

CI/CD pipelines

- [Setting up CI/CD pipelines with GitHub Actions or Jenkins](#)

Setting up CI/CD pipelines with GitHub Actions or Jenkins

Setting up CI/CD pipelines with GitHub Actions or Jenkins involves defining automated workflows for building, testing, and deploying your code.

GitHub Actions:

In your GitHub repository, create a directory named `.github/workflows/` and add a YAML file (e.g., `ci-cd.yml`) inside it. This file will define your workflow.

Specify events that will trigger the workflow, such as `push` to a specific branch (e.g., `main`), `pull_request`, or `workflow_dispatch` for manual triggering.

A workflow consists of one or more jobs, each running in a separate virtual environment (runner).

Within each job, define a sequence of steps. Steps can include:

- Checking out the code (`actions/checkout@v4`).
- Setting up the environment (e.g., `actions/setup-node@v4` for Node.js).
- Installing dependencies.
- Running tests.
- Building the application.
- Deploying to a target environment.

Leverage pre-built actions from the GitHub Marketplace or create custom actions to streamline common tasks.

Code

name: CI/CD Pipeline

```
on:
  push:
    branches:
      - main

jobs:
  build-and-test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '18'
      - name: Install dependencies
        run: npm install
      - name: Run tests
        run: npm test
  deploy:
    runs-on: ubuntu-latest
    needs: build-and-test
    steps:
      - uses: actions/checkout@v4
      - name: Deploy to production
        run: echo "Deploying application..."
        # Add your deployment commands here (e.g., using a deployment script or an action)
```

Jenkins:

Install Jenkins: Set up a Jenkins server on your preferred infrastructure.

Install Plugins: Install necessary plugins like "GitHub Integration Plugin," "Pipeline," and any others required for your specific build and deployment tools.

Create a New Item: In Jenkins, create a new "Pipeline" project.

Configure Pipeline:

Pipeline Definition: Choose "Pipeline script from SCM" and select "Git."

Repository URL: Enter your GitHub repository URL.

Credentials: Add GitHub credentials (username and personal access token).

Branch Specifier: Specify the branch to build (e.g., `main`).

Script Path: Specify the path to your `Jenkinsfile` in the repository.

Create a Jenkinsfile: In your GitHub repository, create a `Jenkinsfile` at the root. This file defines the pipeline stages and steps using Groovy syntax.

```
pipeline {
  agent any
  tools {
    // Define any tools required, e.g., Maven, JDK
  }
  stages {
    stage('Build') {
      steps {
        sh 'mvn clean install' // Example for a Maven project
      }
    }
    stage('Test') {
      steps {
        sh 'mvn test'
      }
    }
    stage('Deploy') {
      steps {
        echo 'Deploying application...'
        // Add your deployment commands here
      }
    }
  }
}
```

Build Triggers: Configure build triggers (e.g., "Poll SCM" to periodically check for changes, or a GitHub webhook for instant triggering)