8. Demonstrate Continuous Integration and development using Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

Zooiiee

☐ Treat username as secret ?

Password ?

●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

ID ?

github-creds

Description ?

GitHub Personal Access Token for Jenkins pipeline

Create

New Item

Enter an item name

web-deploy-pipeline

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.

Configure

- ⚙ General
- ⏱ Triggers
- 🔁 Pipeline
- 🔧 Advanced

Pipeline

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

Definition

| Pipeline script from SCM | ⌄ |

SCM ?

| Git | ⌄ |

Repositories ?

Repository URL ?

[   ] ✕

🛑 Please enter Git repository.

Credentials ?

| - none - | ⌄ |

[+ Add]

[Advanced ⌄]

- 🗎 Status
- </> Changes
- ▷ Build Now
- ⚙ Configure
- 🗑 Delete Pipeline
- 🗇 Stages
- ✏ Rename
- ❓ Pipeline Syntax

web-deploy-pipeline

Permalinks

Configure

⚙ General

🕒 Triggers

🔀 Pipeline

🔧 Advanced

Branch Specifier (blank for 'any') ?

```
*/main
```

Add Branch

Repository browser ?

```
(Auto)
```

Additional Behaviours

Add ⌄
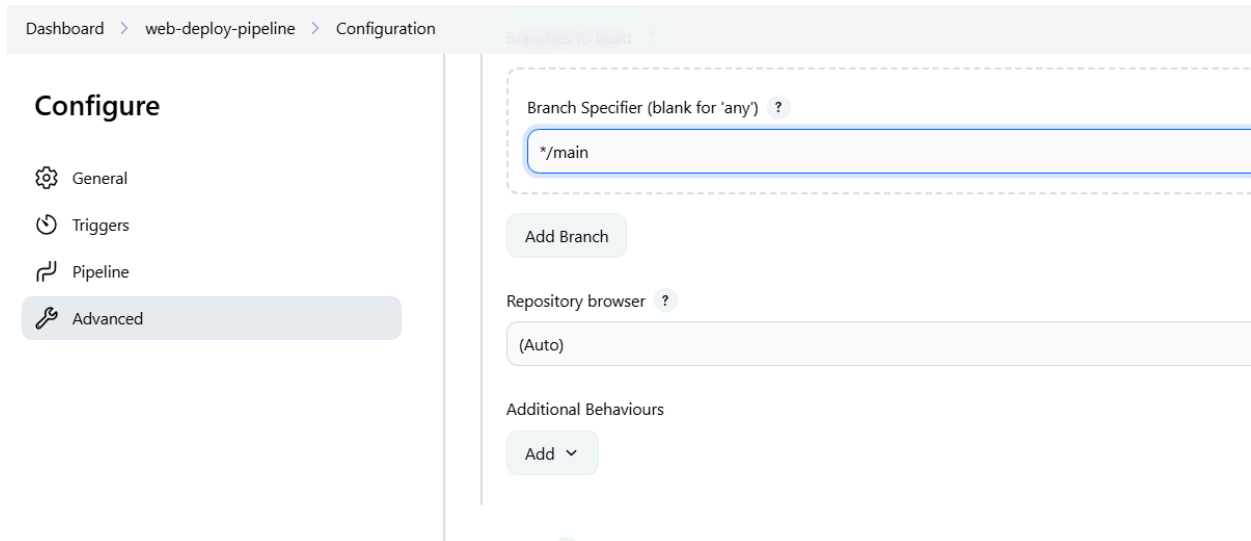
2. Create Your GitHub Repo

Example structure:

pgsql
CopyEdit

```
simple-web-app/
├── index.html
├── Jenkinsfile
```

Sample `index.html`:

html
CopyEdit

```html
<!DOCTYPE html>
<html>
  <head><title>CI/CD Test</title></head>
  <body><h1>Hello from Jenkins CI/CD!</h1></body>
</html>
```

Sample `Jenkinsfile`:

```
pipeline {
    agent any

    stages {
        stage('Clone') {
            steps {
                git url:
'https://github.com/Zooiiee/simple-web-app.git', branch:
'main', credentialsId: 'github-creds'
            }
        }

        stage('Build') {
            steps {
                echo 'Static site - no build step needed.'
            }
        }

        stage('Deploy') {
            steps {
                echo 'Deploying to Jenkins workspace
folder...'
                sh 'mkdir -p deploy'
                sh 'cp index.html deploy/index.html'
            }
        }
    }
}
```

🔁 Replace path (`C:\inetpub\wwwroot`) with your server root (XAMPP or Python folder if needed).

⚙️ 3. Configure Jenkins

a. Install Required Plugins

Go to: `Manage Jenkins → Plugins`
Install:

- Git Plugin

- GitHub Integration Plugin

- Pipeline Plugin

b. Add GitHub Credentials

1. Generate a GitHub Personal Access Token

2. Go to `Manage Jenkins → Credentials → Global`

3. Add:

    - **Kind**: Username & Password

    - **Username**: your GitHub username

    - **Password**: the token

    - **ID**: `github-creds` (must match the `Jenkinsfile`)

🧪 4. Create a Pipeline Job

1. **New Item → Pipeline**

2. Name: `web-deploy-pipeline`

3. Choose: `Pipeline`

4. Scroll to **Pipeline section**

   ○ Definition: `Pipeline script from SCM`

   ○ SCM: `Git`

   ○ Repo URL:
     `https://github.com/YOUR_USERNAME/simple-web-app.git`

   ○ Credentials: Select your GitHub creds

   ○ Script Path: `Jenkinsfile`

Click **Save**.

🚀 5. Run the Pipeline

● Click **Build Now**

● Jenkins will:

   1. Clone your GitHub repo

   2. Copy the `index.html` to your local server folder

3. Show "Hello from Jenkins CI/CD!" at `http://localhost`

4.

✅ Here's how to correctly copy the file from your host:

Then from **PowerShell on your host (Windows)**, run:

powershell

```
docker cp
jenkins-container:/var/jenkins_home/workspace/web-deploy-pipeline
/deploy/index.html .
```

This will copy the index.html file from inside the Jenkins container to your **current folder on your host**.

🧪 Then, serve the file with Python (if you have Python installed):

powershell
CopyEdit

```
python -m http.server 8081
```

Then visit http://localhost:8081 in your browser to see your deployed page.