integrations Build Reports</div> <p>This plugin allows an easy integration of <b>SonarQube</b>, the open source platform for Continuous Inspection of code quality.</p>	2.12	1 mo 6 days ago

Install without restart

Download now and install after restart

Update information obtained: 2 hr 54 min ago

Check now

### Build

Root POM

Goals and options

Advanced...

### Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Add post-build step ▼

### Build Settings

```
[INFO] ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=com.mycompany.app%3Amy-app
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AXTzfwTL502Y9D3g-6\_V
[INFO] Analysis total time: 20.345 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 27.829 s
[INFO] Finished at: 2020-10-04T17:33:43+05:30
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\Users\ARJUN\.jenkins\workspace\CI-CD_check\my-app\pom.xml to com.mycompany.app/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.pom
[JENKINS] Archiving C:\Users\ARJUN\.jenkins\workspace\CI-CD_check\my-app\target\my-app-1.0-SNAPSHOT.jar to com.mycompany.app/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

Perspective: Overall Status

Sort by: Name

Search by project name or key

1 projects

☆ my-app Passed

Last analysis: October 4, 2020, 5:33 PM

0 A  
Bugs

0 A  
Vulnerabilities

– A  
Hotspots Reviewed

2 A  
Code Smells

0.0%  
Coverage

0.0%  
Duplications

83 XS  
XML, Java

1 of 1 shown

## 06. Creating a Pipeline with Jenkinsfile

The image shows two screenshots related to creating a Jenkins pipeline using a Jenkinsfile.

The top screenshot is the Jenkins 'Pipeline' configuration page for a job named 'CICD\_Lab2'. The 'Definition' is set to 'Pipeline script from SCM'. The 'SCM' is 'Git'. The 'Repository URL' is 'https://github.com/kudoabhjeet/EverythingCICD.git'. The 'Credentials' are set to 'none'. The 'Branches to build' are specified as '\*/master'. The 'Repository browser' is '(Auto)'. The 'Script Path' is 'Jenkinsfile'. The 'Lightweight checkout' option is checked. There are 'Save' and 'Apply' buttons at the bottom.

The bottom screenshot is a GitHub repository view for 'kudoabhjeet/EverythingCICD'. It shows the 'Jenkinsfile' file selected. The file content is as follows:

```
1 pipeline{
2   agent any
3   stages{
4     stage('Build'){
5       steps{
6         echo 'Build Stage'
7       }
8     }
9   }
10  stage('Test'){
11    steps{
12      echo 'Test Stage'
13    }
14  }
15  stage('Clean'){
16    steps{
17      echo 'Clean Stage'
18    }
19  }
20 }
21 }
```

The footer of the GitHub page shows '© 2020 GitHub, Inc.' and various links like Terms, Privacy, Security, Status, Help, Contact GitHub, Pricing, API, Training, Blog, and About.

EverythingCICD/jenkinsfile at m...CICD\_Lab3 [jenkins]

localhost:8080/job/CICD\_Lab3/

search

Abhijeet Prasadlog out

JenkinsCICD\_Lab3

Back to Dashboard

Status

Changes

Build Now

Delete Pipeline

Configure

Full Stage View

Open Blue Ocean

Rename

Pipeline Syntax

Pipeline CICD\_Lab3

add description

Disable Project

Recent Changes

Average stage times:  
(Average full run time: ~11s)

Sep 18 11:23

No Changes

Sep 18 08:09

No Changes

Declarative: Checkout SCM	Build	Test	Clean
1s	253ms	154ms	135ms
1s	231ms	118ms	123ms
1s	276ms	190ms	147ms

Build History

trend

find

Sep 18, 2020, 11:23 AM

Sep 18, 2020, 8:09 AM

Atom feed for allAtom feed for failures

Permalinks

Last build (#2) 1 mo 4 days ago

Last stable build (#2) 1 mo 4 days ago

Last successful build (#2) 1 mo 4 days ago

Last completed build (#2) 1 mo 4 days ago

Page generated:  
Oct 23, 2020, 11:18:51 AM IST

REST API

Jenkins 2.235.4



## 07. Creating a Pipeline with BlueOcean

The image displays two screenshots from a Jenkins Blue Ocean interface. The top screenshot shows the 'Pipeline' configuration page for a project named 'CICD\_Lab2'. The 'Definition' is set to 'Pipeline script from SCM'. The 'SCM' is 'Git'. The 'Repository URL' is 'https://github.com/kudoabhjeet/EverythingCICD.git'. The 'Branches to build' is set to '\*/master'. The 'Script Path' is 'Jenkinsfile'. The 'Lightweight checkout' option is checked. The bottom screenshot shows the GitHub repository 'kudoabhjeet/EverythingCICD' on the 'master' branch. It displays the 'Jenkinsfile' content, which is a Jenkins pipeline script. The script defines a pipeline with an agent 'any' and three stages: 'Build', 'Test', and 'Clean'. The 'Build' stage echoes 'Build Stage'. The 'Test' stage echoes 'mvn Test'. The 'Clean' stage echoes 'Clean Stage'.

**Jenkins Pipeline Configuration:**

- Definition: Pipeline script from SCM
- SCM: Git
- Repository URL: https://github.com/kudoabhjeet/EverythingCICD.git
- Credentials: - none -
- Branches to build: Branch Specifier (blank for 'any'): \*/master
- Repository browser: (Auto)
- Script Path: Jenkinsfile
- Lightweight checkout: ☒

**GitHub Repository View:**

EverythingCICD / Jenkinsfile

Latest commit 7917236 on Sep 18

We found a potential security vulnerability in one of your dependencies. Only the owner of this repository can see this message. See Dependabot alert

```
1 pipeline{
2   agent any
3   stages{
4     stage('Build'){
5       steps{
6         echo 'Build Stage'
7       }
8     }
9     stage('Test'){
10      steps{
11        echo 'mvn Test'
12      }
13    }
14    stage('Clean'){
15      steps{
16        echo 'Clean Stage'
17      }
18    }
19  }
20 }
21 }
```

EverythingCICD/jenkinsfile at m... x jenkins / CKD\_Lab3 / #2 x +

localhost:8080/blue/organizations/jenkins/CKD\_Lab3/detail/CKD\_Lab3/2/pipeline

✓ CKD\_Lab3 < 2 Pipeline Changes Tests Artifacts ↺ ⚙️ 📄 🗑️ Logout

Branch: — 9s No changes  
Commit: — a month ago Started by user Abhijeet Prasad

Start Build Test Clean End

Clean - <1s [Restart Clean](#) 📄 🗑️

✓ Clean Stage — Print Message <1s

EverythingCICD/jenkinsfile at m... x Update Center [jenkins] x +

localhost:8080/pluginManager/

Jenkins Jenkins Plugin Manager

Back to Dashboard Manage Jenkins

Search blueo

Updates Available Installed Advanced

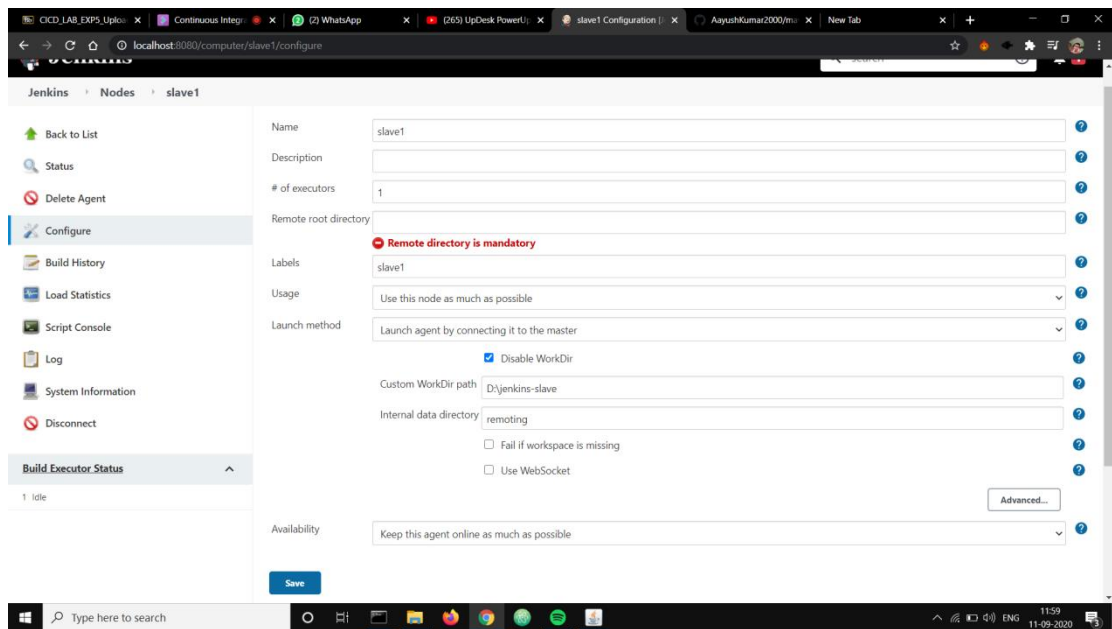
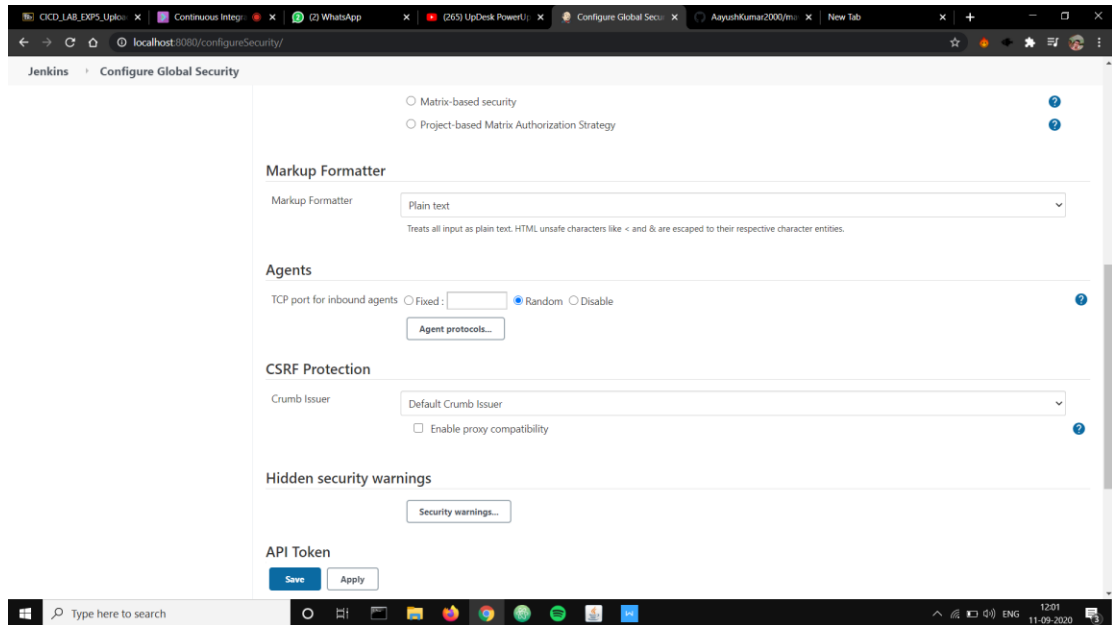
Install	Name	Version	Released	Installed
<input type="checkbox"/>	<a href="#">Bitbucket Pipeline for Blue Ocean</a> BlueOcean Bitbucket pipeline creator	1.24.1	17 days ago	1.23.2
<a href="#">Blue Ocean</a>				
	<a href="#">External Site/Tool Integrations</a>   <a href="#">User Interface</a>			
<input type="checkbox"/>	BlueOcean Aggregator	1.24.1	17 days ago	1.23.2
Applying this update will address security vulnerabilities in the currently installed version.				
<input type="checkbox"/>	<a href="#">Blue Ocean Core JS</a> The Jenkins Plugins Parent POM Project	1.24.1	17 days ago	1.23.2
<input type="checkbox"/>	<a href="#">Blue Ocean Pipeline Editor</a> The Blue Ocean Pipeline Editor is the simplest way for anyone wanting to get started with creating Pipelines in Jenkins	1.24.1	17 days ago	1.23.2
<a href="#">Common API for Blue Ocean</a>				
	<a href="#">External Site/Tool Integrations</a>   <a href="#">User Interface</a>			
<input type="checkbox"/>	This plugin is a part of Blue Ocean UI	1.24.1	17 days ago	1.23.2
<input type="checkbox"/>	<a href="#">Config API for Blue Ocean</a> BlueOcean Analytics Tools plugin	1.24.1	17 days ago	1.23.2
<a href="#">Dashboard for Blue Ocean</a>				
	<a href="#">External Site/Tool Integrations</a>   <a href="#">User Interface</a>			
<input type="checkbox"/>	Blue Ocean Dashboard	1.24.1	17 days ago	1.23.2
<a href="#">Events API for Blue Ocean</a>				
	<a href="#">External Site/Tool Integrations</a>   <a href="#">User Interface</a>			
<input type="checkbox"/>	Blue Ocean Events	1.24.1	17 days ago	1.23.2

Get Dashboard for Blue Ocean

[Download now and install after restart](#) Update information obtained: 15 hr ago [Check now](#)

<https://plugins.jenkins.io/blueocean>


## 08.Implementing Master/Slave architecture in Jenkins





## Agent slave1

Connect agent to Jenkins one of these ways:

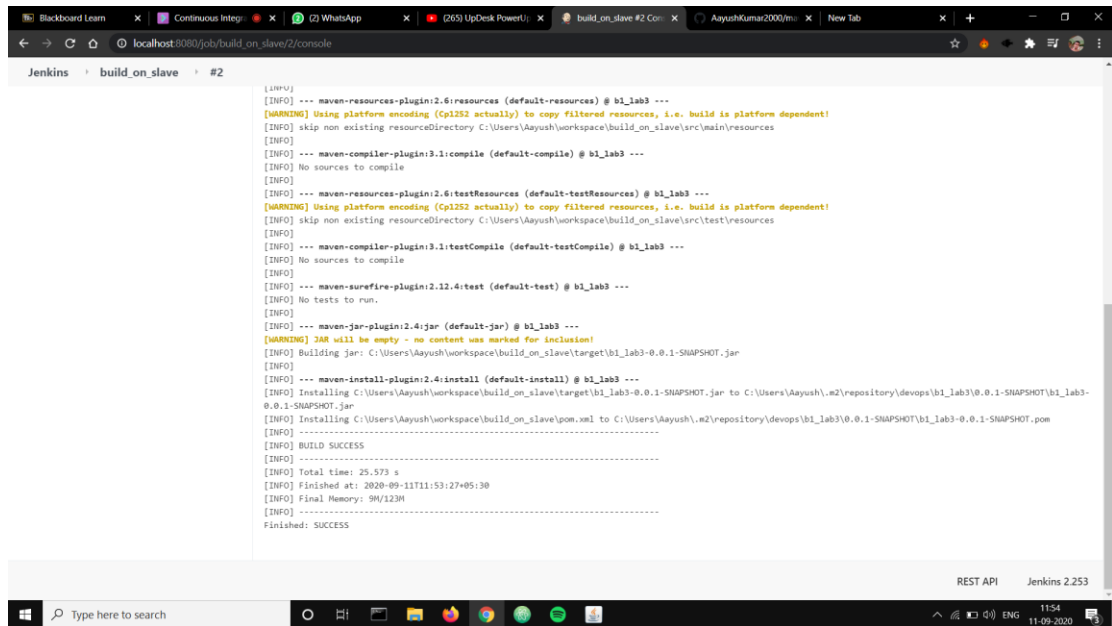
-  Launch Launch agent from browser

- Run from agent command line:

```
javaws http://localhost:8080/computer/slave1/slave-agent.jnlp
```

- Or if the agent is headless:

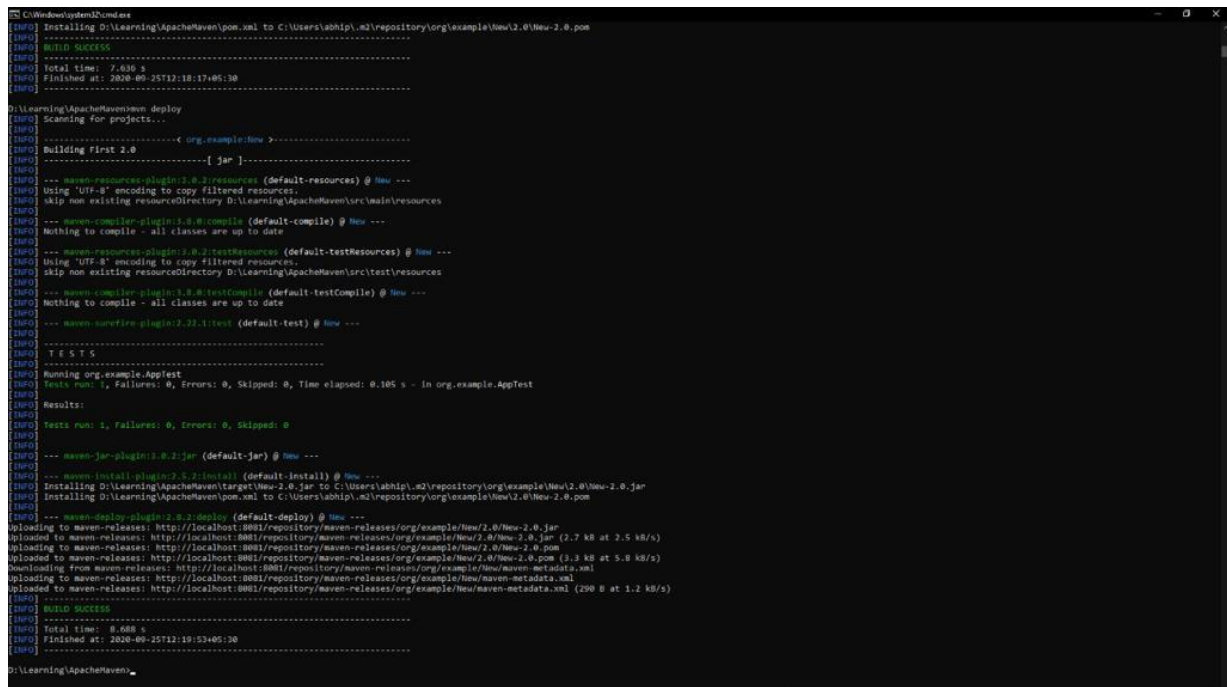
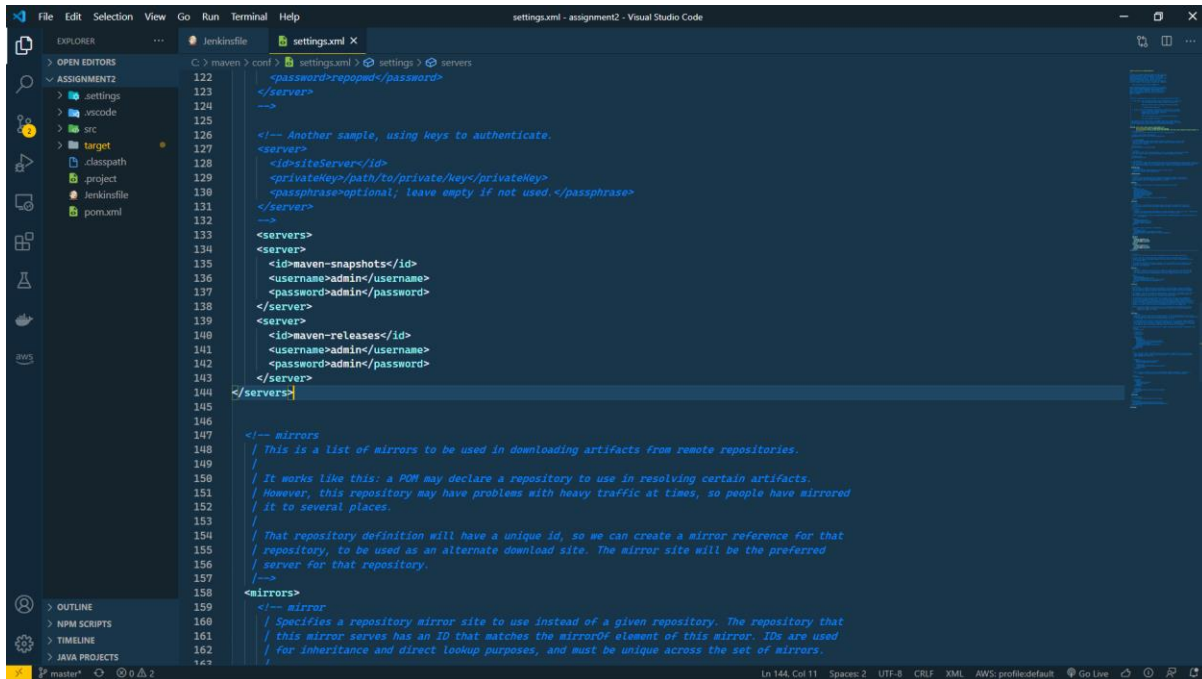
```
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave1/slave-agent.jnlp
```

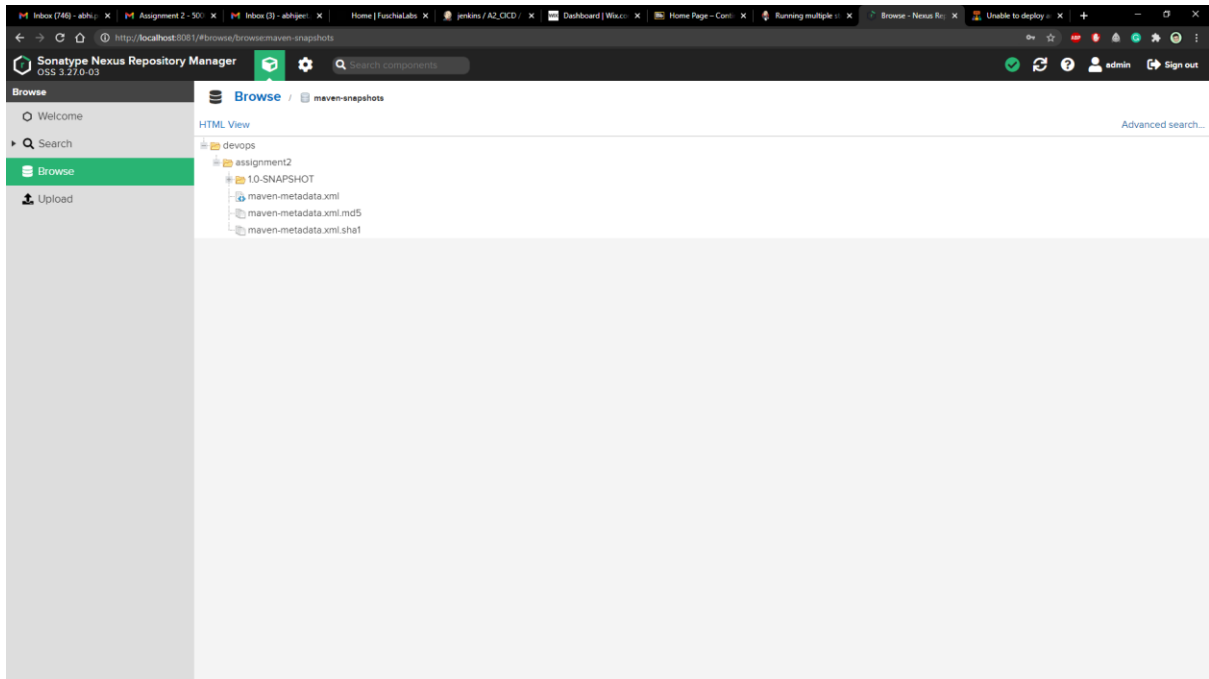


```
Jenkins build_on_slave #2  
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ b1_lab3 ---  
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!  
[INFO] skip non existing resourceDirectory C:\Users\Aayush\workspace\build_on_slave\src\main\resources  
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ b1_lab3 ---  
[INFO] No sources to compile  
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ b1_lab3 ---  
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!  
[INFO] skip non existing resourceDirectory C:\Users\Aayush\workspace\build_on_slave\src\test\resources  
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ b1_lab3 ---  
[INFO] No sources to compile  
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ b1_lab3 ---  
[INFO] No tests to run.  
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ b1_lab3 ---  
[WARNING] JAR will be empty - no content was marked for inclusion!  
[INFO] Building jar: C:\Users\Aayush\workspace\build_on_slave\target\b1_lab3-0.0.1-SNAPSHOT.jar  
[INFO] --- maven-install-plugin:2.4:install (default-install) @ b1_lab3 ---  
[INFO] Installing C:\Users\Aayush\workspace\build_on_slave\target\b1_lab3-0.0.1-SNAPSHOT.jar to C:\Users\Aayush\.m2\repository\devops\b1_lab3\0.0.1-SNAPSHOT\b1_lab3-0.0.1-SNAPSHOT.jar  
[INFO] Installing C:\Users\Aayush\workspace\build_on_slave\pom.xml to C:\Users\Aayush\.m2\repository\devops\b1_lab3\0.0.1-SNAPSHOT\b1_lab3-0.0.1-SNAPSHOT.pom  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 25.573 s  
[INFO] Finished at: 2020-09-11T11:53:27+05:30  
[INFO] Final Memory: 9M/123M  
[INFO] -----  
Finished: SUCCESS
```

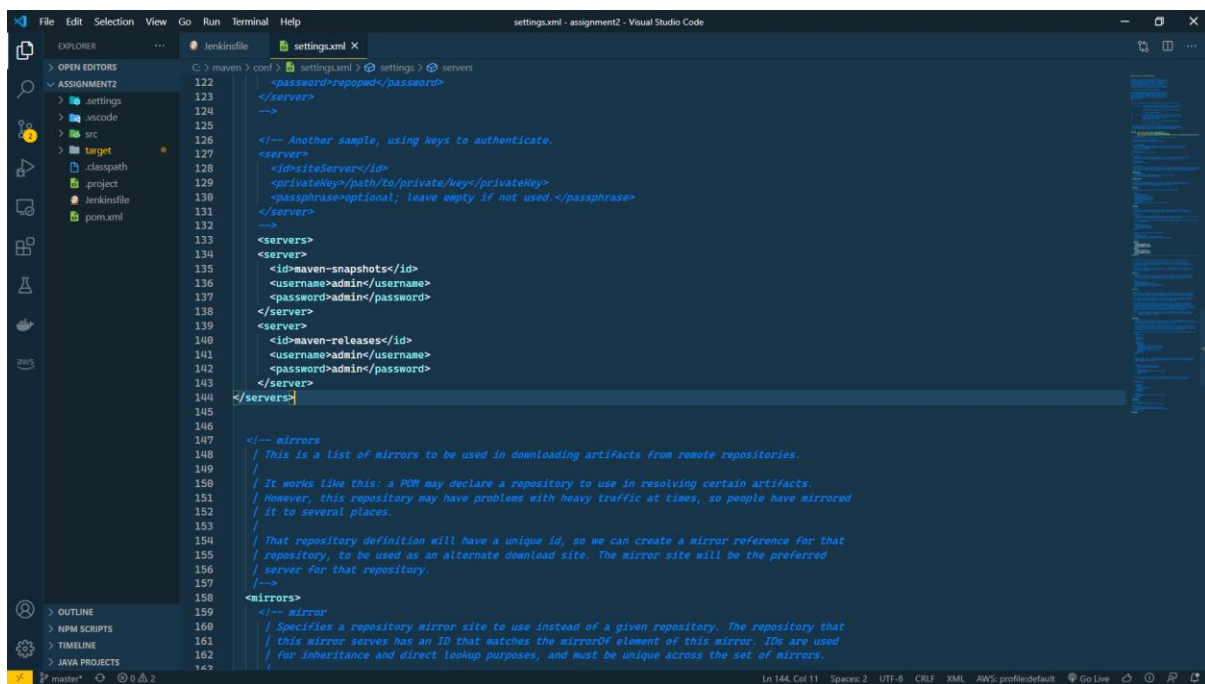
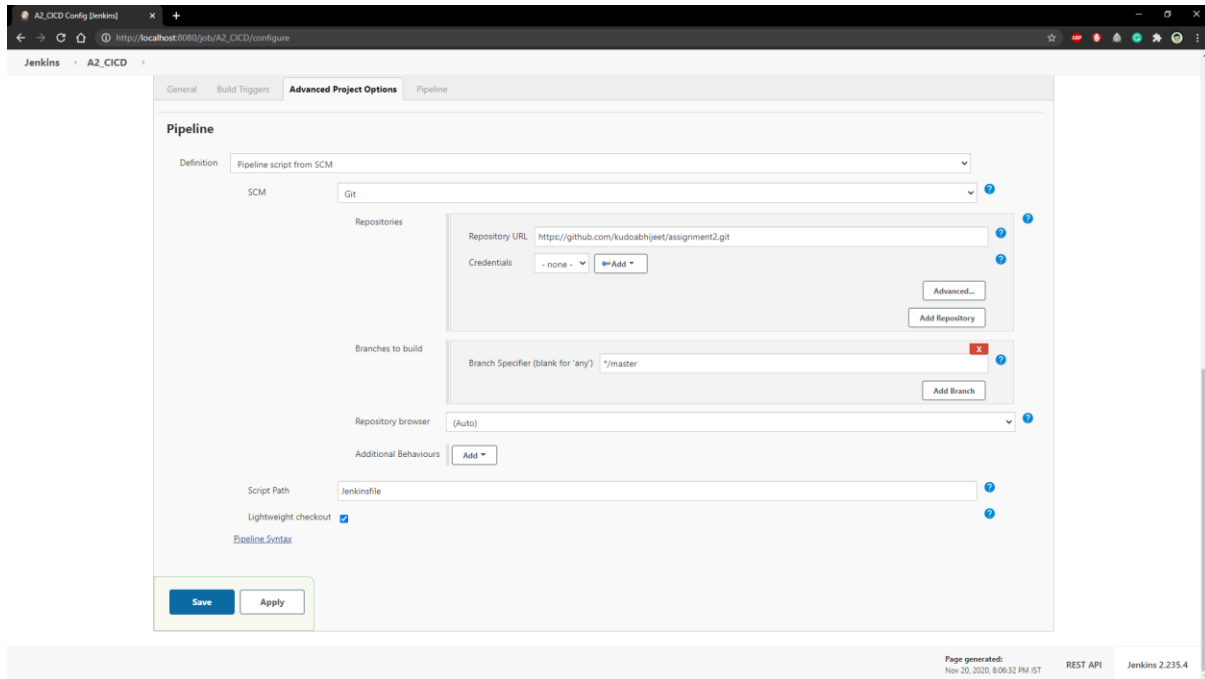
REST API Jenkins 2.253

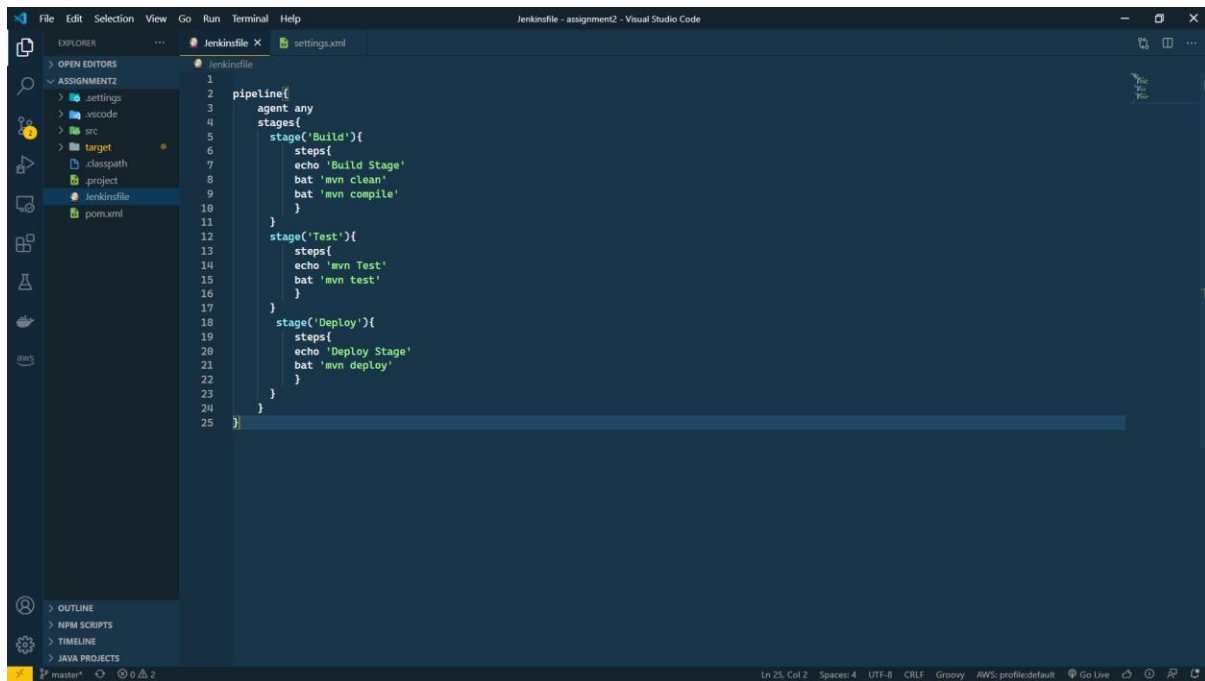
## 09.Uploading artifacts onto Nexus Server using CLI



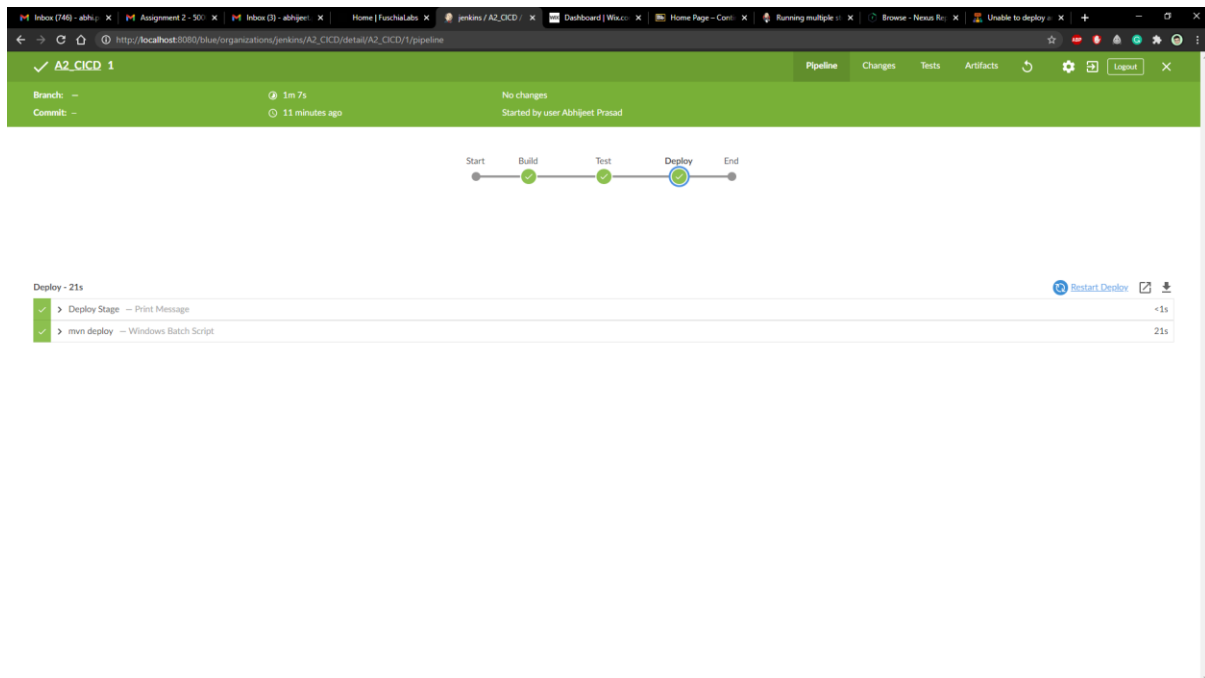


# 10. Jenkins Integration with Nexus





```
1 pipeline{
2   agent any
3   stages{
4     stage('Build'){
5       steps{
6         echo 'Build Stage'
7         bat 'mvn clean'
8         bat 'mvn compile'
9       }
10    }
11  }
12  stage('Test'){
13    steps{
14      echo 'mvn Test'
15      bat 'mvn test'
16    }
17  }
18  stage('Deploy'){
19    steps{
20      echo 'Deploy Stage'
21      bat 'mvn deploy'
22    }
23  }
24 }
25 }
```



✓ A2\_CICD 1

Branch: — 1m 7s No changes  
Commit: — 11 minutes ago Started by user Abhijeet Prasad

Start Build Test Deploy End

Deploy - 21s

- ✓ > Deploy Stage — Print Message <1s
- ✓ > mvn deploy — Windows Batch Script 21s



