[CS304] Lab14. Use CI/CD tools

Using Jenkins for CI/CD

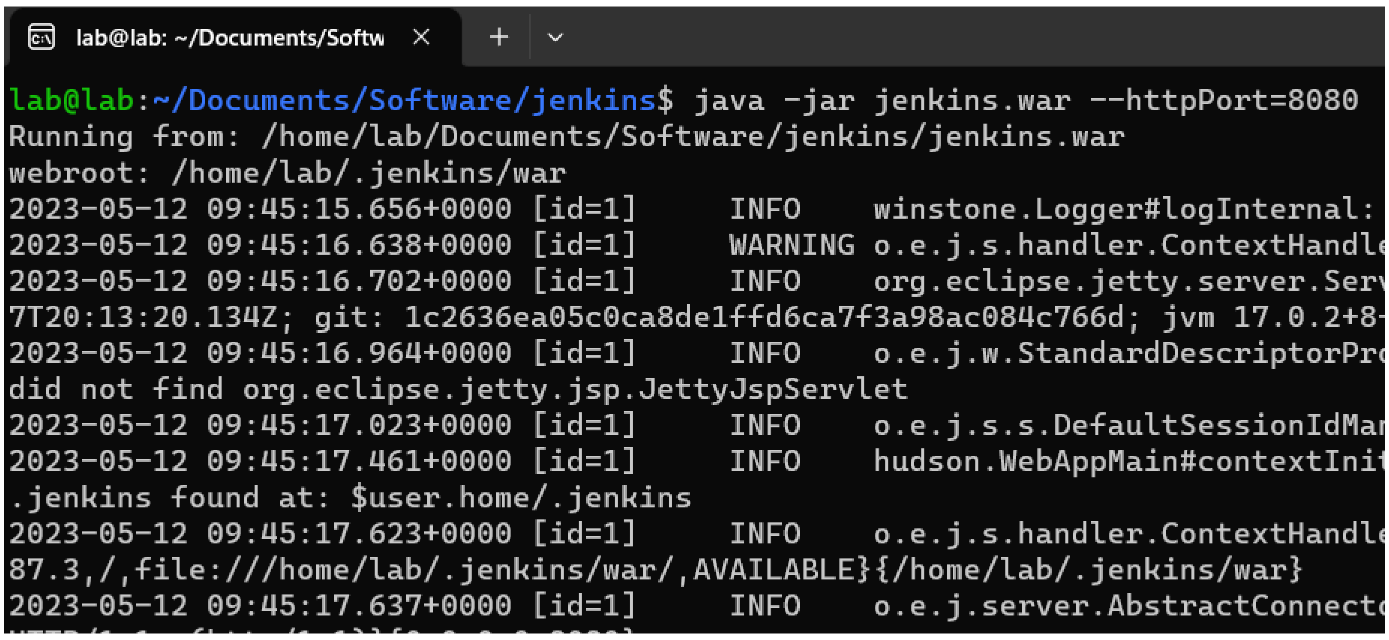
Install and run Jenkins

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:

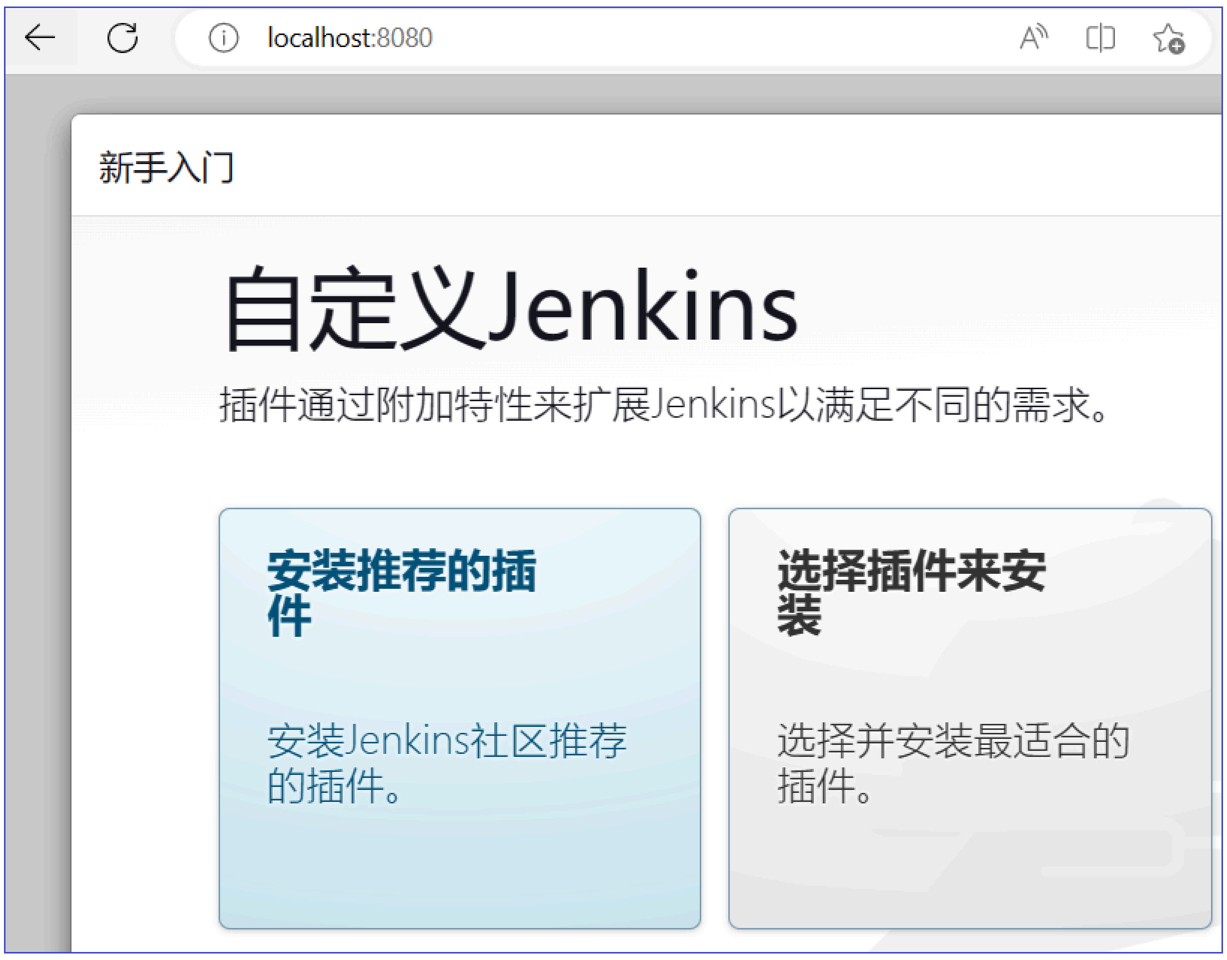


```
lab@lab:~/Documents/Software/jenkins$ java -jar jenkins.war --httpPort=8080
Running from: /home/lab/Documents/Software/jenkins/jenkins.war
webroot: /home/lab/.jenkins/war
2023-05-12 09:45:15.656+0000 [id=1]     INFO    winstone.Logger#logInternal:
2023-05-12 09:45:16.638+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandle
2023-05-12 09:45:16.702+0000 [id=1]     INFO    org.eclipse.jetty.server.Serv
7T20:13:20.134Z; git: 1c2636ea05c0ca8de1ffd6ca7f3a98ac084c766d; 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 browswer 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 Recommanded Plugins" if you are new to Jenkins.



新手入门

自定义Jenkins

插件通过附加特性来扩展Jenkins以满足不同的需求。

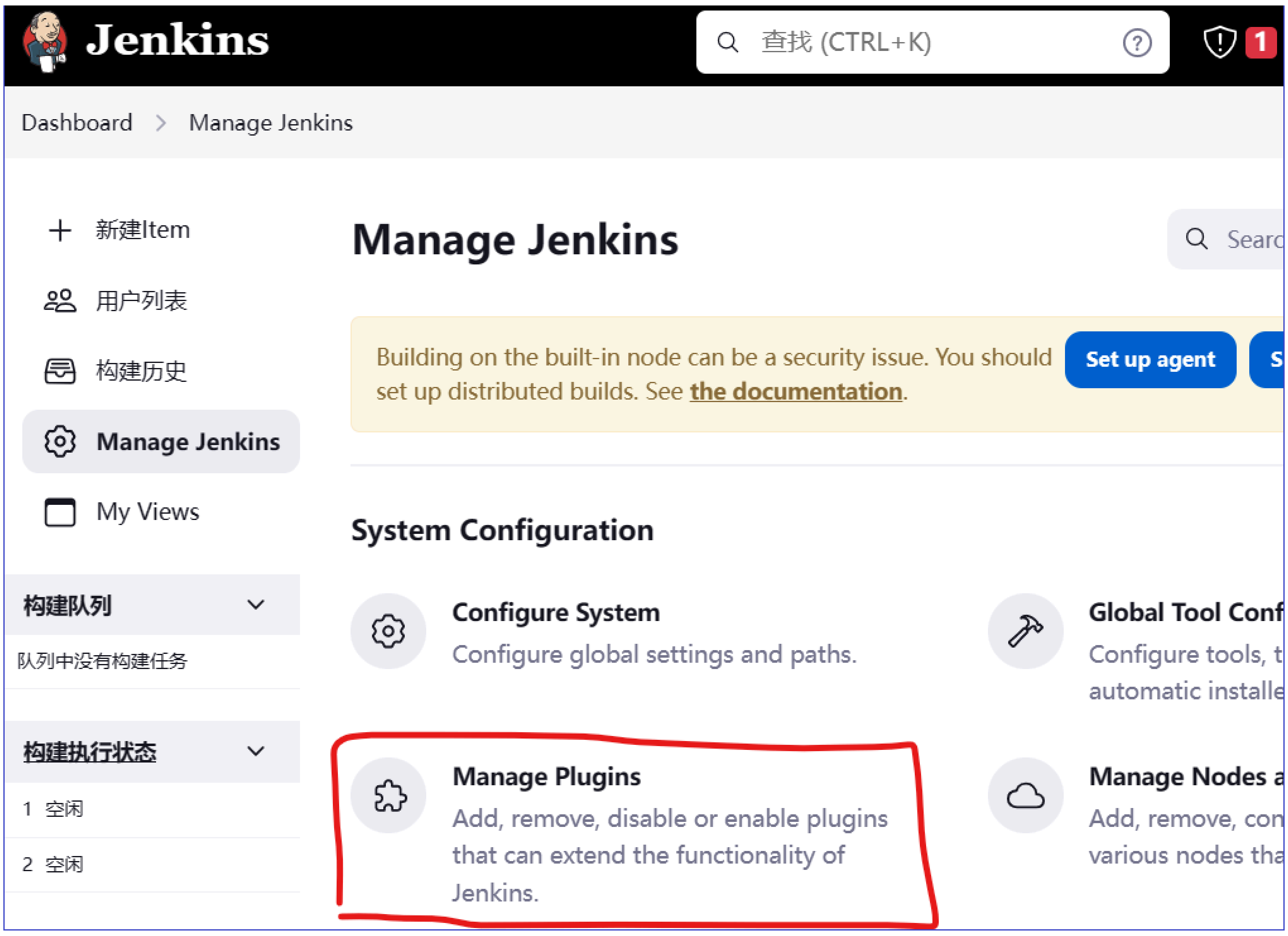
安装推荐的插件

安装Jenkins社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

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:



队列中没有构建任务

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**.

System Configuration

Configure System
Configure global settings and paths.

Global Tool Conf
Configure tools, t
automatic installe

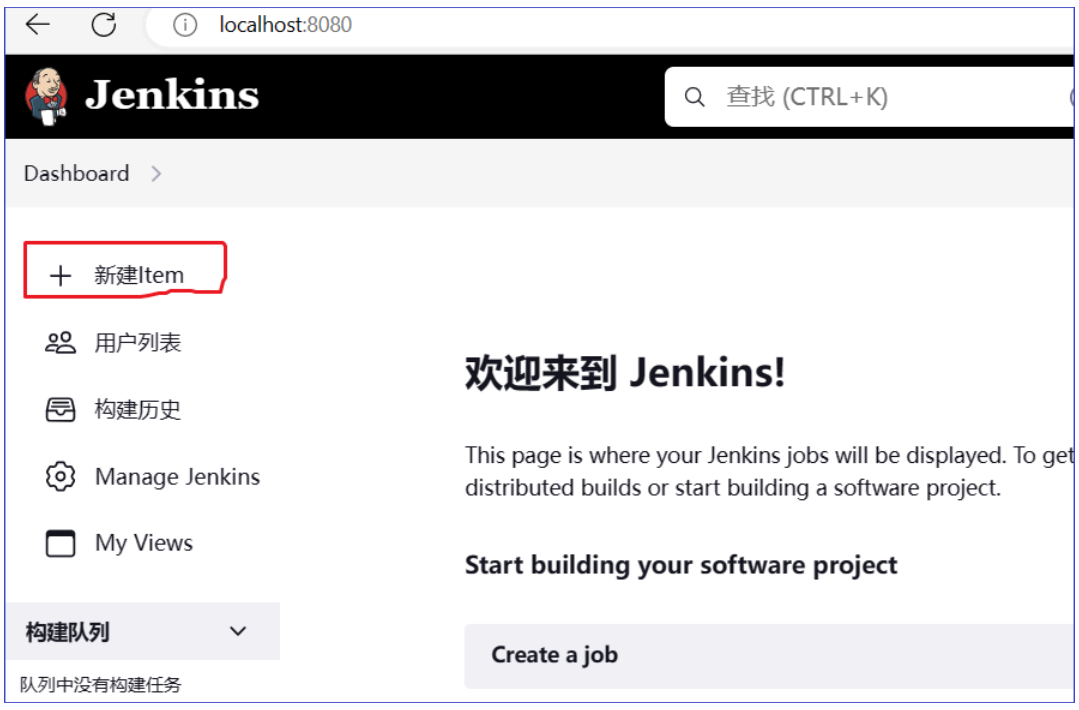
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes a
Add, remove, con
various nodes tha

Restart Jenkins. The plugins needs to restart.

Create Jenkins project

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



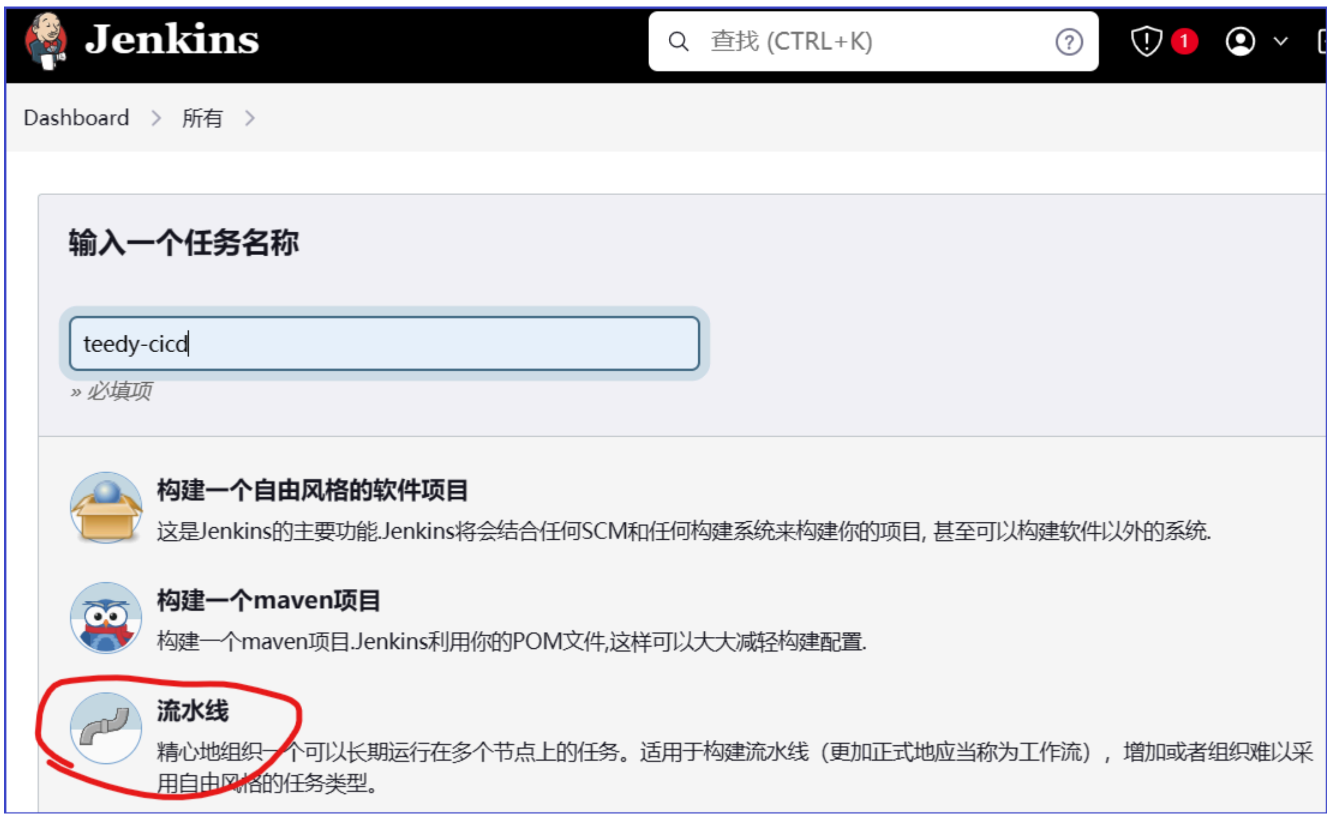
队列中没有构建任务

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

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



输入一个任务名称

teedy-cicd

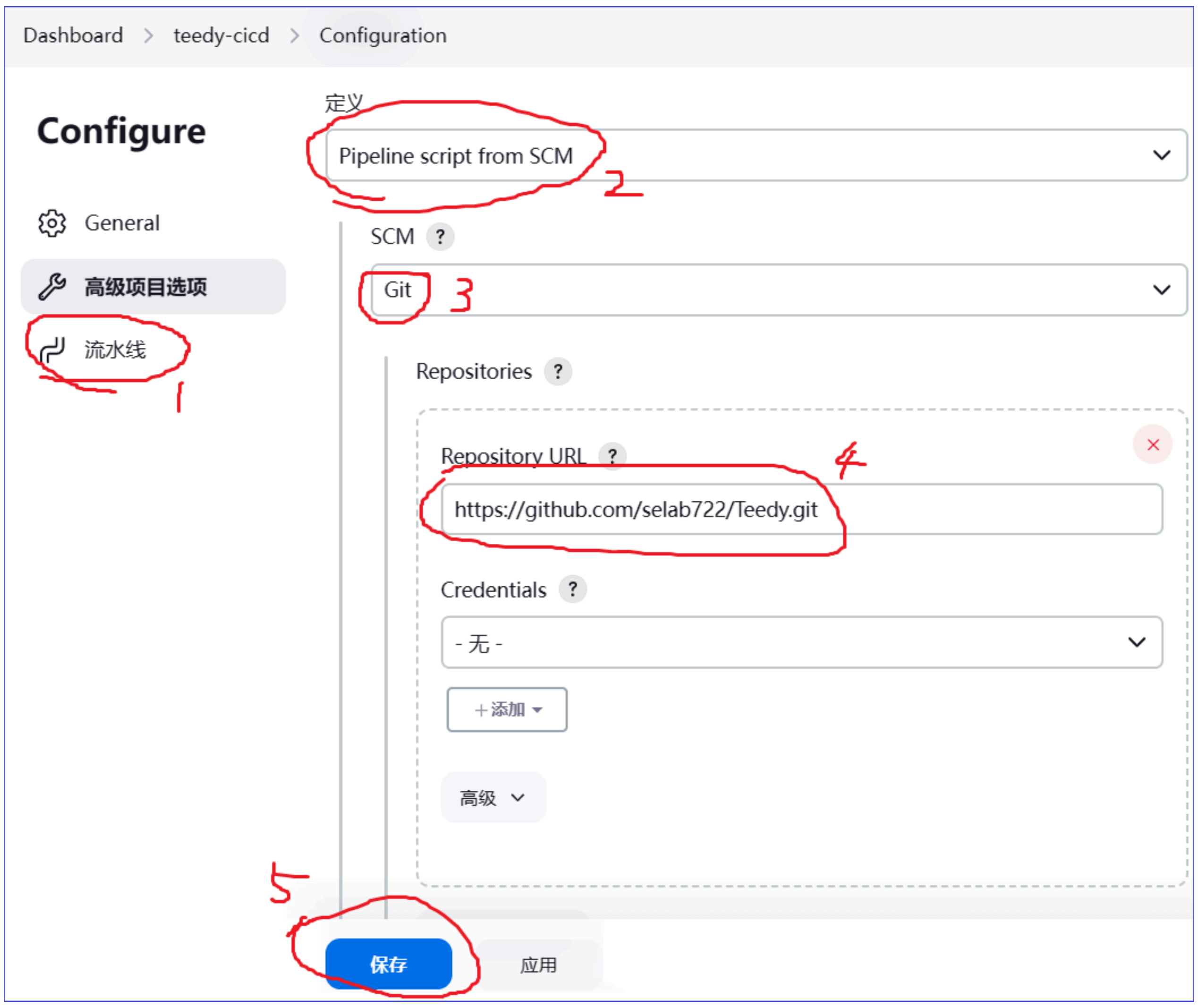
» 必填项

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 -

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

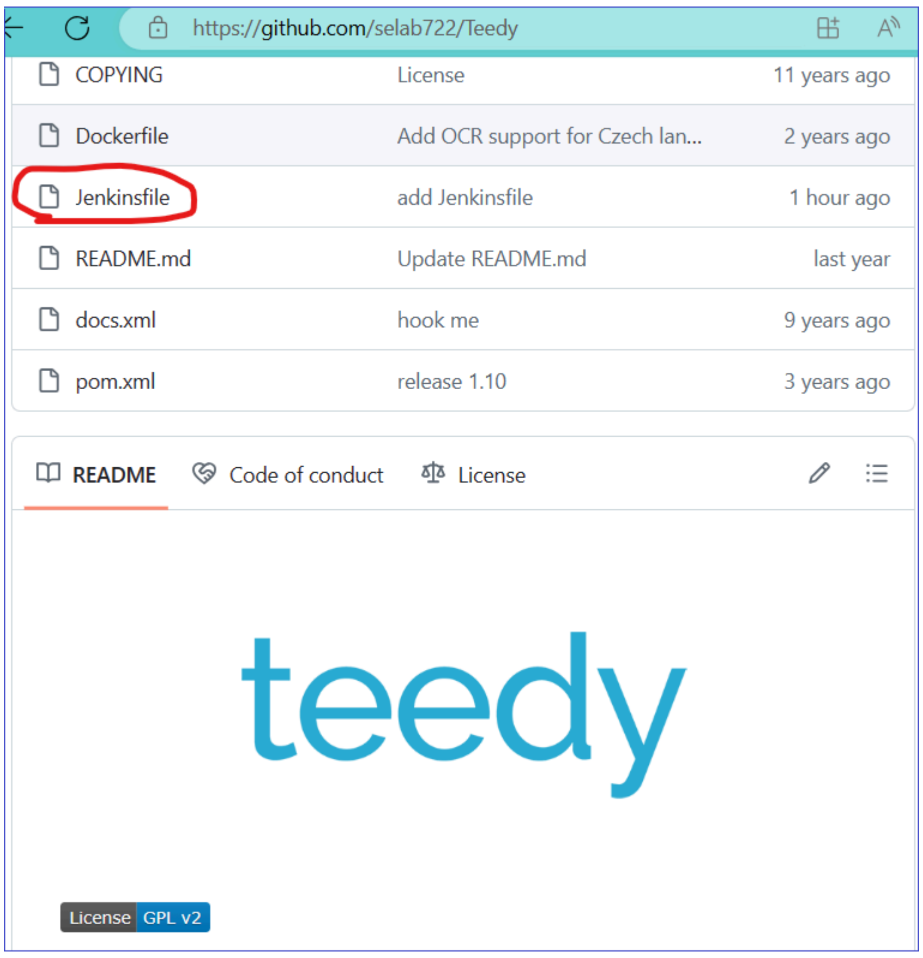
Click "Save" and go to next stage.

Create Jenkins pipeline

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

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                sh 'mvn -B -DskipTests clean package'
            }
        }
    }
}
```

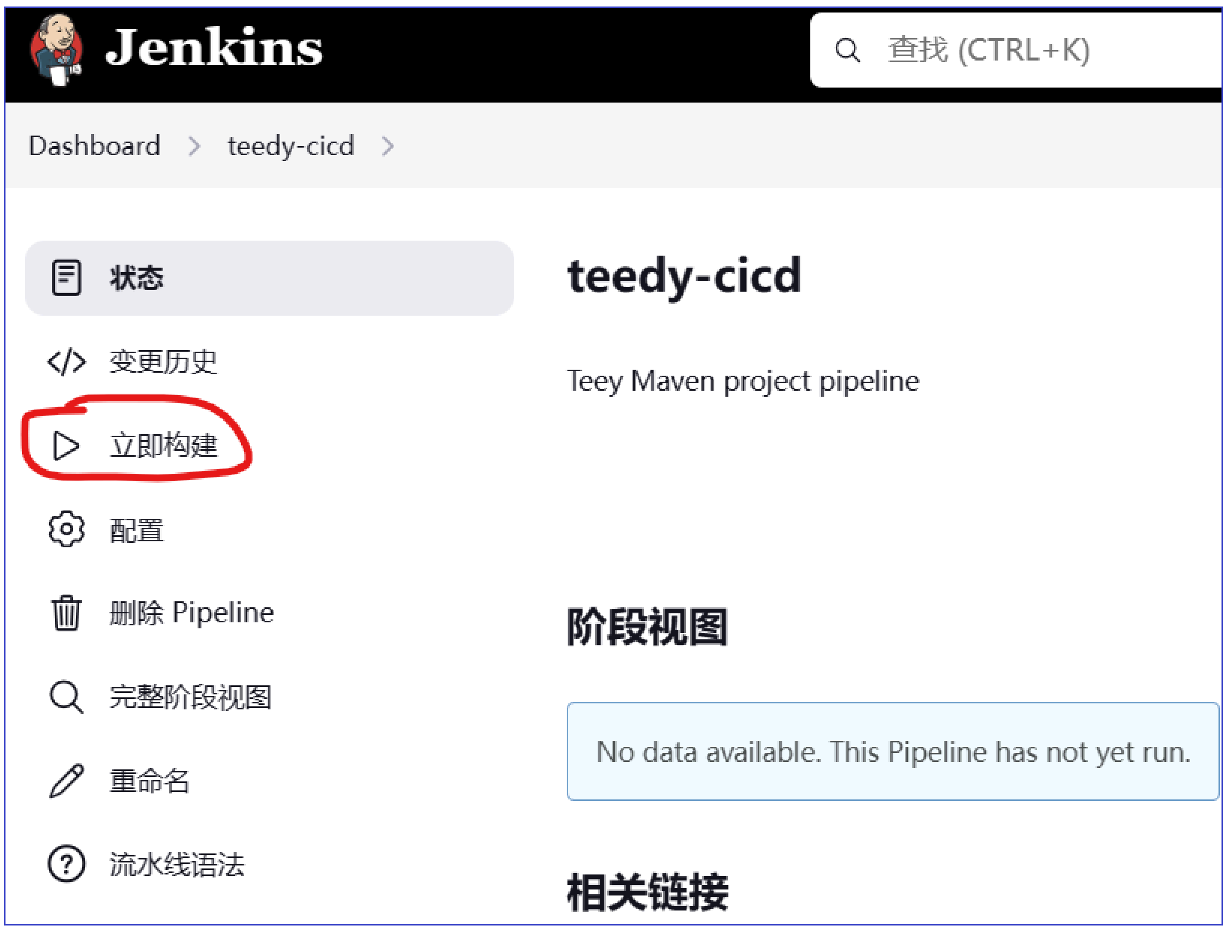
Commit and push the Jenkins as above:



https://github.com/selab722/Teedy

COPYING	License	11 years ago
Dockerfile	Add OCR support for Czech lan...	2 years ago
Jenkinsfile	add Jenkinsfile	1 hour ago
README.md	Update README.md	last year
docs.xml	hook me	9 years ago
pom.xml	release 1.10	3 years ago

Click "Build Now" and wait for results:

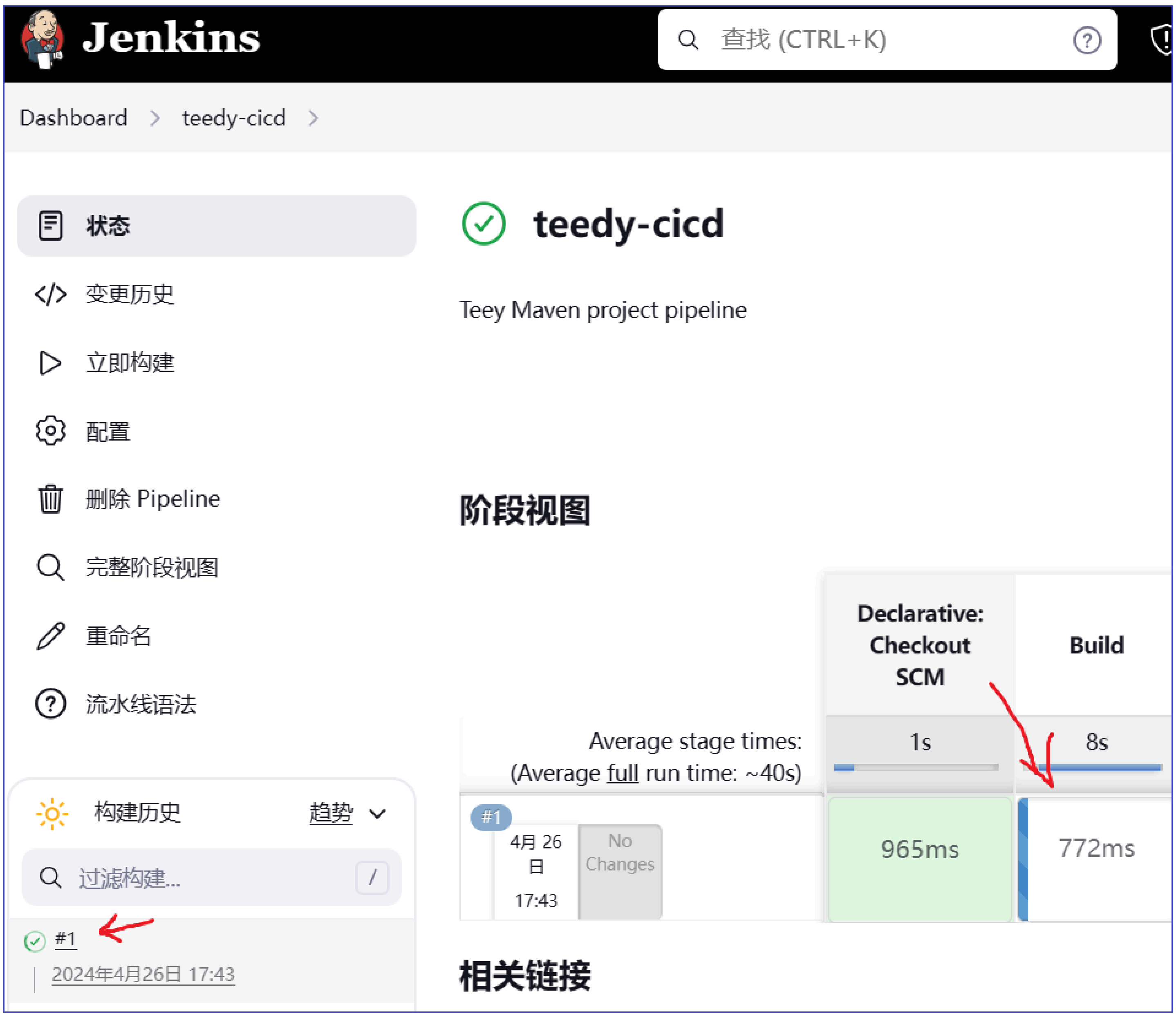


teedy-cicd

Teey Maven project pipeline

No data available. This Pipeline has not yet run.

During your build, you can click "#1" to see this build ("#1" means the first build in your project). You can see the progress on the right.



teedy-cicd

Teey Maven project pipeline

	Declarative: Checkout SCM	Build
Average stage times: (Average full run time: ~40s)	1s	8s
#1 4月 26 日 17:43 No Changes	965ms	772ms

Click the "#1" go

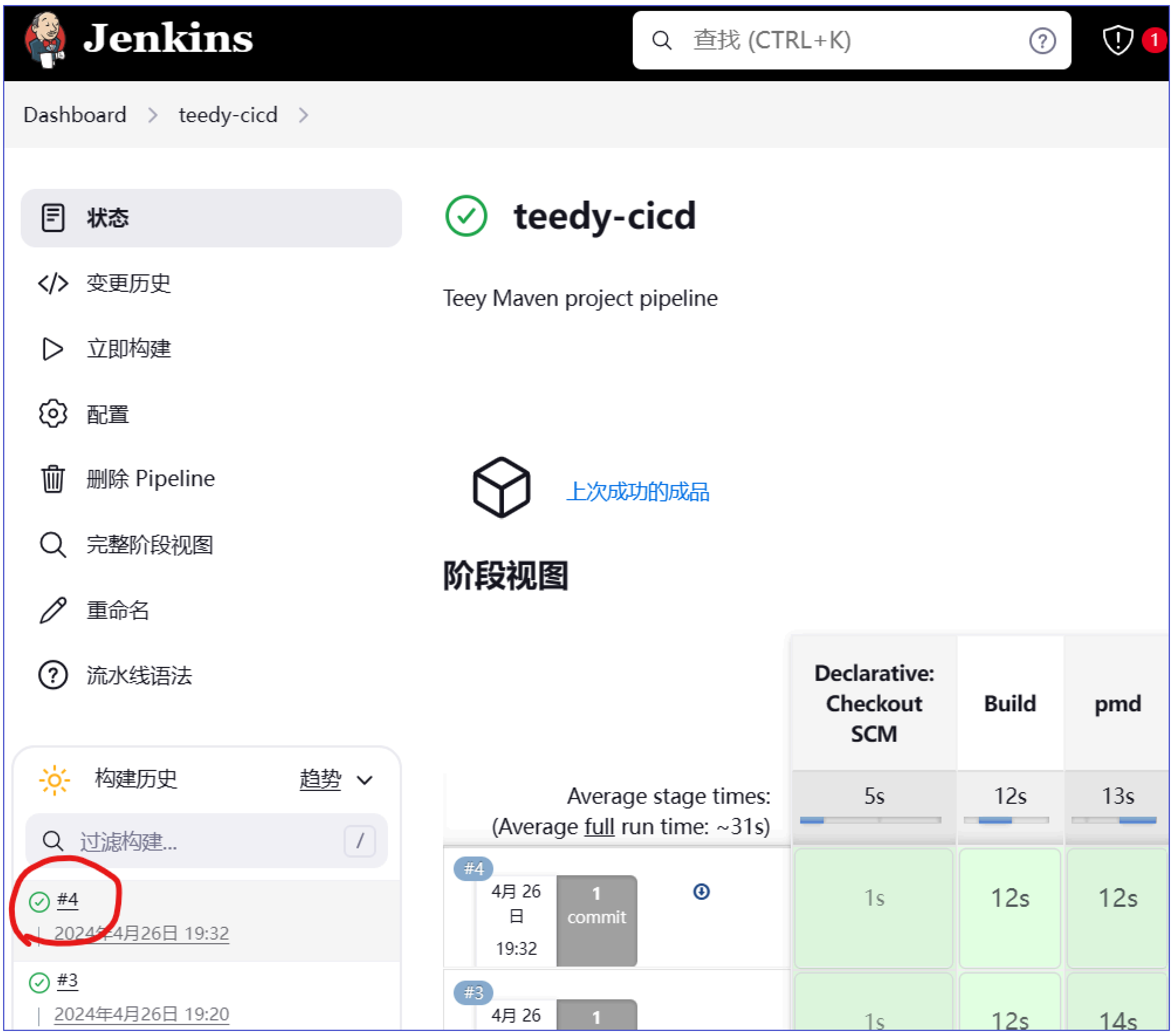
Adding pmd to pipeline

Let's add PMD plugin in our repo. First add "maven-pmd-plugin" to your pom.xml file as we did in week 7. Then update your Jenkins file:

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                sh 'mvn -B -DskipTests clean package'
            }
        }
        stage('pmd') {
            steps {
                sh 'mvn pmd:pmd'
            }
        }
    }

    post {
        always {
            archiveArtifacts artifacts: '**/target/site/**', fingerprint: true
            archiveArtifacts artifacts: '**/target/**/*.jar', fingerprint: true
            archiveArtifacts artifacts: '**/target/**/*.war', fingerprint: true
        }
    }
}
```

Here we tell Jenkins to archive all web pages in "site" directories, jar files and war files.

Again, git add, git commit, git push. Go to jenkins page, click "Build Now":



teedy-cicd

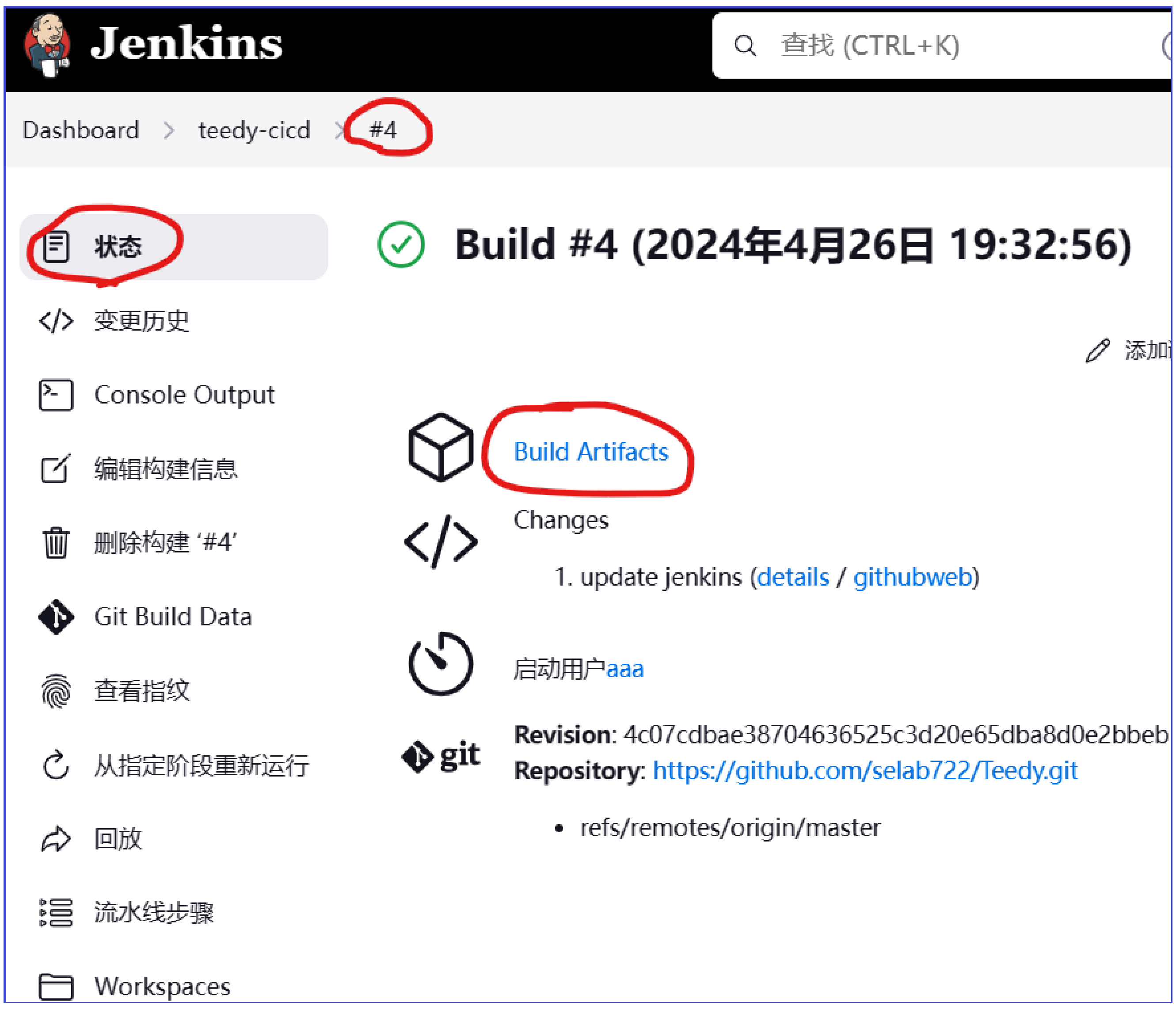
Teey Maven project pipeline

	Declarative: Checkout SCM	Build	pmd
Average stage times: (Average full run time: ~31s)	5s	12s	13s
#4 4月 26 日 19:32 1 commit	1s	12s	12s
#3 4月 26 1	1s	12s	14s

#4
2024年4月26日 19:32

#3
2024年4月26日 19:20

It says successfully build, let's click this "#4" and enter see everything (Here #4 means it's the 4th time I've built this project. On your computer it's different. I believe this won't confuse you):



Build #4 (2024年4月26日 19:32:56)

Build Artifacts

Changes

1. update jenkins (details / githubweb)

启动用户aaa

Revision: 4c07cdbae38704636525c3d20e65dba8d0e2bbeb
Repository: https://github.com/selab722/Teedy.git

- refs/remotes/origin/master

Click "Status" and then "Build Artifacts" and download the pmd reports.

Jenkins and Docker

You may want to run Jenkins in Docker. Docker will be introduced in the future.

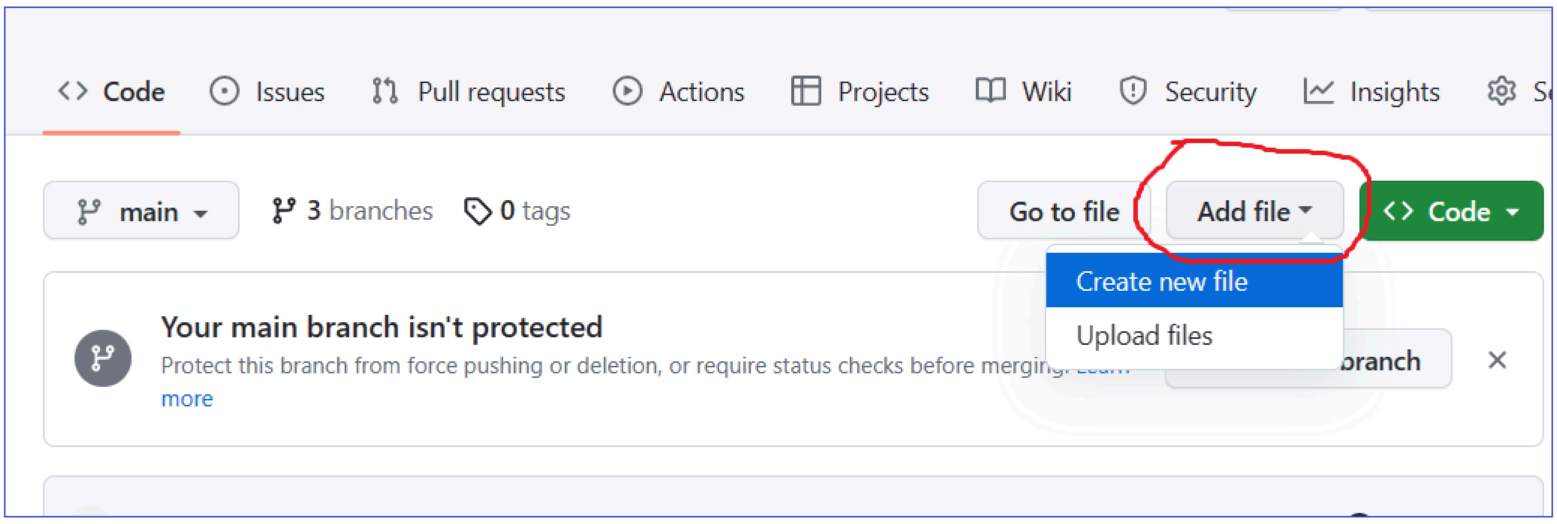
Github Actions

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.

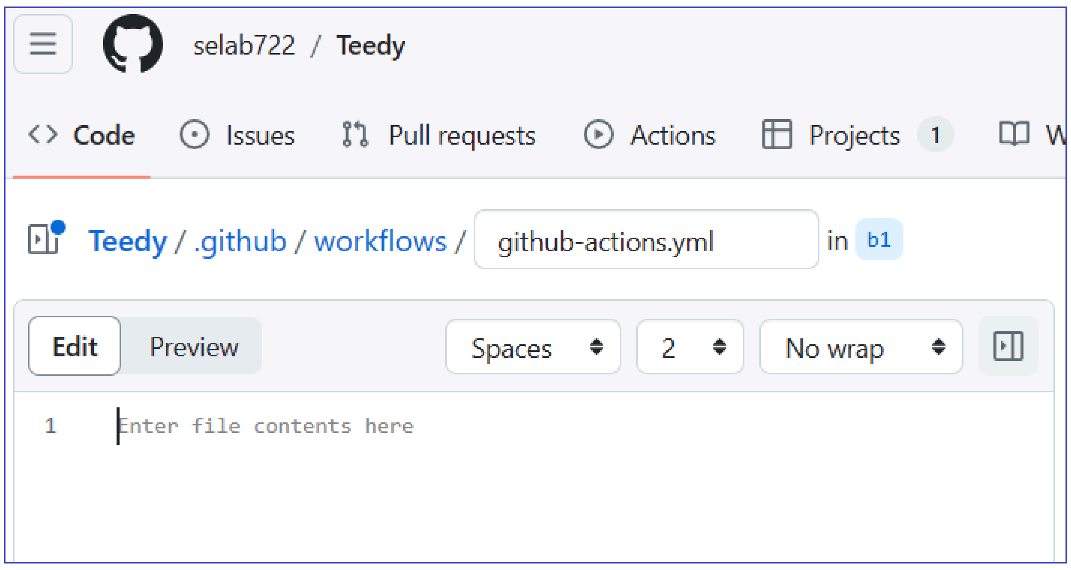
First let's create a ".github/workflows/github-actions.yml" file in the project:



Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

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



Teedy / .github / workflows / github-actions.yml in b1

```
Enter file contents here
```

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.

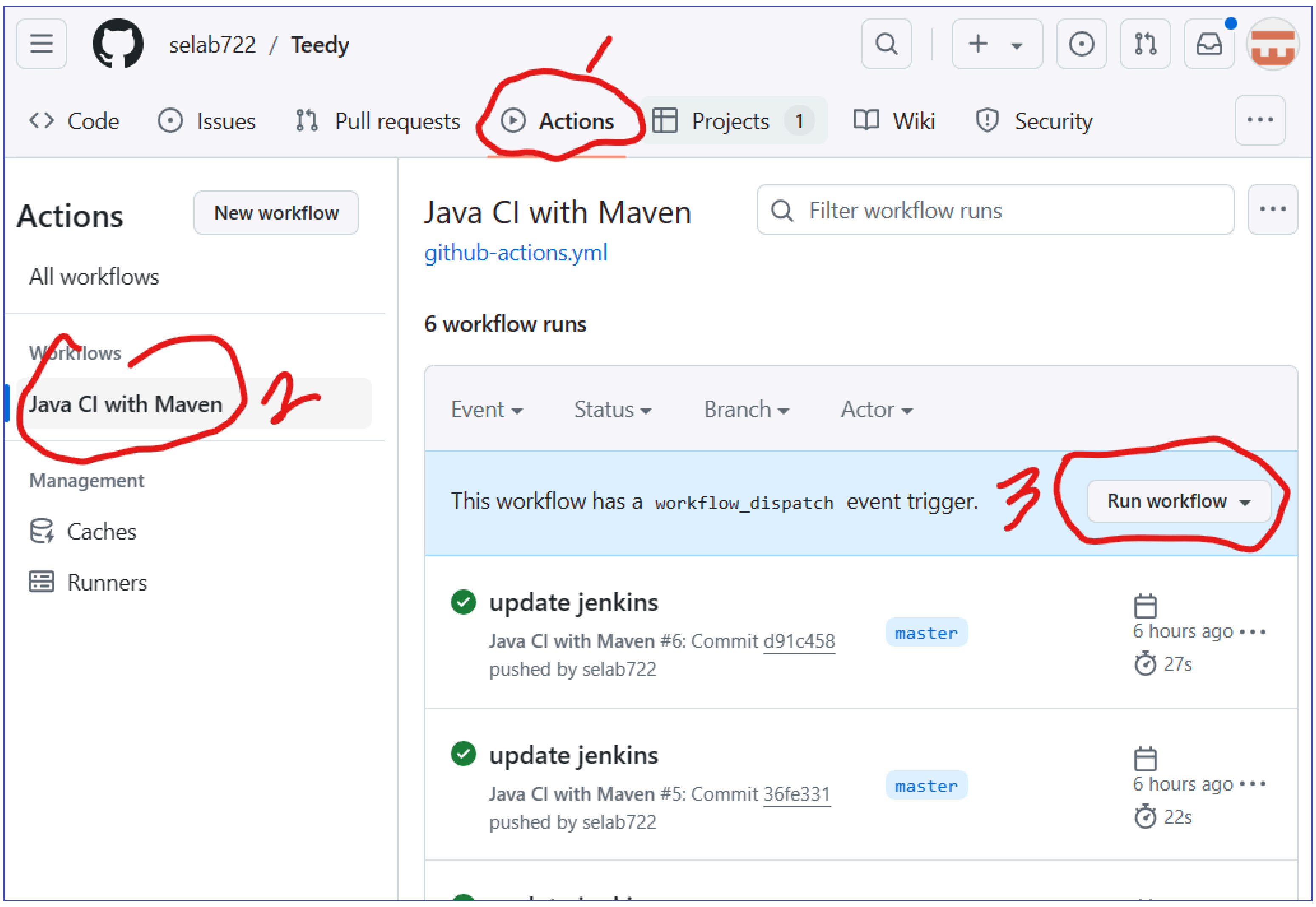
Configure when Github Action is triggered

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:



This workflow has a `workflow_dispatch` event trigger.

update jenkins
Java CI with Maven #6: Commit d91c458 pushed by selab722 · master · 6 hours ago · 27s

update jenkins
Java CI with Maven #5: Commit 36fe331 pushed by selab722 · master · 6 hours ago · 22s

Click and run you will see the result.

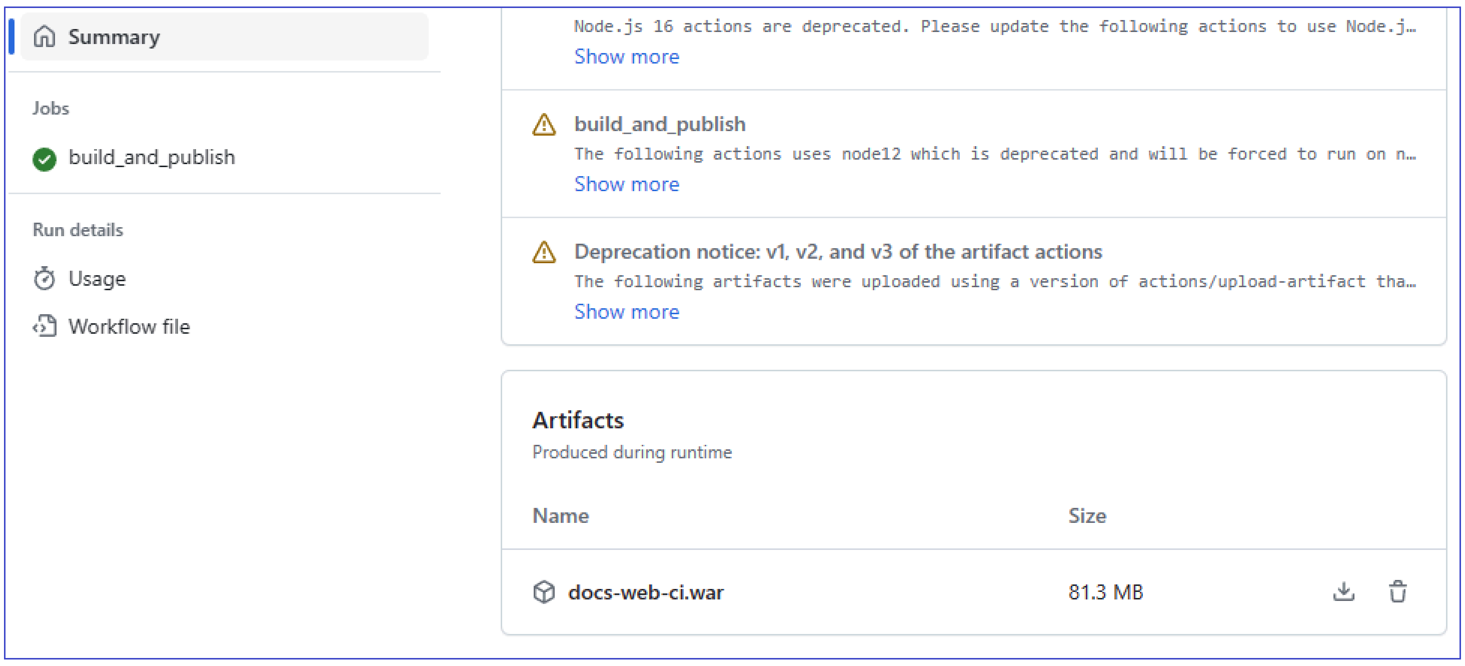
Teedy's default workflow

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 mediainfo tesseract-ocr tesseract-oc
      - 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:



Node.js 16 actions are deprecated. Please update the following actions to use Node.j…
Show more

build_and_publish
The following actions uses node12 which is deprecated and will be forced to run on n…
Show more

Deprecation notice: v1, v2, and v3 of the artifact actions
The following artifacts were uploaded using a version of actions/upload-artifact tha…
Show more

Artifacts
Produced during runtime

Name	Size
docs-web-ci.war	81.3 MB

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