[CS304] Lab14. Use CI/CD tools

Using 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/Softw

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
```

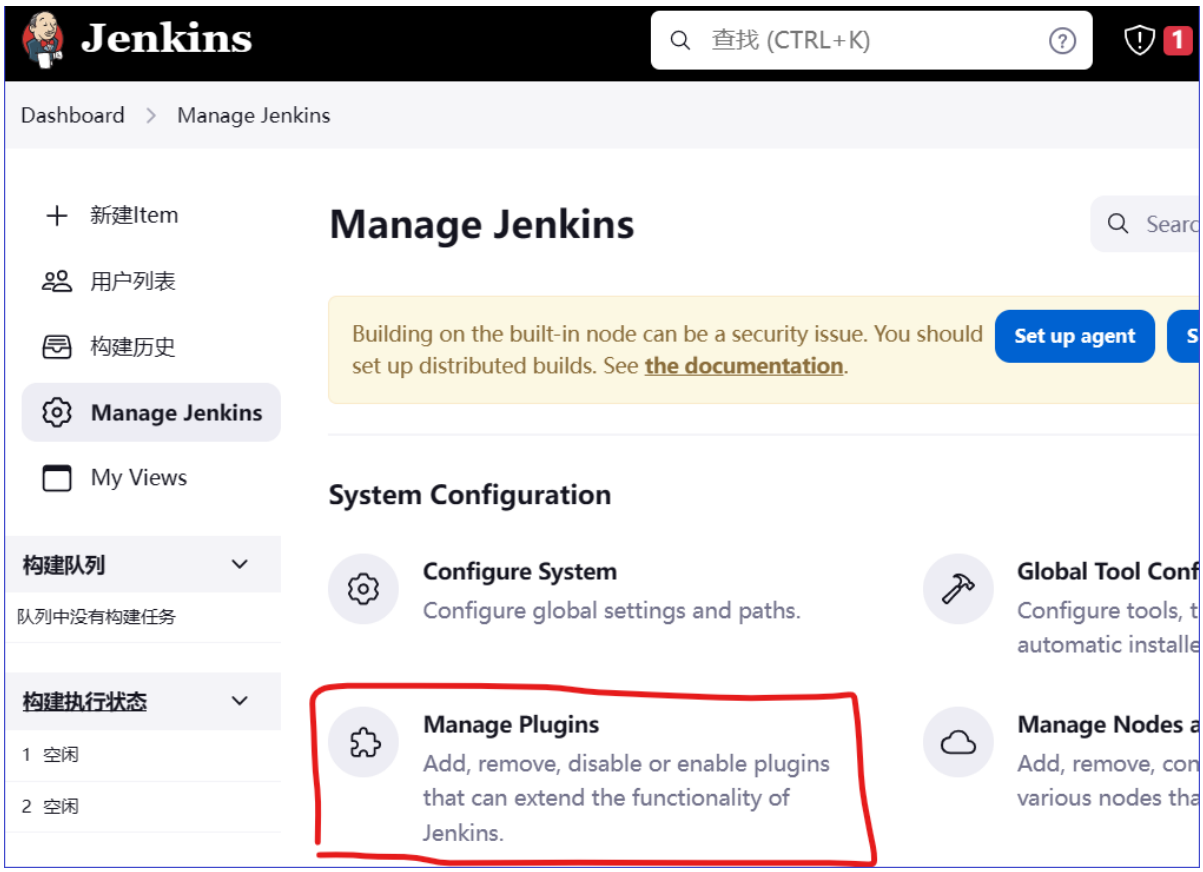
The output in the window will tell you an initial password. Copy that password.

Open you browswer and go to: http://localhost:8080

After you paste the initial password showing that you have access to your computer, Jenkins will ask you to finish some initial steps. Follow the steps to finish installation. Choose "Install

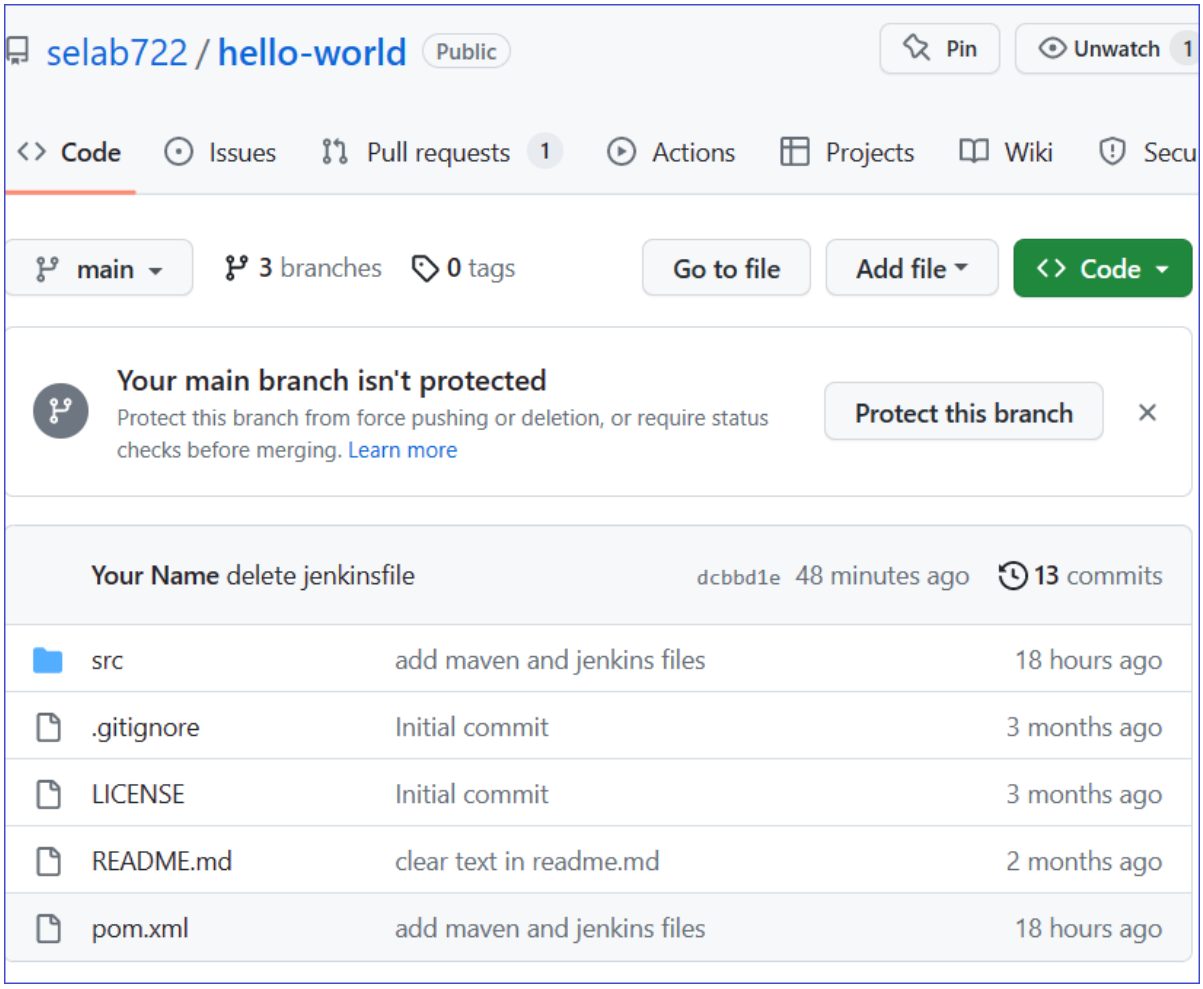
Recommanded Plugins" if you are new to Jenkins.

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:

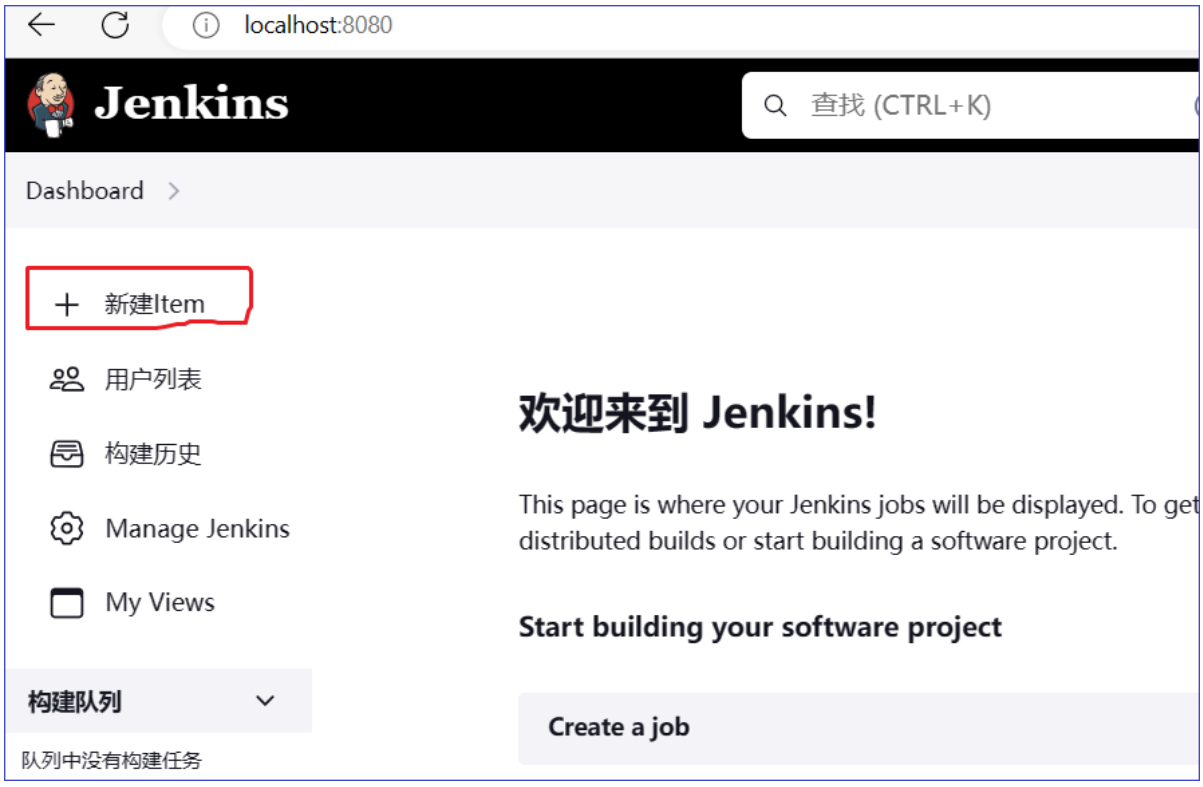


Restart Jenkins. The plugins needs to restart.

Create a Maven project on github, your project may look something like this:



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



Type a name here and choose "Build a Maven Project" here:

Then we enter a long list of configuration. First type your github url:

Then fill in the github url again. Add a credential if you need to access any privilege of your github account, but that not necessary for today:

Dashboard > hello-world > Configuration

Configure

- General
- 源码管理
- 构建触发器
- 构建环境
- Pre Steps
- Build
- Post Steps
- 构建设置
- 构建后操作

Git ?

Repositories ?

Repository URL ?

https://github.com/selab722/hello-world.git

Credentials ?

+ 添加

高级

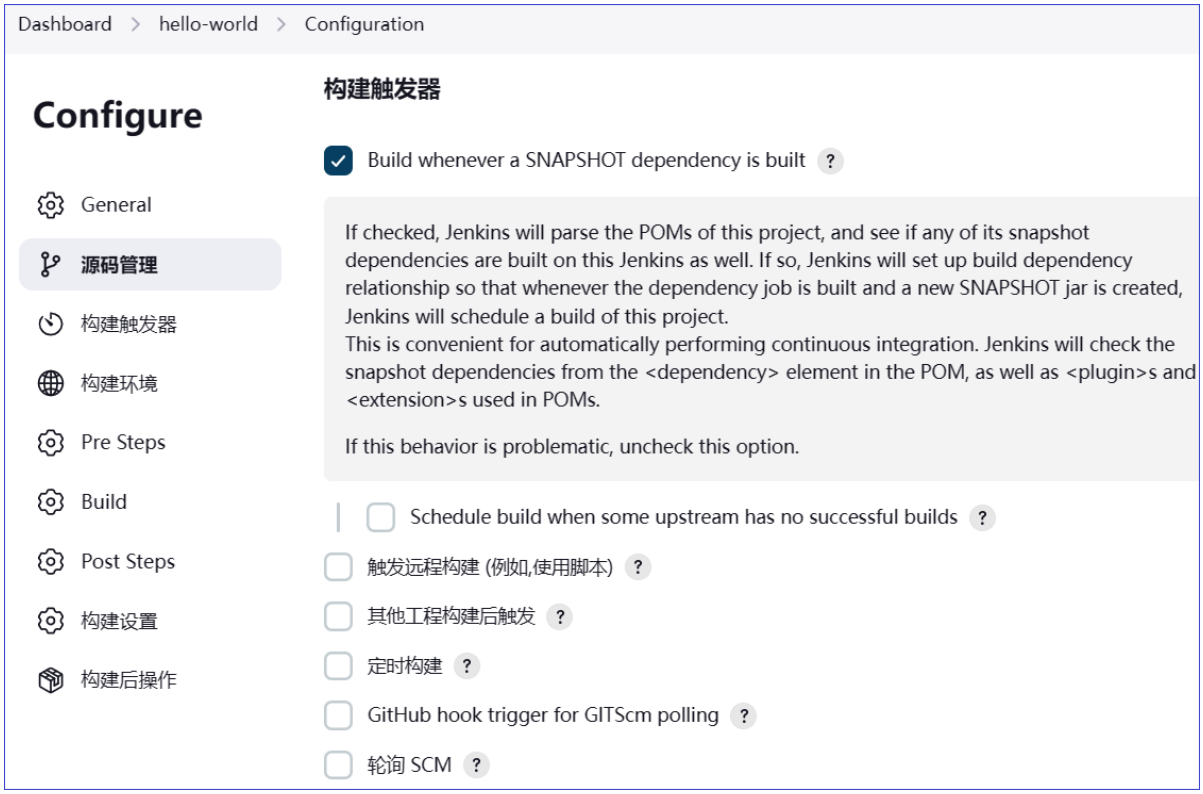
Add Repository

Branches to build ?

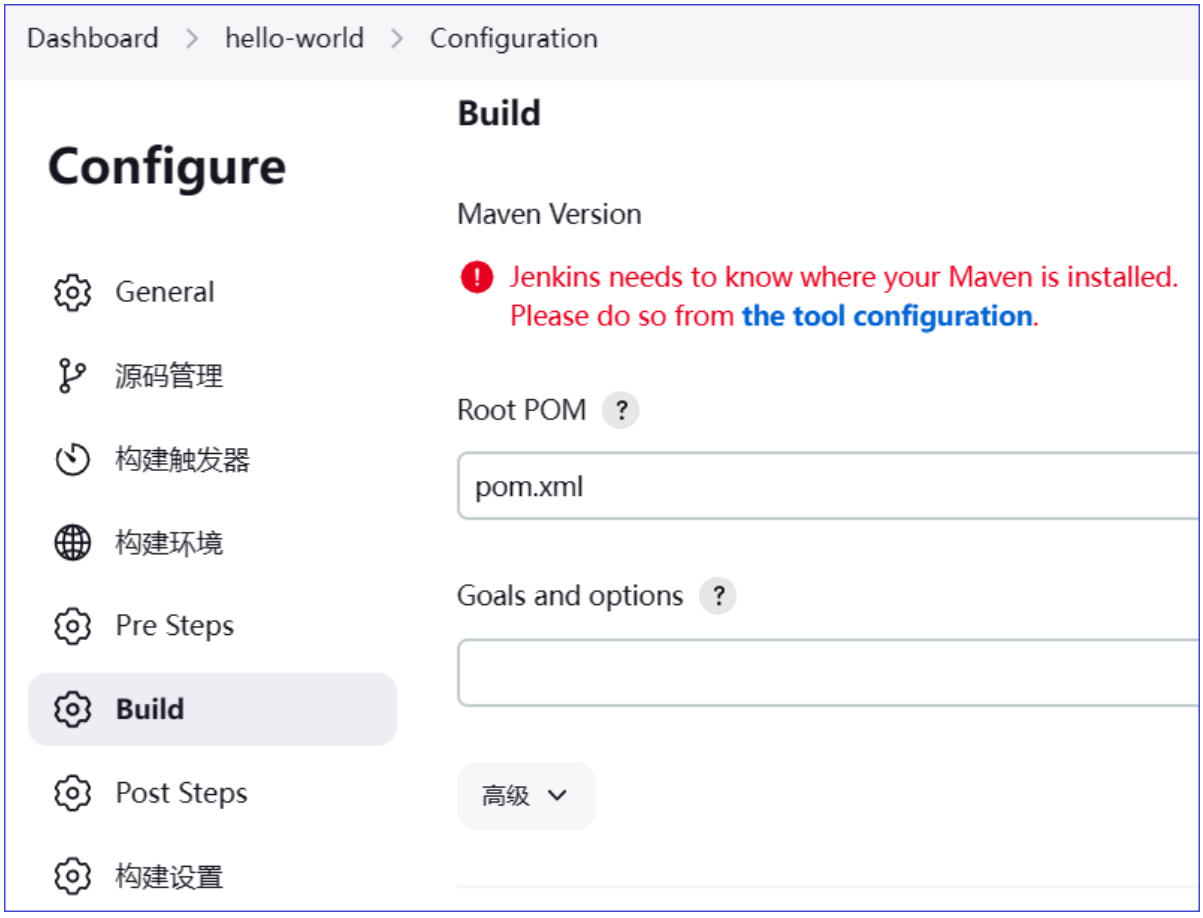
指定分支（为空时代表any） ?

*/main

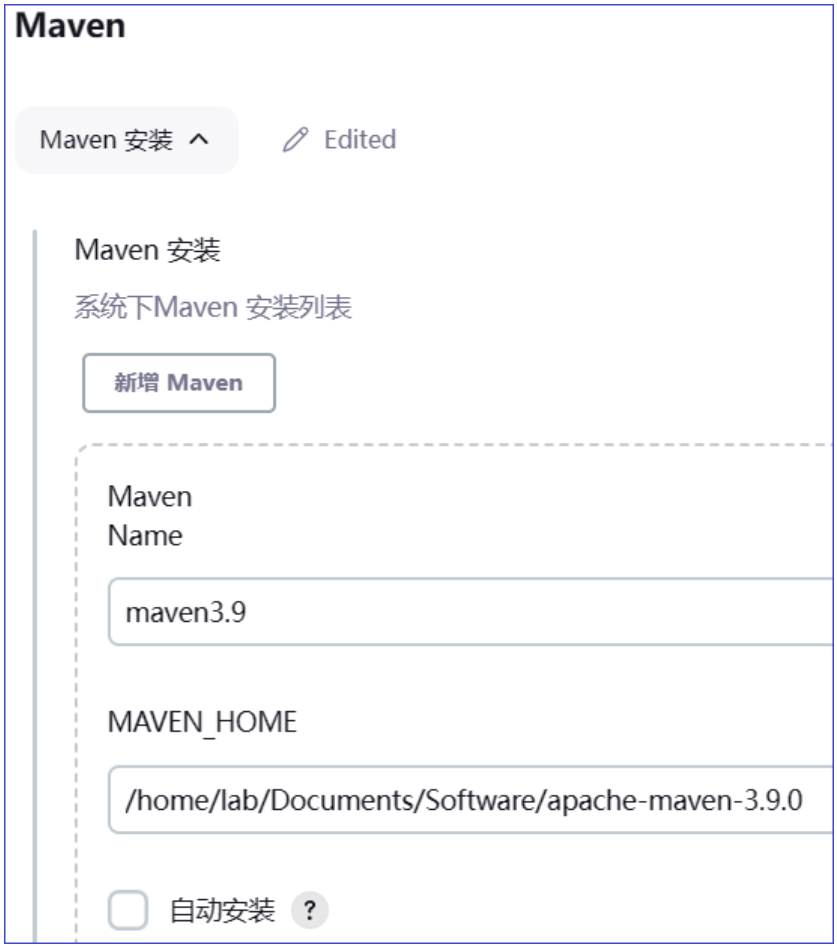
Choose how a new build is triggered.



You need to configure the location of your Maven (the Maven software, not the project). Then add "clean package" to "Goals and options", this is what we want Jenkins to do for now:

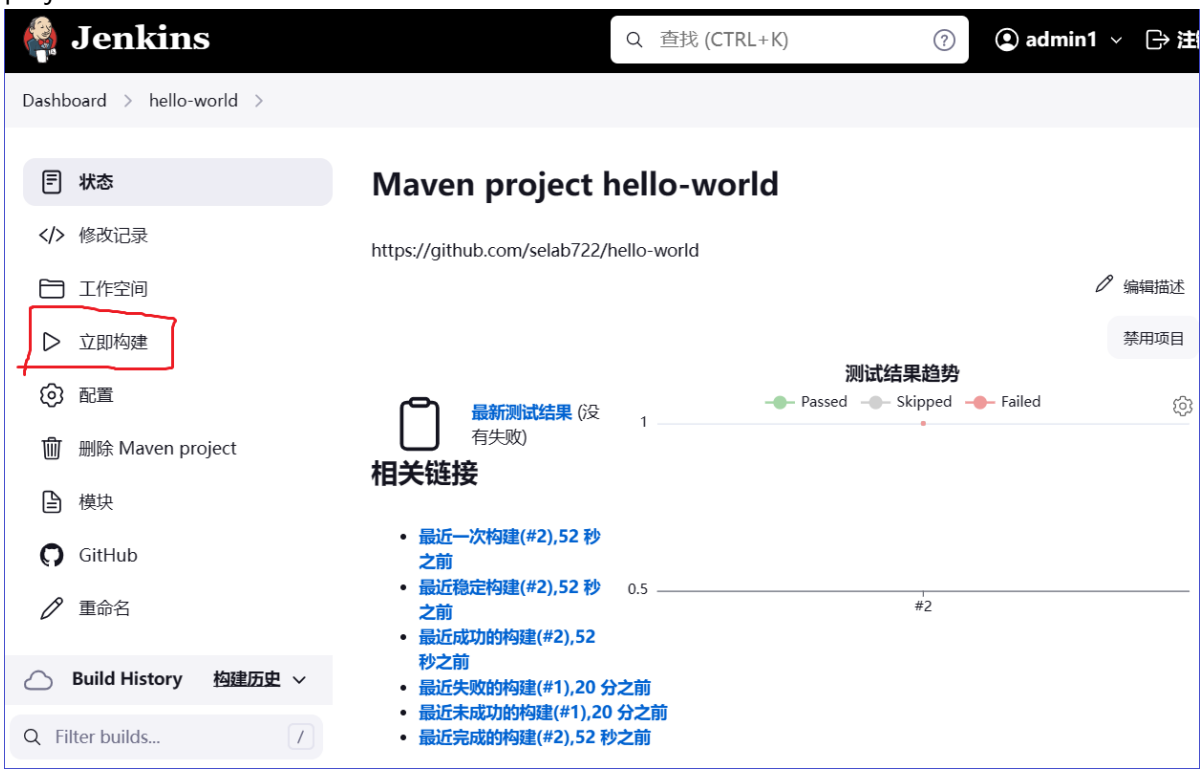


Click "the tool configuration" in the above graph, and configure your maven location:

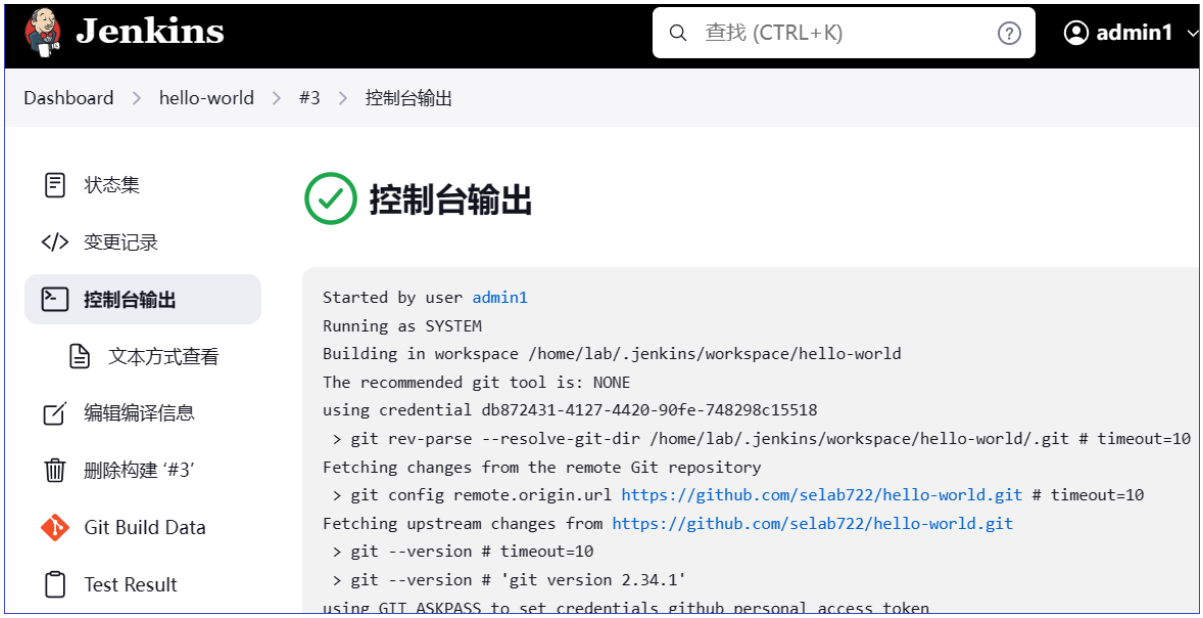


This location should contains "bin/mvn" file that runs Maven. You could run "which mvn" to have a clue where it is.

After you finished all the config, click "Save". Now you can click "Build Instantly" to build this project.



After the build is suscessful or failed we can see the result together with the output.



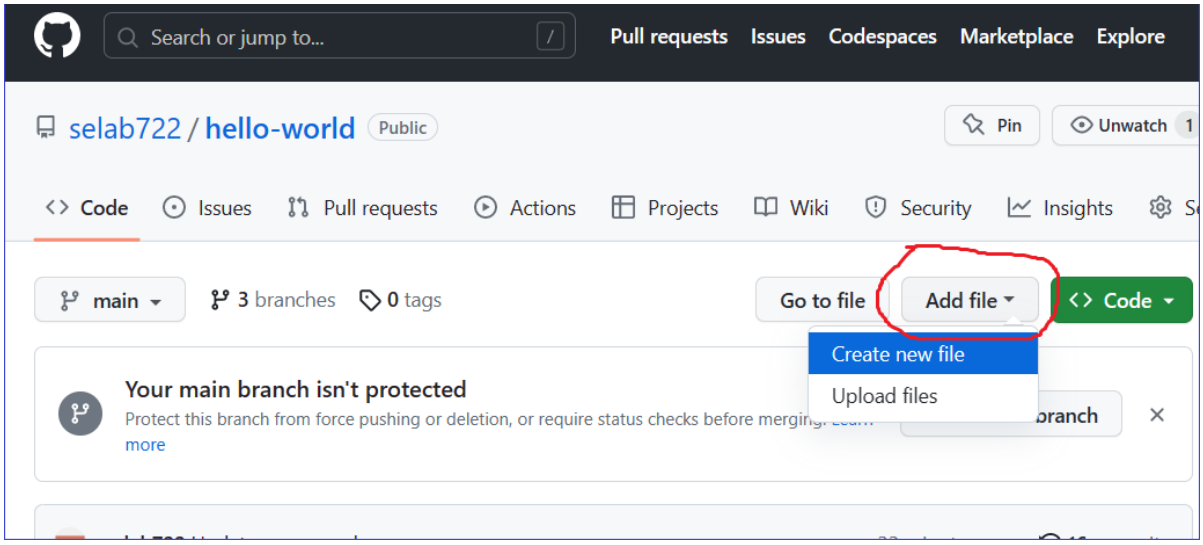
By adding a "Jenkinsfile" file you can ask Jenkins to do more, such as using a docker. Refer to the Jenkins document for a more detailed description: https://www.jenkins.io/doc/pipeline/tour/hello-world/.

Github Actions

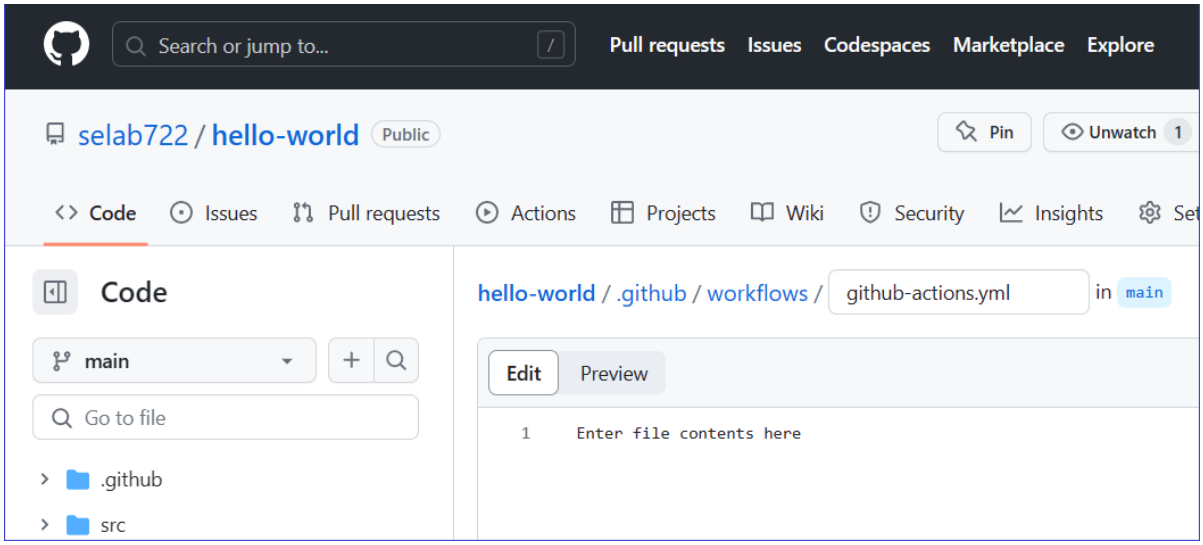
Github also has its own CI/CD tools. It is called "Github Actions".

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:

```yml
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 package --file pom.xml
```

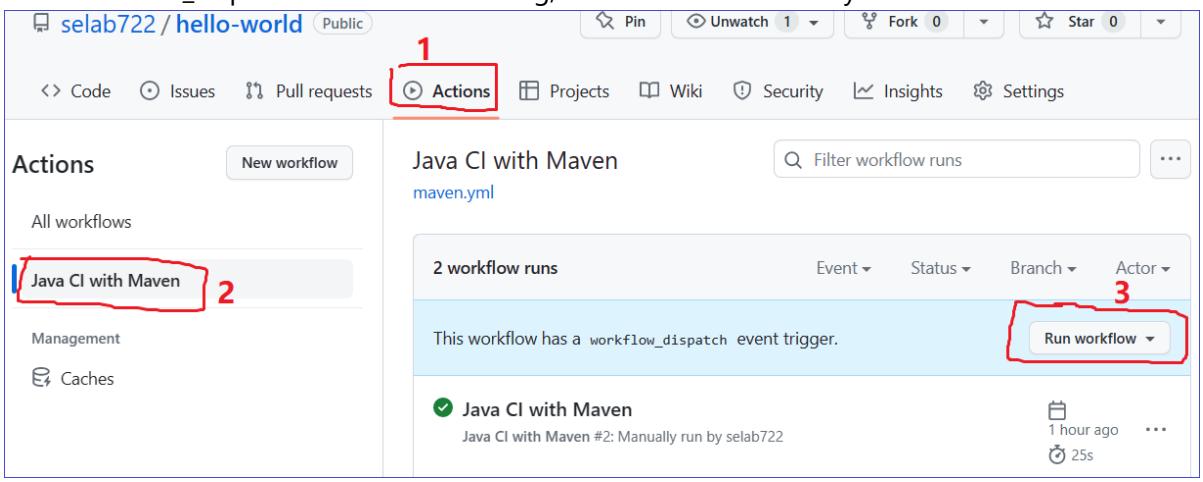
After that, commit this file to a new branch and start a pull request. This will trigger this workflow and the Github Actions will run it.

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:

```yaml
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 package --file pom.xml
```

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



Click and run you will see the result.