

[CS304] Tutorial 11 - Use CI/CD tools

Using Jenkins for CI/CD

Install and run Jenkins

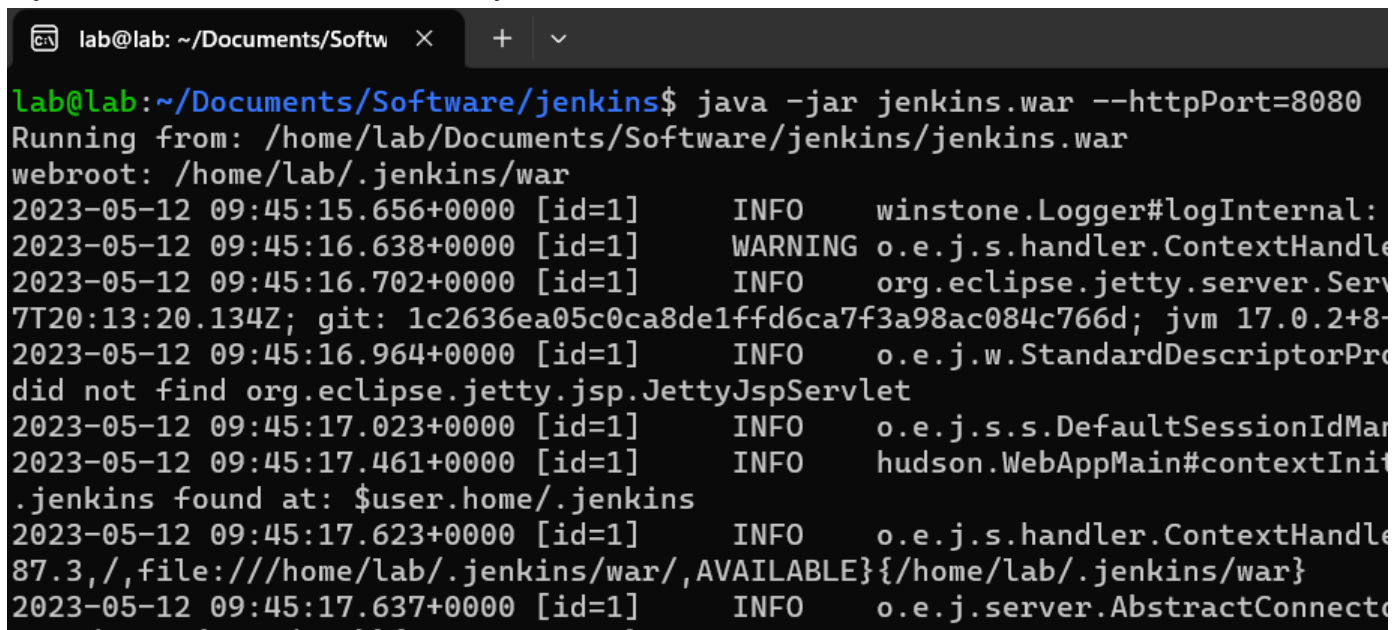
In order to use Jenkins, first download Jenkins Generic Java package (.war)

<https://www.jenkins.io/download/>, put this war file wherever you want.

Open a terminal, "cd" to the directory contains the war file and run:

```
java -jar jenkins.war --httpPort=8080
```

If you installed Maven in wsl, then you should execute the above command in wsl:

A terminal window with a dark background. The title bar shows 'lab@lab: ~/Documents/Softw' and window controls. The terminal text shows the command 'java -jar jenkins.war --httpPort=8080' being executed. The output includes the path to the war file, the webroot, and a series of log messages from the winstone logger. A key message states: 'jenkins found at: \$user.home/.jenkins'. The logs also mention the git commit hash '1c2636ea05c0ca8de1fffd6ca7f3a98ac084c766d' and the JVM version '17.0.2+8'.

```
lab@lab:~/Documents/Software/jenkins$ java -jar jenkins.war --httpPort=8080
Running from: /home/lab/Documents/Software/jenkins/jenkins.war
webroot: /home/lab/.jenkins/war
2023-05-12 09:45:15.656+0000 [id=1] INFO winstone.Logger#logInternal:
2023-05-12 09:45:16.638+0000 [id=1] WARNING o.e.j.s.handler.ContextHandle
2023-05-12 09:45:16.702+0000 [id=1] INFO org.eclipse.jetty.server.Serv
7T20:13:20.134Z; git: 1c2636ea05c0ca8de1fffd6ca7f3a98ac084c766d; jvm 17.0.2+8-
2023-05-12 09:45:16.964+0000 [id=1] INFO o.e.j.w.StandardDescriptorPro
did not find org.eclipse.jetty.jsp.JettyJspServlet
2023-05-12 09:45:17.023+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdMan
2023-05-12 09:45:17.461+0000 [id=1] INFO hudson.WebAppMain#contextInit
.jenkins found at: $user.home/.jenkins
2023-05-12 09:45:17.623+0000 [id=1] INFO o.e.j.s.handler.ContextHandle
87.3,,file:///home/lab/.jenkins/war/,AVAILABLE}{/home/lab/.jenkins/war}
2023-05-12 09:45:17.637+0000 [id=1] INFO o.e.j.server.AbstractConnecto
```

The output in the window will tell you an initial password. Copy that password.

Open you browser and go to: <http://localhost:8080>

The initial username is "admin". After you paste the initial password showing that you have access to your computer, Jenkins will ask you to finish some initial steps. If you fail to see your password, you can still find it in /home/username/.jenkins/secrets/initialAdminPassword. Follow the steps to finish installation. Choose "Install Recommended Plugins" if you are new to Jenkins.



After that you may want to install "Maven Intergration Plugin" in your Jenkins. Open "Manage Plugins" and find it under "Available Plugins" and install it:

Jenkins

Dashboard > Manage Jenkins

+ 新建Item

用户列表

构建历史

Manage Jenkins

My Views

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

Manage Jenkins

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

Set up agent

System Configuration

Configure System

Configure global settings and paths.

Global Tool Configuration

Configure tools, tool automatic installation

Manage Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clusters

Add, remove, configure various nodes that can build

Restart Jenkins. The plugins need to restart.

Create Jenkins project

Now Let's use Jenkins to manage your Teedy project. Click "New Item" on the Jenkins startup page:

Jenkins

localhost:8080

Dashboard >

+ 新建Item

用户列表

构建历史

Manage Jenkins

My Views

构建队列

队列中没有构建任务

欢迎来到 Jenkins!

This page is where your Jenkins jobs will be displayed. To get distributed builds or start building a software project.

Start building your software project

Create a job

Type a name here and choose "Pipeline". Then click OK:

 **Jenkins**

查找 (CTRL+K)

Dashboard > 所有 >

输入一个任务名称

teedy-cicd

» 必填项



构建一个自由风格的软件项目
这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件以外的系统.



构建一个maven项目
构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.





流水线
精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当称为工作流），增加或者组织难以采用自由风格的任务类型。


Select "pipeline" on the left and choose "Pineline script from SCM" option from definition. Choose Git under "SCM" and fill your repo URL, as follows (this tells Jenkins to obtain your project from your git repo):

Dashboard > teedy-cicd > Configuration

Configure

 General

 高级项目选项

 流水线

定义

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/selab722/Teedy.git

Credentials ?

- 无 -

+ 添加

高级

保存

应用

In case your repo is a private one, you need a personal access token: [How to use Github Personal Access Token in Jenkins](#).

Click "Save" and go to next stage.

Create Jenkins pipeline

First add a file named "Jenkinsfile" in your Teedy repo which defines a pipeline with many stages:

```
pipeline {
  agent any

  stages {
    stage('Clean') {
      steps {
        sh 'mvn clean'
      }
    }

    stage('Compile') {
      steps {
        sh 'mvn compile'
      }
    }

    stage('Test') {
      steps {
        sh 'mvn test -Dmaven.test.failure.ignore=true'
      }
    }

    stage('PMD') {
      steps {
        sh 'mvn pmd:pmd'
      }
    }

    stage('JaCoCo') {
      steps {
        sh 'mvn jacoco:report'
      }
    }

    stage('Javadoc') {
      steps {
        sh 'mvn javadoc:javadoc'
      }
    }

    stage('Site') {
      steps {
        sh 'mvn site'
      }
    }
  }
}
```

```

stage('Package') {
    steps {
        sh 'mvn package -DskipTests'
    }
}

post {
    always {
        archiveArtifacts artifacts: '**/target/site/**/*.*', fingerprint: true
        archiveArtifacts artifacts: '**/target/**/*.*.jar', fingerprint: true
        archiveArtifacts artifacts: '**/target/**/*.*.war', fingerprint: true
        junit '**/target/surefire-reports/*.xml'
    }
}
}


```

The "post" section above tells Jenkins to store these files after the build for later use.

Commit and push the Jenkinsfile as above:

The screenshot shows the GitHub repository interface for 'selab722/Teedy'. The file list includes 'COPYING', 'Dockerfile', 'Jenkinsfile' (circled in red), 'README.md', 'docs.xml', and 'pom.xml'. Below the file list, there are tabs for 'README', 'Code of conduct', and 'License'. The main content area displays the 'teedy' logo in blue and a 'License GPL v2' badge at the bottom left.

Click "Build Now" and wait for results:



Jenkins

查找 (CTRL+K)

Dashboard > teedy-cicd >

状态

变更历史

立即构建

配置

删除 Pipeline

完整阶段视图

重命名

流水线语法

teedy-cicd

Teey Maven project pipeline

阶段视图

No data available. This Pipeline has not yet run.

相关链接

Jenkins will pull the Git repo newest commit to Jenkins server (in this case on local computer) to execute the pipeline.

During your build, you can click "#19" to see this build ("#19" means the first build in your project). You can see the progress on the right.

状态

变更历史

立即构建

配置

删除 Pipeline

完整阶段视图

重命名

流水线语法

Pipeline demo-25

上次成功的成品

Build History 构建历史

Filter builds... /

#19

2025年4月25日 下午3:40

#18

2025年4月25日 下午3:04

#15

2025年4月25日 下午2:30

#14

2025年4月25日 下午2:16

#13

2025年4月25日 下午1:53

#12

阶段视图

Average stage times:

	Declarative: Checkout SCM	Clean	Compile	Test	PMD	JaCoCo	Javadoc	Site	Package	Declarative: Post Actions
#19	1s	2s	6s	8min 57s	5s	2s	9s	8min 14s	5s	21s
#18	1s	2s	6s	10min 3s	5s	2s	10s	9min 54s	5s	22s

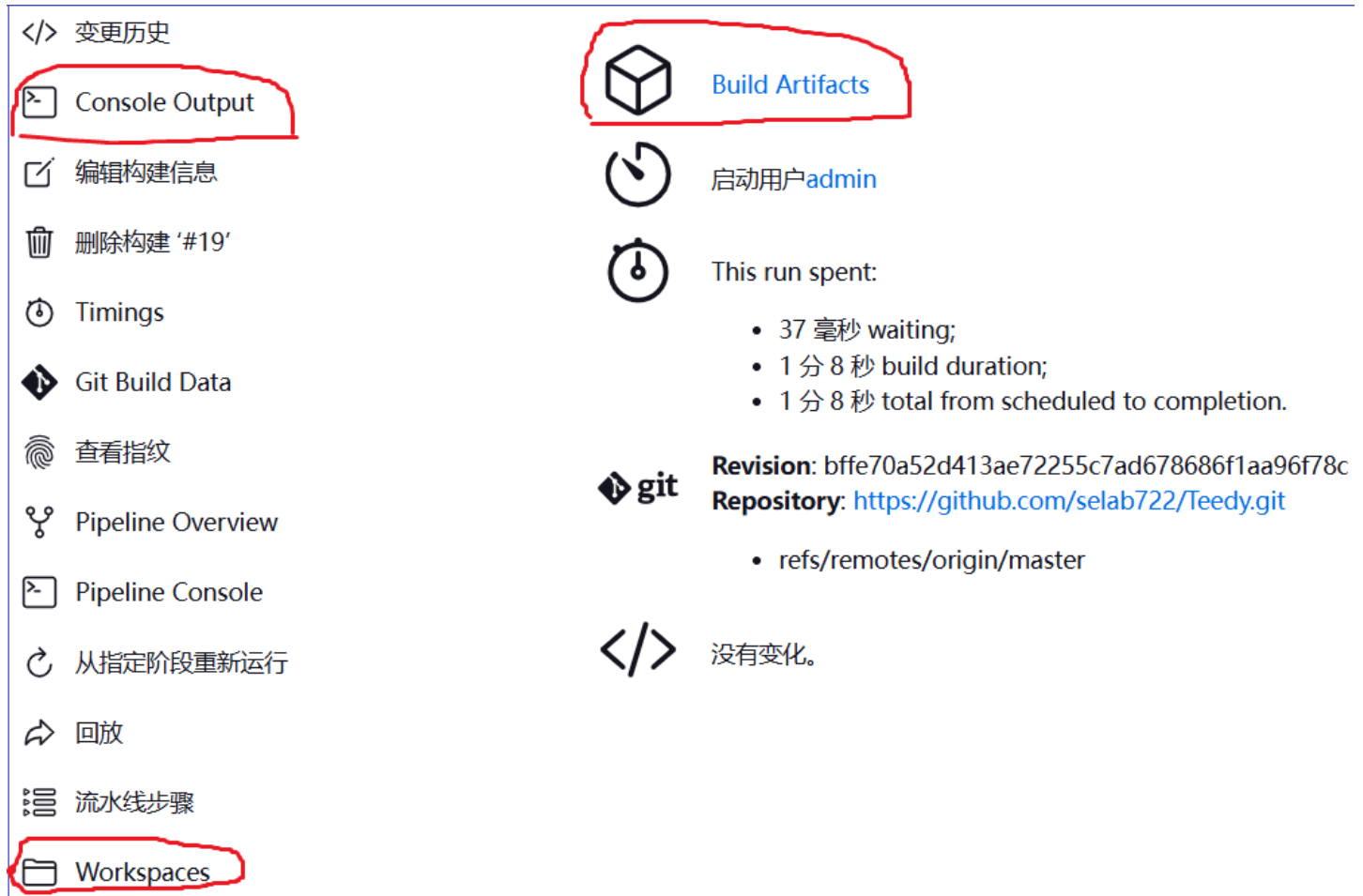
测试结果趋势

Passed Skipped Failed

71 72 73 74 75

#13 #18 #19

Click the "#19" and see what's in all results:



</> 变更历史

Console Output

编辑构建信息

删除构建 '#19'

Timings

Git Build Data

查看指纹

Pipeline Overview

Pipeline Console

从指定阶段重新运行

回放

流水线步骤

Workspaces

Build Artifacts

启动用户 admin

This run spent:

- 37 毫秒 waiting;
- 1 分 8 秒 build duration;
- 1 分 8 秒 total from scheduled to completion.

Revision: bffe70a52d413ae72255c7ad678686f1aa96f78c

Repository: <https://github.com/selab722/Teedy.git>

- refs/remotes/origin/master

没有变化。

Click the "Build Artifacts" link, then all files specified "post" section in the Jenkins file above can be downloaded. In this case jars and sites are stored.

The "Console Output" link contains all the console output during this build.

"Workspaces" lists all files of this repo that stores in Jenkins server.

If you find that your html files in "Build Artifacts" cannot display, this probably because Jenkins refuse to render js and css. Check [Index.html file - is not displaying with CSS and basic html displayed when the file is created during the jenkins build](#) for explanation.

demo-25 #19 的成品

/ target / site /

→

apidocs

css

images

jacoco-aggregate

cpd.html

dependency-convergence.html

dependency-info.html

dependency-management.html

distribution-management.html

index.html

modules.html

plugin-management.html

plugins.html

pmd.html

project-info.html

project-reports.html

scm.html

summary.html

surefire-report.html

2025年4月25日 下午3:49:48

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:48

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:52

2025年4月25日 下午3:49:48

13.46 KB

3.56 KB

3.64 KB

28.11 KB

3.79 KB

4.21 KB

4.04 KB

4.47 KB

7.02 KB

63.44 KB

5.86 KB

4.00 KB

3.95 KB

4.53 KB

66.88 KB

(打包下载全部文件)

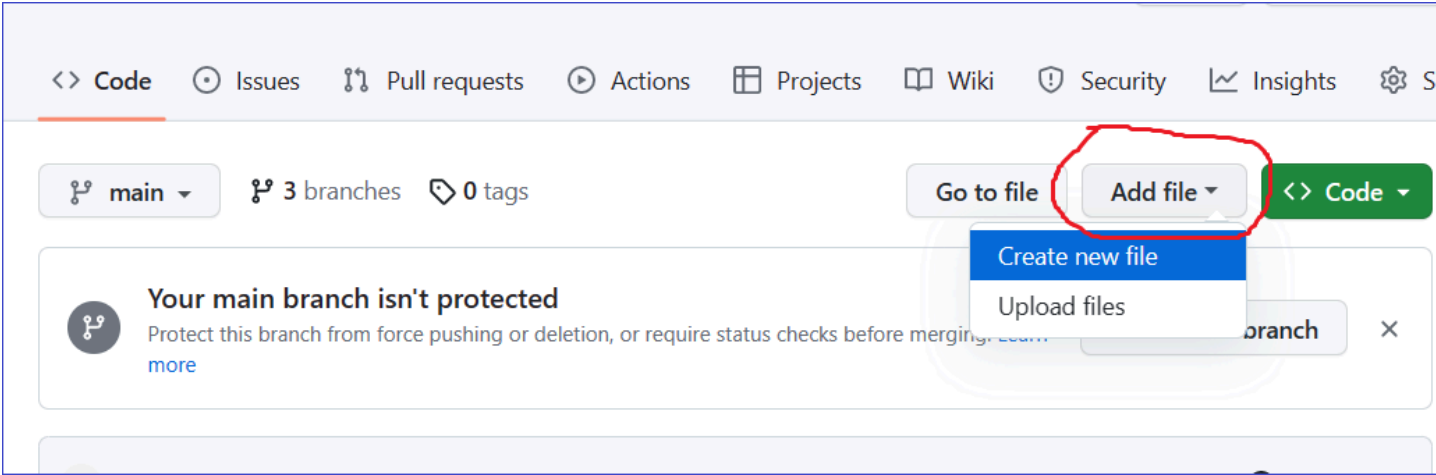
Github Actions

Add Github Actions to Teedy

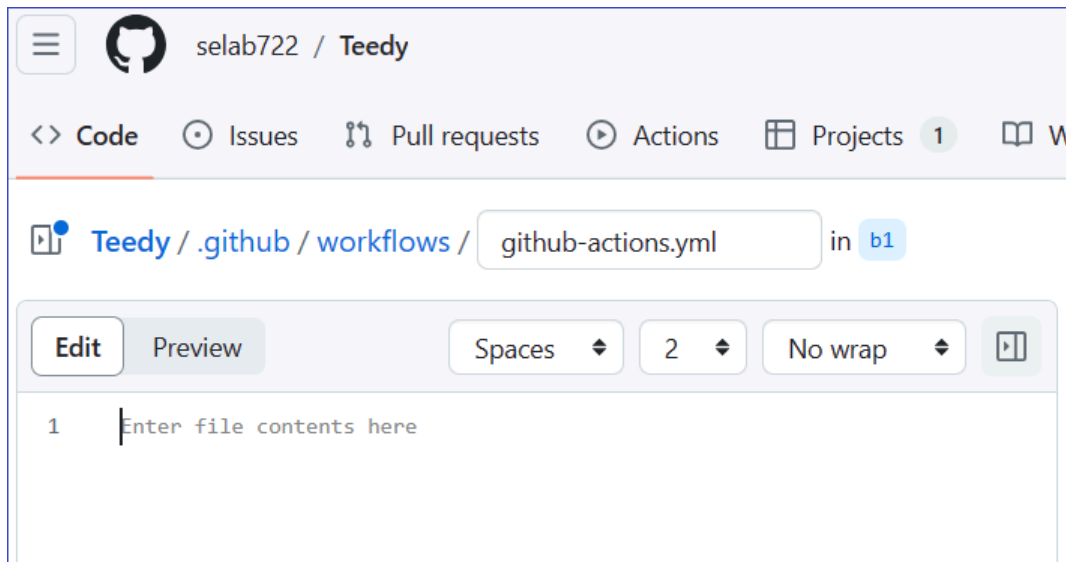
Github also has its own CI/CD tools. It is called "Github Actions".

Since we already created hello-world project, we add actions to this project. Follow the instructions in: <https://docs.github.com/en/actions/quickstart>.

First let's create a ".github/workflows/github-actions.yml" file in the project:



The file name can be different, but make sure it's under ".github/workflows" directory and it ends with ".yml":



Next, what should we put in this file? We can find a simple example of yml for Maven project here: <https://github.com/actions/starter-workflows/blob/main/ci/maven.yml>. Let's copy and paste the contents of the file to our yml file:

```
name: Java CI with Maven

on:
  push:
    branches: [ $default-branch ]
  pull_request:
    branches: [ $default-branch ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 11
      uses: actions/setup-java@v3
      with:
        java-version: '11'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B -DskipTests package --file pom.xml
```

After that, commit this file to a new branch and start a pull request. This will trigger this workflow and the Github Actions will run it.

Configure when Github Action is triggered

However, if you commit this file change to the default branch, you may find out that it didn't run later. You can still run it manually according to:

<https://docs.github.com/en/actions/managing-workflow-runs/manually-running-a-workflow>.

Let's modify our github-actions.yml file to:

```
name: Java CI with Maven

on:
  [ push, pull_request, workflow_dispatch ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3
      - name: Set up JDK 11
        uses: actions/setup-java@v3
        with:
          java-version: '11'
          distribution: 'temurin'
          cache: maven
      - name: Build with Maven
        run: mvn -B -DskipTests package --file pom.xml
```

With workflow_dispatch added to the config, we are able to manually run the action:

The screenshot shows the GitHub Actions interface for the repository 'selab722 / Teedy'. The 'Actions' tab is selected and circled in red. In the left sidebar, 'Java CI with Maven' is selected and circled in red with a handwritten '2'. The main area shows the workflow 'Java CI with Maven' with a search bar and a 'Run workflow' button circled in red with a handwritten '3'. Below this, two workflow runs are listed, both titled 'update jenkins' and pushed by 'selab722'.

Event	Status	Branch	Actor
This workflow has a <code>workflow_dispatch</code> event trigger.			
update jenkins	✓	master	selab722
Java CI with Maven #6: Commit d91c458 pushed by selab722			
update jenkins	✓	master	selab722
Java CI with Maven #5: Commit 36fe331 pushed by selab722			

Click and run you will see the result.

Teedy's default workflow

Teedy also contains its own workflow. See: [build-deploy](#).

In this class we only want the first part. The second part involves Docker, which will be covered in future class.

```
name: Maven CI/CD

on:
  push:
    branches: [master]
    tags: [v*]
  workflow_dispatch:

jobs:
  build_and_publish:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2
      - name: Set up JDK 11
        uses: actions/setup-java@v2
        with:
          java-version: "11"
          distribution: "temurin"
          cache: maven
      - name: Install test dependencies
        run: sudo apt-get update && sudo apt-get -y -q --no-install-recommends install ffmpeg r
      - name: Build with Maven
        run: mvn --batch-mode -Pprod clean install
      - name: Upload war artifact
        uses: actions/upload-artifact@v2
        with:
          name: docs-web-ci.war
          path: docs-web/target/docs*.war
```

Run and see the result. Notice that this time you can download .war archive here:

Summary

Jobs

build_and_publish

Run details

Usage

Workflow file

Node.js 16 actions are deprecated. Please update the following actions to use Node.js 18 or later.
[Show more](#)

⚠️ build_and_publish

The following actions uses node12 which is deprecated and will be forced to run on node16 or later.
[Show more](#)

⚠️ Deprecation notice: v1, v2, and v3 of the artifact actions

The following artifacts were uploaded using a version of actions/upload-artifact that is deprecated.
[Show more](#)

Artifacts

Produced during runtime

Name	Size	
docs-web-ci.war	81.3 MB	Download Delete

Jenkins with Github Action

You can use Github Action to trigger your jenkins, as in: [Build a Jenkins pipeline by using Jenkinsfile Runner GitHub Actions](#)