

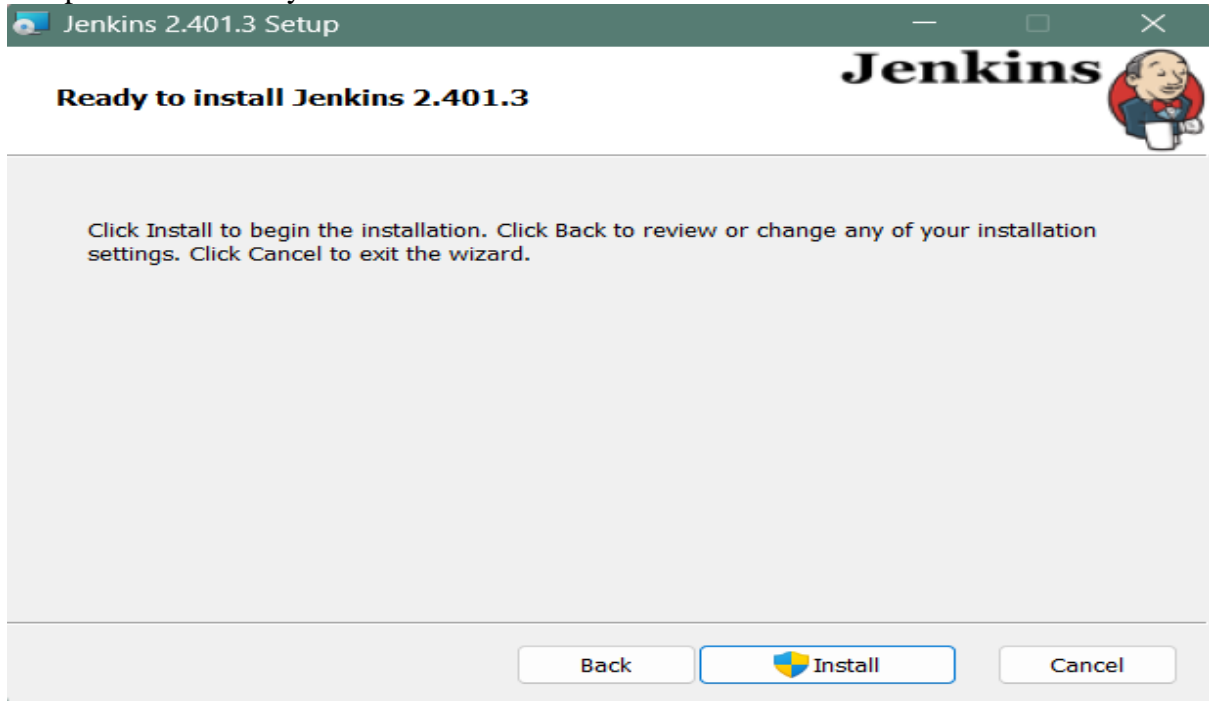
Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Assignment

1.Jenkins Introduction & steps of Jenkins Installation

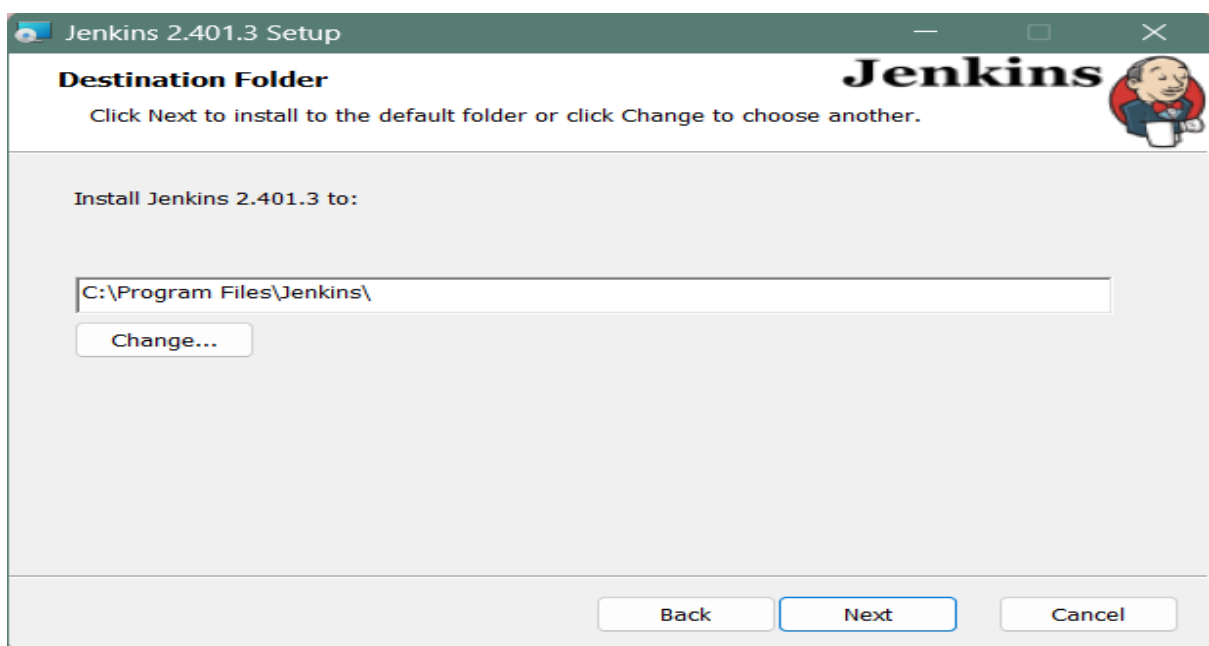
Step 1: Setup wizard:

On opening the Windows Installer, an **Installation Setup Wizard** appears, Click **Next** on the Setup Wizard to start your installation.



Step 2: Select destination folder:

Select the destination folder to store your Jenkins Installation and click **Next** to continue.

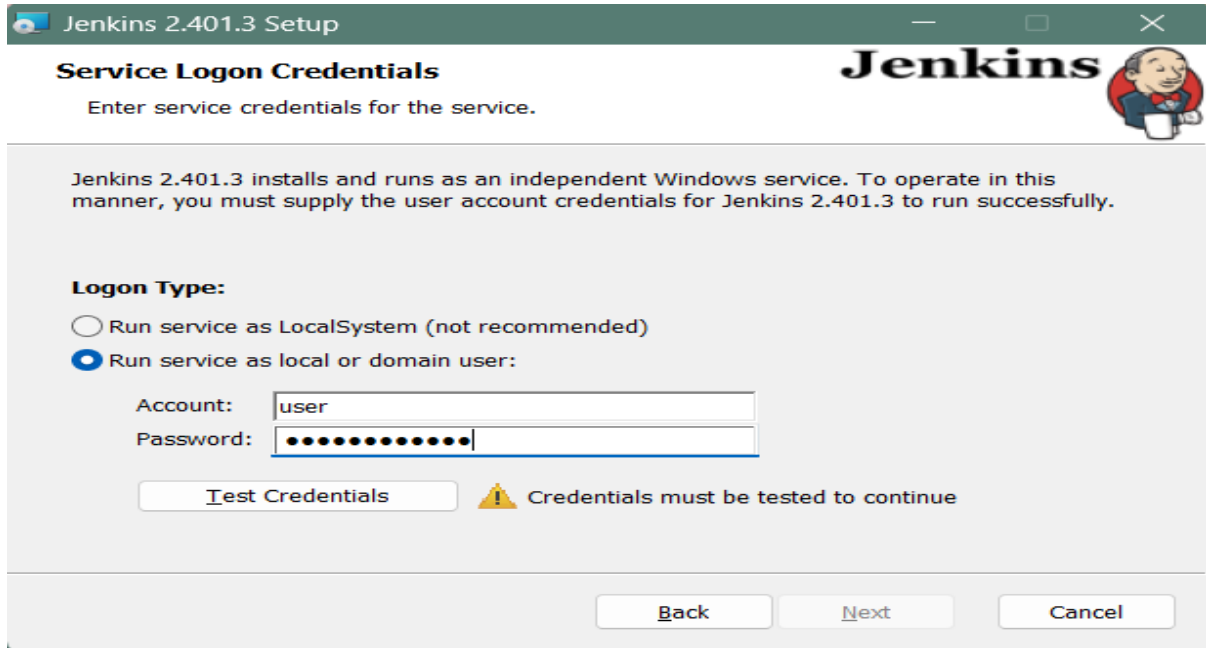


Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Step 3: Service logon credentials

When Installing Jenkins, it is recommended to install and run Jenkins as an independent windows service using a **local or domain user** as it is much safer than running Jenkins using **LocalSystem**(Windows equivalent of root) which will grant Jenkins full access to your machine and services.

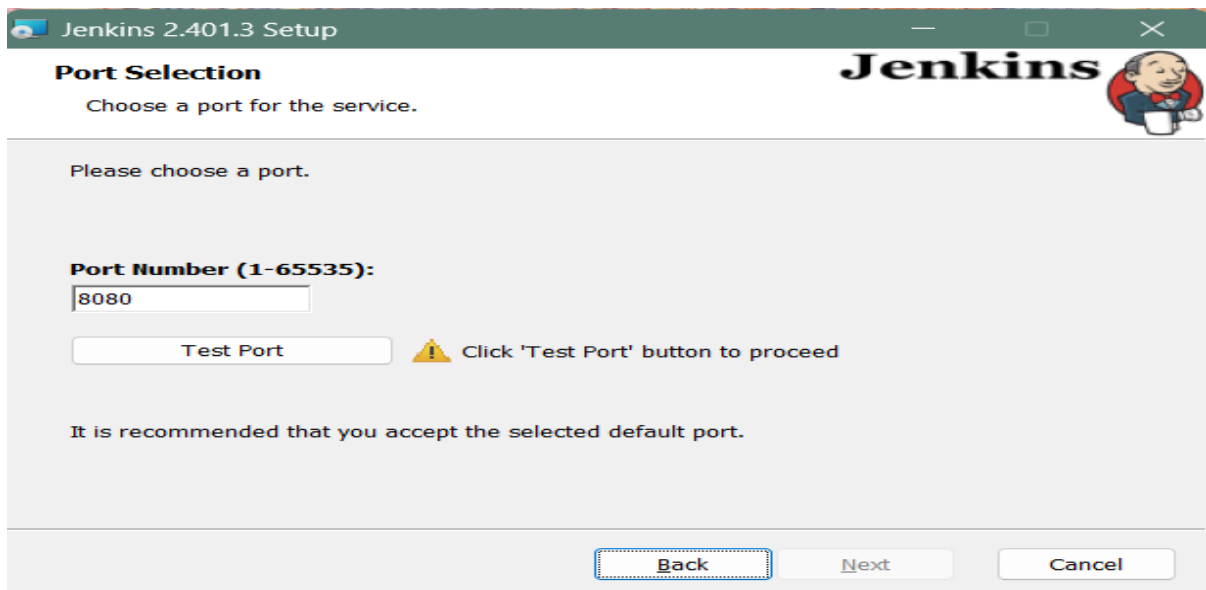
To run Jenkins service using a **local or domain user**, specify the domain user name and password with which you want to run Jenkins, click on **Test Credentials** to test your domain credentials and click on **Next**.



The screenshot shows the 'Service Logon Credentials' window of the Jenkins 2.401.3 Setup. The window title is 'Jenkins 2.401.3 Setup'. The header includes the Jenkins logo and the text 'Jenkins'. Below the header, it says 'Enter service credentials for the service.' The main content area contains the following text: 'Jenkins 2.401.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.401.3 to run successfully.' Under the heading 'Logon Type:', there are two radio buttons. The first is 'Run service as LocalSystem (not recommended)' and the second is 'Run service as local or domain user:', which is selected. Below the selected option, there are two text input fields: 'Account:' with the value 'user' and 'Password:' with masked characters. A 'Test Credentials' button is located below these fields. To the right of the button is a warning icon and the text 'Credentials must be tested to continue'. At the bottom of the window, there are three buttons: 'Back', 'Next', and 'Cancel'.

Step 4: Port selection

Specify the port on which Jenkins will be running, **Test Port** button to validate whether the specified port is free on your machine or not. Consequently, if the port is free, it will show a green tick mark as shown below, then click on **Next**.



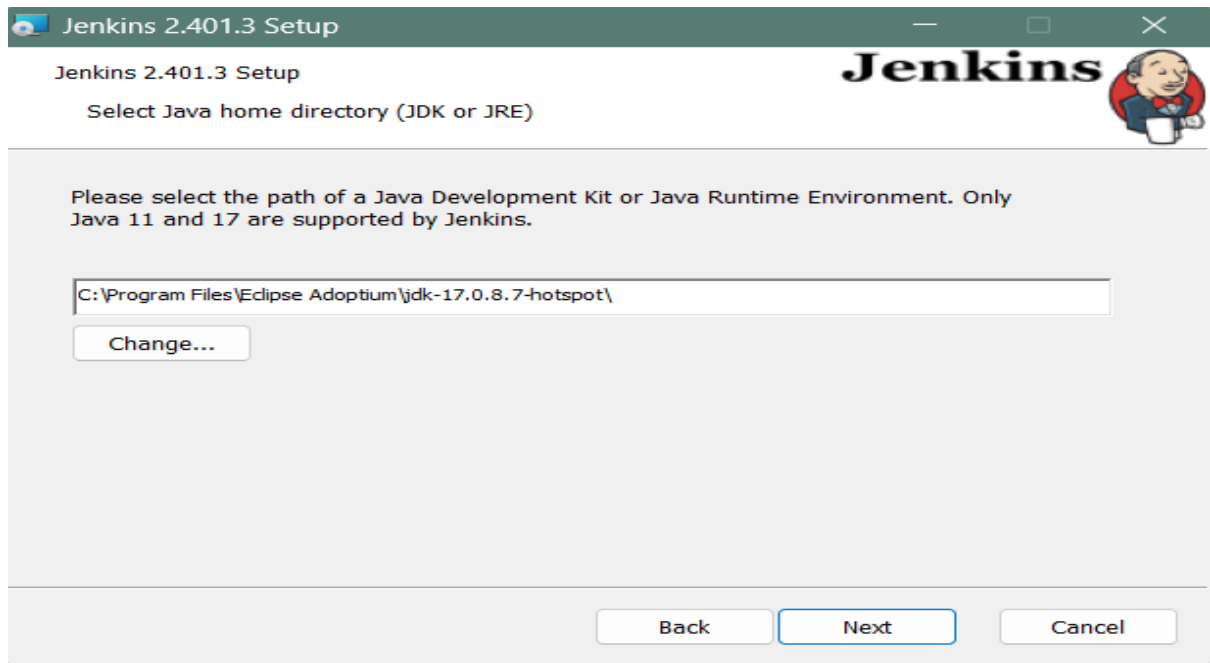
The screenshot shows the 'Port Selection' window of the Jenkins 2.401.3 Setup. The window title is 'Jenkins 2.401.3 Setup'. The header includes the Jenkins logo and the text 'Jenkins'. Below the header, it says 'Choose a port for the service.' The main content area contains the following text: 'Please choose a port.' Under the heading 'Port Number (1-65535):', there is a text input field with the value '8080'. A 'Test Port' button is located below the input field. To the right of the button is a warning icon and the text 'Click 'Test Port' button to proceed'. At the bottom of the window, there are three buttons: 'Back', 'Next', and 'Cancel'.

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Step 5: Select Java home directory

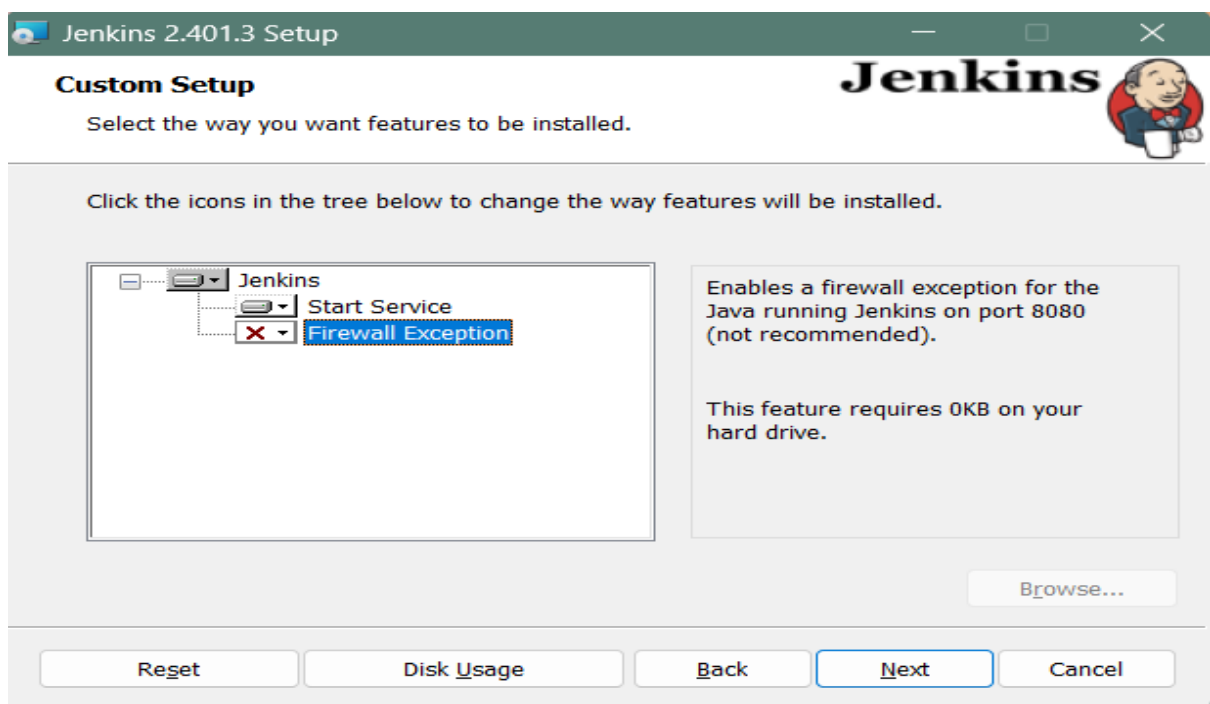
The installation process checks for Java on your machine and prefills the dialog with the Java home directory. If the needed Java version is not installed on your machine, you will be prompted to install it.

Once your Java home directory has been selected, click on **Next** to continue.



Step 6: Custom setup

Select other services that need to be installed with Jenkins and click on **Next**.



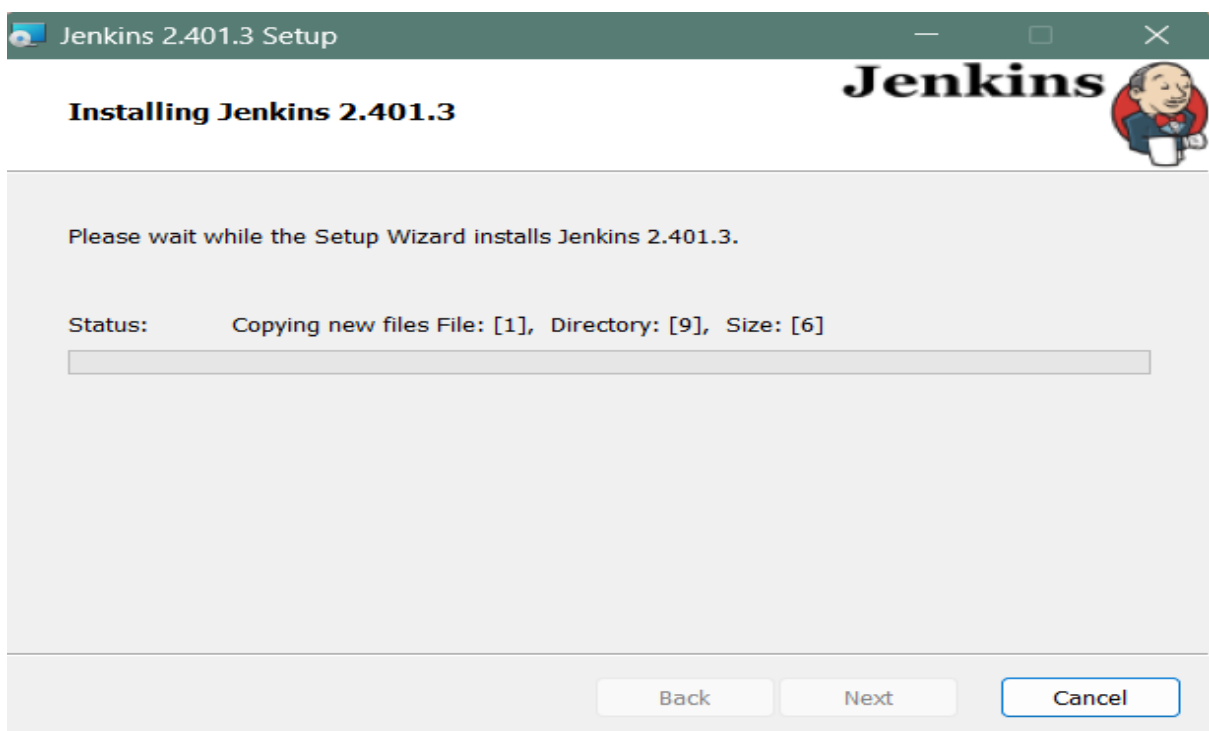
Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Step 7: Install Jenkins

Click on the **Install** button to start the installation of Jenkins.



Additionally, clicking on the **Install** button will show the progress bar of installation, as shown below:



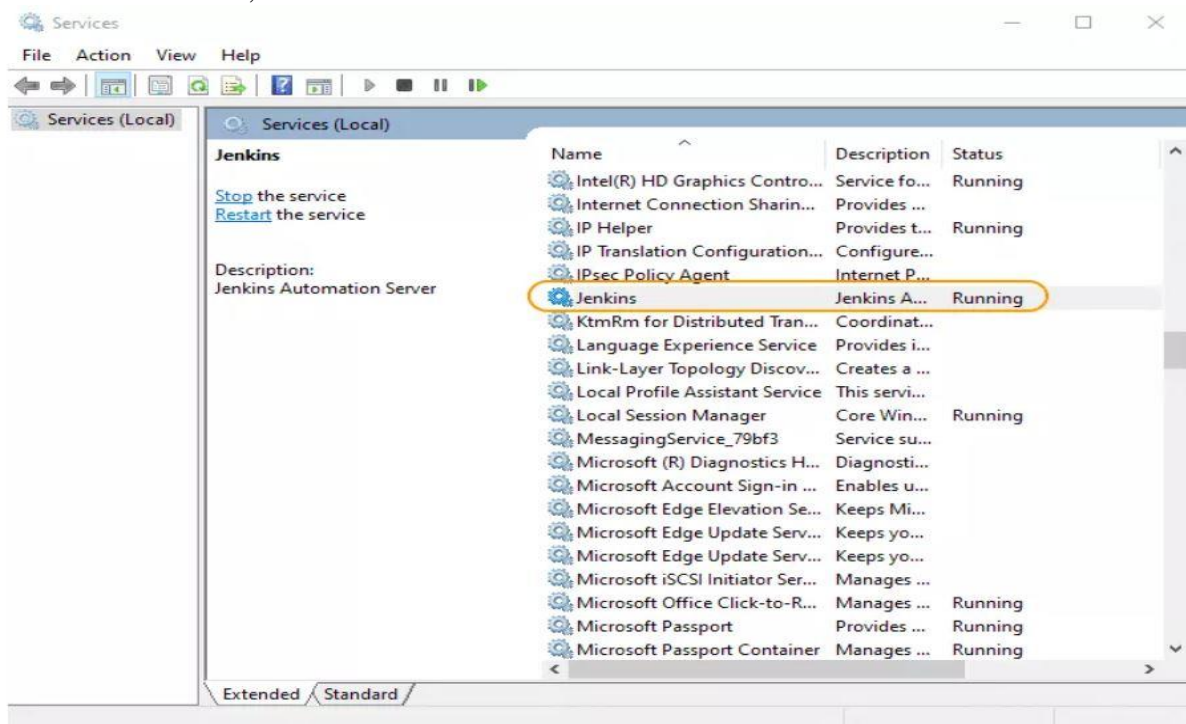
Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Step 8: Finish Jenkins installation

Once the installation completes, click on **Finish** to complete the installation.



Jenkins will be installed as a **Windows Service**. You can validate this by browsing the **services** section, as shown below:



Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

2.Jenkins – build and deploy a web application to a local HTTP server

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

⚠ No schedules so will never run

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Trigger builds remotely (e.g., from scripts) ?

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Hello World.....'
8       }
9     }
10  }
11 }
12
```

try sample Pipeline...
try sample Pipeline...
Hello World
GitHub + Maven
Scripted Pipeline

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Pipeline Script

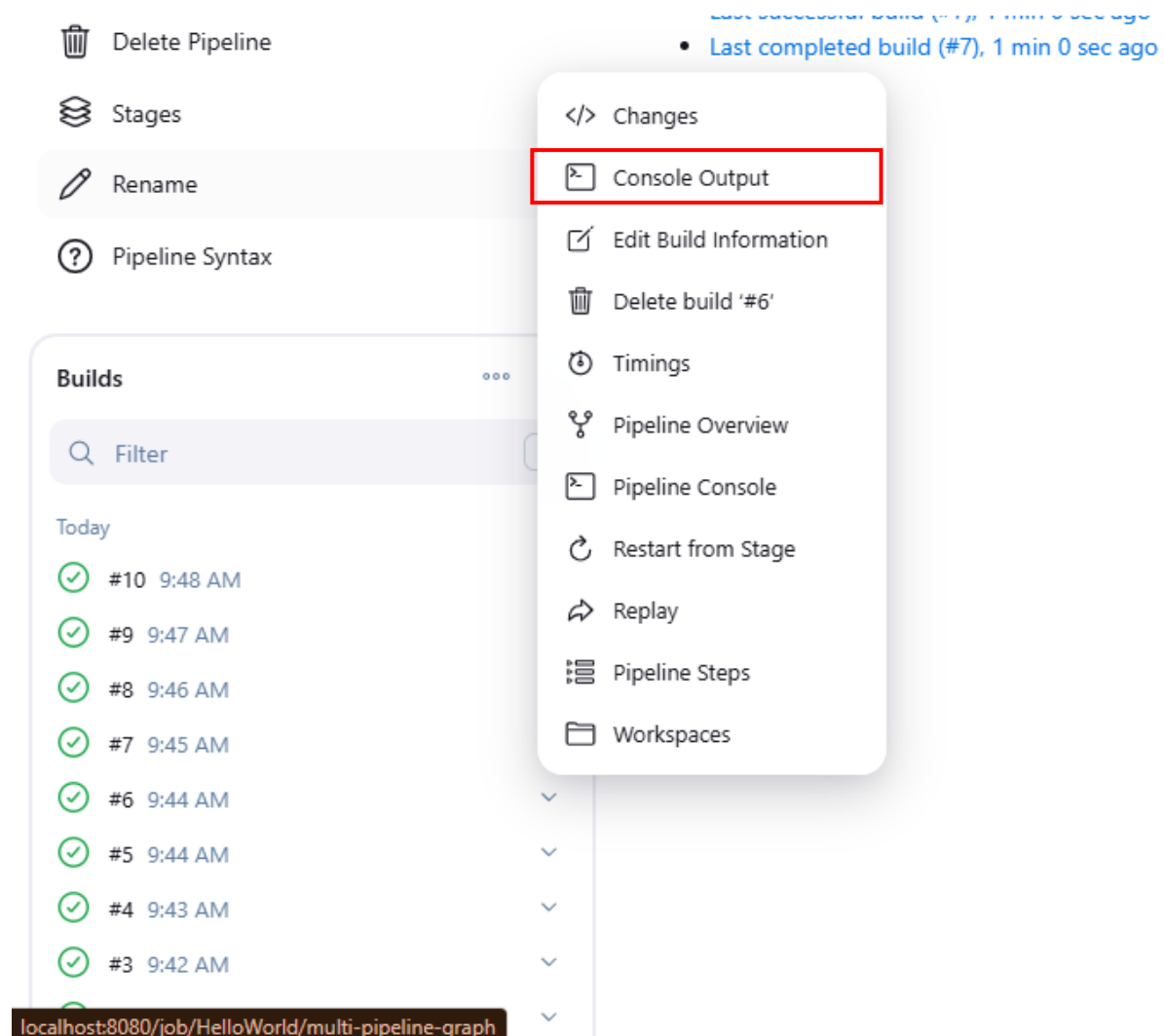
```
pipeline {
  agent any

  stages {
    stage('Hello') {
      steps {
        echo 'Hello World.....'
      }
    }
  }
}
```

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

```
    }  
    stage('Build'){  
        steps{  
            echo 'Building the project....'  
        }  
    }  
    stage('Deploy'){  
        steps{  
            echo 'Deploying the application....'  
        }  
    }  
    post{  
        success{  
            echo 'Pipeline completed successfully!'  
        }  
        failure{  
            echo "Pipeline failed!"  
        }  
    }  
}
```

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A



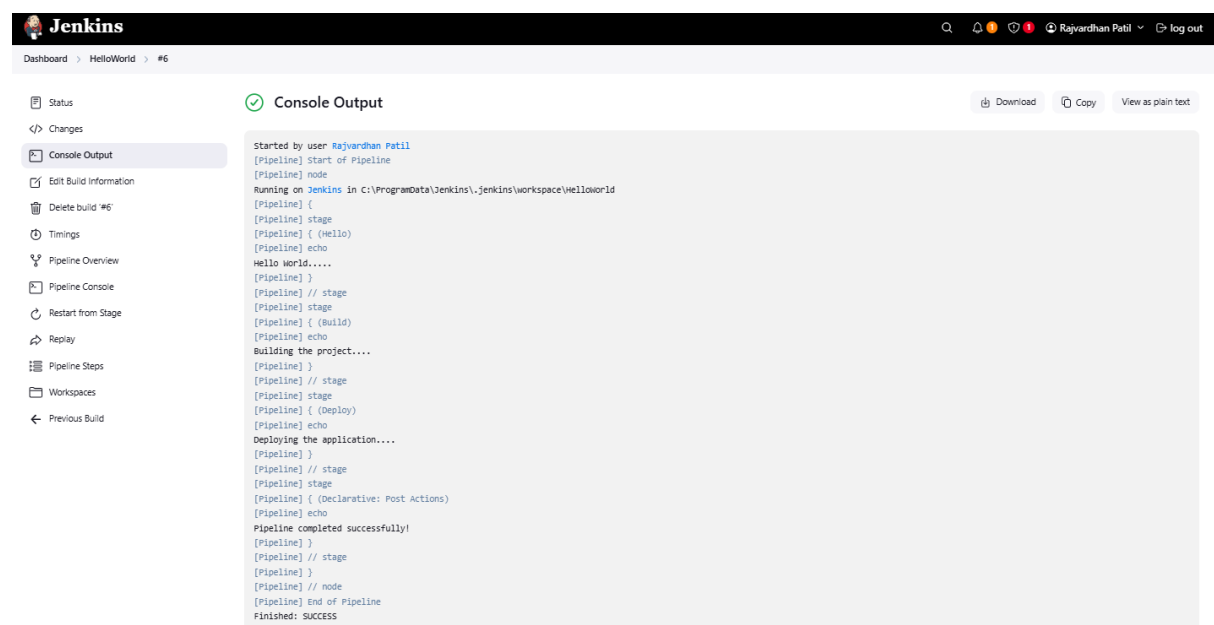
The screenshot shows the Jenkins Pipeline console output menu. The menu is open, displaying various options. The 'Console Output' option is highlighted with a red box. The background shows the Jenkins interface with a list of builds and a sidebar with navigation options.

- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax
- Builds
 - Filter
 - Today
 - #10 9:48 AM
 - #9 9:47 AM
 - #8 9:46 AM
 - #7 9:45 AM
 - #6 9:44 AM
 - #5 9:44 AM
 - #4 9:43 AM
 - #3 9:42 AM

localhost:8080/job/HelloWorld/multi-pipeline-graph

- </> Changes
- Console Output**
- Edit Build Information
- Delete build '#6'
- Timings
- Pipeline Overview
- Pipeline Console
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces

Output:



The screenshot shows the Jenkins Console Output for a pipeline. The output is displayed in a text area with a download button and a copy button. The output text is as follows:

```
Started by user Rajvardhan Patil
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\jenkins\jenkins\workspace\HelloWorld
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World.....
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building the project....
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Deploying the application....
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```


Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

3. Jenkins – Integrating Jenkins with Github

New Item

Enter an item name

github_pipeline_job

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

None

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/rajgpatil/pipelineDemo.git

Credentials ?

rajgpatil/*****

+ Add

Advanced

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

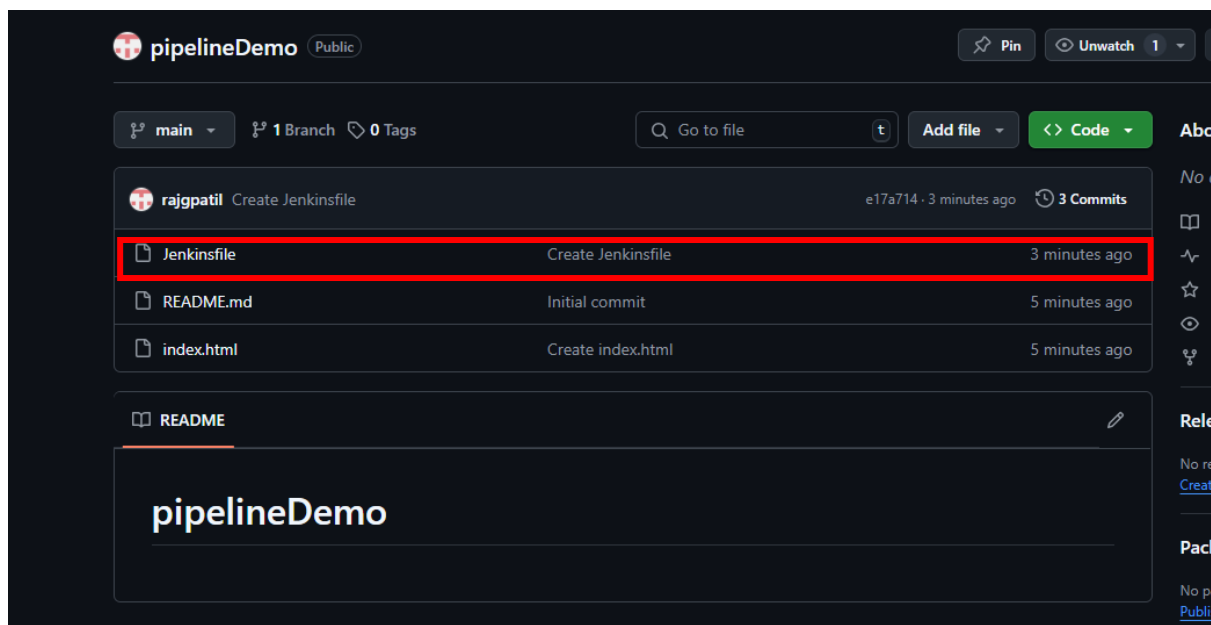
H/5 * * * *

Would last have run at Monday, 10 March, 2025 at 10:03:49 am India Standard Time; would next run at Monday, 10 March, 2025 at 10:08:49 am India Standard Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Trigger builds remotely (e.g., from scripts) ?



Jenkinsfile:

```
pipeline {  
  agent any  
  
  stages {  
    stage('Checkout') {  
      steps {  
        checkout scm  
      }  
    }  
  }  
}
```

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

```
}
```

```
stage('Build') {  
    steps {  
        echo 'Building the project...'  
        // Add build commands here based on your project type  
        // For example, if it's a Java project:  
        // sh 'mvn clean compile'  
    }  
}
```

```
stage('Test') {  
    steps {  
        echo 'Running tests...'  
        // Add test commands here  
        // Example: sh 'mvn test'  
    }  
}
```

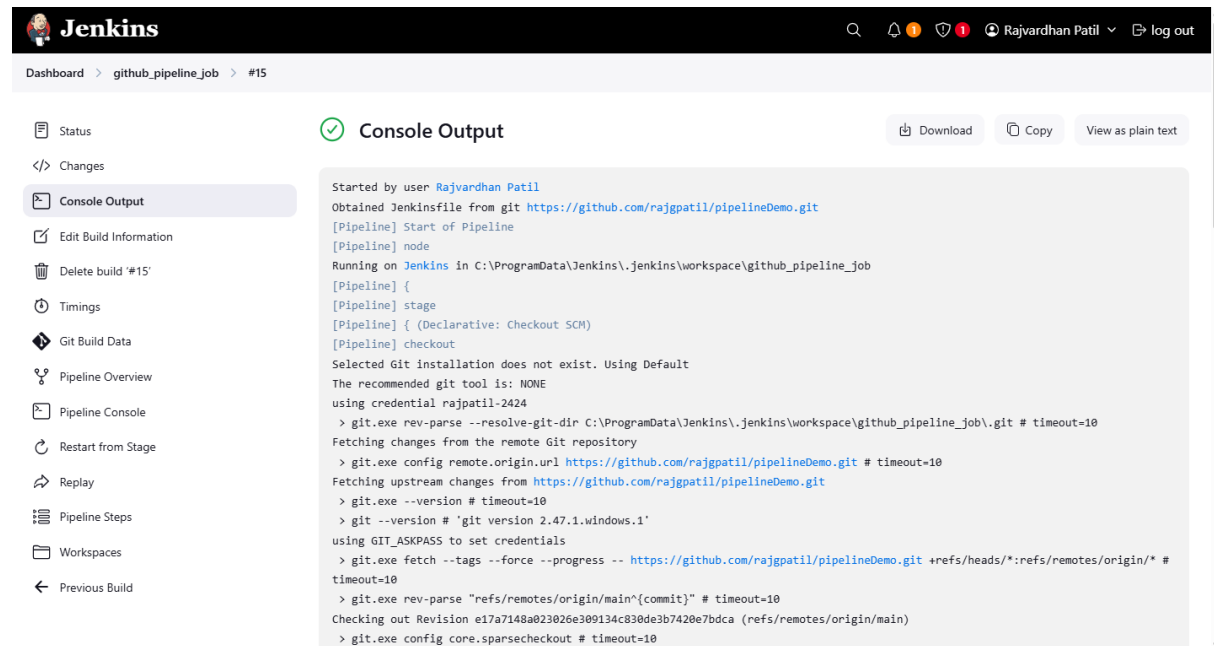
```
stage('Deploy') {  
    steps {  
        echo 'Deploying...'  
        // Add deployment commands  
    }  
}
```

```
post {  
    success {  
        echo 'Pipeline executed successfully!'  
    }  
    failure {
```

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A

```
        echo 'Pipeline execution failed!'
    }
}
}
```

Output



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search icon, notification icons, and the user name 'Rajvardhan Patil' with a 'log out' button. The breadcrumb trail is 'Dashboard > github_pipeline_job > #15'. The left sidebar contains a list of links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#15', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Console Output' with a green checkmark icon and buttons for 'Download', 'Copy', and 'View as plain text'. The console output text is as follows:

```
Started by user Rajvardhan Patil
Obtained Jenkinsfile from git https://github.com/rajgpatil/pipelineDemo.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\github_pipeline_job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential rajpatil-2424
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\github_pipeline_job\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/rajgpatil/pipelineDemo.git # timeout=10
Fetching upstream changes from https://github.com/rajgpatil/pipelineDemo.git
> git.exe --version # timeout=10
> git --version # 'git version 2.47.1.windows.1'
using GIT_ASKPASS to set credentials
> git.exe fetch --tags --force --progress -- https://github.com/rajgpatil/pipelineDemo.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision e17a7148a023026e309134c830de3b7420e7bdca (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
```

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A


4.Demonstrate Continuous Integration and development using Jenkins(login Page html)


New Item


Enter an item name


loginPageDemo


Select an item type


**Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.


**Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/rajgpatil/LoginForm.git

Credentials ?

rajgpatil/**

+ Add

Advanced ▾

Add Repository

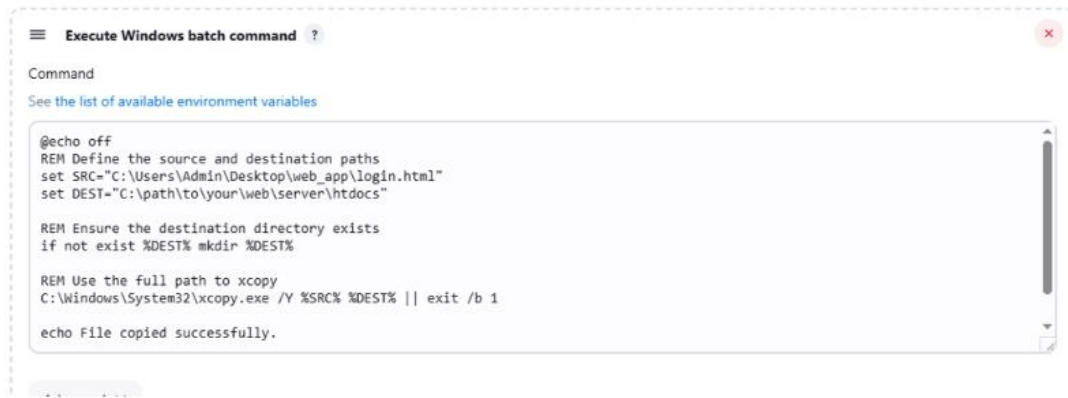
Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Name: Rajvardhan Ganpatrao Patil
Roll No: 40 Div: A



Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.



Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save Apply

Output:

Dashboard > loginPageDemo > #1 > Console Output

Status

<> Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

✓ Console Output

Download

Copy

View as plain text

```
Started by user Rajvardhan Patil
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\loginPageDemo
The recommended git tool is: NONE
using credential rajpatil-2424
Cloning the remote git repository
Cloning repository https://github.com/rajpatil/loginform.git
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace\loginPageDemo # timeout=10
Fetching upstream changes from https://github.com/rajpatil/loginform.git
> git.exe --version # timeout=10
> git --version # 'git version 2.47.1.windows.1'
using GIT_ASKPASS to set credentials
> git.exe fetch --tags --force --progress -- https://github.com/rajpatil/loginform.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/rajpatil/loginform.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main"(commit)" # timeout=10
Checking out Revision 2d8e9023d1cda8c857cd43cef778f35e7bb5d5 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 2d8e9023d1cda8c857cd43cef778f35e7bb5d5 # timeout=10
Commit message: "Update index.html"
First time build, skipping changelog.
[loginPageDemo] $ cmd /c call C:\WINDOWS\TEMP\jenkins6956187772751468611.bat
D:\form\loginform\index.html
1 File(s) copied
File copied successfully.
Finished: SUCCESS
```

REST API Jenkins 2.492.1

Status

Changes

Workspace

Wipe Out Current Workspace

Build Now

Configure

Delete Project

Rename

Builds

Filter

Today

#1 9:52 AM

Workspace of loginPageDemo on Built-In Node

loginPageDemo /

.git

index.html

Mar 17, 2025, 9:52:12 AM

799 B

README.md

Mar 17, 2025, 9:52:12 AM

11 B

style.css

Mar 17, 2025, 9:52:12 AM

1.22 KiB

(all files in zip)

Workspace of loginPageDemo on Built-In Node

loginPageDemo /

.git

index.html

Mar 17, 2025, 9:52:12 AM

799 B

README.md

Mar 17, 2025, 9:52:12 AM

11 B

style.css

Mar 17, 2025, 9:52:12 AM

1.22 KiB

(all files in zip)

Username

Enter Username

Password

Enter Password

Login

☒ Remember me

Cancel

Forgot password?