

CI Pipeline using Github actions

CI (Continuous Integration)

Continuous Integration (CI) is a DevOps practice where:

- Developers **frequently push code** to a shared repository (GitHub)
- Every push or pull request **automatically triggers checks**
- The code is **built, tested, and validated automatically**

CI Pipeline

A **CI pipeline** is a series of automated steps that run when something happens in your repo (like a push).

Typical CI pipeline stages:

1. **Checkout code**
2. **Install dependencies**
3. **Build the application**
4. **Run unit tests**
5. **Generate reports (coverage, artifacts, etc.)**

GitHub Actions

GitHub Actions is GitHub's built-in automation tool that lets you:

- Define CI/CD pipelines using **YAML (Yet Another Markup Language)**
- Run jobs on GitHub-hosted runners (Linux, Windows, macOS)
- Trigger workflows on events like:
 - `Push` , `pull_request` , `workflow_dispatch` (manual)

Core Components of GitHub Actions

1. Workflow

- A YAML file
- Stored inside:
`.github/workflows/`

2. Event (Trigger)

Defines when the pipeline starts

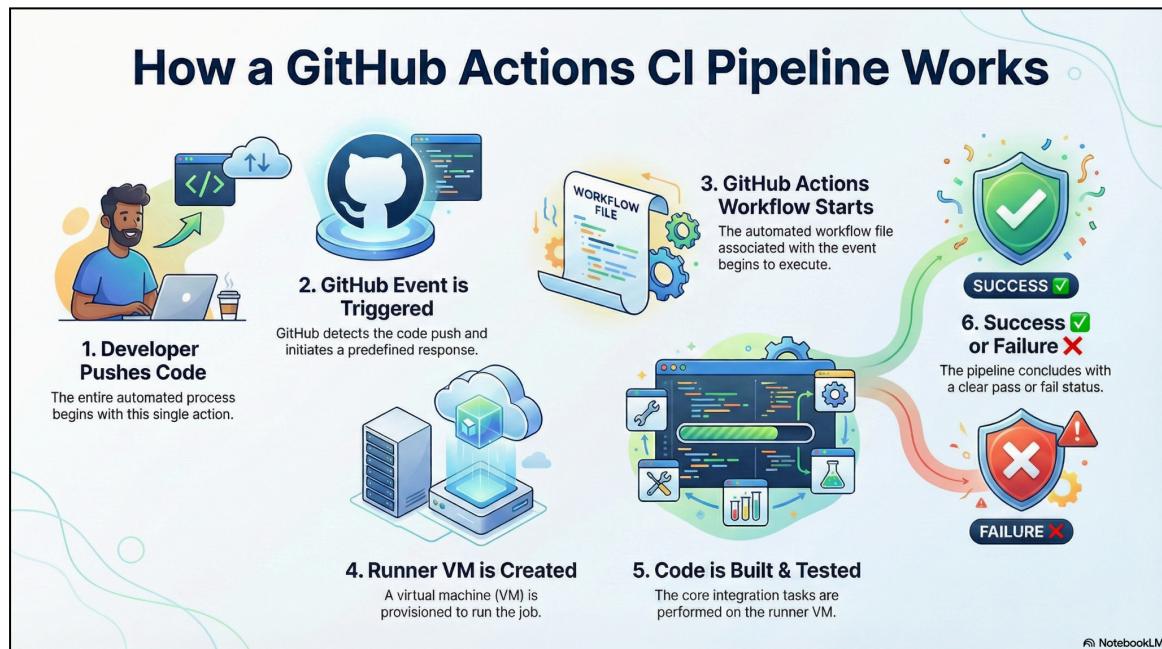
Examples:

- `push` → when code is pushed
- `pull_request` → when PR is opened
- `workflow_dispatch` → manual trigger

3. Job

- A job runs on a virtual machine (runner)

Simple CI Pipeline Flow



CI Pipeline Example (Code)

```
name: Maven CI

on:
  push:
    branches: [ "master", "feature/**" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Set up JDK 21
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '21'
          cache: maven

      - name: Build and run tests with JaCoCo
        run: mvn clean test

      - name: Upload JaCoCo Report
        uses: actions/upload-artifact@v4
        with:
          name: jacoco-report
          path: target/site/jacoco
```

Code Explanation

Workflow Name

name: Maven CI

- This is the **name of the CI pipeline**

- It appears in the **GitHub Actions tab**

Trigger Events (on)

```
on:
  push:
    branches: [ "master", "feature/**" ]
  pull_request:
    branches: [ "master" ]
```

Push Trigger

Pipeline runs when code is **pushed** to:

- **master** branch
- any branch starting with **feature/** (example: **feature/login**)

Pull Request Trigger

- Pipeline runs when a **pull request is created or updated**
- Target branch must be **master**

Jobs Section

- ```
jobs:
 build:
 • Defines a job named build
 • A job is a set of steps that run together on one machine
```

## Runner Environment

```
runs-on: ubuntu-latest
```

- GitHub provides a **virtual machine**
- OS used: **latest Ubuntu Linux**

## Steps (Actual CI Actions)

Steps are executed **top to bottom**.

### Step 1: Checkout Code

```
- name: Checkout code
 uses: actions/checkout@v4
```

- Downloads your **GitHub repository code**
- Without this, the runner has **no project files**

## Step 2: Set up Java (JDK 21)

```
- name: Set up JDK 21
 uses: actions/setup-java@v4
 with:
 distribution: 'temurin'
 java-version: '21'
 cache: maven
```

- Installs **Java JDK 21**
- Uses **Eclipse Temurin** distribution

## Step 3: Build & Test with JaCoCo

- ```
- name: Build and run tests with JaCoCo
  run: mvn clean test
```
- Runs Maven command:
 - `clean` → deletes old build files
 - `test` → runs **JUnit tests**
 - JaCoCo plugin (configured in `pom.xml`) generates **code coverage report**

Step 4: Upload JaCoCo Report

```
- name: Upload JaCoCo Report
  uses: actions/upload-artifact@v4
  with:
    name: jacoco-report
    path: target/site/jacoco
```

Uploads the **JaCoCo HTML report** & Stored as a **build artifact** in GitHub Actions

Github Actions Result

Triggered via push 3 days ago
Xcelevator-Vettrivel pushed -o- 764673d master Status Success Total duration 29s Artifacts 1

maven-ci.yml
on: push

build 27s

[] - +

Artifacts
Produced during runtime

Name	Size	Digest
jacoco-report	39.4 KB	sha256:68aa619bfb59a80c72981c72a05c1028d2...

build
succeeded 3 days ago in 27s

Search logs

> Set up job	2s
> Checkout code	1s
> Set up JDK 21	0s
> Build and run tests with JaCoCo	18s
> Upload JaCoCo Report	1s
> Post Set up JDK 21	2s
> Post Checkout code	0s
> Complete job	0s