

Agile Development Process And Devops Lab

Course Code: ISWE301P

Digital Assessment-2



Name: Tejas C
Reg. No: 23MIS0671
Faculty: Kalaivani S
Slot: L51+L52

TASK 1: Jenkins Familiarization

Objective: Understand Jenkins UI and basic navigation

Tasks:

1. Open Jenkins Dashboard in browser

2. Identify:

- Dashboard
- Manage Jenkins
- New Item
- Build History

3. Check Jenkins version

1. Jenkins Dashboard

The screenshot shows the Jenkins Dashboard at localhost:1924. The main header says "Welcome to Jenkins!". Below it, there are sections for "Build Queue" (No builds in the queue), "Build Executor Status" (0/2), and a "Create a job" button. To the right, there are links for "Set up a distributed build", "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". On the left sidebar, there are links for "New Item" and "Build History". The top navigation bar includes links for Gmail, YouTube, WhatsApp web, Maps, Netflix, Google, News, HTML - GeeksforGeeks, GitHub, Prime Video, and Monkeytype.

2. Manage Jenkins

The screenshot shows the Manage Jenkins page at localhost:1924/manage/. It features a "Manage Jenkins" header and a search bar. A warning message at the top says: "Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)". Below are buttons for "Set up agent", "Set up cloud", and "Dismiss". A note about Content Security Policy (CSP) is displayed: "Jenkins can enforce Content Security Policy (CSP). CSP tells web browsers what they are allowed to do while rendering a web page. This limits or even eliminates the impact of vulnerabilities like cross-site scripting (XSS). CSP is disabled by default for backward compatibility, but it is recommended to enable it, if possible." Buttons for "Learn more" and "Dismiss" are shown. The main area is titled "System Configuration" and contains several sections: "System" (Configure global settings and paths), "Tools" (Configure tools, their locations and automatic installers), "Plugins" (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), "Nodes" (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), "Clouds" (Configure cloud environments), and "Appearance" (Customize the look and feel of the Jenkins interface).

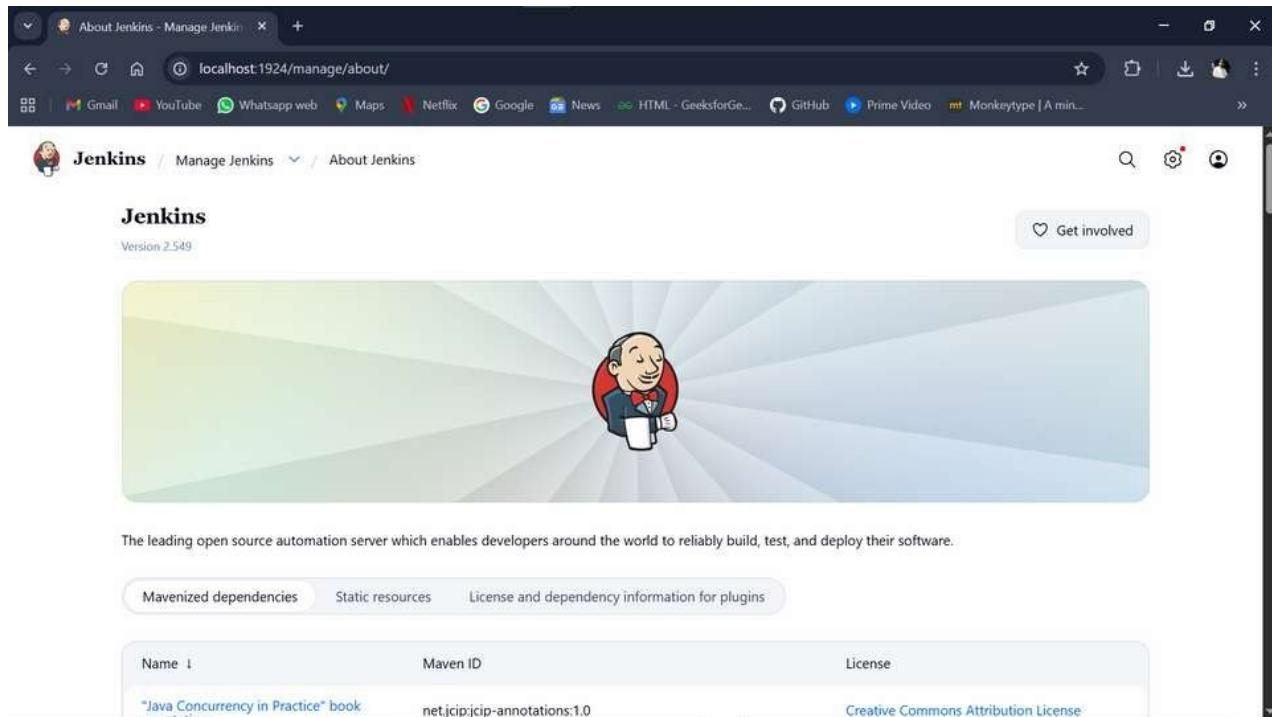
3. New Item

The screenshot shows the Jenkins dashboard at localhost:1924. At the top, there's a header with a Jenkins icon, the title "Dashboard - Jenkins", and a search bar. Below the header, a toolbar has links to various services like Gmail, YouTube, WhatsApp web, Maps, Netflix, Google News, and GitHub. The main content area is titled "Jenkins" and shows a "Build History" section. It includes a "New Item" button, a "Build Queue" section (empty), and a "Build Executor Status" section (0/2). A table displays build information for "FirstCI": Name: FirstCI, Last Success: N/A, Last Failure: N/A, Last Duration: N/A. Below the table are icons for "S" (Stable), "M" (Medium), and "L" (Large). At the bottom right, there are links to "REST API" and "Jenkins 2.549".

4. Build History

The screenshot shows the Jenkins "Build History" page at localhost:1924/view/all/builds. The interface is similar to the dashboard, with a header, toolbar, and "Build History" section. The "Build History" section is titled "Build History of Jenkins" and contains a table with one row for "FirstCI". The table columns are "S" (Status), "Build", "Time Since", and "Status". The entry for "FirstCI" is marked as "stable" with a green checkmark icon. Below the table are icons for "S" (Stable), "M" (Medium), and "L" (Large). At the bottom right, there are links to "REST API" and "Jenkins 2.549".

5. Check Jenkins Version



The screenshot shows the Jenkins 'About Jenkins' page. At the top, it displays 'Jenkins / Manage Jenkins / About Jenkins'. Below that is the Jenkins logo with the text 'Version 2.549'. A large central image features a cartoon character holding a coffee cup. Below the image, a brief description reads: 'The leading open source automation server which enables developers around the world to reliably build, test, and deploy their software.' There are three tabs at the bottom: 'Mavenized dependencies', 'Static resources', and 'License and dependency information for plugins'. The 'License and dependency information for plugins' tab is selected, showing a table with one row:

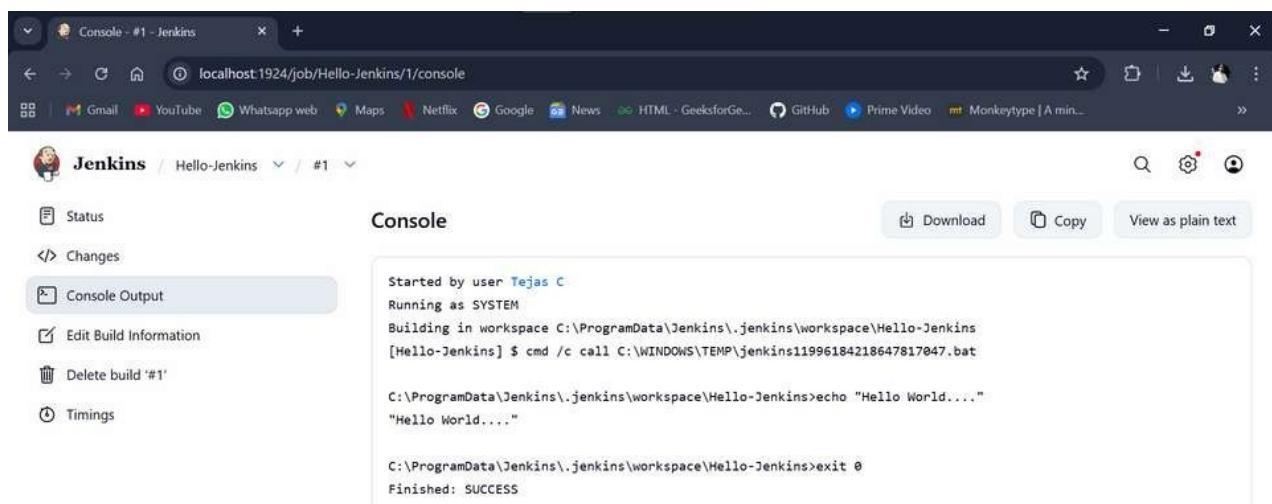
Name	Maven ID	License
"Java Concurrency in Practice" book	net.jcip:jcip-annotations:1.0	Creative Commons Attribution License

TASK 2: Create First Freestyle Job

Objective: Create and run a Jenkins job

Tasks:

1. Create a Freestyle project named Hello-Jenkins
2. Add a description
3. Add build step:
 - Execute shell / Windows batch command
 - Print "Hello Jenkins"
4. Build the job manually



The screenshot shows the Jenkins 'Console' page for the 'Hello-Jenkins' job. The left sidebar includes links for 'Status', 'Changes', 'Console Output' (which is selected), 'Edit Build Information', 'Delete build #1', and 'Timings'. The main area is titled 'Console' and contains the following output:

```
Started by user Tejas C
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins11996184218647817047.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Hello World..."
"Hello World..."

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

TASK 3: Jenkins Workspace & Commands

Objective: Understand workspace usage Tasks:

1. Navigate to job workspace
2. Create a text file using build step
3. Display file contents in console

Job Workspace and a text file

The screenshot shows the Jenkins job configuration page for 'Hello-Jenkins'. The 'Build Steps' section contains two entries:

- Execute Windows batch command**:
Command: echo "Hello World...."
This step outputs "Hello World...." to the console.
- Execute Windows batch command**:
Command: echo "This is Jenkins Workshop" > Jenkins.txt
type Jenkins.txt
This step creates a file named 'Jenkins.txt' containing the text "This is Jenkins Workshop".

At the bottom are 'Save' and 'Apply' buttons.

Console

The screenshot shows the Jenkins console output for build #7. The left sidebar shows build details like Status, Changes, and Console Output. The main area displays the command-line output:

```
Started by user Tejas C
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins16602344171351009561.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Hello World...."
"Hello World...."

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins12067688778512768111.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "This is Jenkins Workshop" > Jenkins.txt
"This is Jenkins Workshop"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type Jenkins.txt
"This is Jenkins Workshop"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

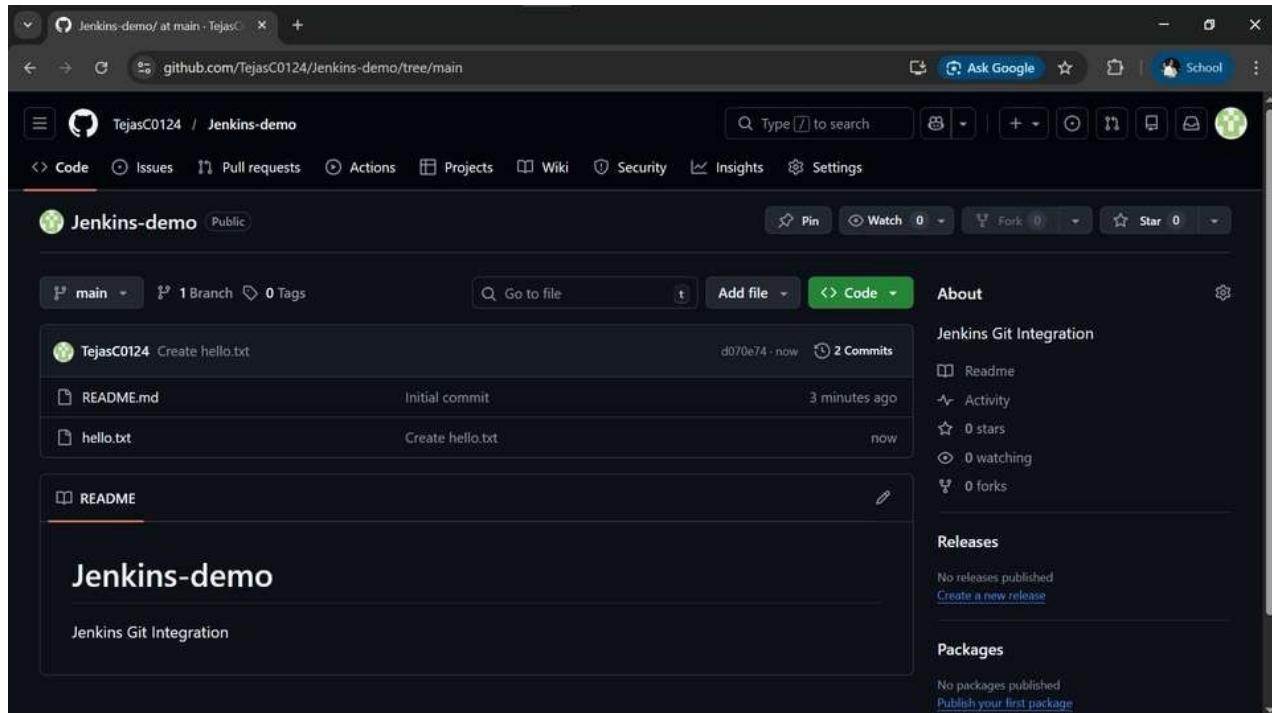
TASK 4: Git Integration

Objective: Integrate Jenkins with GitHub

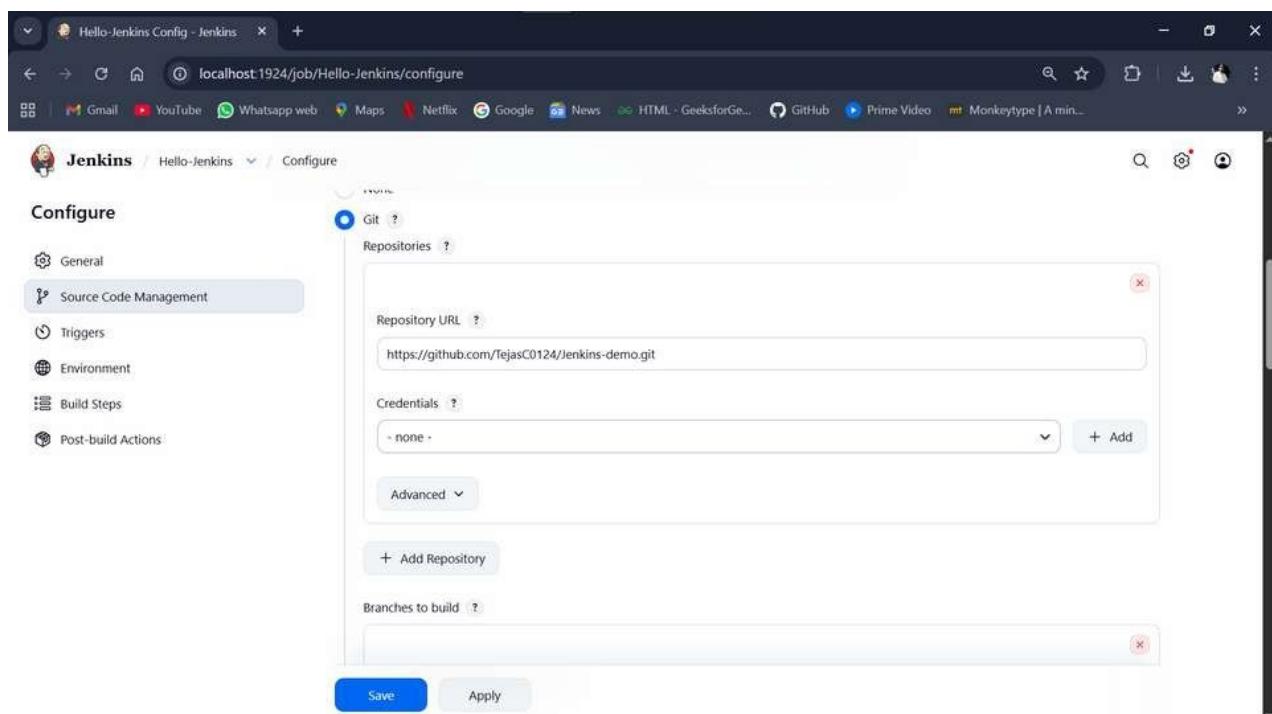
Tasks:

- 1.Create a GitHub repository with sample code
- 2.Configure Git in Jenkins
- 3.Add Git repository URL in job
- 4.Build and verify code checkout

1.Create a GitHub repository with sample code



2.Configure Git in Jenkins



3. Build History

The screenshot shows the Jenkins 'Build History' page at localhost:1924/view/all/builds. The page title is 'Build History of Jenkins'. On the left, there are filters for 'Build Queue' (no builds) and 'Build Executor Status' (0/2). The main area displays a table of builds:

S	Build	Time Since	Status
1	Hello-Jenkins #8	1 min 3 sec	stable
2	Hello-Jenkins #7	18 min	stable
3	Hello-Jenkins #6	27 min	stable
4	Hello-Jenkins #5	38 min	stable
5	FirstCI #1	48 min	stable

At the bottom, there are icons for S, M, and L.

4. Workspace after adding Git

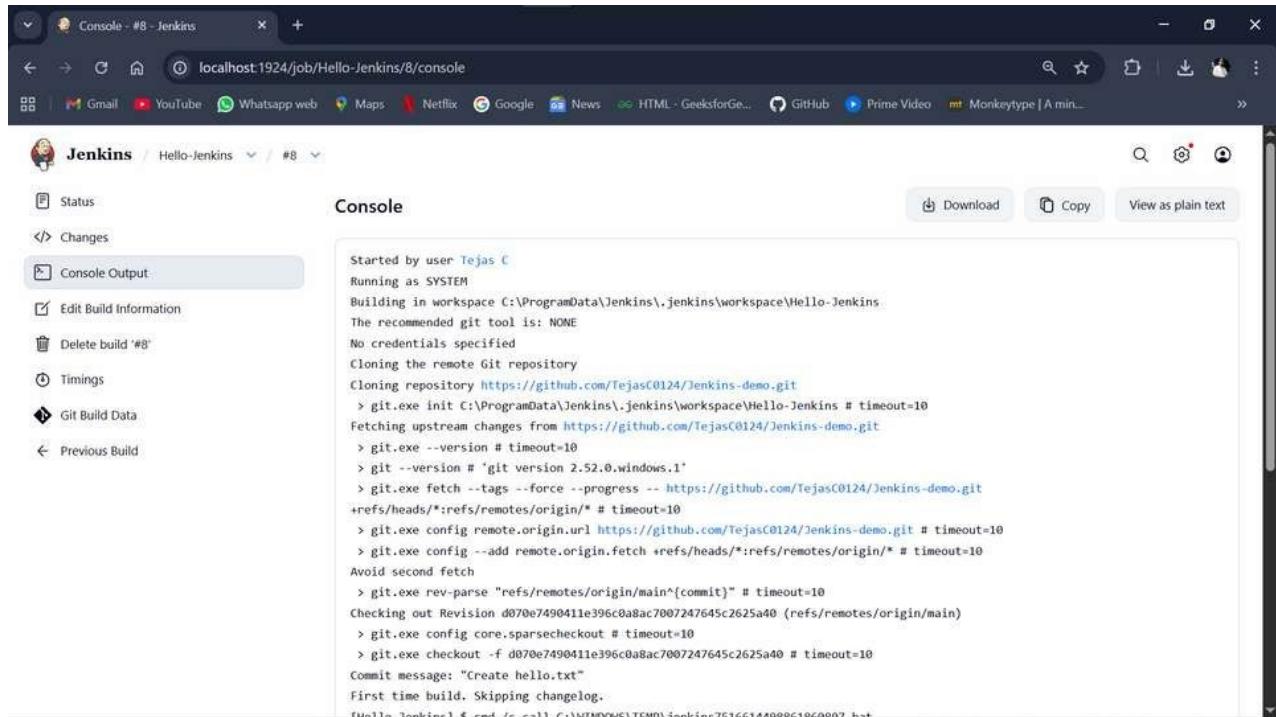
The screenshot shows the Jenkins 'Workspace of Hello-Jenkins on Built-In Node' page at localhost:1924/job/Hello-Jenkins/ws/. The left sidebar has options: Status, Changes, Workspace (selected), Wipe Out Current Workspace, Build Now, Configure, Delete Project, and Rename. The main area shows a file tree under 'Hello-Jenkins /':

- .git
- hello.txt (4 Feb 2026, 22:54:17 29 B)
- Jenkins.txt (4 Feb 2026, 22:54:21 29 B)
- README.md (4 Feb 2026, 22:54:17 41 B)

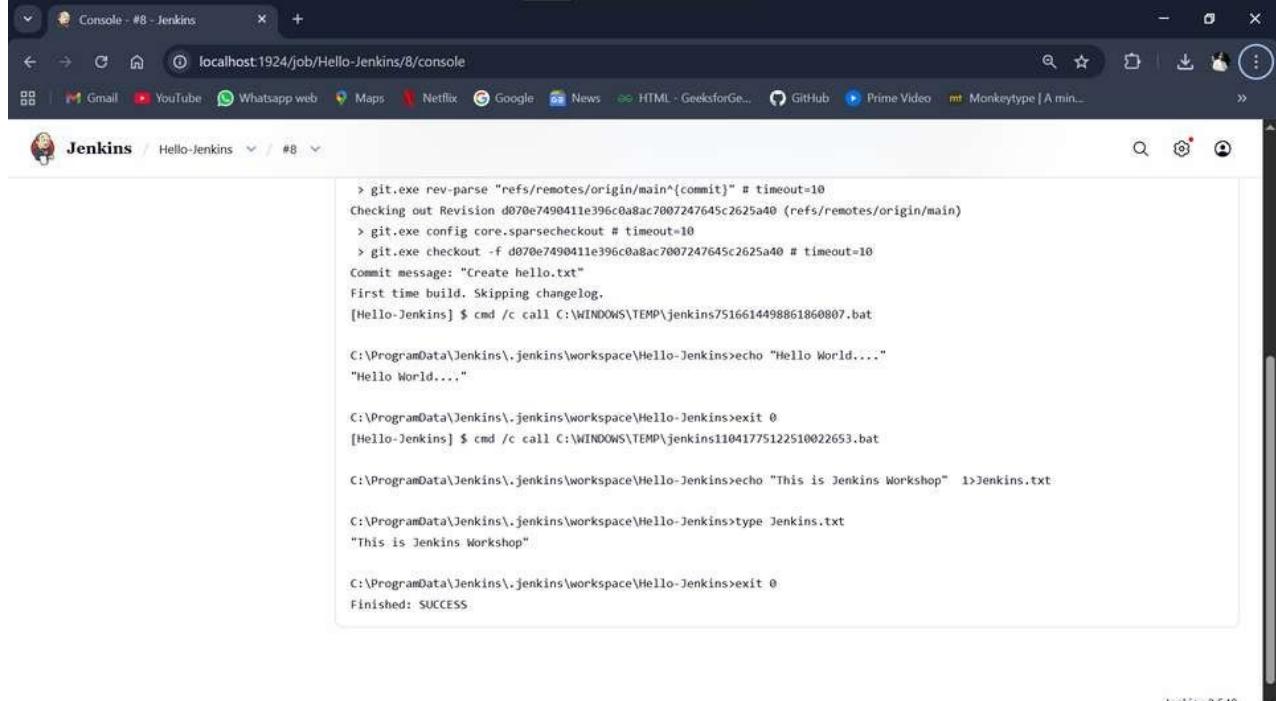
A link to download all files in zip is provided. Below, a 'Builds' section lists recent builds:

Build	Time
#8	22:54
#7	22:37
#6	22:27
#5	22:16

5. Console



```
Started by user Tejas C
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/TejasC0124/Jenkins-demo.git
> git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins # timeout=10
Fetching upstream changes from https://github.com/TejasC0124/Jenkins-demo.git
> git.exe --version # timeout=10
> git --version # 'git version 2.52.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/TejasC0124/Jenkins-demo.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/TejasC0124/Jenkins-demo.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 d070e7490411e396c0a8ac7007247645c2625a40 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f d070e7490411e396c0a8ac7007247645c2625a40 # timeout=10
Commit message: "Create hello.txt"
Commit message: "Create hello.txt"
First time build. Skipping changelog.
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins7516614498861860807.bat
```



```
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision d070e7490411e396c0a8ac7007247645c2625a40 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f d070e7490411e396c0a8ac7007247645c2625a40 # timeout=10
Commit message: "Create hello.txt"
Commit message: "Create hello.txt"
First time build. Skipping changelog.
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins7516614498861860807.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Hello World...."
"Hello World...,"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins11041775122510022653.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "This is Jenkins Workshop" >> Jenkins.txt
C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type Jenkins.txt
"This is Jenkins Workshop"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

Jenkins 2.549

TASK 5: Poll SCM Trigger

Objective: Automatically trigger builds on code change

Tasks:

1. Enable Poll SCM
2. Set schedule: * * * * *
3. Modify GitHub file and commit
4. Observe automatic build

Enable SCM Poll and Set Schedule “* * * * *”

The screenshot shows the Jenkins configuration interface for a job named "Hello-Jenkins". Under the "Triggers" section, the "Poll SCM" option is selected, indicated by a checked checkbox. The schedule field contains the value "* * * * *". A warning message below the schedule field states: "⚠️ Do you really mean 'every minute' when you say '* * * * *'? Perhaps you meant 'H * * * *' to poll once per hour. Would last have run at Thursday, 5 February 2026, 11:10:00 am India Standard Time; would next run at Thursday, 5 February 2026, 11:11:00 am India Standard Time." There are "Save" and "Apply" buttons at the bottom.

Modify Git File

The screenshot shows a GitHub repository named "Jenkins-demo". A commit titled "Update hello.txt" was made by user "TejasC0124". The commit message contains two lines of text: "Jenkins integration repo..." and "I have updated for the SCM Poll". The commit was pushed at "c82cf7c · now". The GitHub interface includes standard navigation and search features.

Console Output

The screenshot shows the Jenkins console output for build #11. The log contains the following commands and their outputs:

```
COMMIT message: 'update hello.txt
> git.exe rev-list --no-walk c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins6753643988672896964.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Hello World...."
"Hello World...."

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins7850386444702788261.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "This is Jenkins Workshop" 1>Jenkins.txt
C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type Jenkins.txt
"This is Jenkins Workshop"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type hello.txt
Jenkins integration repo...
I have updated for the SCM Poll

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

Jenkins 2.549

Jenkins Status

The screenshot shows the Jenkins job #11 status page. Key details include:

- Status:** #11 (5 Feb 2026, 11:18:52) - Build was successful.
- Started by user:** Tejas C.
- This run spent:**
 - 3 ms waiting;
 - 2 sec build duration;
 - 2 sec total from scheduled to completion.
- Git Build Data:** Revision: c82cf7cc0988ace0b4a1690d3b96655d374ad2b2, Repository: <https://github.com/TejasC0124/Jenkins-demo.git>, refs/remotes/origin/main
- Changes:** No changes.

REST API Jenkins 2.549

TASK 6: Parameterized Build

Objective: Use parameters in Jenkins job

Tasks:

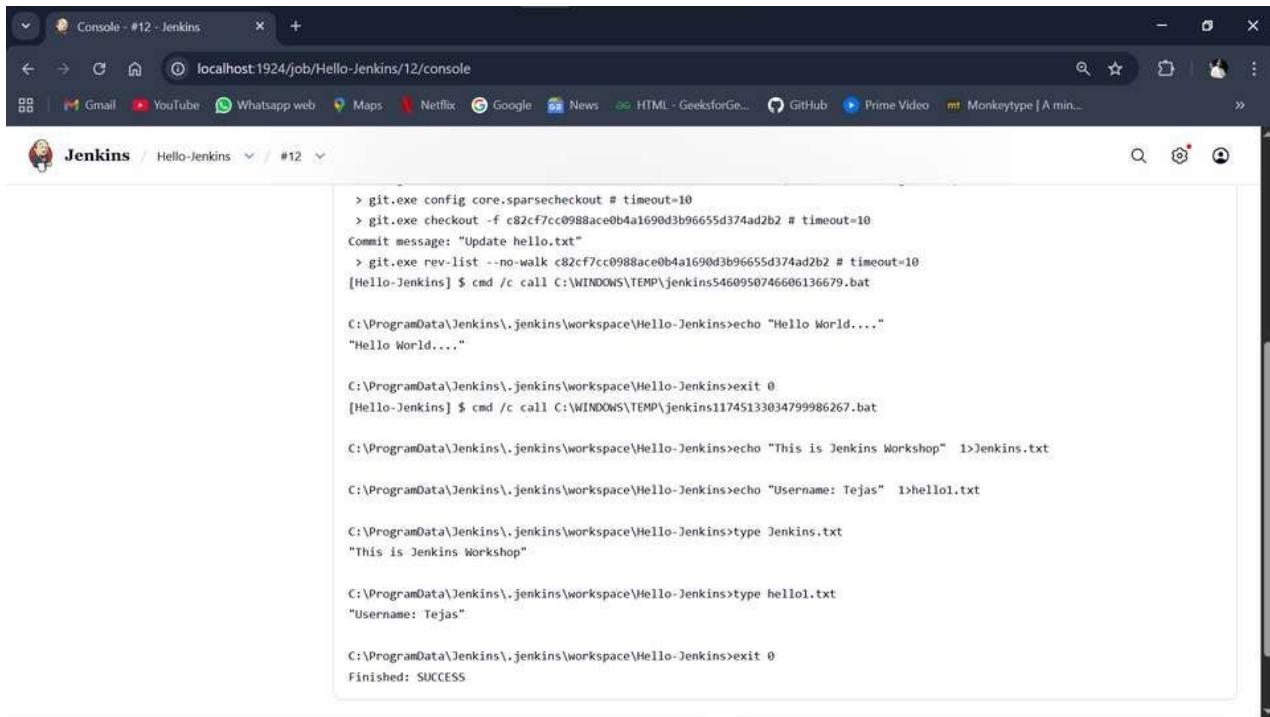
1. Enable parameterized build
2. Add String parameter USERNAME
3. Print parameter value in build step

Enable parameterized build

The screenshot shows the Jenkins configuration interface for a job named "Hello-Jenkins". In the left sidebar, under "General", the "This project is parameterised" checkbox is checked. A "String Parameter" is defined with the name "USERNAME" and a default value "Tejas". The "Description" is "Username Parameter". At the bottom, there are "Save" and "Apply" buttons.

The screenshot shows the Jenkins build page for the "Hello-Jenkins" project. It displays a "Build with Parameters" step. The "USERNAME" parameter is set to "Tejas". Below the form are "Build" and "Cancel" buttons. The main area shows a list of builds, with the most recent ones being #11, #10, and #9, all from today at 11:18, 11:17, and 11:14 respectively. A "Builds > ***" link and a "Filter" input field are also visible.

Console Output



The screenshot shows a browser window with the address bar set to `localhost:1924/job/Hello-Jenkins/12/console`. The main content area displays the Jenkins console output for build #12. The output shows a series of commands being run in a Windows command prompt, including Git operations, file creation, and system calls. The final message indicates a successful build.

```
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
Commit message: "Update hello.txt"
> git.exe rev-list --no-walk c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins5460950746606136679.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Hello World...."
"Hello World...."

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins11745133034799986267.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "This is Jenkins Workshop" 1>Jenkins.txt

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo "Username: Tejas" 1>hello1.txt

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type Jenkins.txt
"This is Jenkins Workshop"

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>type hello1.txt
"Username: Tejas"

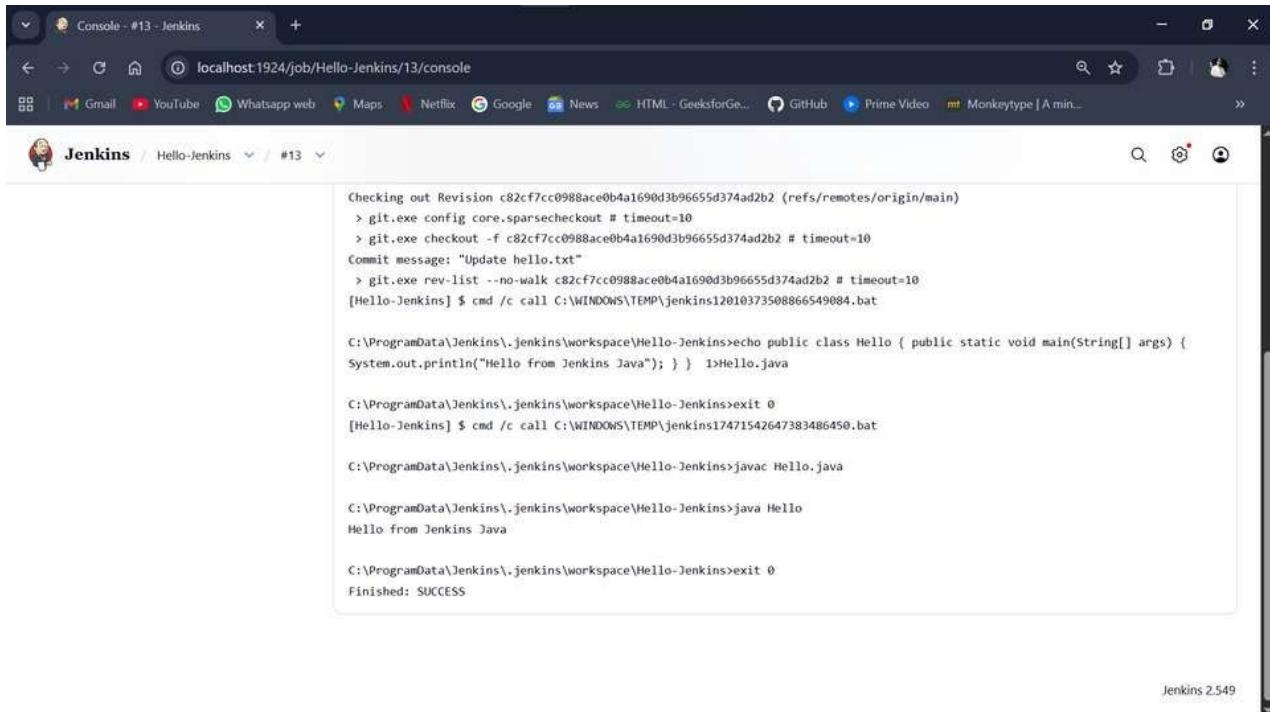
C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

TASK 7: Java Build Using Jenkins

Objective: Compile Java program using Jenkins

Tasks:

1. Create simple Hello.java
2. Compile using javac
3. Run Java program



The screenshot shows a browser window with the address bar set to `localhost:1924/job/Hello-Jenkins/13/console`. The main content area displays the Jenkins console output for build #13. The output shows a series of commands being run in a Windows command prompt, including Git operations, file creation, and system calls. The final message indicates a successful build.

```
Checking out Revision c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
Commit message: "Update hello.txt"
> git.exe rev-list --no-walk c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins12010373508866549084.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo public class Hello { public static void main(String[] args) {
System.out.println("Hello from Jenkins Java"); } } 1>Hello.java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins17471542647383486450.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>javac Hello.java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>java Hello
Hello from Jenkins Java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

TASK 8: Archive Artifacts

Objective: Store build outputs

Tasks:

1. Generate .class or .jar file
2. Archive artifacts in post-build action
3. Download artifact from Jenkins UI

The screenshot shows the Jenkins configuration interface for the 'Hello-Jenkins' job. The 'Post-build Actions' section is active. Under 'Archive the artifacts', the 'Hello.class' file is listed. There is an 'Advanced' dropdown button. At the bottom are 'Save' and 'Apply' buttons.

```
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
Commit message: "Update hello.txt"
> git.exe rev-list --no-walk c82cf7cc0988ace0b4a1690d3b96655d374ad2b2 # timeout=10
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins18363792652459676248.bat

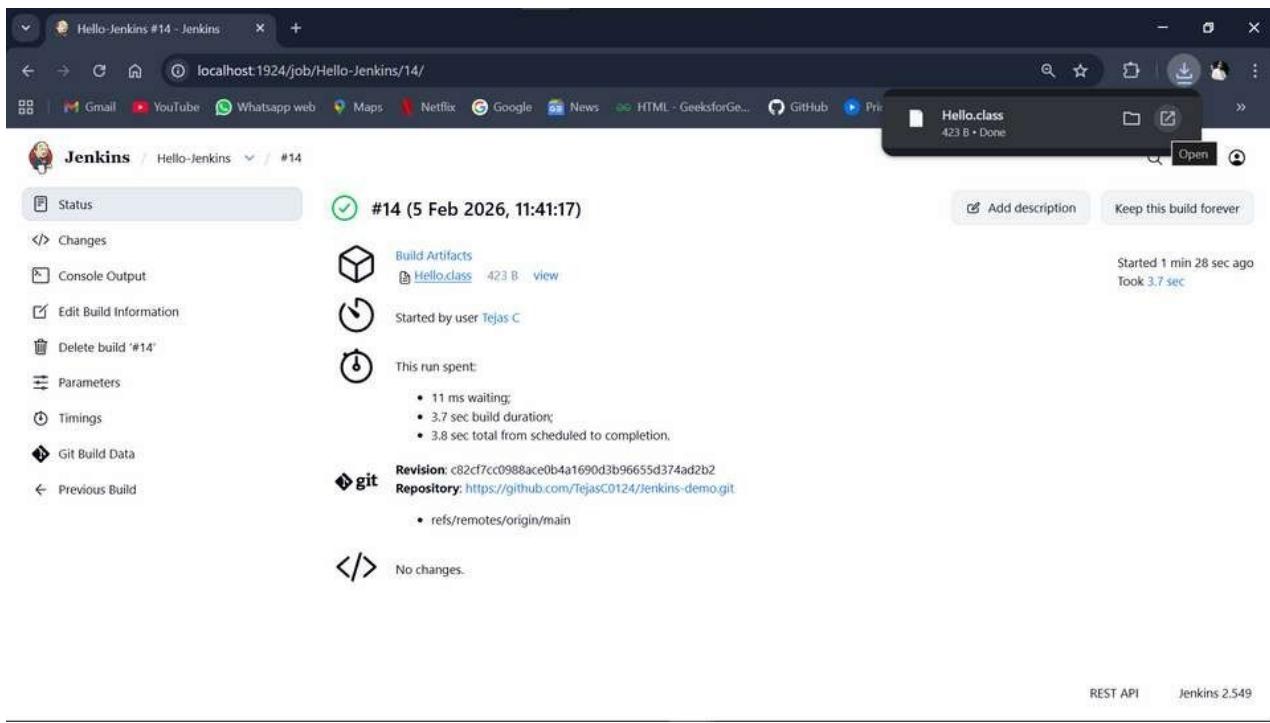
C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>echo public class Hello { public static void main(String[] args) {
System.out.println("Hello from Jenkins Java"); } } >Hello.java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins69865905826085774.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>javac Hello.java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>java Hello
Hello from Jenkins Java

C:\ProgramData\Jenkins\.jenkins\workspace\Hello-Jenkins>exit 0
Archiving artifacts
Finished: SUCCESS
```



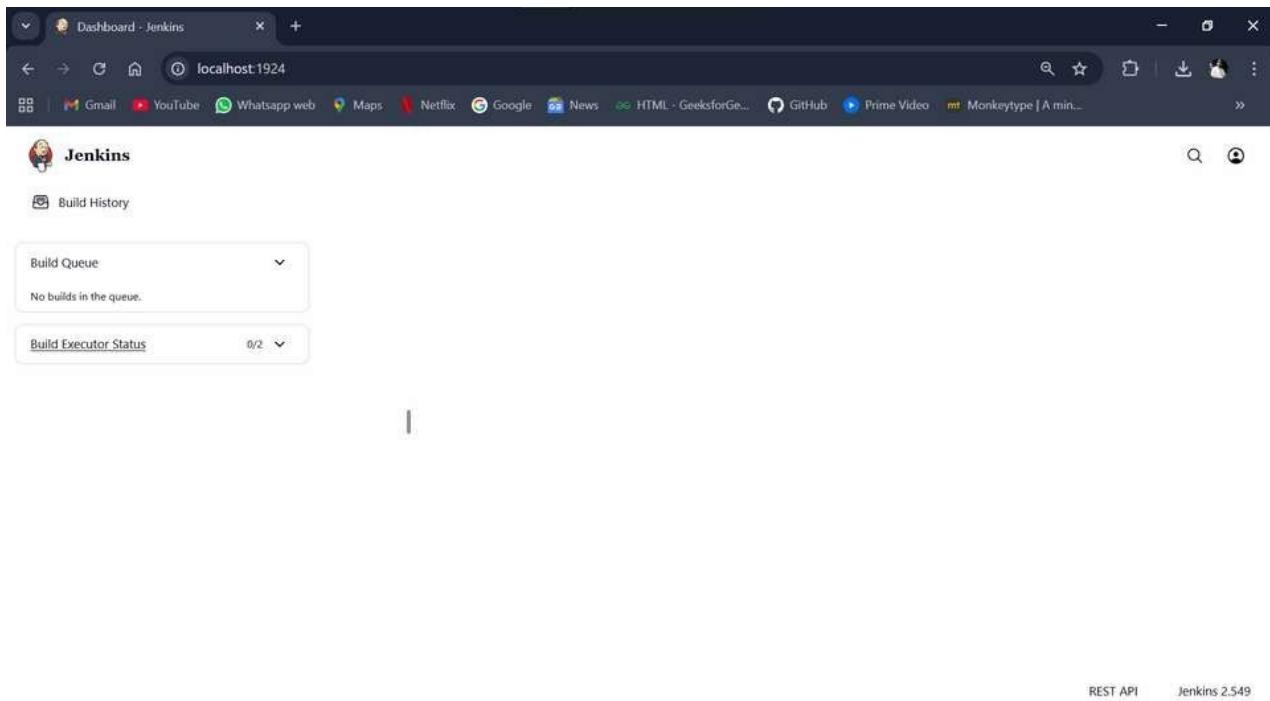
The screenshot shows the Jenkins interface for a completed build. The top navigation bar includes links for Gmail, YouTube, Whatsapp web, Maps, Netflix, Google, News, HTML - GeeksforGeeks, GitHub, Prime Video, and Monkeytype. The main content area displays build #14, which was successful and completed on Feb 5, 2026, at 11:41:17. It shows a build artifact named 'Hello.class' (423 B) with a 'view' link. The build was started by user 'Tejas C'. A summary of run times indicates 11 ms waiting, 3.7 sec build duration, and 3.8 sec total from scheduled to completion. The Git information shows revision c82c17cc0988ace0b4a1690d3b96655d374ad2b2 and repository https://github.com/TejasC0124/jenkins-demo.git. The 'Changes' section notes 'No changes.' A sidebar on the left provides links for Status, Changes, Console Output, Edit Build Information, Delete build '#14', Parameters, Timings, Git Build Data, and Previous Build.

TASK 9: Users & Roles

Objective: Manage Jenkins users

Tasks:

1. Create two users
2. Assign read-only permission to one user
3. Assign build permission to another user



The screenshot shows the Jenkins dashboard. The top navigation bar includes links for Gmail, YouTube, Whatsapp web, Maps, Netflix, Google, News, HTML - GeeksforGeeks, GitHub, Prime Video, and Monkeytype. The main content area displays the 'Build Queue' section, which states 'No builds in the queue.' Below it is the 'Build Executor Status' section, showing '0/2' executors. A sidebar on the left provides links for Build History, Configuration as Code, Pipeline, and Pipeline as Code.

The screenshot shows the Jenkins 'Build History' page at localhost:1924/view/all/builds. The interface includes a navigation bar with links like Gmail, YouTube, WhatsApp web, Maps, Netflix, Google, News, HTML - GeeksforGeeks, GitHub, Prime Video, and Monkeytype. The main content area is titled 'Build History of Jenkins' and displays a summary of the build queue. It shows 'No builds in the queue.' and 'Build Executor Status' with 0/2 available. There are filters for 'Build Queue' (set to 'S'), 'Build' (set to 'S'), 'Time Since' (set to '1'), and 'Status'. A search bar and a refresh button are also present.

TASK 10: Simple Jenkins Pipeline

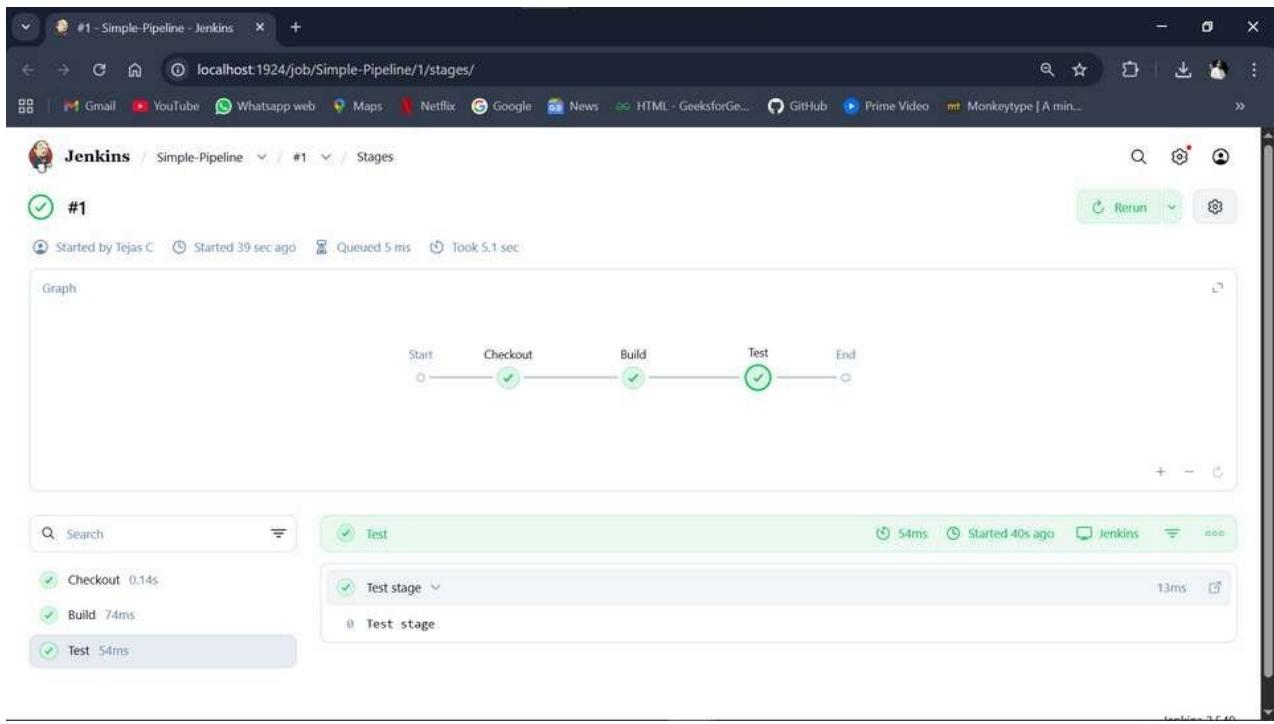
Objective: Create basic pipeline

Tasks:

1. Create Pipeline job
2. Write pipeline with stages:
 - Checkout
 - Build
 - Test
3. Run pipeline

The screenshot shows the Jenkins 'Configure' page for a 'Simple-Pipeline' job at localhost:1924/job/Simple-Pipeline/configure. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The right panel contains a 'Pipeline script' section with a code editor. The code defines a pipeline with two stages: 'Build' and 'Test', each containing an 'echo' step. A 'try sample Pipeline...' button is visible in the top right of the code editor. At the bottom, there are 'Save' and 'Apply' buttons, and a checked checkbox for 'Use Groovy Sandbox'.

```
7      }
8      }
9      stage('Build') {
10     steps {
11       echo 'Build stage'
12     }
13   }
14   stage('Test') {
15     steps {
16       echo 'Test stage'
17     }
18   }
19 }
20 }
21 }
```



TASK 11: Jenkinsfile from Git

Objective: Pipeline as Code

Tasks:

1. Create Jenkinsfile in Git repo
2. Configure pipeline from SCM
3. Trigger build

The screenshot shows a GitHub repository named 'Jenkins-demo'. The 'main' branch is selected, and the file 'JenkinsFile' is open. The code content is as follows:

```

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

```

Simple-Pipeline #3 - Jenkins

localhost:1924/job/Simple-Pipeline/3/

Status: #3 (5 Feb 2026, 12:11:01)

Started by user Tejas C. Started 10 sec ago Took 6.7 sec

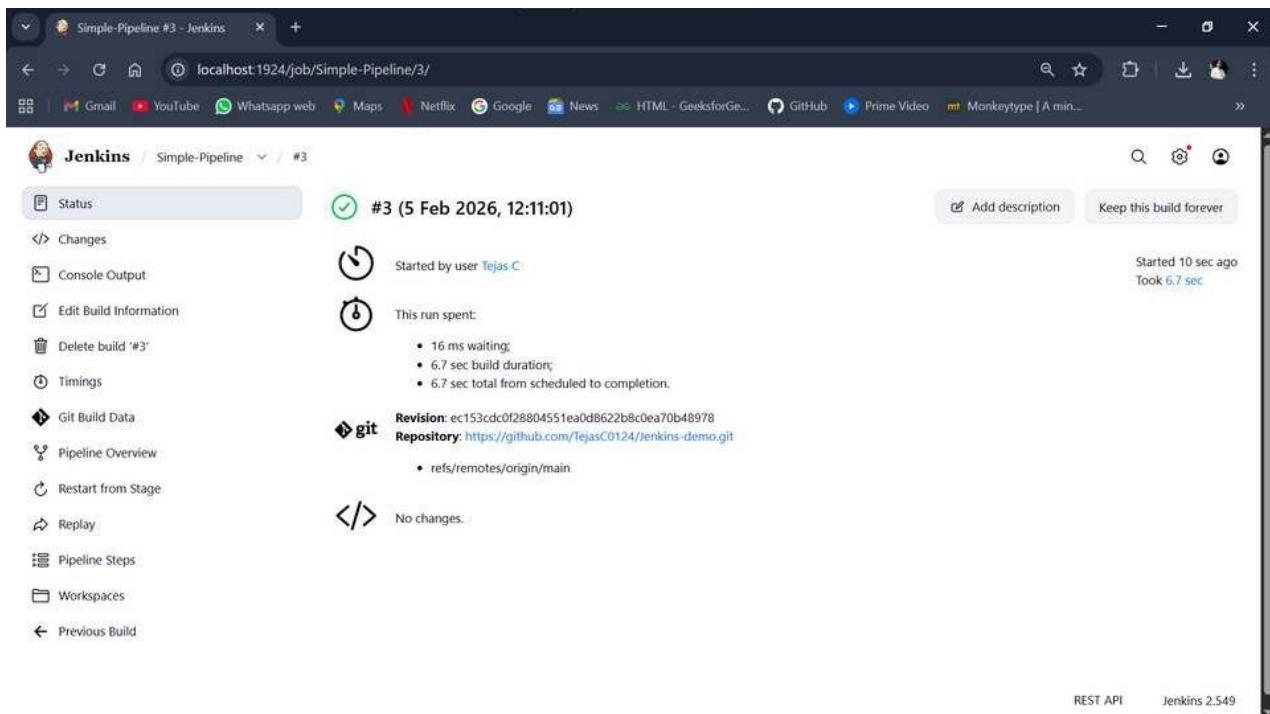
Changes: No changes.

Console Output: No output.

Edit Build Information: Pipeline Overview: Pipeline Steps: Workspaces: Previous Build: Git Build Data: Timings: Pipeline Overview: Restart from Stage: Replay: Pipeline Steps: Workspaces: Previous Build:

Revision: ec153cdc0f28804551ea0d8622b8c0ea70b48978
Repository: https://github.com/TejasC0124/jenkins-demo.git
refs/remotes/origin/main

REST API Jenkins 2.549



#3 - Simple-Pipeline - Jenkins

localhost:1924/job/Simple-Pipeline/3/stages/

Stages: #3

Started by Tejas C. Started 57 sec ago Queued 4 ms Took 6.7 sec

Graph:

```
graph LR; Start((Start)) --> SCM[Checkout SCM]; SCM --> Checkout[Checkout]; Checkout --> Build[Build]; Build --> Test[Test]; Test --> End((End))
```

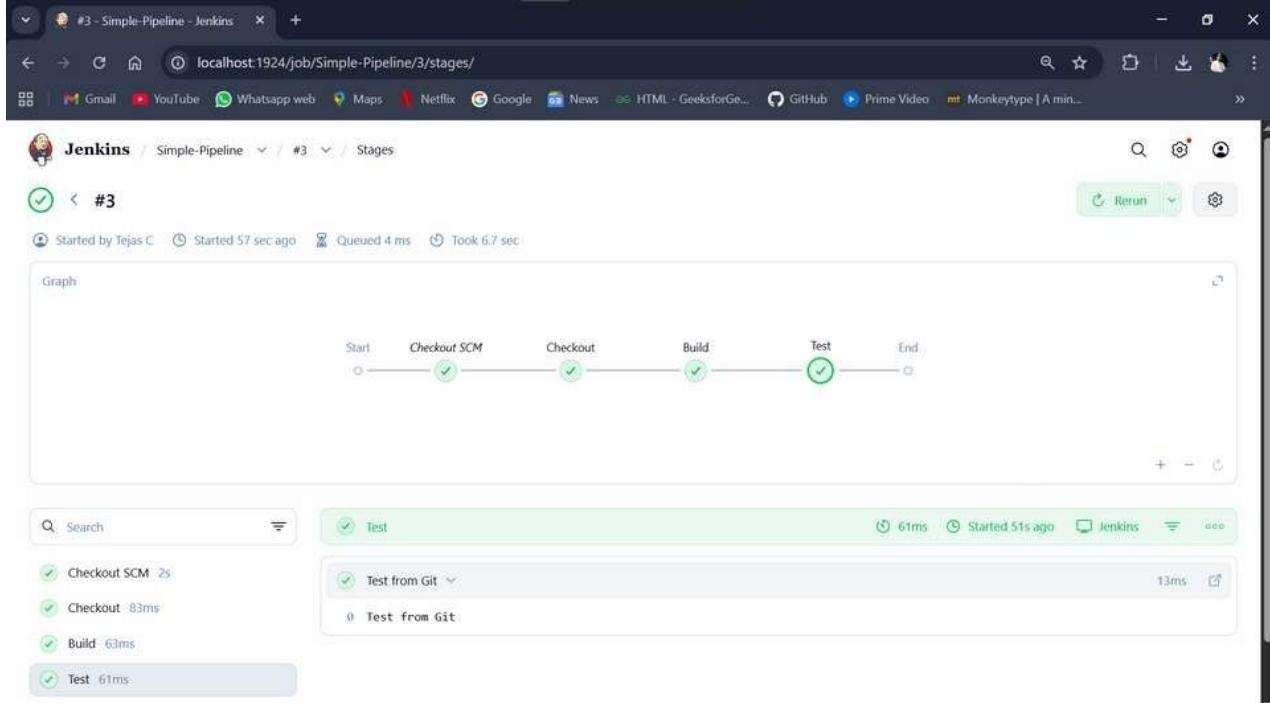
Search: Test

Test: 61ms (Started 51s ago) Jenkins 600

Checkout SCM: 2s (Checkout: 83ms) Jenkins 13ms

Build: 63ms Jenkins 0ms

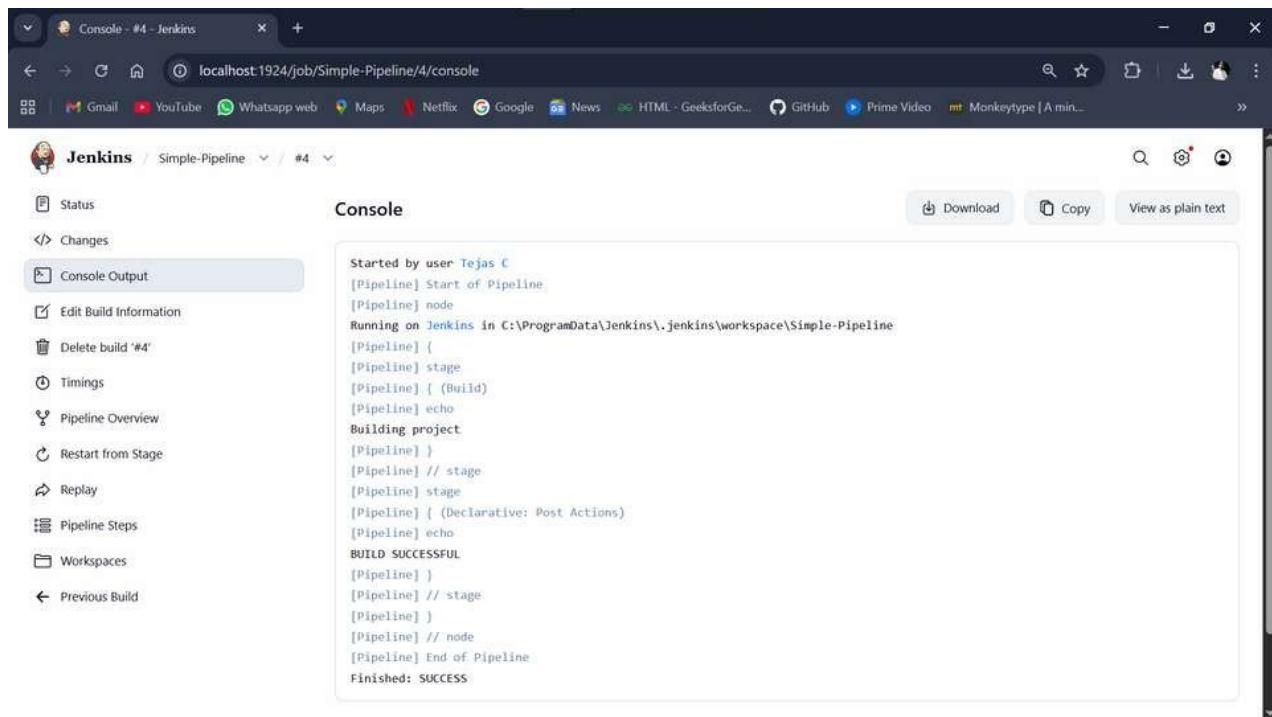
Test: 61ms Jenkins 600



TASK 12: Post-Build

Actions Objective: Handle build result Tasks:

1. Add post section
2. Print message on success/failure



The screenshot shows the Jenkins console output for a pipeline job named "Simple-Pipeline". The build number is #4. The output shows the pipeline starting, running on Jenkins, and executing a stage that prints "Building project". It then reaches a declarative post action section, which includes a "BUILD SUCCESSFUL" message. The entire process is completed successfully.

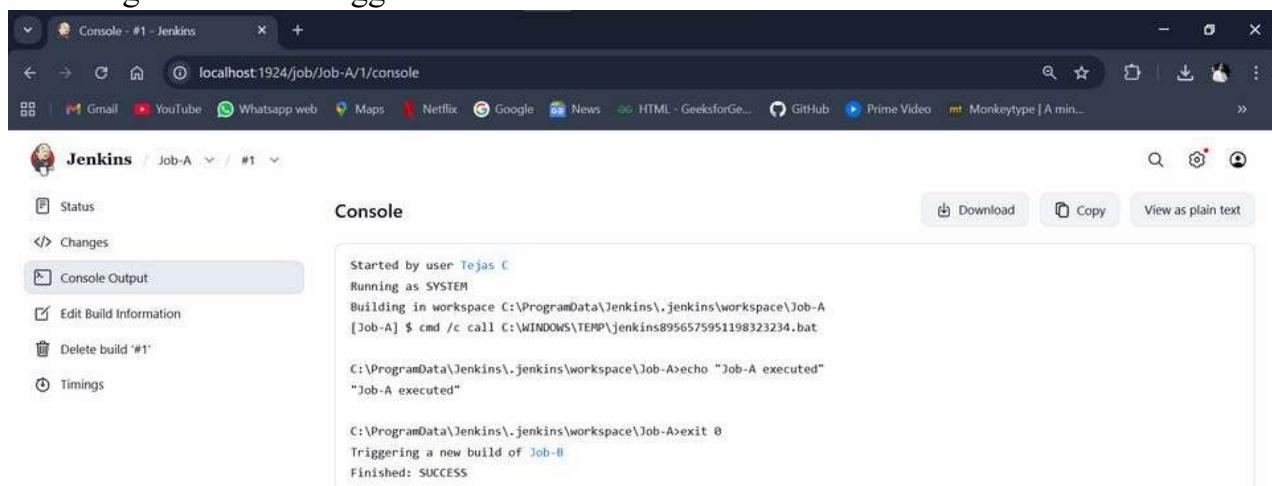
```
Started by user Tejas C
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\Simple-Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building project
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
[BUILD SUCCESSFUL]
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

TASK 13: Trigger Job from Another Job

Objective: Job chaining

Tasks:

1. Create Job-A and Job-B
2. Configure Job-B to trigger after Job-A



The screenshot shows the Jenkins console output for a job named "Job-A". The build number is #1. The output shows the job starting, running as SYSTEM, and building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Job-A. It executes a command to trigger Job-B, which is triggered and successfully completed.

```
Started by user Tejas C
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Job-A
[Job-A] $ cmd /c call C:\WINDOWS\TEMP\jenkins8956575951198323234.bat

C:\ProgramData\Jenkins\jenkins\workspace\Job-A>echo "Job-A executed"
"Job-A executed"

C:\ProgramData\Jenkins\jenkins\workspace\Job-A>exit 0
Triggering a new build of Job-B
Finished: SUCCESS
```

TASK 14: Workspace

Cleanup Objective: Manage disk usage Tasks:

1. Install Workspace Cleanup plugin
2. Clean workspace before build

The screenshot shows the Jenkins 'Plugins' management page. In the search bar at the top right, 'Workspace Cle' is typed. A table lists the 'Workspace Cleanup Plugin' version 0.49, which is described as deleting project workspace when invoked. The plugin is marked as 'Enabled' with a green status icon and a blue toggle switch.

The screenshot shows the Jenkins 'Console' output for build #16 of the 'Hello-Jenkins' project. The left sidebar shows navigation options like Status, Changes, and Console Output, with 'Console Output' selected. The main area displays the build logs:

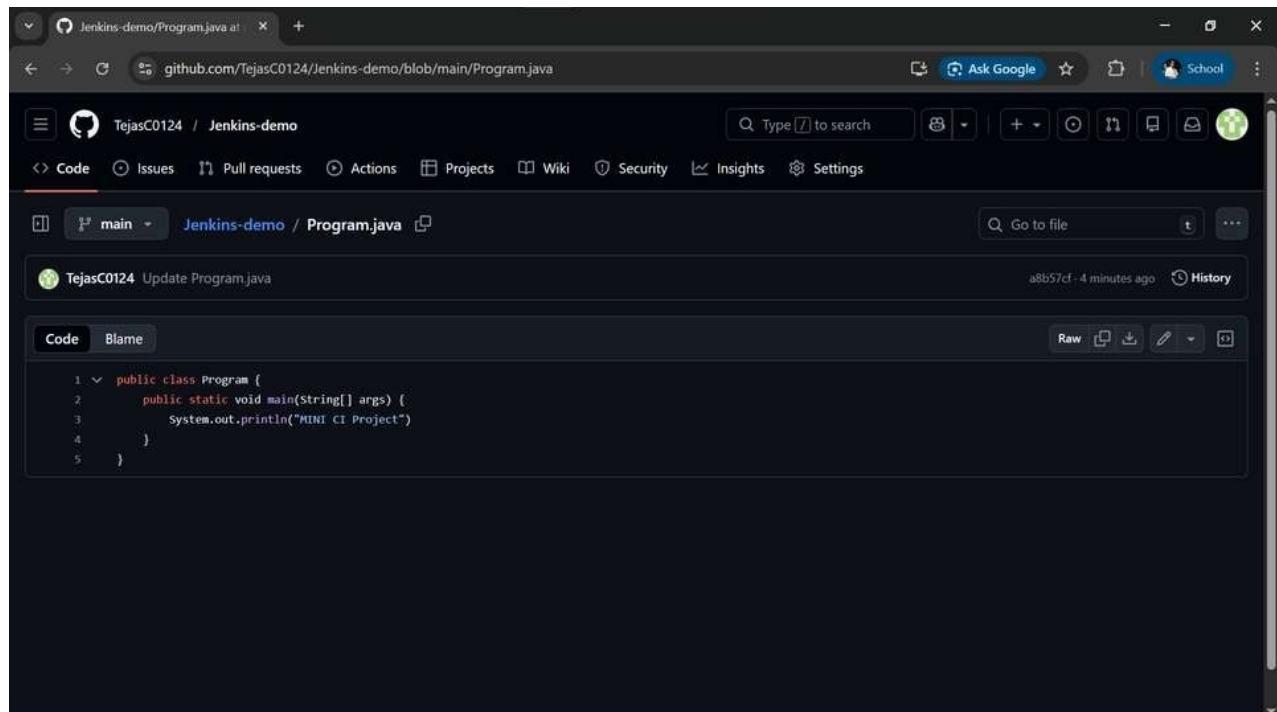
```
Started by user Tejas C
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace>Hello-Jenkins
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/TejasC0124/Jenkins-demo.git
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace>Hello-Jenkins # timeout=10
Fetching upstream changes from https://github.com/TejasC0124/Jenkins-demo.git
> git.exe --version # timeout=10
> git --version # git version 2.52.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/TejasC0124/Jenkins-demo.git
+refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.url https://github.com/TejasC0124/Jenkins-demo.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 ec153cdc0f28804551ea0d8622b8c0ea70b48978 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
```

TASK 15: Mini CI Project

Objective: Implement basic CI flow

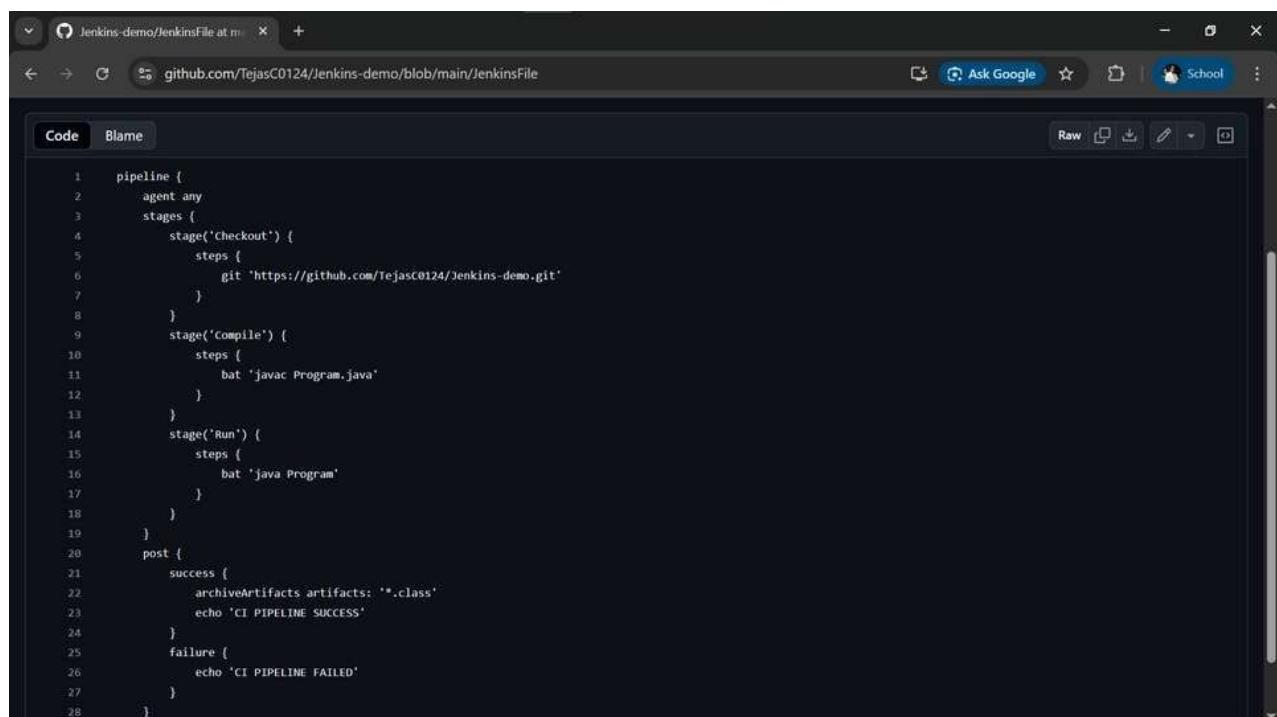
Tasks:

1. Git commit → Jenkins build
2. Compile code
3. Archive artifacts
4. Fail build on error



A screenshot of a web browser displaying a GitHub repository page. The URL in the address bar is `github.com/TejasC0124/Jenkins-demo/blob/main/Program.java`. The page shows a single-line Java program:

```
1 public class Program {  
2     public static void main(String[] args) {  
3         System.out.println("MINI CI Project");  
4     }  
5 }
```



A screenshot of a web browser displaying a GitHub repository page. The URL in the address bar is `github.com/TejasC0124/Jenkins-demo/blob/main/JenkinsFile`. The page shows a Jenkins pipeline configuration:

```
1 pipeline {  
2     agent any  
3     stages {  
4         stage('Checkout') {  
5             steps {  
6                 git 'https://github.com/TejasC0124/Jenkins-demo.git'  
7             }  
8         }  
9         stage('Compile') {  
10            steps {  
11                bat 'javac Program.java'  
12            }  
13        }  
14         stage('Run') {  
15            steps {  
16                bat 'java Program'  
17            }  
18        }  
19    }  
20    post {  
21        success {  
22            archiveArtifacts artifacts: "*.class"  
23            echo 'CI PIPELINE SUCCESS'  
24        }  
25        failure {  
26            echo 'CI PIPELINE FAILED'  
27        }  
28    }  
}
```

Simplifying Jenkins Pipeline - Jenkins

localhost:1924/job/Simple-Pipeline/changes

Changes

#7 (5 Feb 2026, 12:47:56)

1. Create Program.java — noreply / githubweb
2. Update JenkinsFile — noreply / githubweb
3. Update Program.java — noreply / githubweb

Builds

Filter

Today

- #7 12:47
- #6 12:43
- #5 12:42
- #4 12:16

Console - #7 - Jenkins

localhost:1924/job/Simple-Pipeline/7/console

Console

Started by user Tejas C
Obtained JenkinsFile from git https://github.com/TejasC0124/Jenkins-demo.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\Simple-Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\Simple-Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/TejasC0124/Jenkins-demo.git # timeout=10
Fetching upstream changes from https://github.com/TejasC0124/Jenkins-demo.git
> git.exe --version # timeout=10
> git --version # git version 2.52.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/TejasC0124/Jenkins-demo.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision a8b57cf3e4a7b7f3515e8173acbf2e11d6defb1 (refs/remotes/origin/main)
> git checkout -q --force --track a8b57cf3e4a7b7f3515e8173acbf2e11d6defb1 # timeout=10

Console - #7 - Jenkins

localhost:1924/job/Simple-Pipeline/7/console

Stage "Compile" skipped due to earlier failure(s)

[Pipeline] getContext
[Pipeline]]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run)
Stage "Run" skipped due to earlier failure(s)
[Pipeline] getContext
[Pipeline]]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
CI PIPELINE FAILED
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.

Finished: FAILURE

Git Lab

Task 1:

Study the GitLab user interface and identify the main components such as Repository, CI/CD Pipelines, Jobs, and Project Settings. Note the GitLab version used.

The image consists of three vertically stacked screenshots of the GitLab web interface, demonstrating the user interface for managing a project.

Top Screenshot: Repository View

This screenshot shows the "Repository" section of a project named "Gitlab-CI-Demo". The sidebar on the left is open, showing options like Issues, Merge requests, Manage, Plan, Code, Repository (which is selected), README.md, Build, What's new, Help, and Collapse sidebar. The main area displays the "Files" view for the "main" branch, showing a single file "README.md". A commit titled "Initial commit" by "Tejas C 23MIS0671" is shown, dated one hour ago. Below the commit is a table of files with their last commit details. A "Getting started" section provides basic instructions for new users.

Middle Screenshot: Pipelines View

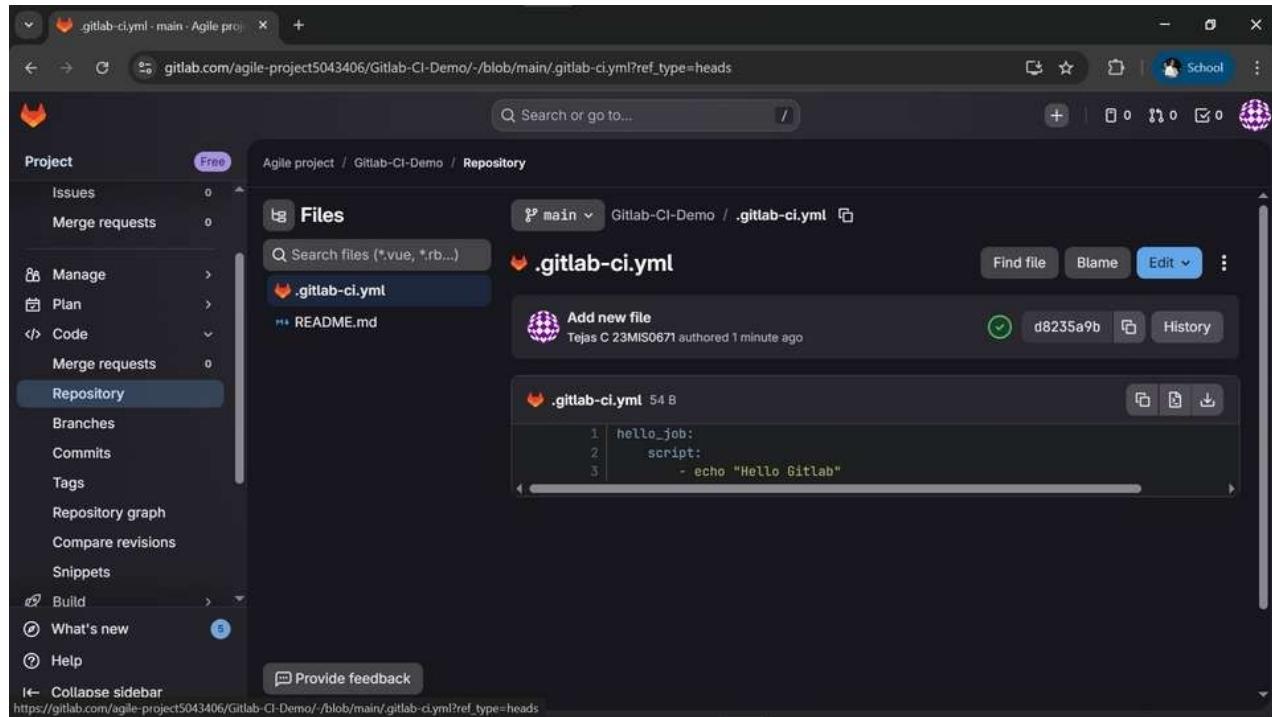
This screenshot shows the "Pipelines" section of the same project. The sidebar is open with the "Pipelines" option selected. The main area features a "Get started with GitLab CI/CD" section with three cards: "Hello world" with GitLab CI (a simple pipeline template), "Learn CI/CD with GitLab University" (guides and videos), and "Easy configuration with CI/CD Catalog" (exploring CI components). Buttons for "Try test template", "Access GitLab University", and "Explore CI/CD Catalog" are present.

Bottom Screenshot: Jobs View

This screenshot shows the "Jobs" section of the project. The sidebar is open with the "Jobs" option selected. The main area displays a list of jobs under the heading "All (0) Finished". A search bar at the top allows filtering by "Kind" (Build, Test, etc.). Below the search bar, a section titled "Use jobs to automate your tasks" explains what jobs are and how to add them to a pipeline. A button "Create CI/CD configuration file" is available.

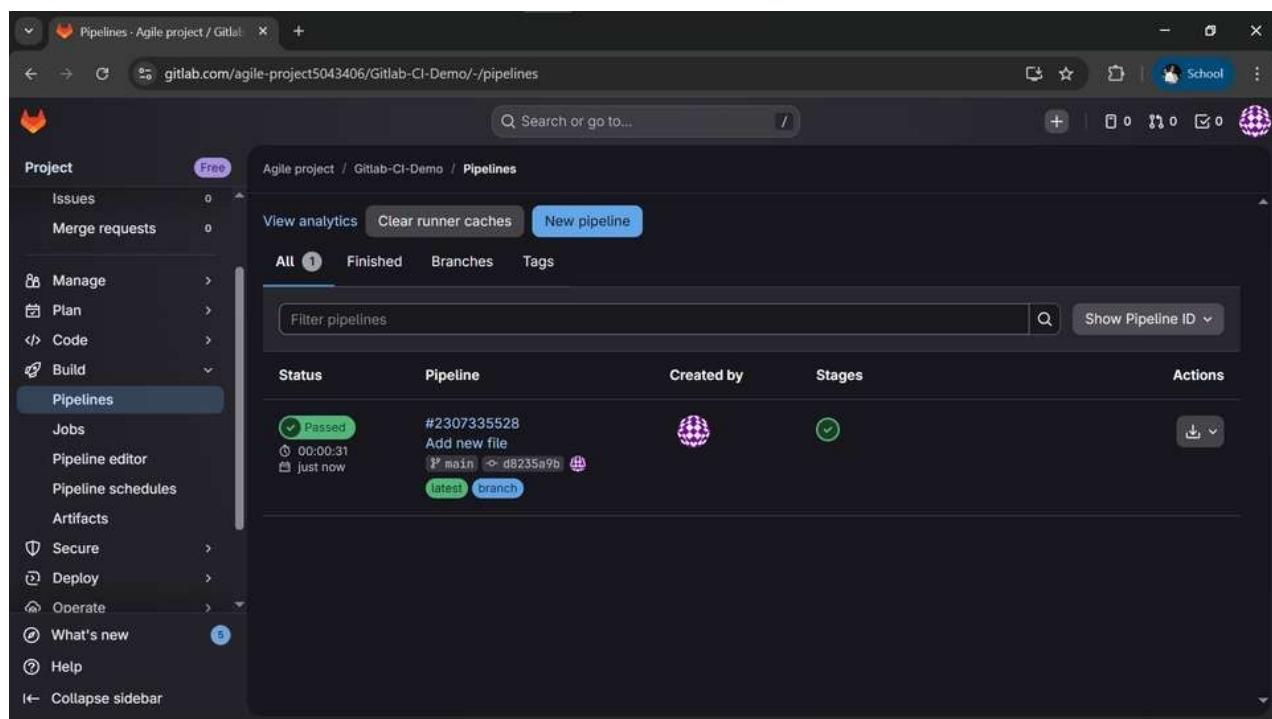
Task 2

Create a basic GitLab CI job that prints a simple message (“Hello GitLab”) using a ` `.gitlab-ci.yml` file and execute the pipeline.



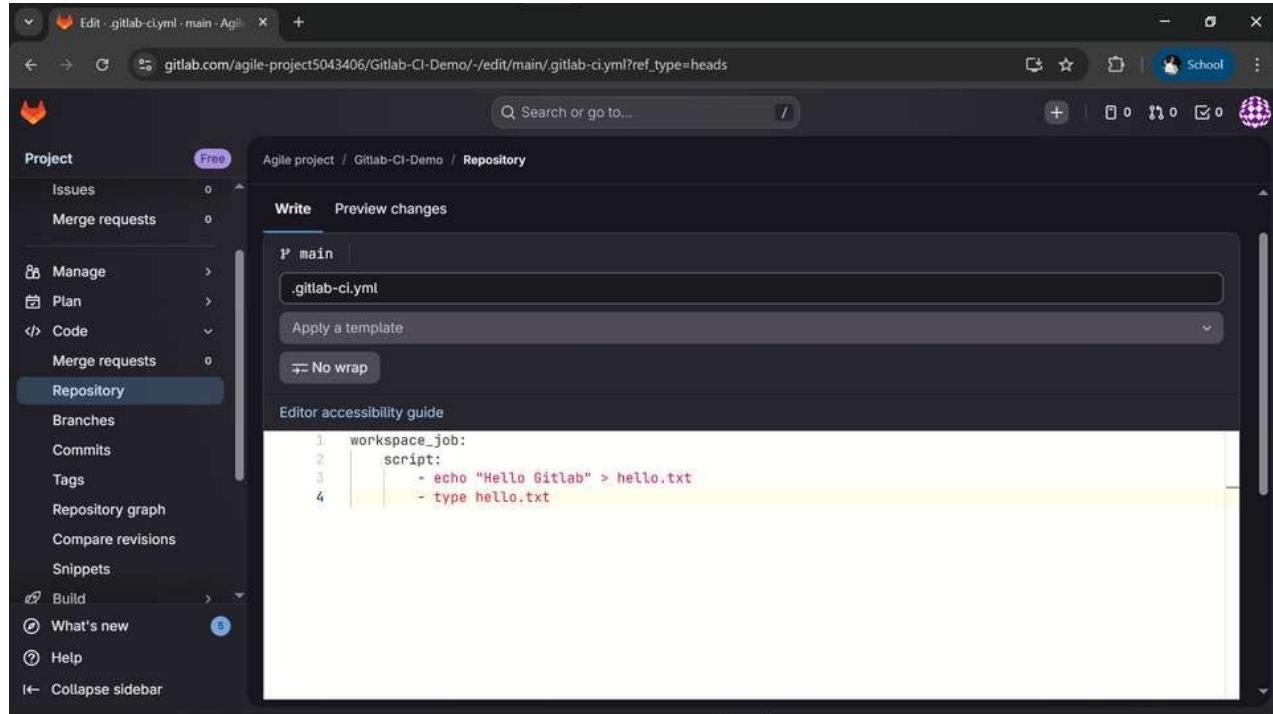
The screenshot shows a GitLab repository interface. On the left, a sidebar menu is open under the 'Project' heading, with 'Repository' selected. The main area displays a file named '.gitlab-ci.yml'. The code content is as follows:

```
hello_job:
  script:
    - echo "Hello Gitlab"
```



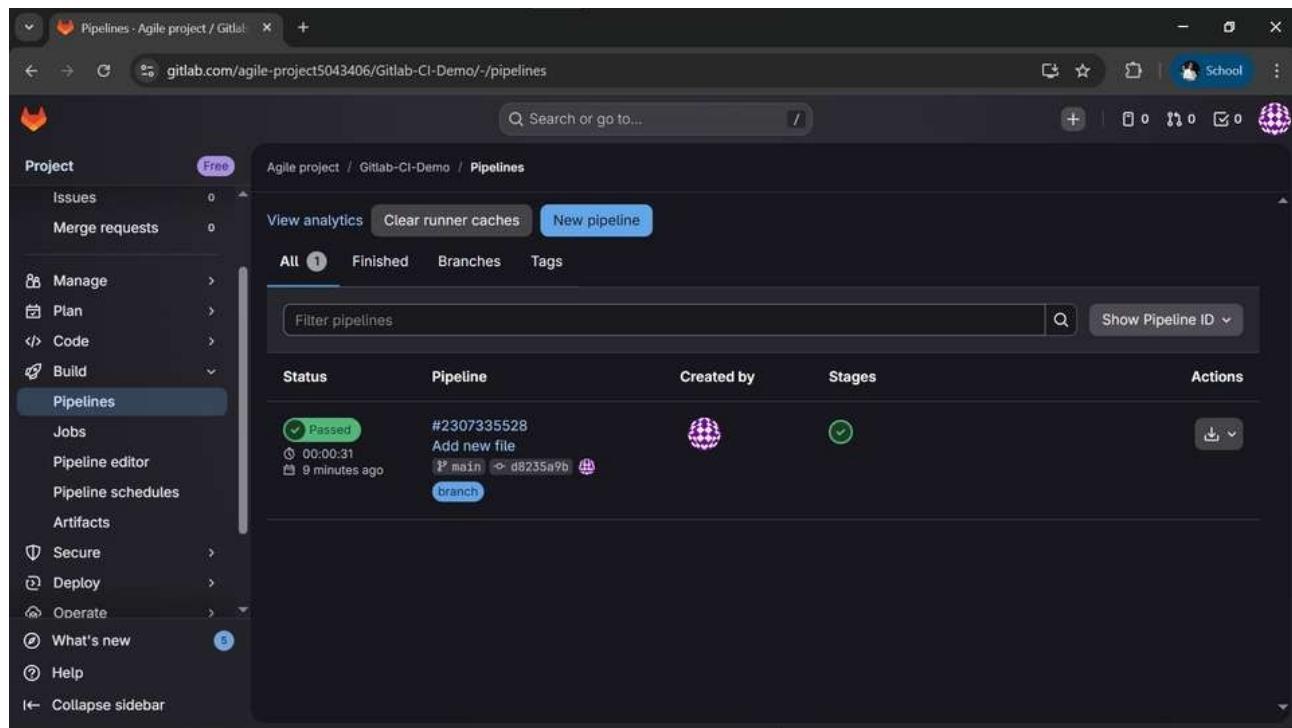
Task 3

Demonstrate workspace usage in GitLab CI by creating a text file during a CI job and displaying its contents in the job logs.



The screenshot shows the GitLab CI editor interface. On the left, a sidebar lists project navigation options like Issues, Merge requests, Manage, Plan, Code, Merge requests, Repository, Branches, Commits, Tags, Repository graph, Compare revisions, Snippets, Build, What's new, Help, and Collapse sidebar. The Repository section is currently selected. The main area displays a code editor titled 'main' containing a '.gitlab-ci.yml' file. The file content is:

```
workspace_job:
  script:
    - echo "Hello Gitlab" > hello.txt
    - type hello.txt
```



The screenshot shows the Pipelines page. The sidebar on the left includes Project, Issues, Merge requests, Manage, Plan, Code, Build (selected), Pipelines, Jobs, Pipeline editor, Pipeline schedules, Artifacts, Secure, Deploy, Operate, What's new, Help, and Collapse sidebar. The main area shows a table of pipelines. One pipeline is listed:

Status	Pipeline	Created by	Stages	Actions
Passed	#2307335528 Add new file 0 00:00:31 9 minutes ago	 d8235a9b		

Task 4

Demonstrate Git integration in GitLab by committing a file to the repository and verifying that the source code is automatically available to the CI pipeline.

The screenshot shows the GitLab interface with the URL gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/commits/main?ref_type=heads. The sidebar on the left is open, showing various project management and development options like 'Manage', 'Plan', 'Code', 'Merge requests', 'Repository', 'Branches', 'Commits' (which is selected), 'Tags', 'Repository graph', 'Compare revisions', 'Snippets', 'Build', 'Secure', 'Deploy', 'What's new', 'Help', and 'Collapse sidebar'. The main area displays the commit history for the 'main' branch. The commits listed are:

- Add new file (Tejas C 23MIS0671 authored just now)
- Edit .gitlab-ci.yml (Tejas C 23MIS0671 authored 5 minutes ago)
- Add new file (Tejas C 23MIS0671 authored 12 minutes ago)
- Initial commit (Tejas C 23MIS0671 authored 1 hour ago)

Each commit has a blue circular icon with a person icon, a commit hash (e.g., 2841ccab, c2c05b79, d8235a9b, 48542924), and standard GitLab file operations buttons (copy, download, etc.).

Task 5

Configure GitLab CI to automatically trigger a pipeline whenever a change is committed to the repository.

The screenshot shows the GitLab interface with the URL gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/edit/main/git.txt?ref_type=heads. The sidebar is open, showing 'Issues' (0), 'Merge requests' (0), 'Manage', 'Plan', 'Code', 'Merge requests' (0), 'Repository' (selected), 'Branches', 'Commits', 'Tags', 'Repository graph', 'Compare revisions', 'Snippets', 'Build', 'What's new' (5), 'Help', and 'Collapse sidebar'. The main area shows an 'Edit file' dialog for the 'git.txt' file in the 'main' branch. The file content is:

```
git integration test
Trigger pipeline
```

The 'Commit changes' button is highlighted in blue at the top right of the dialog. Below the editor, there is an 'Editor accessibility guide' section with two items: 'Git integration test' and 'Trigger pipeline'.

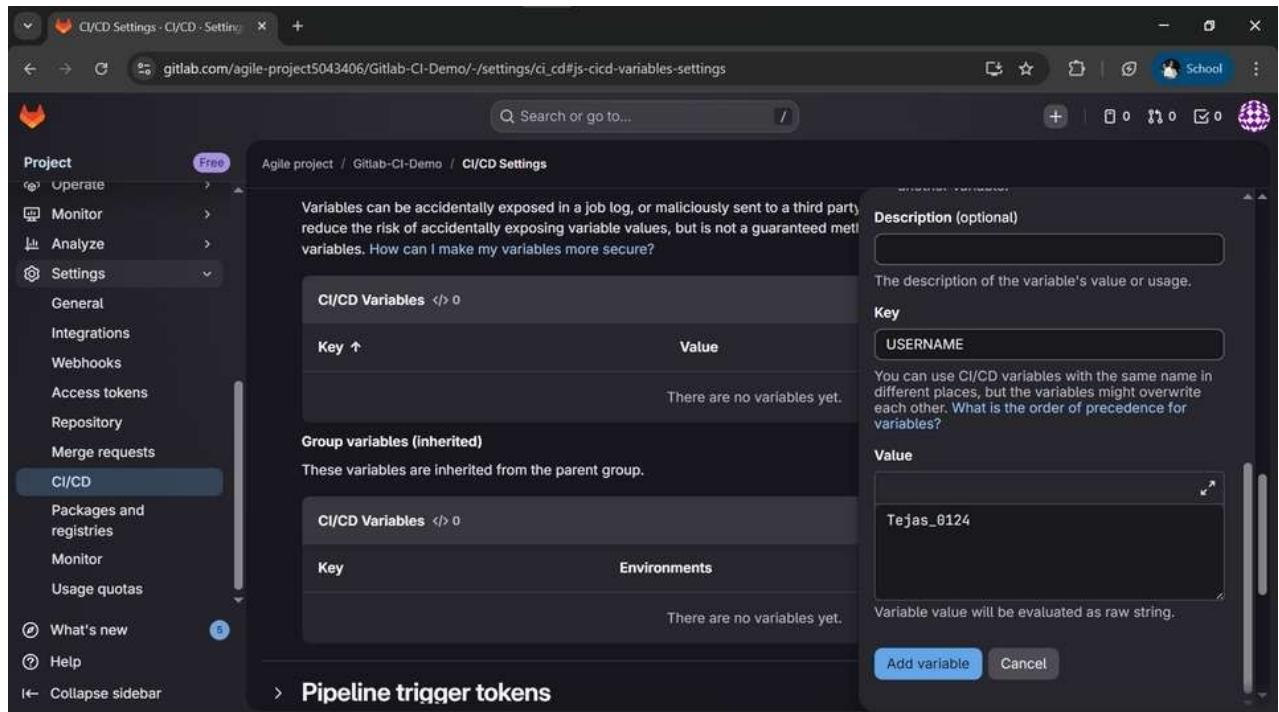
The screenshot shows the GitLab interface with the URL gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/pipelines. The sidebar is open, showing 'Issues' (0), 'Merge requests' (0), 'Manage', 'Plan', 'Code', 'Build', 'Pipelines' (selected), 'Jobs', 'Pipeline editor', and 'Pipeline schedules'. The main area displays the pipeline status. A single pipeline is shown with the following details:

Status	Pipeline	Created by	Stages	Actions
Passed	#2307408087	Edit git.txt		
00:00:30	70 seconds ago	↑ main > 53c6bdfa		
		LateUp branch		

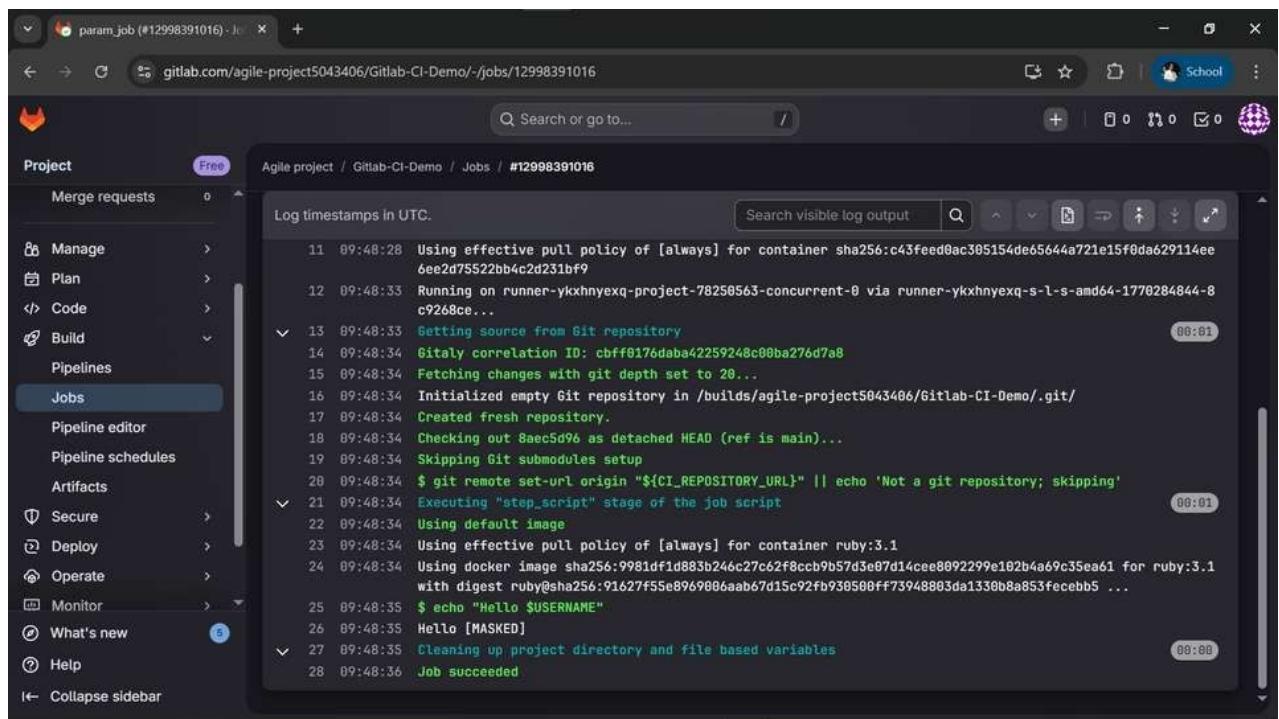
The pipeline status is 'Passed' with a green checkmark icon. The pipeline ID is #2307408087. The stages are represented by small purple globe icons. The 'Actions' column contains three small circular icons with arrows pointing up, down, and right.

Task 6

Implement a parameterized build in GitLab CI using CI/CD variables and display the variable value during pipeline execution.



The screenshot shows the 'CI/CD Variables' settings page in GitLab. On the left sidebar, 'CI/CD' is selected under 'Project'. The main area displays a table for 'CI/CD Variables' with one entry: 'Key: USERNAME' and 'Value: Tejas_8124'. A note states: 'Variables can be accidentally exposed in a job log, or maliciously sent to a third party; reduce the risk of accidentally exposing variable values, but is not a guaranteed method.' A 'Description (optional)' field is present but empty.

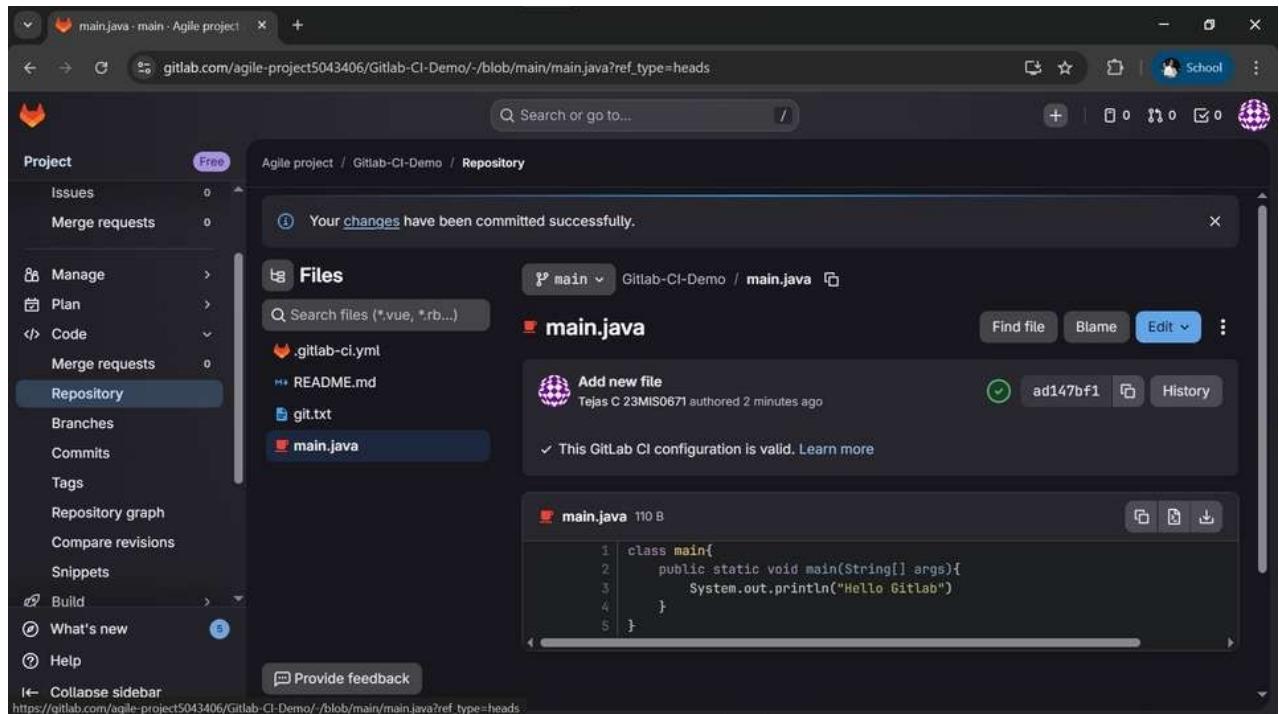


The screenshot shows the log output for a job named 'param_job' with ID '#12998391016'. The log is timestamped in UTC. It shows the execution of a Git repository, setting up environments, and running a step script that prints 'Hello [MASKED]'. The log ends with 'Job succeeded'.

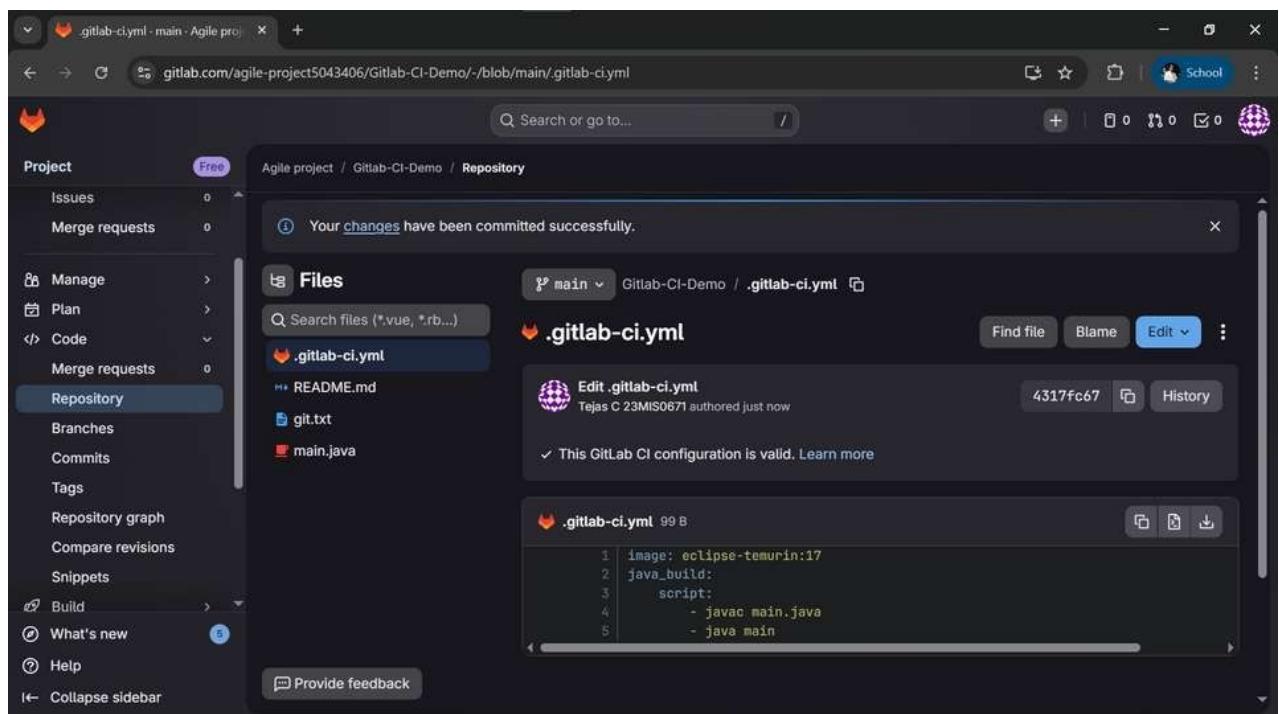
```
11 09:48:28 Using effective pull policy of [always] for container sha256:c43feedbac305154de65644a721e15f0da629114ee6ee2d75522bb4c2d231bf9
12 09:48:33 Running on runner-ykxhnyexq-project-78258563-concurrent-0 via runner-ykxhnyexq-s-l-s-amd64-1770284844-8c9268ce...
13 09:48:33 Getting source from Git repository
14 09:48:34 Gitlab correlation ID: cbff0176daba42259248c00ba276d7a8
15 09:48:34 Fetching changes with git depth set to 20...
16 09:48:34 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
17 09:48:34 Created fresh repository.
18 09:48:34 Checking out 8aec5d96 as detached HEAD (ref is main)...
19 09:48:34 Skipping Git submodules setup
20 09:48:34 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
21 09:48:34 Executing "step_script" stage of the job script
22 09:48:34 Using default image
23 09:48:34 Using effective pull policy of [always] for container ruby:3.1
24 09:48:34 Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969806aab67d15c92fb938500ff73948803da1330b8a853fecebb5 ...
25 09:48:35 $ echo "Hello $USERNAME"
26 09:48:35 Hello [MASKED]
27 09:48:35 Cleaning up project directory and file based variables
28 09:48:36 Job succeeded
```

Task 7

Create a simple Java program and use GitLab CI/CD to compile and execute the Java program using an appropriate Docker image.



The screenshot shows a GitLab repository interface. The sidebar on the left is collapsed. The main area displays a message: "Your changes have been committed successfully." Below this, the "Files" section shows a file named "main.java". The code content is:1 class main{2 public static void main(String[] args){3 System.out.println("Hello GitLab");4 }5 }A status bar at the bottom indicates: "This GitLab CI configuration is valid. Learn more".



The screenshot shows a GitLab repository interface. The sidebar on the left is collapsed. The main area displays a message: "Your changes have been committed successfully." Below this, the "Files" section shows a file named ".gitlab-ci.yml". The code content is:1 image: eclipse-temurin:172 java_build:3 script:4 - javac main.java5 - java mainA status bar at the bottom indicates: "This GitLab CI configuration is valid. Learn more".

The screenshot shows a GitLab pipeline page. The pipeline has passed, created 22 seconds ago by Tejas C 23MIS0671, finished just now. It's for commit a90e88c2, with an option to edit prog.java. The pipeline is in the main branch, with latest branch, containing 1 job (0.34 seconds, queued for 0 seconds). The job test contains a single step java_build, which has passed.

Task 8

Archive build artifacts in GitLab CI by storing the compiled Java class file and make it available for download after the pipeline execution.

The screenshot shows the .gitlab-ci.yml file in the repository. The file defines a job named java_build with a script section containing javac prog.java and java prog. It also specifies an artifacts section with a path to Hello.class. The file was authored by Tejas C 23MIS0671 7 minutes ago.

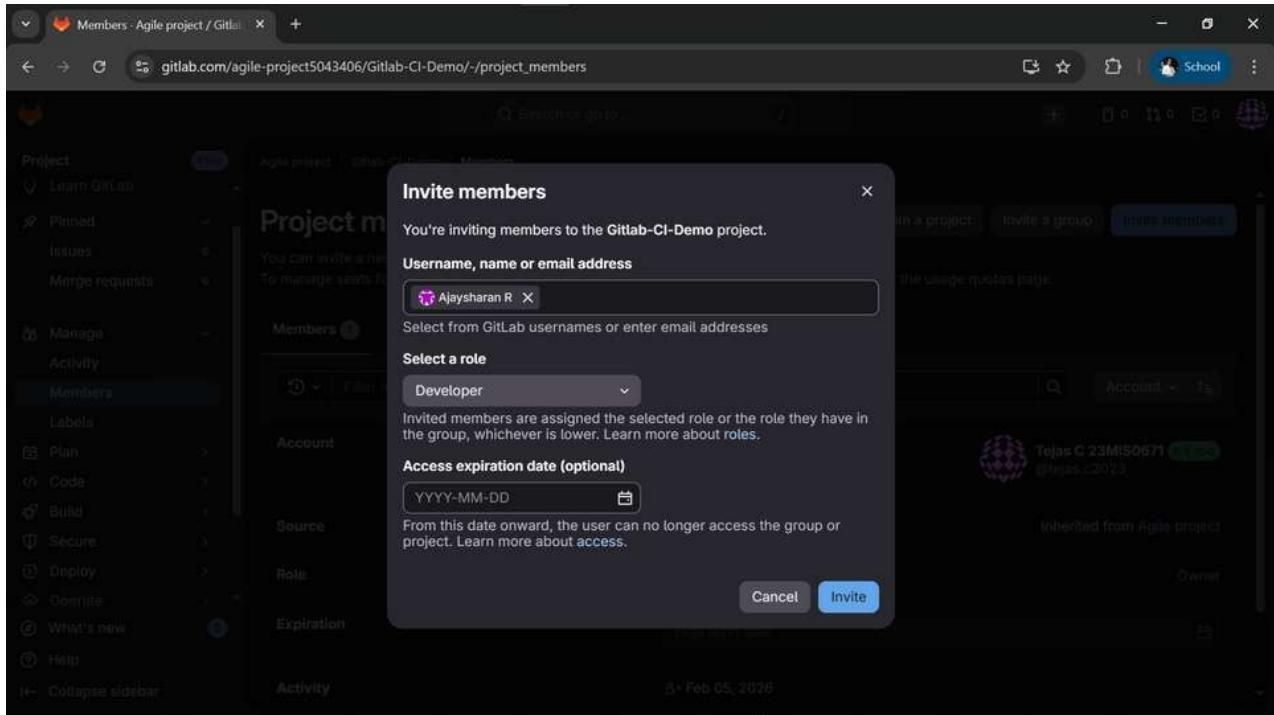
```
image: eclipse-temurin:17
java_build:
  script:
    - javac prog.java
    - java prog
  artifacts:
    paths:
      - Hello.class
```

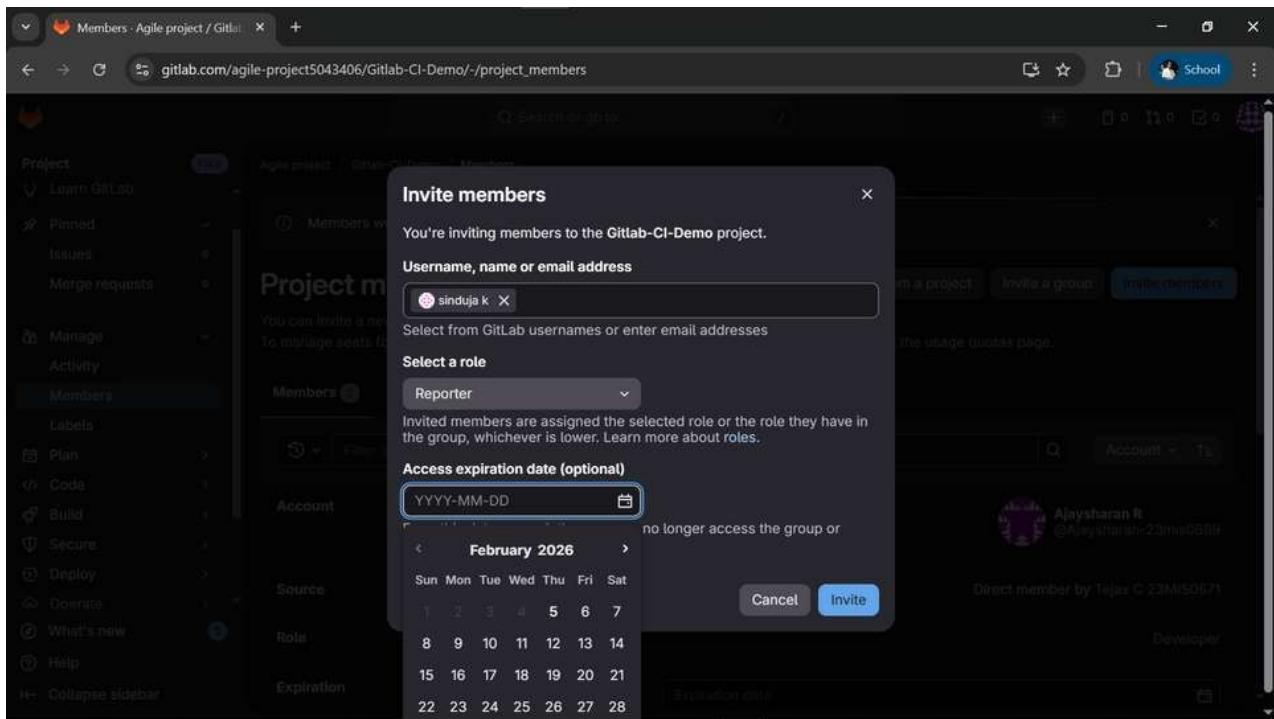
A screenshot of a web browser displaying a GitLab job log. The URL is gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/12998636288. The sidebar on the left shows project navigation options like Merge requests, Pipelines, and Jobs. The main area displays a log of command executions:

```
13 10:03:48 Fetching changes with git depth set to 20...
14 10:03:48 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
15 10:03:48 Created fresh repository.
16 10:03:49 Checking out 29514e27 as detached HEAD (ref is main)...
17 10:03:49 Skipping Git submodules setup
18 10:03:49 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
19 10:03:49 Executing "step_script" stage of the job script
20 10:03:49 Using effective pull policy of [always] for container eclipse-temurin:17
21 10:03:49 Using docker image sha256:53c2311899d38e8908f3c7cf494520df8de320402bc4770d10e1d25fab1fff15 for eclipse-temurin:17 with digest eclipse-temurin@sha256:710bbe5d41a4c48ecd1e8d5be5f05d49132a102ab70961006d0675ed8b387d86 ...
22 10:03:50 $ javac prog.java
23 10:03:50 $ java prog
24 10:03:50 Hello GitLab
25 10:03:51 Uploading artifacts for successful job
26 10:03:51 Uploading artifacts...
27 10:03:51 WARNING: Hello.class: no matching files. Ensure that the artifact path is relative to the working directory (/builds/agile-project5043406/Gitlab-CI-Demo)
28 10:03:51 ERROR: No files to upload
29 10:03:51 Cleaning up project directory and file based variables
30 10:03:52 Job succeeded
```

Task 9

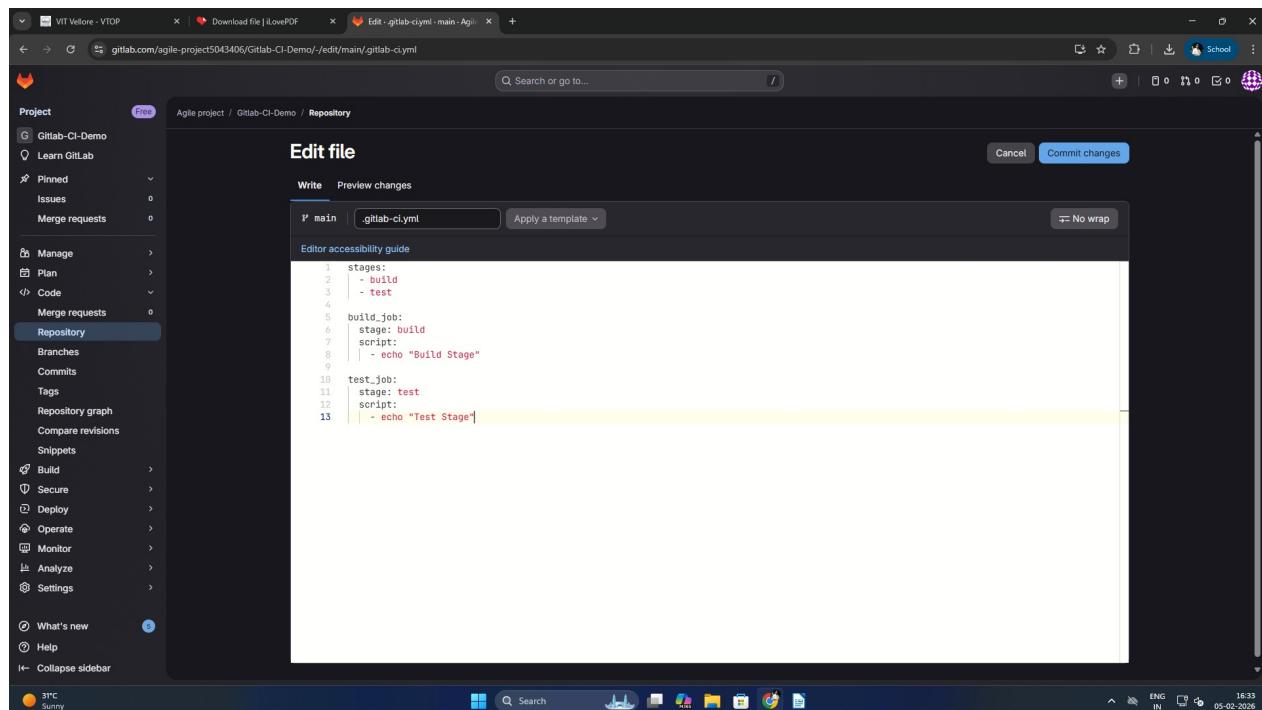
Demonstrate user and role management in GitLab by assigning read-only access to one user and build access to another user within a project.





Task 10

Create a simple GitLab CI pipeline consisting of multiple stages such as Build and Test, and visualize the stage-wise execution.



gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/pipelines

Pipelines

Status	Pipeline	Created by	Stages	Actions
Passed	#2307596896	Scal3r	build, test	View pipeline

#2307596896

Passed Created 1 minute ago by Tejas C 23MIS0671, finished 32 seconds ago

For commit Scal3r

In main

Latest branch: go 2 jobs 1.03 1 minute 1 second, queued for 1 seconds

Pipeline Jobs Tests

```

graph LR
    build[build] --> test[test]
    build --> build_job[build_job]
    test --> test_job[test_job]

```

What's new Help Collapse sidebar

31°C Sunny

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/12999603348

test_job

Passed Started 2 minutes ago by Tejas C 23MIS0671

Log timestamps in UTC.

```

1 11:04:13 Running with gitlab-runner 18.7.0-pre.433.g3af2314 (3a5f2314)
2 11:04:13 on green-3.saa5-linux-small-and64.runners-manager.gitlab.com/default Jhc_Jxvh8, system ID: s_0e6850b20ce1
3 11:04:13 Preparing the "docker-machine" executor
4 11:04:13 Using default image
5 11:04:13 Using Docker executor with image ruby:3.1 ...
6 11:04:18 Using default image
7 11:04:18 Pulling docker image sha256:99810ff10883b246c27cd2f8cc9b057d3d07d14cce8092299e102b4ad0c35e61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92fb930500ff7394803da1330ba053fce0b5 ...
8 11:04:34 Using effective pull policy of [always] for container ruby:3.1 ...
9 11:04:34 Using docker image sha256:99810ff10883b246c27cd2f8cc9b057d3d07d14cce8092299e102b4ad0c35e61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92fb930500ff7394803da1330ba053fce0b5 ...
10 11:04:34 Preparing environment
11 11:04:34 Using effective pull policy of [always] for container sha256:b43fc5a326b924bf2b2fe5b2d4a32454398e67dca27d83c82d4497fad0ba4
12 11:04:39 Running on runner-jhcJxvh8-project-7820563-concurrent-0 via runner-jhcJxvh8-s-l-s-amd64-1778289326-f8f9db2b...
13 11:04:39 Getting source from Git repository
14 11:04:39 Gitely correlation ID: 4e31e5e0f294cb9056db42ac92f25f
15 11:04:39 Fetching changes with git depth set to 20
16 11:04:39 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
17 11:04:39 Created fresh repository
18 11:04:40 Checking out Scal3r as detached HEAD (ref is main)...
19 11:04:40 Skipping Git submodules setup
20 11:04:40 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
21 11:04:41 Executing "step_script" stage of the job script
22 11:04:41 Using default image
23 11:04:41 Using effective pull policy of [always] for container ruby:3.1
24 11:04:41 Using docker image sha256:99810ff10883b246c27cd2f8cc9b057d3d07d14cce8092299e102b4ad0c35e61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92fb930500ff7394803da1330ba053fce0b5 ...
25 11:04:41 $ echo "Test Stage"
26 11:04:41 Test Stage
27 11:04:41 Cleaning up project directory and file based variables
28 11:04:42 Job succeeded

```

Duration: 31 seconds
 Finished: 1 minute ago
 Queued: 0 seconds
 Timeout: 1h (from project)
 Runner: #12270852 (Jhc_Jxvh8)-3-green.saa5-linux-small-and64.runners-manager.gitlab.com/default
 Source: Push
 Commit Scal3r
 Edit .gitlab-ci.yml
 Pipeline #2307596896 Passed for main
 test
 Related jobs → test_job

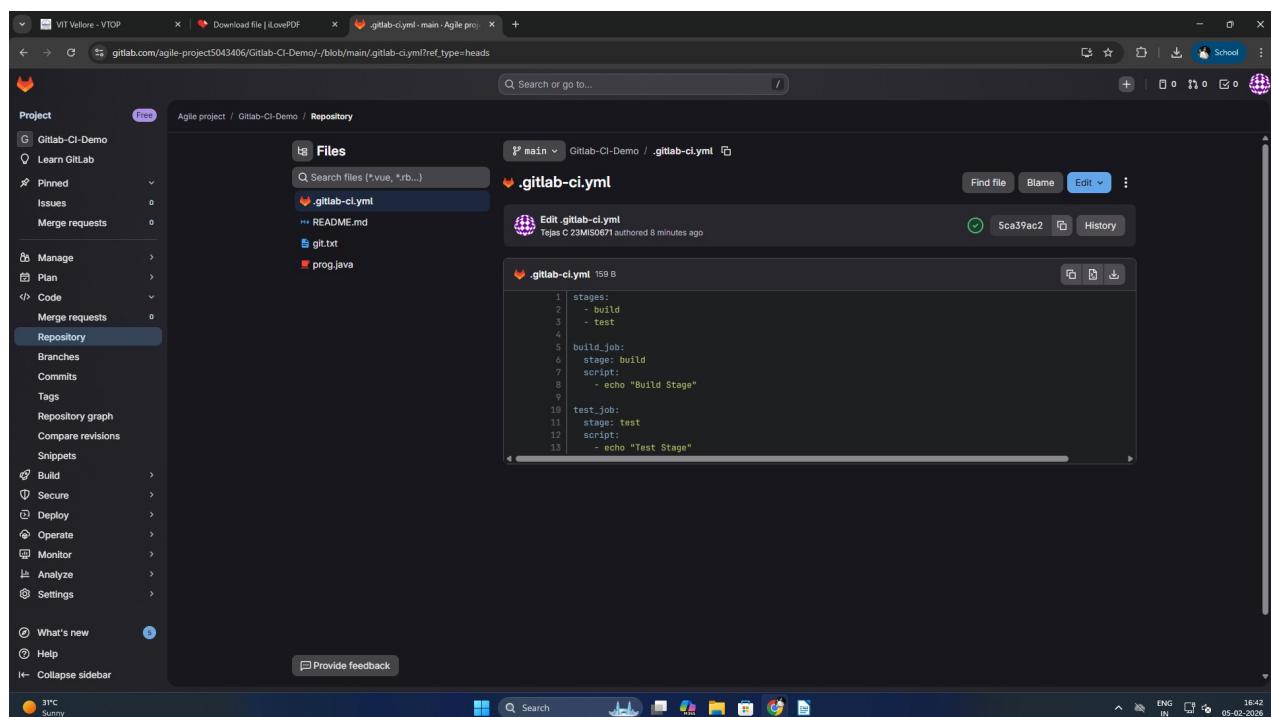
What's new Help Collapse sidebar

31°C Sunny

Task 11

Implement Pipeline as Code in GitLab by defining the CI pipeline using a ‘.gitlab-ci.yml’ file stored in the Git repository and executing it automatically on commit.

In GitLab CI/CD, pipeline are expressed through code in the form of a configuration file named .gitlab-ci.yml, which is maintained in the Git repository of the project. This makes it possible to manage the pipeline logic through version control, along with the source code. Every time there is a commit to the repository, GitLab automatically reads the .gitlab-ci.yml file and performs the pipeline stages and jobs as defined. This enhances consistency and traceability in the continuous integration process.

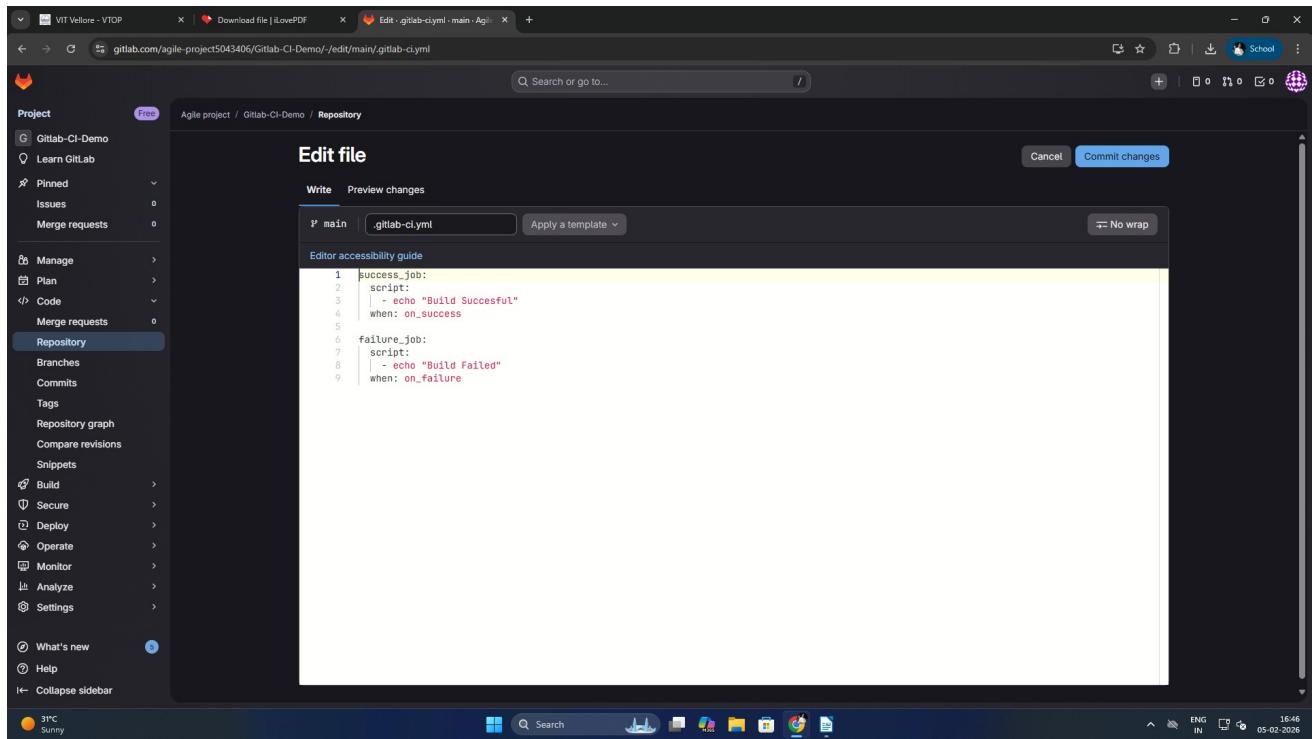


The screenshot shows a GitLab repository interface. The sidebar on the left is titled 'Project' and lists various sections: Gittab-CI-Demo, Learn GitLab, Pinned Issues (0), Merge requests (0), Manage Plan, Code (prog.java), Repository (selected), Branches, Commits, Tags, Repository graph, Compare revisions, Snippets, Build, Secure, Deploy, Operate, Monitor, Analyze, Settings, What's new (5), and Help. The main area shows a file tree for 'main' branch: .gitlab-ci.yml (selected), README.md, git.txt, and prog.java. A preview window for '.gitlab-ci.yml' shows the following YAML code:

```
stages:
- build
- test
build_job:
stage: build
script:
- echo "Build Stage"
test_job:
stage: test
script:
- echo "Test Stage"
```

Task 12

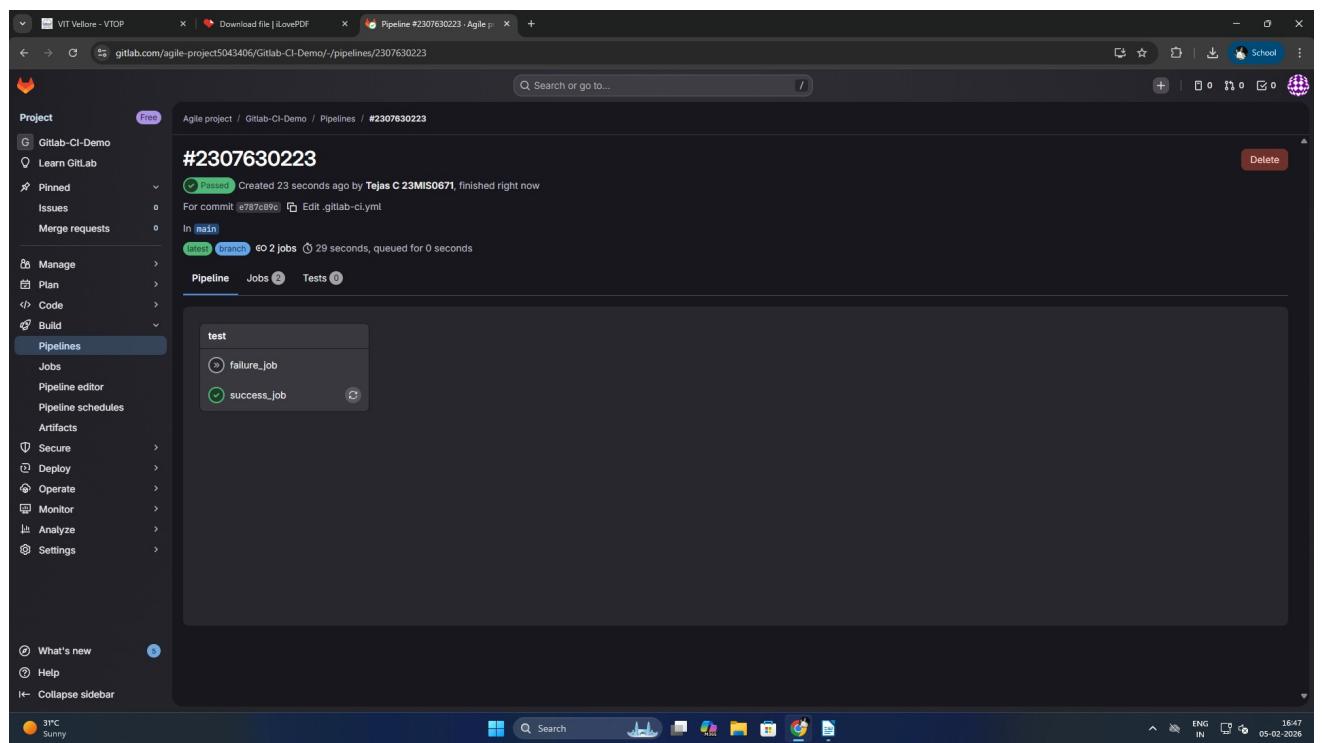
Demonstrate post-build actions in GitLab CI by displaying appropriate messages based on pipeline success or failure.



The screenshot shows the GitLab web interface with the project 'Gitlab-CI-Demo'. The user is editing the '.gitlab-ci.yml' file in the 'main' branch. The code contains two jobs: 'success_job' which prints 'Build Successful' on success, and 'failure_job' which prints 'Build Failed' on failure.

```
success_job:
  script:
    - echo "Build Successful"
  when: on_success

failure_job:
  script:
    - echo "Build Failed"
  when: on_failure
```



VIT Vellore - VTOP | Download file | iLovePDF | success_job (#12999822738) | +

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/12999822738

Project Free

G Gitlab-CI-Demo

Learn GitLab

Pinned

Issues

Merge requests

Manage

Plan

Code

Build

Pipelines

Jobs

Pipeline editor

Pipeline schedules

Artifacts

Secure

Deploy

Operate

Monitor

Analyze

Settings

What's new

Help

Collapse sidebar

success_job

Passed Started 49 seconds ago by Tejas C 23MIS0671

Log timestamps in UTC.

Search visible log output

Duration: 29 seconds

Finished: 19 seconds ago

Queued: 0 seconds

Timeout: 1h (from project)

Runner: #1270945 (JLgUopmM) 1-green.saas-linux-small-amd64.runners-manager.gitlab.com/default

Source: Push

Commit #787c99c Edit .gitlab-ci.yml

Pipeline #2307630223 Passed for main

test

Related jobs

→ success_job

↳ failure.job

Log timestamp in UTC.

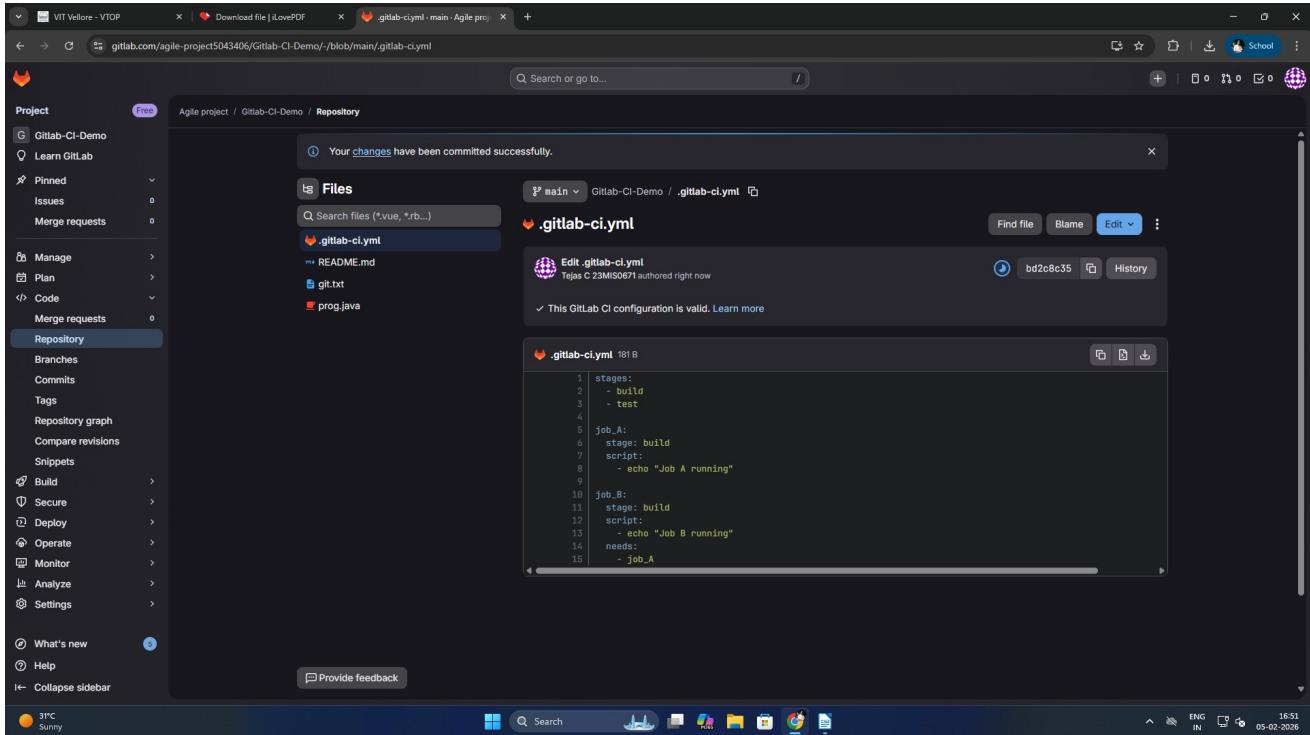
1 11:16:46 Running with gitlab-runner 18.7.0-pre.433.g3a5f2314 (3a5f2314)
2 11:16:46 On green-1.saas-linux-small-amd64.runners-manager.gitlab.com/default JLgUopmM, system ID: s_deaa2ca09de7
3 11:16:46 Preparing the "docker-machine" executor
4 11:16:46 Using default image
5 11:16:46 Using Docker executor with image ruby:3.1 ...
6 11:16:50 Using default image
7 11:16:50 Using effective pull policy of [always] for container ruby:3.1 ...
8 11:16:50 Pulling docker image ruby:3.1 ...
9 11:17:06 Using docker image sha256:9981df1d0883b24cc27c62ff8ccb9b57d3e07d14cee8892299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e896986aab67d15c92fb930580ff73948803da1330b8a853fecebb5 ...
10 11:17:06 Preparing environment
11 11:17:06 Using effective pull policy of [always] for container sha256:57200d11762d004e23bbb8d00ea72ccf4317cb65677db3699c0cab4fa8a846
12 11:17:10 Running on runner-jlguopmvm-project-78250563-concurrent-0 via runner-jlguopmvm-s-1-s-and64-1770298141-ab172125 ...
13 11:17:11 Getting source from Git repository
14 11:17:11 GitLab correlation ID: 8833fc39fcfa94340ba1a08219378869b
15 11:17:11 Fetching changes with git depth set to 20...
16 11:17:11 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
17 11:17:11 Created fresh repository.
18 11:17:12 Checking out e787c99c as detached HEAD (ref is main)...
19 11:17:12 Skipping Git submodules setup
20 11:17:12 \$ git remote set-url origin "\$CI_REPOSITORY_URL" || echo 'Not a git repository: skipping'
21 11:17:12 Executing "step_script" stage of the job script
22 11:17:12 Using default image
23 11:17:12 Using effective pull policy of [always] for container ruby:3.1
24 11:17:12 Using docker image sha256:9981df1d0883b24cc27c62ff8ccb9b57d3e07d14cee8892299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e896986aab67d15c92fb930580ff73948803da1330b8a853fecebb5 ...
25 11:17:12 \$ echo "Build Successful"
26 11:17:12 Build Successful
27 11:17:12 Cleaning up project directory and file based variables
28 11:17:13 Job succeeded

31°C Sunny

ENG IN 05-02-2026 16:47

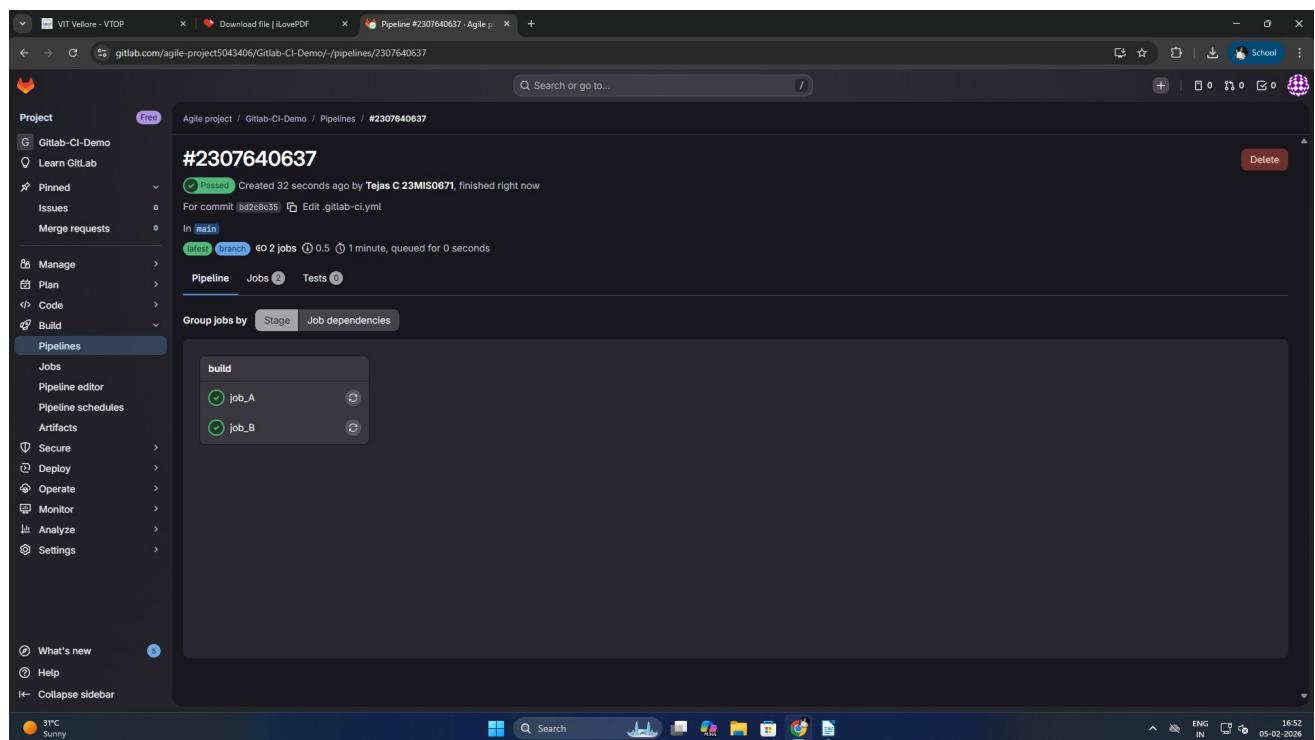
Task 13

Implement job chaining in GitLab CI such that one job is triggered only after the successful completion of another job.



The screenshot shows the GitLab repository interface for the project "Gitlab-CI-Demo". The sidebar on the left is collapsed. The main area displays the ".gitlab-ci.yml" file. A message at the top says "Your changes have been committed successfully." The file content is as follows:

```
stages:
- build
- test
job_A:
stage: build
script:
- echo "Job A running"
job_B:
stage: build
script:
- echo "Job B running"
needs:
- job_A
```



VIT Vellore - VTOP

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/12999894441

job_A

Passed Started 1 minute ago by Tejas C 23MIS0671

Log timestamps in UTC.

```

1 11:21:40 Running with gitlab-runner 18.7.0-pre.433.g3a5f2314 (3a5f2314)
2 11:21:40 on green-3.saaS-linux-small-and64.runners-manager.gitlab.com/default Jhc_Jxvh8, system ID: s_0e6850b2bce1
3 11:21:40 Preparing the "docker-machine" executor
4 11:21:40 Using default image
5 11:21:40 Using Docker executor with image ruby:3.1 ...
6 11:21:45 Using default image
7 11:21:45 Using effective pull policy of [always] for container ruby:3.1
8 11:21:45 Pulling docker image ruby:3.1...
9 11:22:00 Using docker image sha256:9981df1d883b24cc27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92d97394883d1330b8a853fecebb5 ...

10 11:22:00 Preparing environment
11 11:22:00 Using effective pull policy of [always] for container sha256:db2d8388a18732d6877cf7fy9alba4ccaf8fffa5d2b2b29397249ecdb6cef1c8
12 11:22:05 Running on runner-jhcjxvh8-project-7625953-concurrent-0 via runner-jhcjxvh8-s-l-s-amd64-1770290455-8ce351fa...
13 11:22:05 Getting source from Git repository
14 11:22:05 GitLab correlation ID: 220912ca80d428288407ed921aba95
15 11:22:05 Fetching changes with git depth set to 20...
16 11:22:05 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
17 11:22:05 Created fresh repository.
18 11:22:05 Checking out b62c8c35 as detached HEAD (ref is main)...
19 11:22:05 Skipping Git submodules setup
20 11:22:05 $ git remote set-url origin "$CI_REPOSITORY_URL" || echo 'Not a git repository; skipping'
21 11:22:07 Executing "step_script" stage of the job script
22 11:22:07 Using default image
23 11:22:07 Using effective pull policy of [always] for container ruby:3.1
24 11:22:07 Using docker image sha256:9981df1d883b24cc27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92d97394883d1330b8a853fecebb5 ...
25 11:22:07 $ echo "Job A running"
26 11:22:07 Job A running
27 11:22:07 Cleaning up project directory and file based variables
28 11:22:08 Job succeeded

```

Duration: 30 seconds
 Finished: 1 minute ago
 Queued: 0 seconds
 Timeout: 1h (from project)
 Runner: #12270852 (Jhc_Jxvh8)-green.saaS-linux-small-and64.runners-manager.gitlab.com/default
 Source: Push
 Commit bd2c8c35
 Edit.gitlab-ci.yml
 Pipeline #2307640637 Passed
 for main build
 Related jobs
 → job_A
 → job_B

16:53 ENG IN 05-02-2026

VIT Vellore - VTOP

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/12999894445

job_B

Passed Started 1 minute ago by Tejas C 23MIS0671

Log timestamps in UTC.

```

1 11:22:10 Running with gitlab-runner 18.7.0-pre.433.g3a5f2314 (3a5f2314)
2 11:22:10 on green-3.saaS-linux-small-and64.runners-manager.gitlab.com/default Jhc_Jxvh8, system ID: s_0e6850b2bce1
3 11:22:10 Preparing the "docker-machine" executor
4 11:22:10 Using default image
5 11:22:10 Using Docker executor with image ruby:3.1 ...
6 11:22:15 Using default image
7 11:22:15 Using effective pull policy of [always] for container ruby:3.1
8 11:22:15 Pulling docker image ruby:3.1...
9 11:22:31 Using docker image sha256:9981df1d883b24cc27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92d97394883d1330b8a853fecebb5 ...

10 11:22:31 Preparing environment
11 11:22:31 Using effective pull policy of [always] for container sha256:954fb36176dc14cb373540ee7b6a611589b4ad0fb588ea7863029f3611af0a3
12 11:22:35 Running on runner-jhcjxvh8-project-7625953-concurrent-0 via runner-jhcjxvh8-s-l-s-amd64-1770290488-cb18de67...
13 11:22:36 Getting source from Git repository
14 11:22:36 GitLab correlation ID: 3ee50fc8c04a340e8a4cc6bb1c19e11
15 11:22:36 Fetching changes with git depth set to 20...
16 11:22:36 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
17 11:22:36 Created fresh repository.
18 11:22:37 Checking out b62c8c35 as detached HEAD (ref is main)...
19 11:22:37 Skipping Git submodules setup
20 11:22:37 $ git remote set-url origin "$CI_REPOSITORY_URL" || echo 'Not a git repository; skipping'
21 11:22:37 Executing "step_script" stage of the job script
22 11:22:37 Using default image
23 11:22:37 Using effective pull policy of [always] for container ruby:3.1
24 11:22:37 Using docker image sha256:9981df1d883b24cc27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92d97394883d1330b8a853fecebb5 ...
25 11:22:37 $ echo "Job B running"
26 11:22:37 Job B running
27 11:22:38 Cleaning up project directory and file based variables
28 11:22:38 Job succeeded

```

Duration: 29 seconds
 Finished: 52 seconds ago
 Queued: 0 seconds
 Timeout: 1h (from project)
 Runner: #12270852 (Jhc_Jxvh8)-green.saaS-linux-small-and64.runners-manager.gitlab.com/default
 Source: Push
 Commit bd2c8c35
 Edit.gitlab-ci.yml
 Pipeline #2307640637 Passed
 for main build
 Related jobs
 → job_A
 → job_B

16:53 ENG IN 05-02-2026

Task 14

Demonstrate workspace cleanup in GitLab CI to ensure that previous build files are removed before executing a new pipeline.

The screenshot shows the 'Edit file' interface in GitLab. The left sidebar is for the 'Gittab-CI-Demo' project, with 'Repository' selected. The main area shows a file named 'gitlab-ci.yml' with the following content:

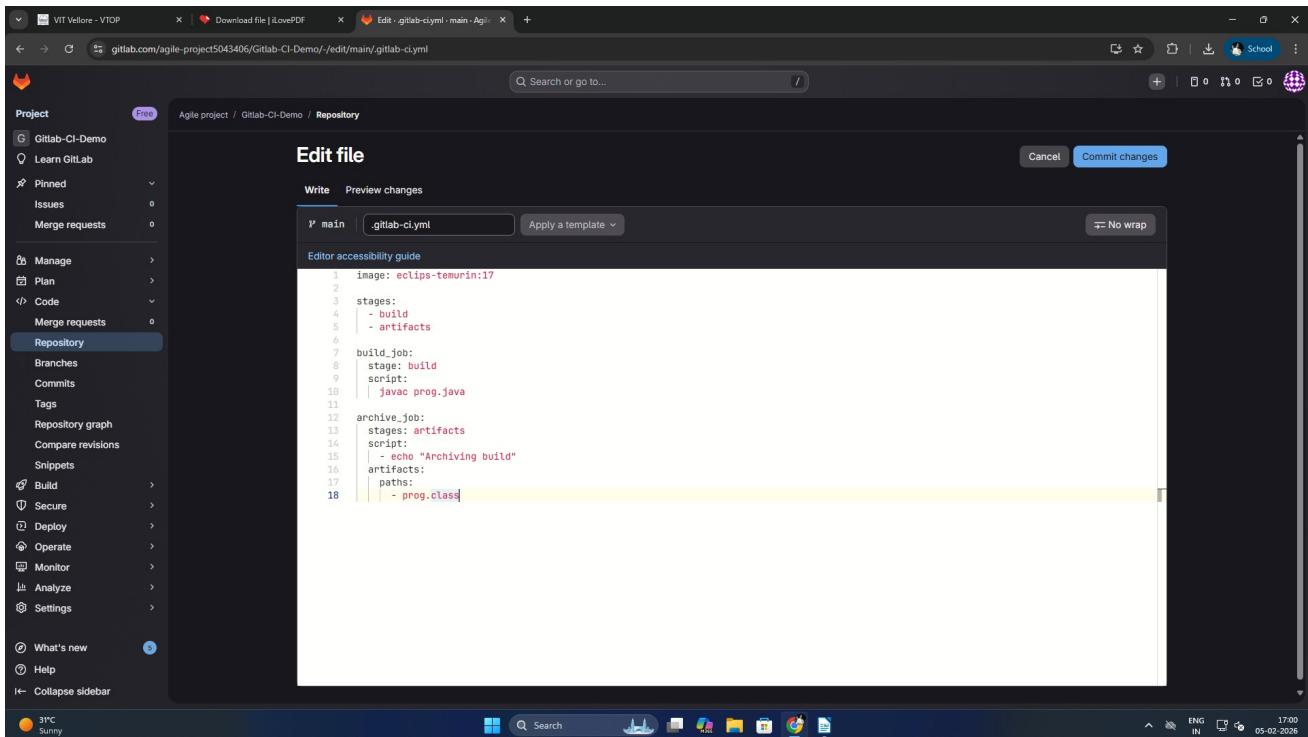
```
cleanup_job:
  script:
    - rm -rf *
```

The screenshot shows the 'Jobs' interface for the 'cleanup_job' in the 'Gittab-CI-Demo' project. The log output shows the execution of the cleanup job:

```
Running with gitlab-runner 18.7.0-pre.433.g3a5f2314 (3a5f2314)
on green-6.saa5-linux-small-amd64.runners-manager.gitlab.com/default YKxHnyexq, system ID: s_a201ab37b78a
Preparing the 'docker-machine' executor
Using default image
Using Docker executor with image ruby:3.1 ...
Using default image
Using effective pull policy of [always] for container ruby:3.1
Pulling docker image ruby:3.1 ...
Using docker image sha256:9881df1d0883b246c27c62f8ccb9b57d3e07d14ceee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92f938500f73948883d1330b8a853feceeb5 ...
Preparing environment
Using effective pull policy of [always] for container sha256:e8e623b8c929d85ab783ed3e13ce0cb58#883538263d8979de7bbdcfb9897c38
Running on runner-ykxhnyexq-project-78250563-concurrent-0 via runner-ykxhnyexq-s-1s-amd64-1770290726-f31f8fb...
Getting source from Git repository
GitLab correlation ID: 26814ffaf07ad4f83a24f07afacd96f8f
Fetching changes with git depth set to 20...
Initialized empty Git repository in /builds/agile-project5043406/gittab-CI-Demo/.git/
Created fresh repository.
Checking out 0@ed631 as detached HEAD (ref is main)...
Skipping Git submodules setup
$ git remote set-url origin "$CI_REPOSITORY_URL" || echo 'Not a git repository, skipping'
Executing 'step_script' stage of the job script
Using default image
Using effective pull policy of [always] for container ruby:3.1
Using docker image sha256:9881df1d0883b246c27c62f8ccb9b57d3e07d14ceee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c92f938500f73948883d1330b8a853feceeb5 ...
$ rm -rf *
Cleaning up project directory and file based variables
Job succeeded
```

Task 15

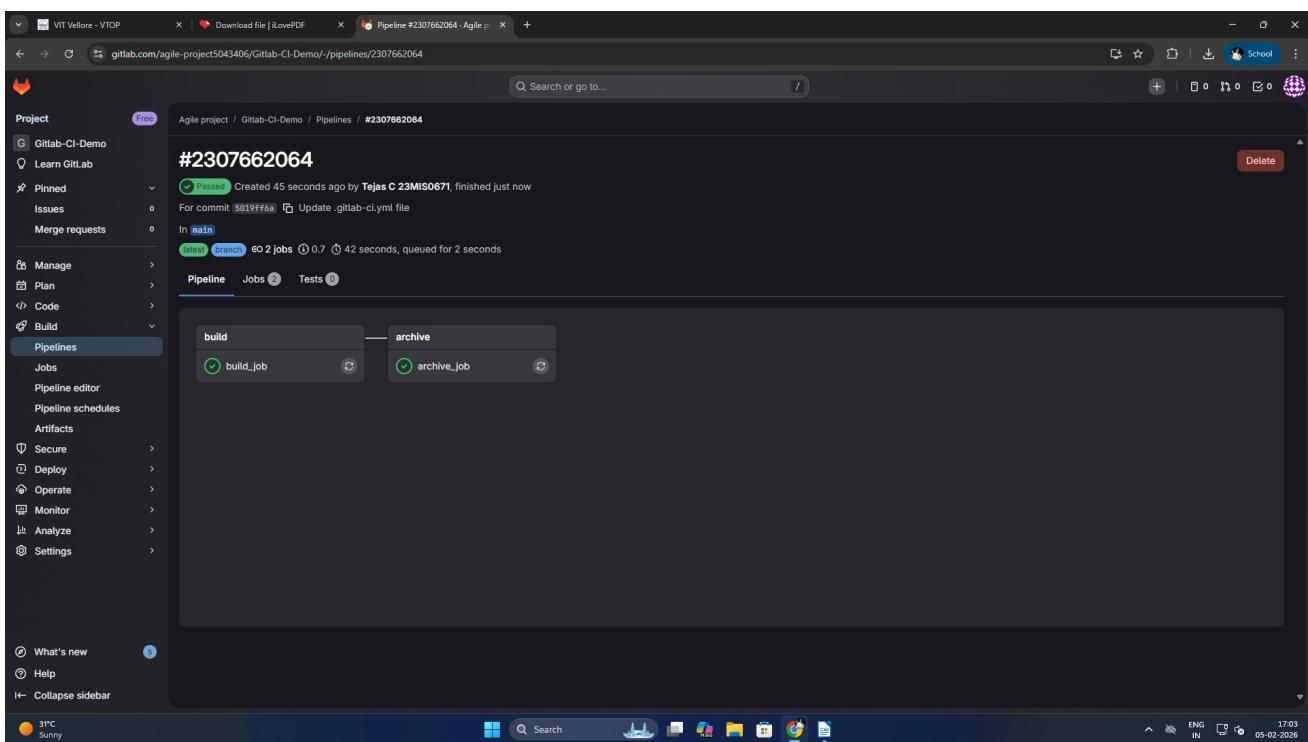
Implement a Mini Continuous Integration (CI) project using GitLab CI/CD where a code commit automatically triggers a pipeline to compile a java program, archive build artifacts, and fail the build in case of errors.



The screenshot shows the GitLab web interface with a dark theme. On the left is a sidebar with various project management and development tools like Issues, Merge requests, Plan, Code, Repository, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main area is titled "Edit file" and shows the ".gitlab-ci.yml" file content:

```
image: eclipse-temurin:17
stages:
- build
- artifacts
build_job:
stage: build
script:
| - javac prog.java
archive_job:
stages: artifacts
script:
| - echo "Archiving build"
artifacts:
paths:
| - prog.class
```

At the top right of the editor are "Cancel" and "Commit changes" buttons. Below the editor is an "Editor accessibility guide". The bottom right corner of the screen shows system status: 31°C, Sunny, ENG IN, 05-02-2026, 17:00.



VIT Vellore - VTOP | Download file | iLovePDF | build_job (#13000041846) · Job

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/13000041846

Project Agile project / Gitlab-CI-Demo / Jobs / #13000041846

build_job

Passed Started 1 minute ago by Tejas C 23MIS0671

Log timestamps in UTC.

```

1 11:37:59 Running with gitlab-runner 18.7.0-pre.433.g3af5f2314 (3af5f2314)
2 11:37:59 On green-4.saaS-linux-small-amd64.runners-manager.gitlab.com/default ntHFEtyXQ, system ID: s_8999de21c550
3 11:37:59 Preparing the "docker-machine" executor
4 11:37:59 Using Docker executor with image eclipse-temurin:17 ...
5 11:33:04 Using effective pull policy of [always] for container eclipse-temurin:17
6 11:33:04 Pulling docker image eclipse-temurin:17 ...
7 11:33:11 Using docker image sha256:53c231189938e8988f3c7cf494520fd8de328402bc4778d10e1d25fab1ff15 for eclipse-temurin:17 with digest eclipse-temurin@sha256:a710bbe5d1a1c4c48ed1e8d5be5f05d49132a102ab79961086d0675ed8b387d8d ...
8 11:33:11 Preparing environment
9 11:33:11 Using effective pull policy of [always] for container sha256:1967ecd593e5b212258593a03280fd3c8b1ca9c183e7ce40f645ceccbd1d135b226
10 11:33:11 Running on runner-ntHFEtyXQ-project-78250563-concurrent-0 via runner-ntHFEtyXQ-s-l-s-amd64-1778291134-fd6c88bb...
11 11:33:15 Getting source from Git repository
12 11:33:15 GitLab correlation ID: 83a4b6f03e744d69fc462f6ca7d74f0
13 11:33:15 Fetching changes with git depth set to 20...
14 11:33:15 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
15 11:33:15 Created fresh repository.
16 11:33:16 Checking out 5019ff6a as detached HEAD (ref is main)...
17 11:33:16 Skipping Git submodules setup
18 11:33:16 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
19 11:33:16 Executing "step_script" stage of the job script
20 11:33:16 Using effective pull policy of [always] for container eclipse-temurin:17
21 11:33:16 Using docker image sha256:53c231189938e8988f3c7cf494520fd8de328402bc4778d10e1d25fab1ff15 for eclipse-temurin:17 with digest eclipse-temurin@sha256:a710bbe5d1a1c4c48ed1e8d5be5f05d49132a102ab79961086d0675ed8b387d8d ...
22 11:33:17 $ javac prog.java
23 11:33:17 Cleaning up project directory and file based variables
24 11:33:18 Job succeeded

```

Duration: 21 seconds
Finished: 44 seconds ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270857 (ntHFEtyX) 4-green.saaS-linux-small-amd64.runners-manager.gitlab.com/default
Source: Push
Commit 5019ff6a
Update .gitlab-ci.yml file
Pipeline #230762064 Passed
for main build
Related jobs → build_job

What's new Help Collapse sidebar

31°C Sunny

17:04 05-02-2026

VIT Vellore - VTOP | Download file | iLovePDF | archive_job (#13000041852) · Job

gitlab.com/agile-project5043406/Gitlab-CI-Demo/-/jobs/13000041852

Project Agile project / Gitlab-CI-Demo / Jobs / #13000041852

archive_job

Passed Started 1 minute ago by Tejas C 23MIS0671

Log timestamps in UTC.

```

1 11:33:21 Running with gitlab-runner 18.7.0-pre.433.g3af5f2314 (3af5f2314)
2 11:33:21 On green-1.saaS-linux-small-amd64.runners-manager.gitlab.com/default JlGuOpMV, system ID: s_deaa2ca09de7
3 11:33:21 Preparing the "docker-machine" executor
4 11:33:21 Using Docker executor with image eclipse-temurin:17 ...
5 11:33:21 Using effective pull policy of [always] for container eclipse-temurin:17
6 11:33:26 Pulling docker image eclipse-temurin:17 ...
7 11:33:33 Using docker image sha256:53c231189938e8988f3c7cf494520fd8de328402bc4778d10e1d25fab1ff15 for eclipse-temurin:17 with digest eclipse-temurin@sha256:a710bbe5d1a1c4c48ed1e8d5be5f05d49132a102ab79961086d0675ed8b387d8d ...
8 11:33:33 Preparing environment
9 11:33:33 Using effective pull policy of [always] for container sha256:b9947c6088c8224ad98fb91248b64d188ae5c7203025a1ea76e740f1879a75b
10 11:33:33 Running on runner-jlGuOpMV-project-78250563-concurrent-0 via runner-jlGuOpMV-s-l-s-amd64-1778291123-5401b899...
11 11:33:37 Getting source from Git repository
12 11:33:37 GitLab correlation ID: 8a5c1b87c509298ba035Seab
13 11:33:37 Fetching changes with git depth set to 20...
14 11:33:37 Initialized empty Git repository in /builds/agile-project5043406/Gitlab-CI-Demo/.git/
15 11:33:37 Created fresh repository.
16 11:33:38 Checking out 5019ff6a as detached HEAD (ref is main)...
17 11:33:38 Skipping Git submodules setup
18 11:33:38 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
19 11:33:38 Executing "step_script" stage of the job script
20 11:33:38 Using effective pull policy of [always] for container eclipse-temurin:17
21 11:33:38 Using docker image sha256:53c231189938e8988f3c7cf494520fd8de328402bc4778d10e1d25fab1ff15 for eclipse-temurin:17 with digest eclipse-temurin@sha256:a710bbe5d1a1c4c48ed1e8d5be5f05d49132a102ab79961086d0675ed8b387d8d ...
22 11:33:38 $ echo "Archiving build"
23 11:33:38 Archiving build
24 11:33:39 Uploading artifacts for successful job
25 11:33:39 Uploading artifacts...
26 11:33:39 WARNING: prog.class: no matching files. Ensure that the artifact path is relative to the working directory (/builds/agile-project5043406/Gitlab-CI-Demo)
27 11:33:39 ERROR: No files to upload
28 11:33:39 Cleaning up project directory and file based variables
29 11:33:40 Job succeeded

```

Duration: 21 seconds
Finished: 48 seconds ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270845 (JlGuOpMV) 1-green.saaS-linux-small-amd64.runners-manager.gitlab.com/default
Source: Push
Commit 5019ff6a
Update .gitlab-ci.yml file
Pipeline #230762064 Passed
for main archive
Related jobs → archive_job

What's new Help Collapse sidebar

31°C Sunny

17:04 05-02-2026