



**VIT**<sup>®</sup>  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

---

**NAME:** K. Manoj Kumar

**REGISTER NUMBER:** 23MIS0159

**COURSE NAME:** Agile Development Process and DevOps Lab

**COURSE CODE:** ISWE406P

**SLOT:** L51+L52

## TASK 1: Jenkins Familiarization Objective:

### Understand Jenkins UI and basic navigation

#### Tasks:

#### 1. Open Jenkins Dashboard in browser

#### 2. Identify:

- o Dashboard

- o Manage Jenkins

- o New Item

- o Build History

#### 3. Check Jenkins version

#### Expected Output:

#### Screenshot or note of Jenkins version

#### DASHBOARD, Jenkins Version:

The screenshot shows the Jenkins Dashboard interface. The top navigation bar includes links for 'New Item', 'Build History', and 'Add description'. The left sidebar contains 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0 of 2 executors busy). The main content area displays a table of recent builds with columns for Status, Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	demo1	37 min #4	N/A	66 ms
✓	☀	GIT	38 min #5	N/A	1.6 sec
✓	☀	GIT DEMO	3 days 3 hr #1	N/A	1.7 sec
✓	☀	Task1	38 min #5	N/A	1.9 sec

At the bottom right, the text 'REST API Jenkins 2.528.3' is visible.

## NEWITEMS:

Jenkins / All / New Item

New Item

Enter an item name

agilelab

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.

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.

If you want to create a new item from other existing, you can use this option:

Copy from

Type to autocomplete

OK

## Build History:

Jenkins / All / Build History

New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status

(0 of 2 executors busy)

Build History of Jenkins

S	Build	Time Since	1	Status
✓	demo1 #4	42 min		stable
✓	GIT #5	42 min		stable
✓	Task1 #6	42 min		stable
✓	demo1 #3	3 days 3 hr		stable
✓	GIT #4	3 days 3 hr		stable
✓	Task1 #5	3 days 3 hr		stable
✓	demo1 #2	3 days 3 hr		stable
✓	GIT #3	3 days 3 hr		stable
✓	Task1 #4	3 days 3 hr		stable
✓	GIT #2	3 days 3 hr		stable
✓	Task1 #3	3 days 3 hr		stable
✓	Task1 #2	3 days 3 hr		stable
✓	GIT DEMO #1	3 days 3 hr		stable
✓	GIT #1	3 days 3 hr		stable
✓	demo1 #1	3 days 3 hr		stable

## MANAGE JENKINS:

Jenkins / Manage Jenkins

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Get up report Get up cloud Overview

System Configuration

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

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins.

Security

Security

Secure Jenkins; define who is allowed to access/use the system.

Credentials

Configure credentials.

Credential Providers

Configure the credential providers and types.

Users

Create/delete/modify users that can log in to this system.

Status Information

System Information

Display various environmental information to assist trouble-shooting.

System Log

System log captures output from Java, util, logging output related to Jenkins.

Load Statistics

Check your resource utilization and see if you need more computers for your builds.

About Jenkins

See the version and license information.

Troubleshooting

Manage Old Data

Backup configuration files to remove remnants from old plugins and earlier versions.

Tools and Actions

Reset Configuration from Disk

Discard all the loaded data in memory and reload everything from the system. Useful when your modified config files directly on disk.

Jenkins CLI

Access/manage Jenkins from your shell, or from your script.

Script Console

Executes arbitrary script for administrative/troubleshooting/diagnostics.

Prepare for Shutdown

Stops executing new builds, so that the system can be eventually shut down safely.

Jenkins 2.526.3

## TASK 2: Create First Freestyle Job

### Objective:

### Create and run a Jenkins job

### Tasks:







#### 1. Create a Freestyle project named Hello-Jenkins

**New Item**

Enter an item name

Hello-Jenkins

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.
-  **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.

If you want to create a new item from other existing, you can use this option:

Copy from

Type to autocomplete

OK

#### 2. Add a description

Jenkins / Hello-Jenkins / Configuration

Configure

General

Enabled

Description

Hello-Jenkins freestyle project

Plain text Preview

☐ Discard old builds ?

☐ GitHub project

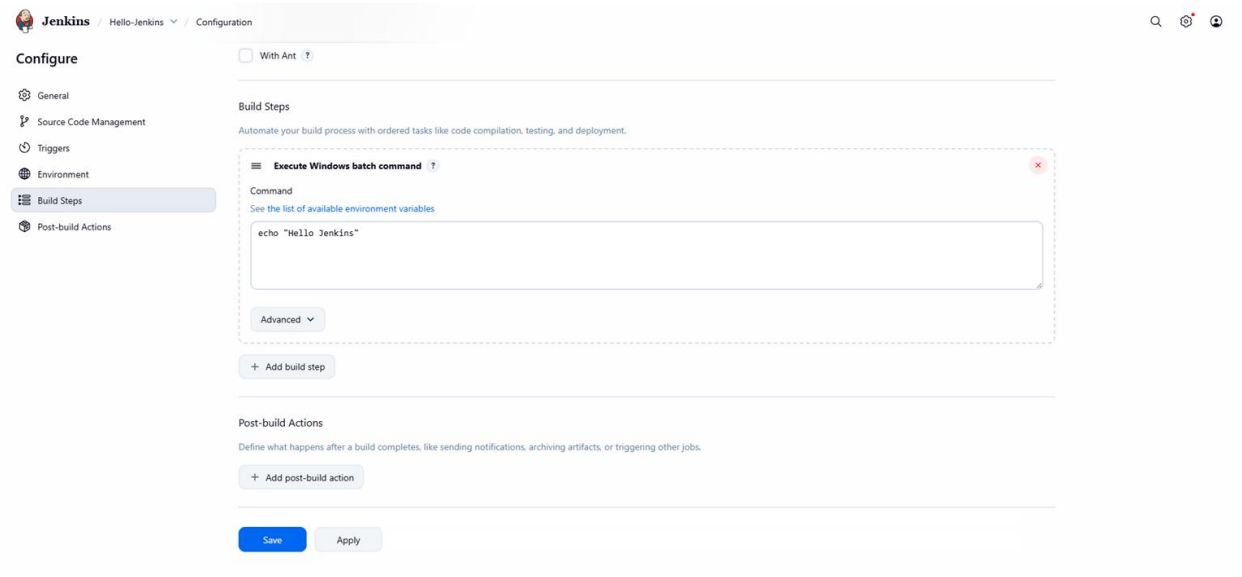
☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced

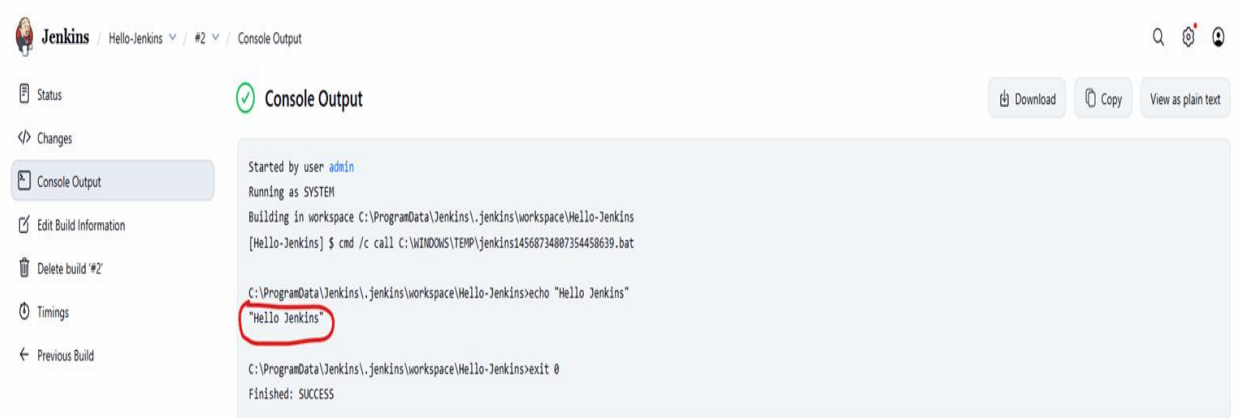
### 3. Add build step: o Execute shell / Windows batch command o Print "Hello Jenkins"



The screenshot shows the Jenkins Configuration page for a job named 'Hello-Jenkins'. The left sidebar contains a 'Configure' menu with options: General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions. The main content area is titled 'Build Steps' and includes a sub-section 'Execute Windows batch command'. A new build step has been added with the command 'echo "Hello Jenkins"'. Below this, there is a 'Post-build Actions' section with an 'Add post-build action' button. At the bottom, there are 'Save' and 'Apply' buttons.

### 4. Build the job manually

Expected Output: Console output showing message



The screenshot shows the Jenkins Console Output page for the 'Hello-Jenkins' job. The left sidebar contains a 'Console Output' menu with options: Status, Changes, Console Output (selected), Edit Build Information, Delete build #2, Timings, and Previous Build. The main content area is titled 'Console Output' and shows the build output. The output text is as follows:

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins
[Hello-Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins14568734807354458639.bat
C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins>echo "Hello Jenkins"
"Hello Jenkins"
C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins>exit 0
Finished: SUCCESS
```

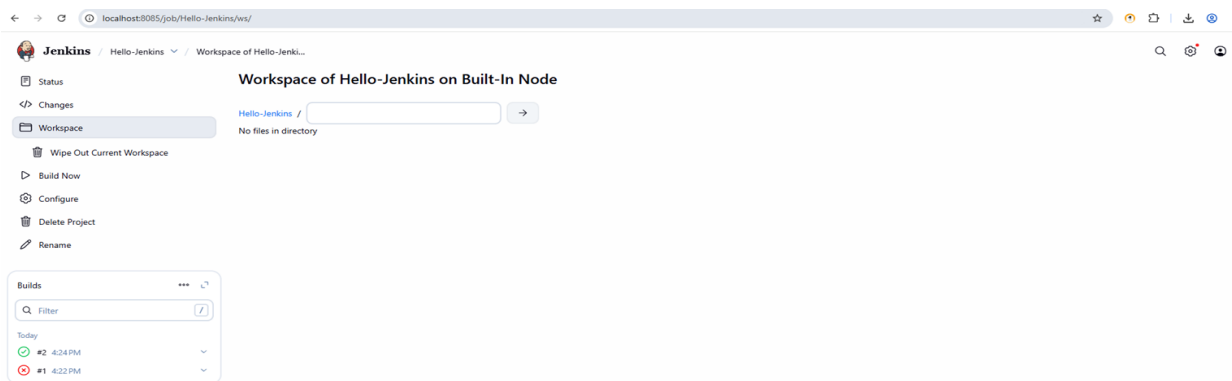
The output shows the build process starting with the user 'admin', running as 'SYSTEM', and building in the workspace 'C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins'. The build command is 'cmd /c call C:\WINDOWS\TEMP\jenkins14568734807354458639.bat'. The output of the command is 'C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins>echo "Hello Jenkins"', which results in the message '"Hello Jenkins"'. The build ends with 'C:\ProgramData\Jenkins\jenkins\workspace\Hello-Jenkins>exit 0' and 'Finished: SUCCESS'.

# TASK 3: Jenkins Workspace & Commands

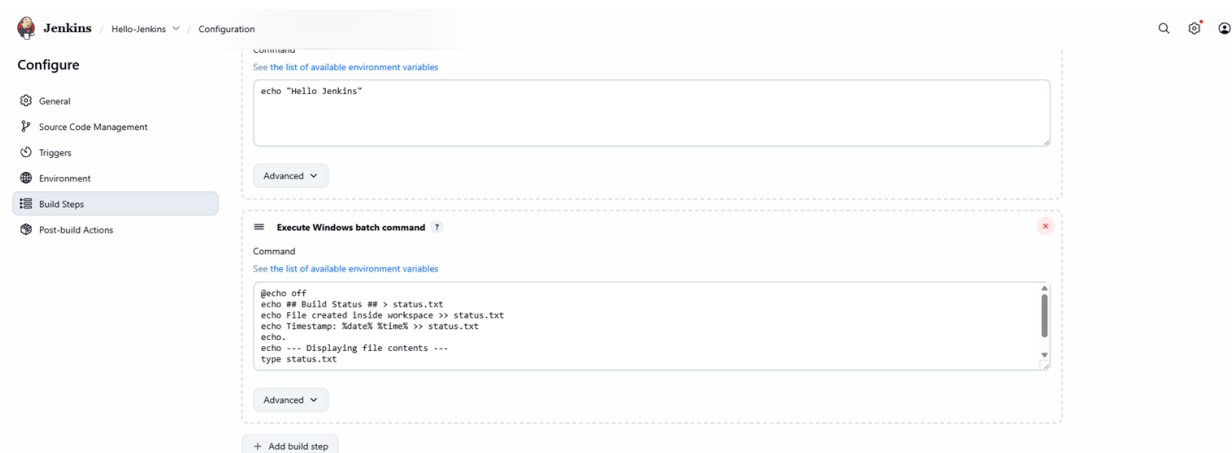
## Objective: Understand workspace usage

### Tasks:

#### 1. Navigate to job workspace



#### 2. Create a text file using build step



#### 3. Display file contents in console



## Expected Output: File created inside workspace

The screenshot shows the Jenkins web interface. At the top, it says 'Jenkins / Hello-Jenkins / Workspace of Hello-Jenkins...'. Below this, there's a sidebar with options: Status, Changes, Workspace (selected), Wipe Out Current Workspace, Build Now, Configure, Delete Project, and Rename. The main area is titled 'Workspace of Hello-Jenkins on Built-In Node'. It shows a file named 'status.txt' with a timestamp 'Jan 22, 2026, 4:36:35 PM' and a size of '89 B'. There's a button to download '(all files in zip)'. On the left, there's a 'Builds' section with a filter and a list of builds: #5 (4:36 PM), #4 (4:36 PM), #3 (4:34 PM), #2 (4:24 PM), and #1 (4:22 PM).

## TASK 4: Git Integration

### Objective: Integrate Jenkins with GitHub

#### Tasks: 1. Create a GitHub repository with sample code

##### Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

1

### General

**Owner \*** **Repository name \***

ManojMIS

/ Jenkins-admin

✓ Jenkins-admin is available.

Great repository names are short and memorable. How about [psychic-potato?](#)

**Description**

Jenkins admin repository

24 / 350 characters

2

### Configuration

**Choose visibility \***

Choose who can see and commit to this repository

Public

**Add README**

READMEs can be used as longer descriptions. [About READMEs](#)

Off

**Add .gitignore**

.gitignore tells git which files not to track. [About ignoring files](#)

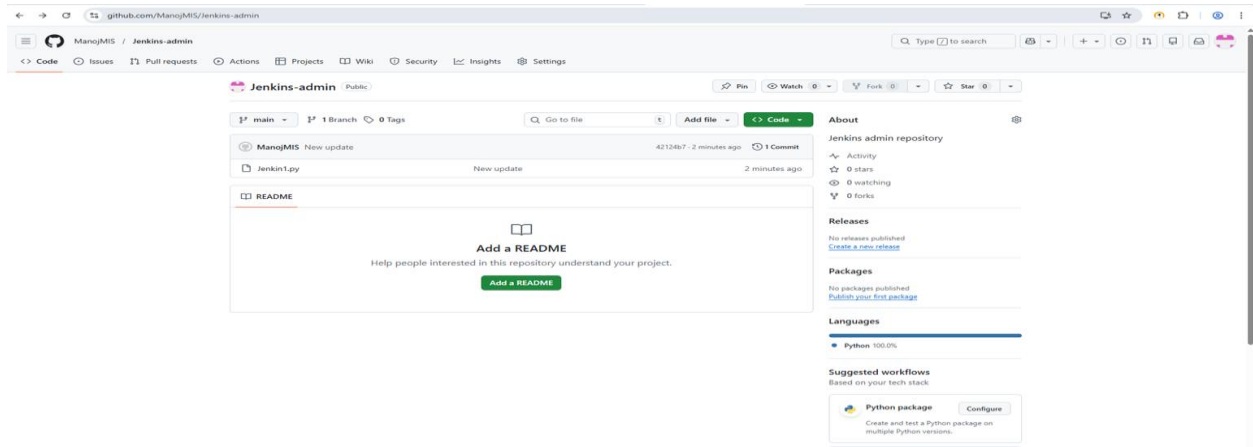
No .gitignore

**Add license**

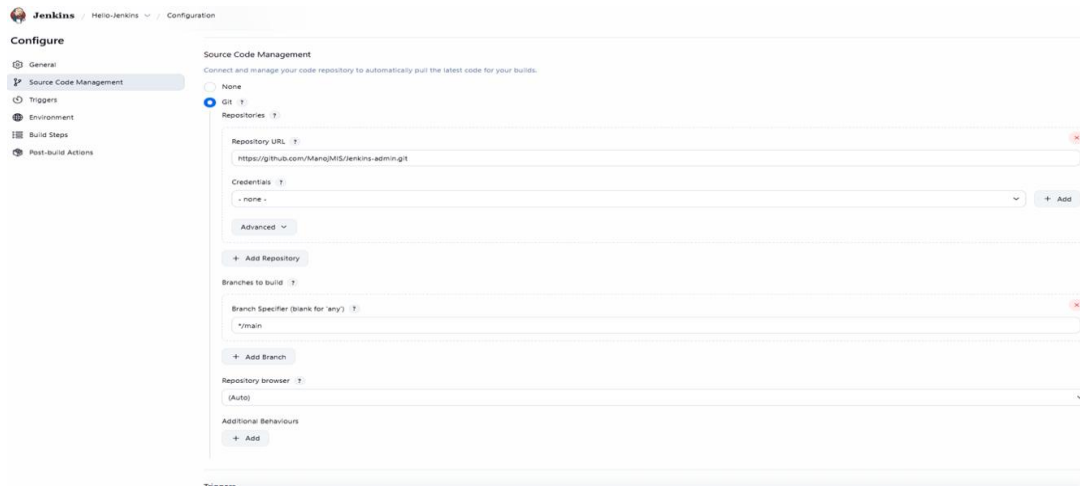
Licenses explain how others can use your code. [About licenses](#)

No license

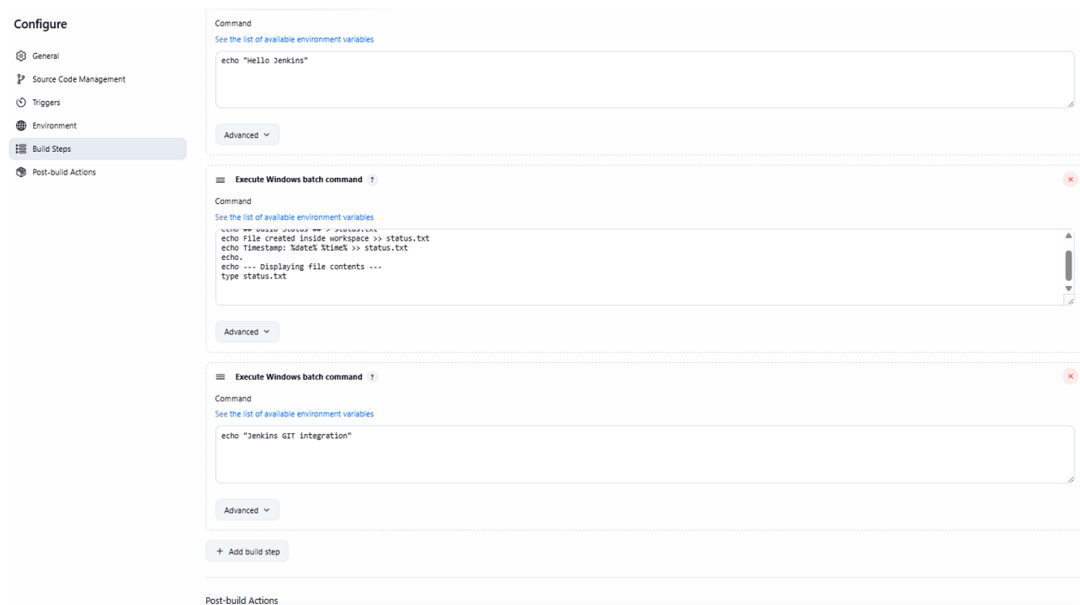
Create repository



## 2. Configure Git in Jenkins



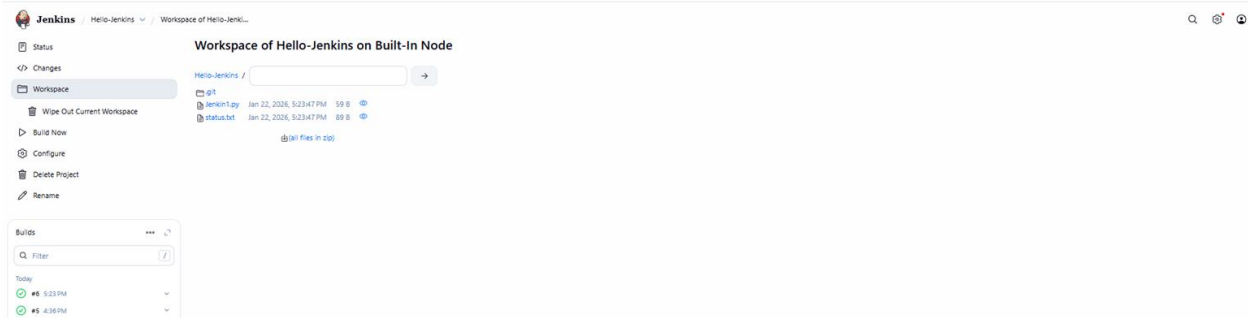
## 3. Add Git repository URL in job





#### 4. Build and verify code checkout

**Expected Output: Source code visible in workspace**



#### 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 Expected Output: • Build triggered without manual action**

#### 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 Expected Output: • Console output showing parameter value**

#### 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 Expected Output: • Java output in console**

#### **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 **Expected Output:** • Artifact available for download

#### **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 **Expected Output:** • Permission differences verified

#### **TASK 10: Simple Jenkins Pipeline**

**Objective:** Create basic pipeline

- Tasks:**
1. Create Pipeline job
  2. Write pipeline with stages: o Checkout o Build o Test
  3. Run pipeline **Expected Output:** • Pipeline stage view

#### **TASK 11: Jenkinsfile from Git**

**Objective:** Pipeline as Code

- Tasks:**
1. Create Jenkinsfile in Git repo
  2. Configure pipeline from SCM
  3. Trigger build **Expected Output:** • Pipeline executed from Git

#### **TASK 12: Post-Build Actions**

**Objective:** Handle build result

- Tasks:**
1. Add post section

2. Print message on success/failure Expected Output: • Appropriate message displayed

#### **TASK 13: Trigger Job from Another Job**

**Objective:** Job chaining

**Tasks:** 1. Create Job-A and Job-B

3. Configure Job-B to trigger after Job-A Expected Output: • Job-B triggered automatically

#### **TASK 14: Workspace Cleanup**

**Objective:** Manage disk usage

**Tasks:** 1. Install Workspace Cleanup plugin

2. Clean workspace before build Expected Output: • Workspace cleared before execution
- 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 Expected Output: • Automated CI pipeline

TOOL:GIT ACTIONS

1.)GIT ACTIONS DASBOARD:

ManojMIS / GITACTIONS-DA2

Q Type [Z] to search

+

+

+

+

+

+

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Actions

New workflow

All workflows

My First Workflow

Version Check

Management

Caches

Attestations

Runners

Usage metrics

Performance metrics

All workflows

Showing runs from all workflows

3 workflow runs

Event Status Branch Actor

Create version-check.yml

My First Workflow #3: Commit e5d759e pushed by ManojMIS

main

now

Queued

...

Update hello.yml

My First Workflow #2: Commit 0f9d07b pushed by ManojMIS

main

4 minutes ago

7s

...

Create hello.yml

My First Workflow #1: Commit 28f2cf4 pushed by ManojMIS

main

4 minutes ago

Failure

...

VERSION AND OUTPUT:

ManojMIS / GITACTIONS-DA2

Q Type [Z] to search

+

+

+

+

+

+

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Version Check

Version Check #2

Re-run all jobs

Summary

All jobs

version-job

Run details

Usage

Workflow file

version-job

succeeded 1 minute ago in 3s

Search logs

Set up job

1 Current runner version: '2.331.0'

2 Runner Image Provisioner

3 Operating System

4 Runner Image

5 GITHub\_TOKEN Permissions

6 Secret source: Actions

7 Prepare workflow directory

8 Prepare all required actions

9 Complete job name: version-job

Show runner version

1 Run uname -a

2 Linux runner00130n 6.11.0-1010-azure #10-24.04.1-Ubuntu SMP Sat Jun 20 04:46:03 UTC 2025 x86\_64 x86\_64 GNU/Linux

3 git version 2.52.0

4 openjdk version "17.0.18" 2024-01-20

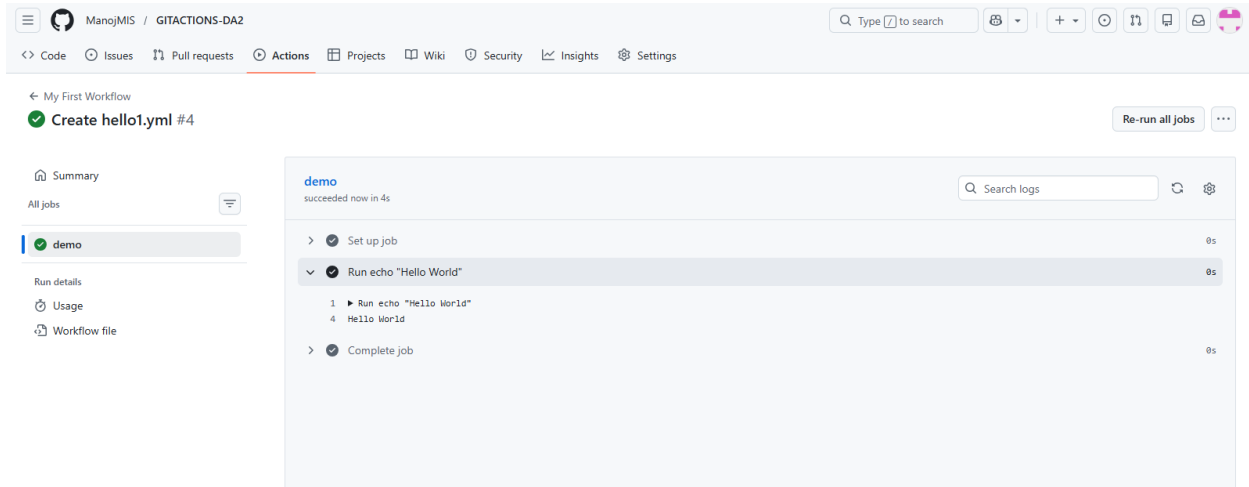
5 OpenJDK Runtime Environment Temurin-17.0.18+8 (build 17.0.18+8)

6 OpenJDK 64-Bit Server VM Temurin-17.0.18+8 (build 17.0.18+8, mixed mode, sharing)

Complete job

1 Cleaning up orphan processes

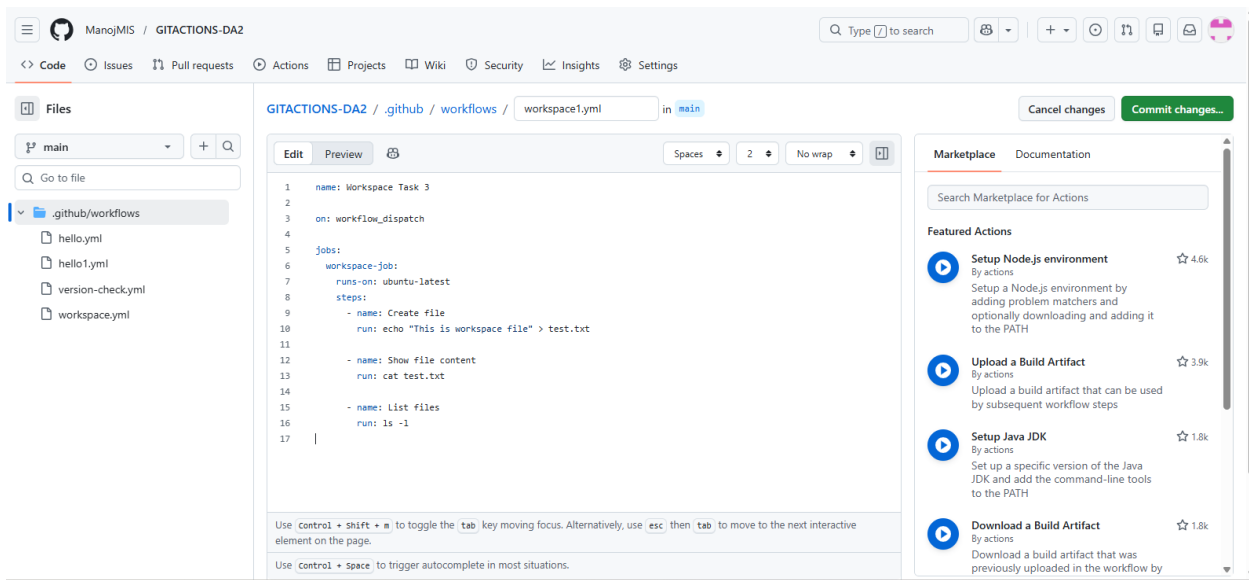
## 2.)First Freestyle job created and Output:



The screenshot shows the GitHub Actions interface for a workflow named "My First Workflow". The job "demo" is completed successfully. The output shows the following steps:

- Set up job (0s)
- Run echo "Hello World" (0s)
  - Run echo "Hello World"
  - Hello World
- Complete job (0s)

## 3.)Text File Created:



The screenshot shows the GitHub Actions interface for a workflow named "GITATIONS-DA2". The file "workspace1.yml" is being edited. The workflow is defined as follows:

```
1 name: workspace Task 3
2
3 on: workflow_dispatch
4
5 jobs:
6   workspace-job:
7     runs-on: ubuntu-latest
8     steps:
9       - name: Create file
10        run: echo "This is workspace file" > test.txt
11
12       - name: Show file content
13        run: cat test.txt
14
15       - name: List files
16        run: ls -l
17
```

## Job Workspace:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

**Actions** New workflow

All workflows Filter workflow runs

Showing runs from all workflows

10 workflow runs

	Event	Status	Branch	Actor
Workspace Task 3 Workspace Task 3 #1: Manually run by ManojMIS	main	2 minutes ago 7s	...	
Hello Jenkins Equivalent Hello Jenkins Equivalent #2: Manually run by ManojMIS	main	5 minutes ago 7s	...	
Hello Jenkins Equivalent Hello Jenkins Equivalent #1: Manually run by ManojMIS	main	6 minutes ago 8s	...	
Create workspace.yml Workspace Demo #5: Commit 64377bd pushed by ManojMIS	main	14 minutes ago 9s	...	
Create hello1.yml Workspace Demo #4: Commit 78b4035 pushed by ManojMIS	main	15 minutes ago 7s	...	
Version Check Version Check #2: Manually run by ManojMIS	main	18 minutes ago 7s	...	

## OUTPUT:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Workspace Task 3

**Workspace Task 3 #1** Re-run all jobs Latest #2

Summary

All jobs

**workspace-job**

Run details

Usage

Workflow file

**workspace-job**  
succeeded 8 minutes ago in 2s

Search logs

Set up job 0s

Create file 0s

1 Run echo "This is workspace file" > test.txt

Show file content 0s

1 Run cat test.txt

4 This is workspace file

List files 0s

1 Run ls -l

4 total 4

5 -rw-r--r-- 1 runner runner 23 Feb 4 14:11 test.txt

Complete job 0s

## 4.)Git Integration Workflow created:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

**Actions** New workflow

All workflows

**Git Integration**

Hello Jenkins Equivalent

Version Check

Workspace Demo

Workspace Demo

Workspace Task 3

Management

Caches

Attestations

Runners

Usage metrics

Performance metrics

**Git Integration**  
git-checkout.yml

Filter workflow runs

2 workflow runs

This workflow has a workflow\_dispatch event trigger. Run workflow

	Event	Status	Branch	Actor
Git Integration Git Integration #2: Manually run by ManojMIS	main	now 8s	...	
Git Integration Git Integration #1: Manually run by ManojMIS	main	now 8s	...	

## OUTPUT:

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Git Integration

**Git Integration #2** Re-run all jobs

Summary

All jobs

**git-job**

Run details

Usage

Workflow file

**git-job**  
succeeded 5 minutes ago in 5s

Search logs

- Set up job 1s
- Checkout code 0s
- List repo files 0s
- Post Checkout code 1s
- Complete job 0s

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Git Integration

**Git Integration #2** Re-run all jobs

Summary

All jobs

**git-job**

Run details

Usage

**Workflow file**

**Workflow file for this run**  
[github/workflows/git-checkout.yml #1 @1b67a3](#)

```
1 name: Git Integration
2
3 on: workflow_dispatch
4
5 jobs:
6   git-job:
7     runs-on: ubuntu-latest
8     steps:
9       - name: Checkout code
10        uses: actions/checkout@v4
11
12       - name: List repo files
13        run: ls -R
```

## 5.)Auto Trigger Workflow created:

**Actions** New workflow

All workflows

**Auto Trigger on Push**

Git Integration

Hello Jenkins Equivalent

Version Check

Workspace Demo

Workspace Demo

Workspace Task 3

Management

Caches

Attestations

Runners

Usage metrics

Performance metrics

Auto Trigger on Push  
[auto-trigger.yml](#)

Filter workflow runs

1 workflow run

Event Status Branch Actor

**Create auto-trigger.yml** main

Auto Trigger on Push #1: Commit [a4da673](#) pushed by [ManojMIS](#)

now  
7s

## OUTPUT:

The screenshot shows the GitHub Actions interface for a workflow named 'Auto Trigger on Push'. The 'build-job' step is highlighted, showing its logs. The logs indicate that the job succeeded 1 minute ago. The steps performed are: Set up job (1s), Run actions/checkout@v4 (0s), Show commit message (0s), Post Run actions/checkout@v4 (0s), and Complete job (1s). The commit message is displayed as follows:

```
1 ▶ Run git log -1
4 commit a4da67379e9e8e195da07c1387737a9c1f021ff
5 Author: ManojMIS <manojkumar.k2023@vitstudent.ac.in>
6 Date: Wed Feb 4 19:48:12 2026 +0530
7
8 Create auto-trigger.yml
```

## 6.)Parameterized Workflow Created:

The screenshot shows the GitHub Actions interface for a workflow named 'Auto Trigger on Push'. The 'Auto Trigger on Push' workflow is selected, and the '2 workflow runs' section is visible. The runs are listed as follows:

Run Name	Event	Status	Branch	Actor	Time
Create params.yml	Auto Trigger on Push #2: Commit 03939cc pushed by ManojMIS	Success	main	ManojMIS	now
Create auto-trigger.yml	Auto Trigger on Push #1: Commit adda673 pushed by ManojMIS	Success	main	ManojMIS	6 minutes ago

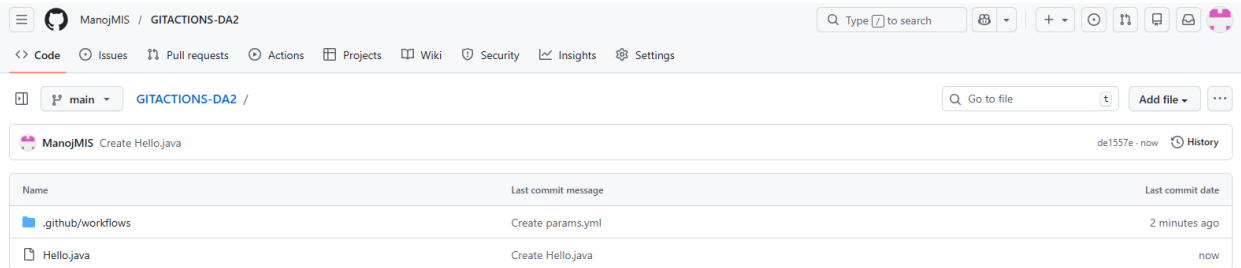
## OUTPUT:

The screenshot shows the GitHub Actions interface for a workflow named 'Parameterized Workflow #2'. The 'param-job' step is highlighted, showing its logs. The logs indicate that the job succeeded now in 4s. The steps performed are: Set up job (0s), Print username (0s), and Complete job (0s). The output of the 'Print username' step is displayed as follows:

```
1 ▶ Run echo "Hello K.Manoj Kumar 23m1s0159"
4 Hello K.Manoj Kumar 23m1s0159
```



## 7.)Hello.java FILE:



ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

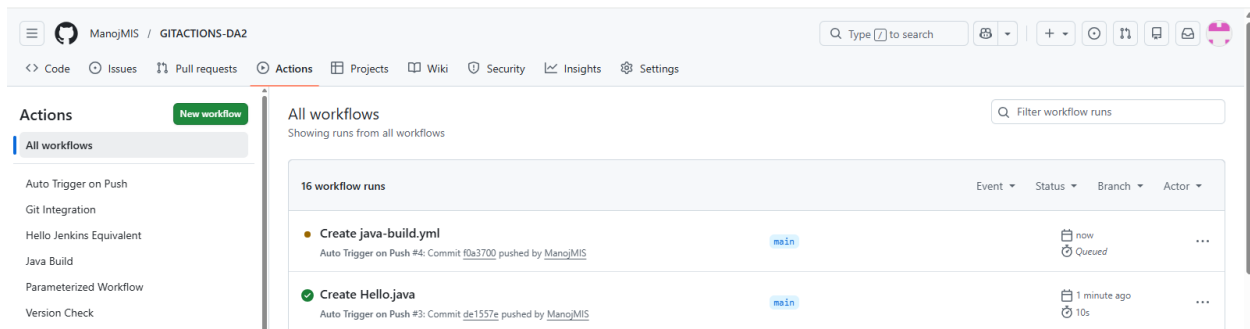
main GITACTIONS-DA2 /

Go to file Add file

ManojMIS Create Hello.java de1557e · now History

Name	Last commit message	Last commit date
.github/workflows	Create params.yml	2 minutes ago
Hello.java	Create Hello.java	now

## JAVA workflow created:



ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Actions New workflow

All workflows

Auto Trigger on Push

Git Integration

Hello Jenkins Equivalent

Java Build

Parameterized Workflow

Version Check

All workflows

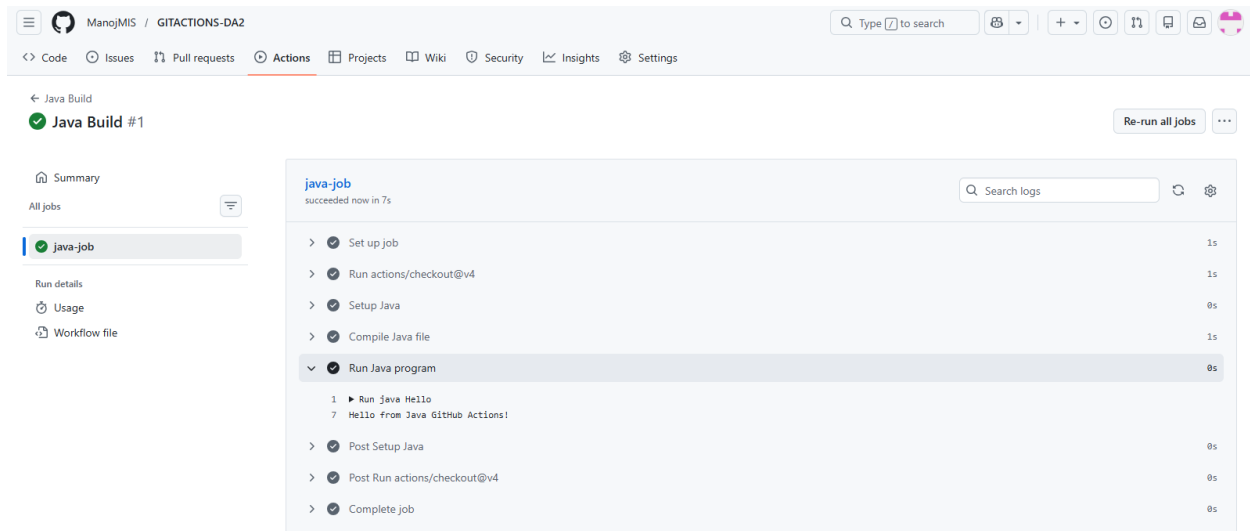
Showing runs from all workflows

Filter workflow runs

16 workflow runs

	Event	Status	Branch	Actor
Create java-build.yml	Auto Trigger on Push #4: Commit f0a3700 pushed by ManojMIS	now	main	Queued
Create Hello.java	Auto Trigger on Push #3: Commit de1557e pushed by ManojMIS	1 minute ago	main	10s

## OUTPUT:



ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Java Build

Java Build #1

Re-run all jobs

Summary

All jobs

java-job

Run details

Usage

Workflow file

java-job

succeeded now in 7s

Search logs

Set up job 1s

Run actions/checkout@v4 1s

Setup Java 0s

Compile Java file 1s

Run Java program 0s

1 ▶ Run Java Hello

7 Hello from Java GitHub Actions!

Post Setup Java 0s

Post Run actions/checkout@v4 0s

Complete job 0s

## 8.) Artifacts workflow Created:

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Actions New workflow

All workflows

Archive Artifacts

Auto Trigger on Push

Git Integration

Hello Jenkins Equivalent

All workflows

Showing runs from all workflows

21 workflow runs

Event Status Branch Actor

✓ Create artifacts.yml

Auto Trigger on Push #5: Commit bededa9 pushed by ManojMIS

main

1 minute ago

9s

## OUTPUT:

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Archive Artifacts

✓ Archive Artifacts #1

Summary

All jobs

artifact-job

Run details

Usage

Workflow file

Re-run all jobs

artifact-job

succeeded now in 6s

Search logs

Set up job 0s

Run actions/checkout@v4 1s

Compile Java 1s

Upload artifact 1s

1 Run actions/upload-artifact@v4

9 With the provided path, there will be 1 file uploaded

10 Artifact name is valid!

11 Root directory input is valid!

12 Beginning upload of artifact content to blob storage

13 Uploaded bytes 432

14 Finished uploading artifact content to blob storage!

15 SHA256 digest of uploaded artifact zip is 271178d1e9d77b8d82ee44e771f7320e6a3488ce68b6fcbf0f62d95ced

16 Finalizing artifact upload

17 Artifact compiled-class.zip successfully finalized. Artifact ID 5375482547

18 Artifact compiled-class has been successfully uploaded! Final size is 432 bytes. Artifact ID is 5375482547

19 Artifact download URL: <https://github.com/ManojMIS/GITACTIONS-DA2/actions/runs/21675478718/artifacts/5375482547>

Post Run actions/checkout@v4 0s

Complete job 0s

## 10.)

## Pipeline workflow created:

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Actions New workflow

All workflows

Archive Artifacts

Auto Trigger on Push

Git Integration

Hello Jenkins Equivalent

Java Build

Parameterized Workflow

Simple Pipeline

Version Check

Workspace Demo

Workspace Demo

Show more workflows...

All workflows

Showing runs from all workflows

23 workflow runs

Event Status Branch Actor

✓ Create pipeline.yml

Auto Trigger on Push #6: Commit c7acbc2 pushed by ManojMIS

main

now

8s

✓ Archive Artifacts

Archive Artifacts #1: Manually run by ManojMIS

main

7 minutes ago

10s

✓ Create artifacts.yml

Auto Trigger on Push #5: Commit bededa9 pushed by ManojMIS

main

9 minutes ago

9s

✓ Parameterized Workflow

Parameterized Workflow #2: Manually run by ManojMIS

main

10 minutes ago

9s

## OUTPUT:

Simple Pipeline #1

Summary

All jobs

pipeline-job

Run details

Usage

Workflow file

Re-run all jobs

pipeline-job succeeded 1 minute ago in 3s

Search logs

Set up job 1s

Checkout Stage 8s

```
1 ▶ Run actions/checkout@v4
16 Syncing repository: ManojMIS/GITACTIONS-DA2
17 ▶ Getting Git version info
21 Temporarily overriding HOME='/home/runner/work/_temp/cd67792-b05c-46bb-85a2-865cdaeb8755' before making global git config changes
22 Adding repository directory to the temporary git global config as a safe directory
23 /usr/bin/git config --global --add safe.directory /home/runner/work/GITACTIONS-DA2/GITACTIONS-DA2
24 Deleting the contents of '/home/runner/work/GITACTIONS-DA2/GITACTIONS-DA2'
25 ▶ Initializing the repository
42 ▶ Disabling automatic garbage collection
44 ▶ Setting up auth
52 ▶ Fetching the repository
56 ▶ Determining the checkout info
57 /usr/bin/git sparse-checkout disable
58 /usr/bin/git config --local --unset-all extensions.worktreeConfig
59 ▶ Checking out the ref
63 /usr/bin/git log -1 --format=%H
64 c7acbc28f5084e7a21355c4f08fb0c45f12645
```

Build Stage 8s

```
1 ▶ Run echo "Building project..."
4 Building project...
```

Test Stage 8s

```
1 ▶ Run echo "Running tests..."
4 Running tests...
```

Post Checkout Stage 1s

Complete job 8s

11.)

## Pipeline from GIT workflow created:

ManojMIS / GITACTIONS-DA2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Filter workflow runs

All workflows

Showing runs from all workflows

26 workflow runs

Event Status Branch Actor

Create from-gityml Pipeline from Git #1: Commit 0be160c pushed by ManojMIS main now Queued ...

Create from-gityml Auto Trigger on Push #7: Commit 0be160c pushed by ManojMIS main now Queued ...

Simple Pipeline Simple Pipeline #1: Manually run by ManojMIS main 2 minutes ago 7s ...

Create pipeline.yml Auto Trigger on Push #6: Commit c7acbc2 pushed by ManojMIS main 4 minutes ago 8s ...

Archive Artifacts main 11 minutes ago ...

## OUTPUT:

The screenshot shows the GitHub Actions interface for a workflow named 'Create from-git.yml #1'. The workflow is in a 'succeeded' state. The left sidebar shows the workflow file and a list of jobs, with 'job1' selected. The main area displays the job details for 'job1', which succeeded 1 minute ago. The job steps are: 'Set up job' (1s), 'Run actions/checkout@v4' (0s), 'Run echo "Pipeline executed from repo YAML"' (0s), 'Post Run actions/checkout@v4' (0s), and 'Complete job' (0s). The logs for the 'Run echo' step show the output: 'Run echo "Pipeline executed from repo YAML"' and 'Pipeline executed from repo YAML'.

12.)

## Post-build workflow created:

The screenshot shows the 'All workflows' page in GitHub Actions. The left sidebar lists various workflow templates, with 'Post Build Actions' highlighted. The main area shows a list of 28 workflow runs. The runs are filtered by 'main' branch and show the status of each workflow. The runs are: 'Create post-build.yml' (now in progress), 'Create post-build.yml' (now in progress), 'Create from-git.yml' (2 minutes ago), 'Create from-git.yml' (2 minutes ago), and 'Simple Pipeline' (5 minutes ago).

## OUTPUT:

The screenshot shows the GitHub Actions interface for a workflow named 'Post Build Actions #1'. The workflow is in a 'succeeded' state. The left sidebar shows the workflow file and a list of jobs, with 'post-job' selected. The main area displays the job details for 'post-job', which succeeded now in 4s. The job steps are: 'Set up job' (0s), 'Build Step' (0s), 'Success Message' (0s), 'Failure Message' (0s), and 'Complete job' (0s). The logs for the 'Success Message' step show the output: 'Run echo "BUILD SUCCESSFUL"' and 'BUILD SUCCESSFUL'.

13.)

## Chaining workflow created:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

**Actions** New workflow

All workflows

- Archive Artifacts
- Auto Trigger on Push
- Git Integration
- Hello Jenkins Equivalent
- Java Build
- Job Chaining**
- Parameterized Workflow
- Pipeline from Git
- Post Build Actions
- Simple Pipeline
- Version Check
- Workspace Demo
- Workspace Demo
- Workspace Task 3

**All workflows**  
Showing runs from all workflows

34 workflow runs

	Event	Status	Branch	Actor
● Create chaining.yml Pipeline from Git #4: Commit c8922ef pushed by ManojMIS	main	now Queued		...
● Create chaining.yml Auto Trigger on Push #10: Commit c8922ef pushed by ManojMIS	main	now Queued		...
● Create chaining.yml Pipeline from Git #3: Commit 63cc2d7 pushed by ManojMIS	main	now 8s		...
● Create chaining.yml Auto Trigger on Push #9: Commit 63cc2d7 pushed by ManojMIS	main	now 7s		...
● Post Build Actions Post Build Actions #2: Manually run by ManojMIS	main	2 minutes ago 7s		...

## OUTPUT:

### JOB-A:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Job Chaining

**Job Chaining #1** Re-run all jobs ...

Summary

All jobs

- jobA
- jobB

Run details

Usage

Workflow file

**jobA**  
succeeded now in 4s

Search logs

- Set up job 0s
- Run echo "Job A running..." 0s
  - 1 Run echo "Job A running..."
  - 4 Job A running...
- Complete job 1s

### JOB-B:

ManojMIS / GITACTIONS-DA2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Job Chaining

**Job Chaining #1** Re-run all jobs ...

Summary

All jobs

- jobA
- jobB**

Run details

Usage

Workflow file

**jobB**  
succeeded now in 4s

Search logs

- Set up job 1s
- Run echo "Job B triggered after Job A" 0s
  - 1 Run echo "Job B triggered after Job A"
  - 4 Job B triggered after Job A
- Complete job 0s

14.)

## Cleanup Workflow created:

The screenshot shows the GitHub Actions interface for the repository 'ManojMIS / GITACTIONS-DA2'. The 'Actions' tab is selected, and the 'All workflows' page is displayed. In the left sidebar, under 'All workflows', the 'Workspace Cleanup' workflow is highlighted with a yellow circle. The main area shows a list of 37 workflow runs. The first few runs are:

Event	Status	Branch	Actor
Create cleanup.yml	now	main	...
Create cleanup.yml	now	main	...
Job Chaining	18 minutes ago	main	...
Create chaining.yml	20 minutes ago	main	...
Create chaining.yml	20 minutes ago	main	...
Create chaining.yml	20 minutes ago	main	...

## OUTPUT:

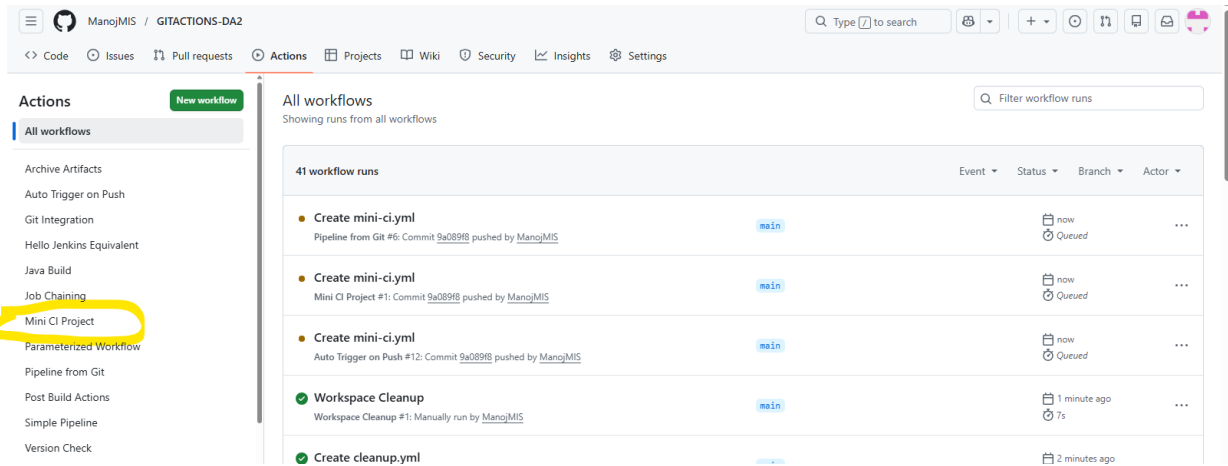
The screenshot shows the GitHub Actions interface for the repository 'ManojMIS / GITACTIONS-DA2'. The 'Workspace Cleanup' workflow is selected, and the 'cleanup-job' is highlighted. The output shows the steps of the workflow:

```
cleanup-job
succeeded now in 3s

> Set up job 1s
> Run actions/checkout@v4 0s
> Create temp file 0s
  1 ▶ Run echo "temp data" > temp.txt
> Clean workspace 0s
  1 ▶ Run rm -rf *
> Confirm cleanup 0s
  1 ▶ Run ls -la
  4 total 16
  5 drwxr-xr-x 4 runner runner 4096 Feb  4 15:10 .
  6 drwxr-xr-x 3 runner runner 4096 Feb  4 15:10 ..
  7 drwxr-xr-x 7 runner runner 4096 Feb  4 15:10 .git
  8 drwxr-xr-x 3 runner runner 4096 Feb  4 15:10 .github
> Post Run actions/checkout@v4 1s
> Complete job 0s
```

15.)

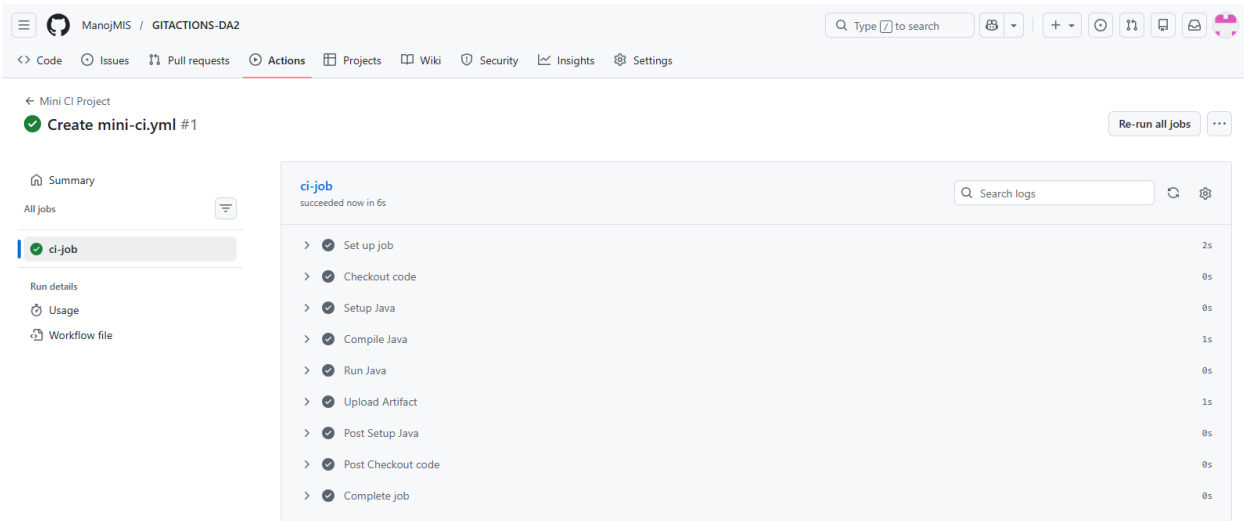
## CI workflow created:



The screenshot shows the GitHub Actions interface for the repository 'ManojMIS / GITACTIONS-DA2'. The 'Actions' tab is selected, displaying a list of workflows on the left and a table of workflow runs on the right. The 'Mini CI Project' workflow is highlighted in the left sidebar. The table on the right shows several runs of the 'Create mini-ci.yml' workflow, all with a status of 'now' and 'Queued'. A 'Workspace Cleanup' workflow run is also visible, completed '1 minute ago'.

Workflow	Branch	Status	Actor
Create mini-ci.yml	main	now	ManojMIS
Create mini-ci.yml	main	now	ManojMIS
Create mini-ci.yml	main	now	ManojMIS
Workspace Cleanup	main	1 minute ago	ManojMIS
Create cleanup.yml	main	2 minutes ago	ManojMIS

## OUTPUT:



The screenshot shows the GitHub Actions interface for the repository 'ManojMIS / GITACTIONS-DA2'. The 'Actions' tab is selected, displaying a list of workflows on the left and a table of workflow runs on the right. The 'Mini CI Project' workflow is highlighted in the left sidebar. The table on the right shows several runs of the 'Create mini-ci.yml' workflow, all with a status of 'now' and 'Queued'. A 'Workspace Cleanup' workflow run is also visible, completed '1 minute ago'.

Workflow	Branch	Status	Actor
Create mini-ci.yml	main	now	ManojMIS
Create mini-ci.yml	main	now	ManojMIS
Create mini-ci.yml	main	now	ManojMIS
Workspace Cleanup	main	1 minute ago	ManojMIS
Create cleanup.yml	main	2 minutes ago	ManojMIS