

Experiment No. 5

Title : Applying CI/CD Principles to Web Development Using Jenkins, Git, and LocalHTTP Server.

Step 1) Creating a Jenkins Job using “ Freestyle Project ” type :

Jenkins

Search (CTRL+K)


Dashboard > All > New Item


New Item


Enter an item name


Calculator-Project

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 namespaces.

OK


Set up the Jenkins job ;


Jenkins


Search (CTRL+K)

Dashboard >

+ New Item

 Build History

 Manage Jenkins

 My Views

Build Queue

All +

S	W	Name ↓	Last Success
✓	☀	Calculator-Project	3 min 14 sec #2

Running Jenkins job;

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Calculator-Project	3 min 14 sec #2	N/A	1.5 sec

## Step 2 ) Set up a Webhook in Git Repository

Actions

Projects

Wiki

Security

Insights

Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Integrations

Webhooks / Manage webhook

SettingsRecent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL \*

https://jenkins-server/github-webhook/

Content type \*

application/json

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification

Disable (not recommended)

Which events would you like to trigger this webhook?

Just the push event.

Or

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Triggers

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

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

## Step 3 ) Trigger the CI/CD Pipeline:

Git Repository before push changes;

calculatorPublic

PinUnwatch1

master1 Branch0 Tags

Go to file

Add file

<> Code

Omkar1236-bit	Resolved merge conflicts	0d67e2d · 2 minutes ago	8 Commits
.vscode	Resolved merge conflicts	2 minutes ago	
index.html	changed	yesterday	
script.js	Initial commit	4 days ago	
style.css	Adding change style to Git Repo	5 minutes ago	


## Commit the changes;

```
PS D:\Calculator> git init
Reinitialized existing Git repository in D:/Calculator/.git/
PS D:\Calculator> git add insertion_sort.java
PS D:\Calculator> git commit -m "add content insertion_sort.java"
[master b3be40b] add content insertion_sort.java
1 file changed, 40 insertions(+)
create mode 100644 insertion_sort.java
```

## Pushing changes to Git Repository;

```
PS D:\Calculator> git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 693 bytes | 693.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Omkar1236-bit/calculator.git
0d67e2d..b3be40b master -> master
```

## After push Git Repository become;

 calculator PublicPinUnwatch1

🔗 master ▾







🌿 1 Branch

🏷 0 Tags

🔍 Go to file t

Add file ▾

<> Code ▾

 Omkar1236-bit	add content insertion_sort.java	b3be40b · 1 minute ago	🕒 9 Commits
 .vscode	Resolved merge conflicts	6 minutes ago	
 index.html	changed	yesterday	
 insertion_sort.java	add content insertion_sort.java	1 minute ago	
 script.js	Initial commit	4 days ago	
 style.css	Adding change style to Git Repo	8 minutes ago	

Monitor the Jenkins job's progress in the Jenkins web interface;

The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo and a search bar. Below the header is a breadcrumb trail: Dashboard > Calculator-Project >. The main content area is divided into two columns. The left column contains a sidebar with icons and labels for various actions: Status (selected), Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, GitHub, and Rename. The right column displays the job name 'Calculator-Project' with a green checkmark icon. Below the job name is a section titled 'Permalinks' containing a list of links: 'Last build (#2), 29 sec ago', 'Last stable build (#2), 29 sec ago', 'Last successful build (#2), 29 sec ago', and 'Last completed build (#2), 29 sec ago'.

#### Step 4) Verify the CI/CD Pipeline.

The webhook should trigger the Jenkins job automatically, executing the build and deployment;

The screenshot shows a GitHub repository page for a repository named 'calculator' (Public). The repository is owned by 'Omkar1236-bit'. The page shows the 'master' branch with 1 branch and 0 tags. There is a search bar and buttons for 'Go to file', 'Add file', and 'Code'. Below this, a commit history table is displayed. The table has columns for the commit author, the commit message, the commit hash, the time ago, and the number of commits. The commit history shows the following entries:

Author	Message	Hash	Time Ago	Commits
Omkar1236-bit	add content insertion_sort.java	b3be40b	4 minutes ago	9
	Resolved merge conflicts		9 minutes ago	
	changed		yesterday	
	add content insertion_sort.java		4 minutes ago	
	Initial commit		4 days ago	
	Adding change style to Git Repo		11 minutes ago	

Below the commit history, there is a section for the README file, which is currently empty.

Hence the local HTTP server to verify that the web application has been updated with the latest changes.