

Lab : Jenkins
Introduction to Software Engineering
Usa Sammapun

Jenkins เป็นเครื่องมือช่วยในการ build application, test, deploy และรัน application

Installing Jenkins

1. อ่านการติดตั้ง Jenkins ตามลิงก์นี้ <https://www.jenkins.io/doc/book/installing/#war-file>

- ดาวน์โหลดไฟล์ jenkins.war
- รันคำสั่ง

```
$ java -jar jenkins.war --httpPort=9090
```
- ต้องใช้ Java 11 or 17 เท่านั้น
 - ดังนั้น ถ้าใช้เวอร์ชันอื่นอยู่ ให้ดาวน์โหลดเวอร์ชัน 11/17 มา แล้วรันจากเวอร์ชัน 11/17 นี้
 - เช่น ถ้าใช้ Mac ให้ดูในโฟลเดอร์ /Library/Java/JavaVirtualMachines/ ว่ามีจาวาเวอร์ชันใดบ้าง หลังจากนั้น ให้เรียกใช้คำสั่ง java เช่น

```
/Library/Java/JavaVirtualMachines/jdk-11.0.13.jdk/Contents/Home/bin/java -jar jenkins.war --httpPort=9090
```

```
"C:\Program Files\Java\jdk11.0.13\bin\java.exe" -jar jenkins.war --httpPort=9090
```

2. Setup Jenkins ตามนี้ (Ref: <https://www.jenkins.io/doc/book/installing/war-file/>)

2.1 Go to <http://localhost:9090/> you will see the following page

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

2.2. Get the password from CMD/Terminal windows. Then copy and paste passwords to the web.

```
~/Downloads — java -jar jenkins.war
rk.beans.factory.support.DefaultListableBeanFactory@1d4bd203: defining beans [filter,legacy]; root of factory hierarchy
2020-10-15 11:22:46.649+0000 [id=31] INFO jenkins.install.SetupWizard#init:
*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

4a8ad2f6208748619026a88d6ae8fd0e

This may also be found at: /Users/usa/.jenkins/secrets/initialAdminPassword

*****
*****
*****

2020-10-15 11:22:55.844+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAt
tained: Completed initialization
2020-10-15 11:22:55.859+0000 [id=21] INFO hudson.WebAppMain$3#run: Jenkins
```

3. Choose “Install suggested plugin”



Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

4. You will see it installing, which will take a while

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	⌚ Build Timeout	⌚ Credentials Binding	** Trilead API ** JAXB Folders
⌚ Timestampers	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	** Oracle Java SE Development Kit Installer ** Script Security ** Command Agent Launcher OWASP Markup Formatter
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View	** Structs
⌚ Git	⌚ Subversion	⌚ SSH Slaves	⌚ Matrix Authorization Strategy	** Pipeline: Step API ** Token Macro ** bouncycastle API Build Timeout
⌚ PAM Authentication	⌚ LDAP	⌚ Email Extension	⌚ Mailer	

** - required dependency

Jenkins 2.190.1

5. Create an admin user with any username/password. (just don't forget it !)¹

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.190.1

[Continue as admin](#) [Save and Continue](#)

¹ ถ้าเคยใช้ jenkins มาแล้ว และจำ admin password ไม่ให้ ให้ลบไฟล์ \$HOME/.jenkins แล้ว setup ใหม่

6. Just click “Save and Finish”

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible.

The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.249.2

Not now

Save and Finish

7. We are ready!

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

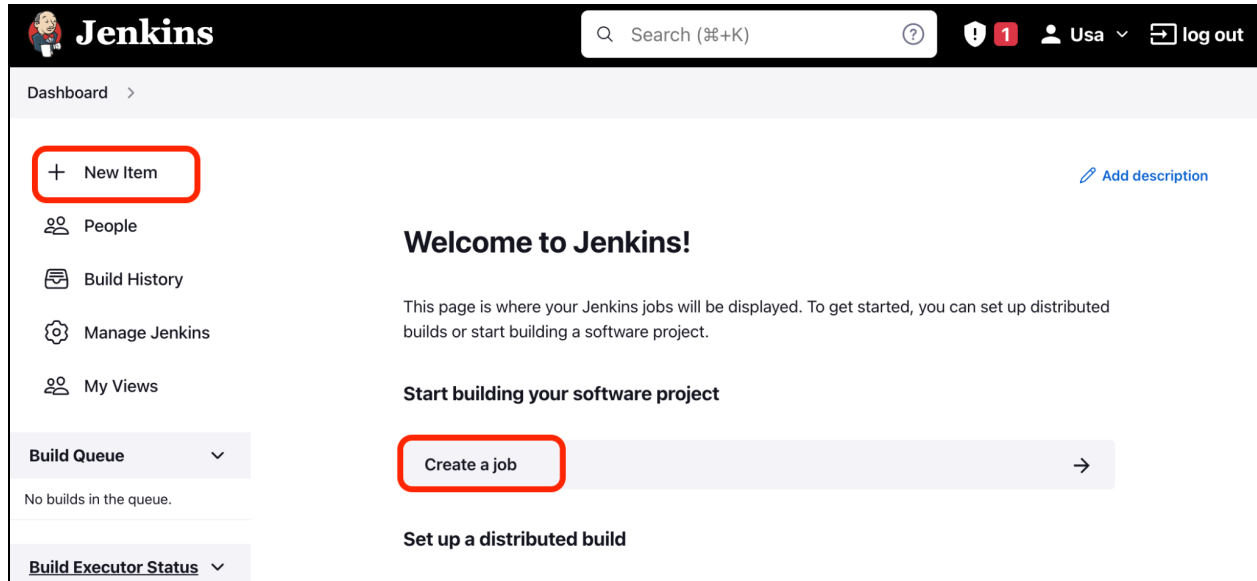
[Start using Jenkins](#)

Jenkins 2.190.1

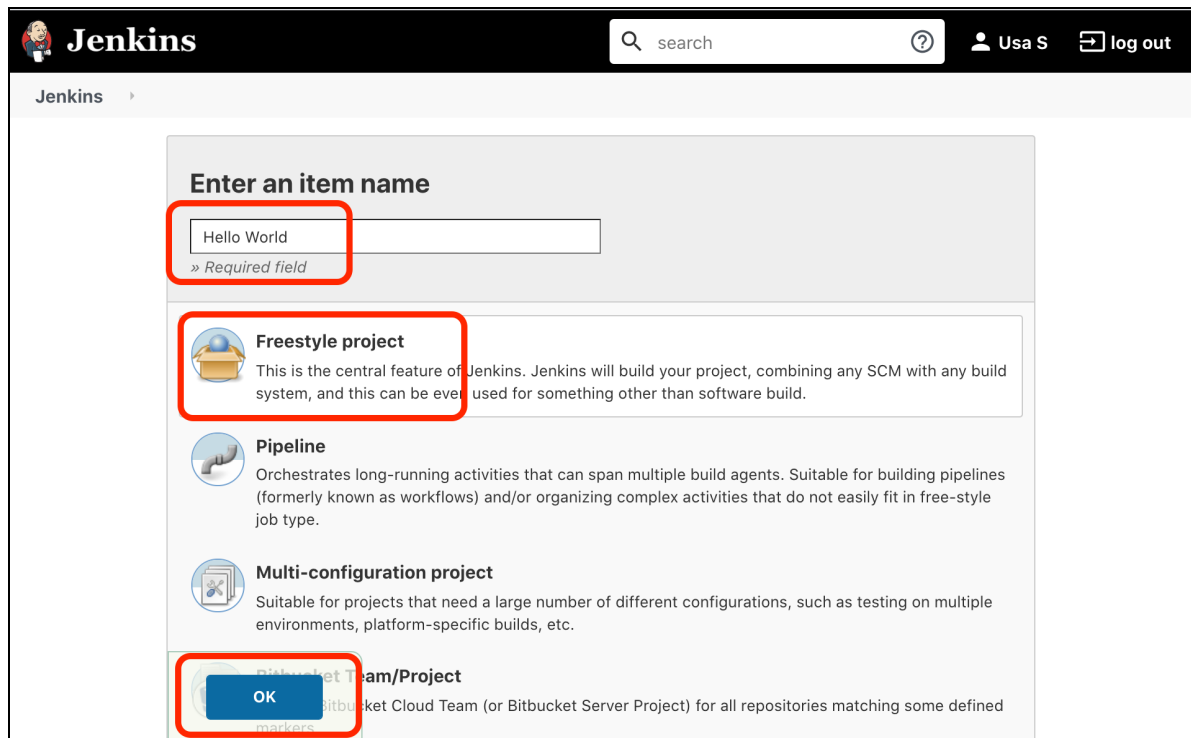
Using Jenkins

I. Hello World on Jenkins

1. Open web browser at: <http://localhost:9090/>
2. Login with admin user you've just created
3. We will create a job that will test, integrate and deploy for us. Click Create a job / New Item



4. Enter an item name, click Freestyle project, click OK



The image shows the Jenkins 'Enter an item name' dialog. At the top, there's a search bar and user information. The main section is titled 'Enter an item name'. Below this, there's a text input field containing 'Hello World', which is highlighted with a red box. Below the input field is a link that says '> Required field'. Underneath, there are three project type options, each with an icon and a description. The 'Freestyle project' option is highlighted with a red box. Below these options is an 'OK' button, also highlighted with a red box.

Enter an item name

Hello World
> Required field

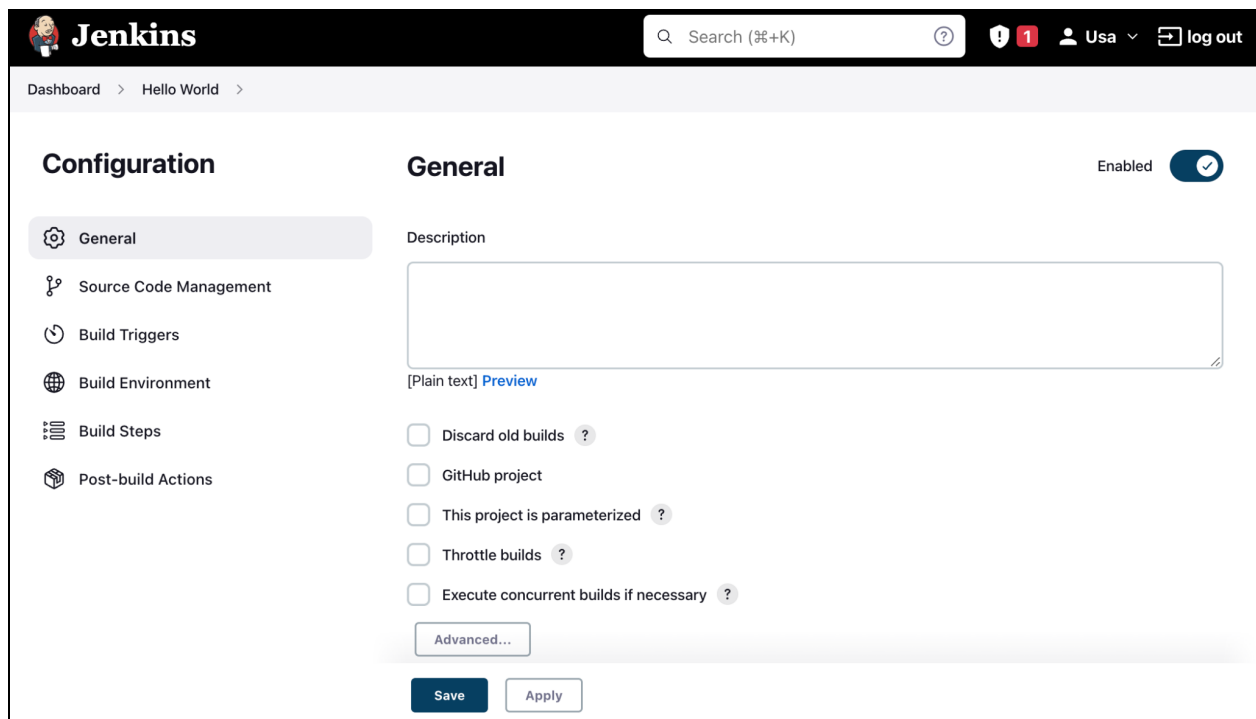
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

5. You will see this page



The image shows the Jenkins Configuration page for a new item named 'Hello World'. The page has a dark header with the Jenkins logo, a search bar, and user information. Below the header, there's a breadcrumb trail: 'Dashboard > Hello World >'. The main content area is titled 'Configuration' and 'General'. On the left, there's a sidebar with various configuration options: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'General' option is selected. The main area shows the 'General' configuration. It includes a 'Description' field, which is currently empty. Below the description field, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. Each checkbox has a help icon. At the bottom, there are 'Save' and 'Apply' buttons. The 'Enabled' toggle switch is turned on.

Jenkins

Search (%+K)

Dashboard > Hello World >

Configuration

General

Enabled

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

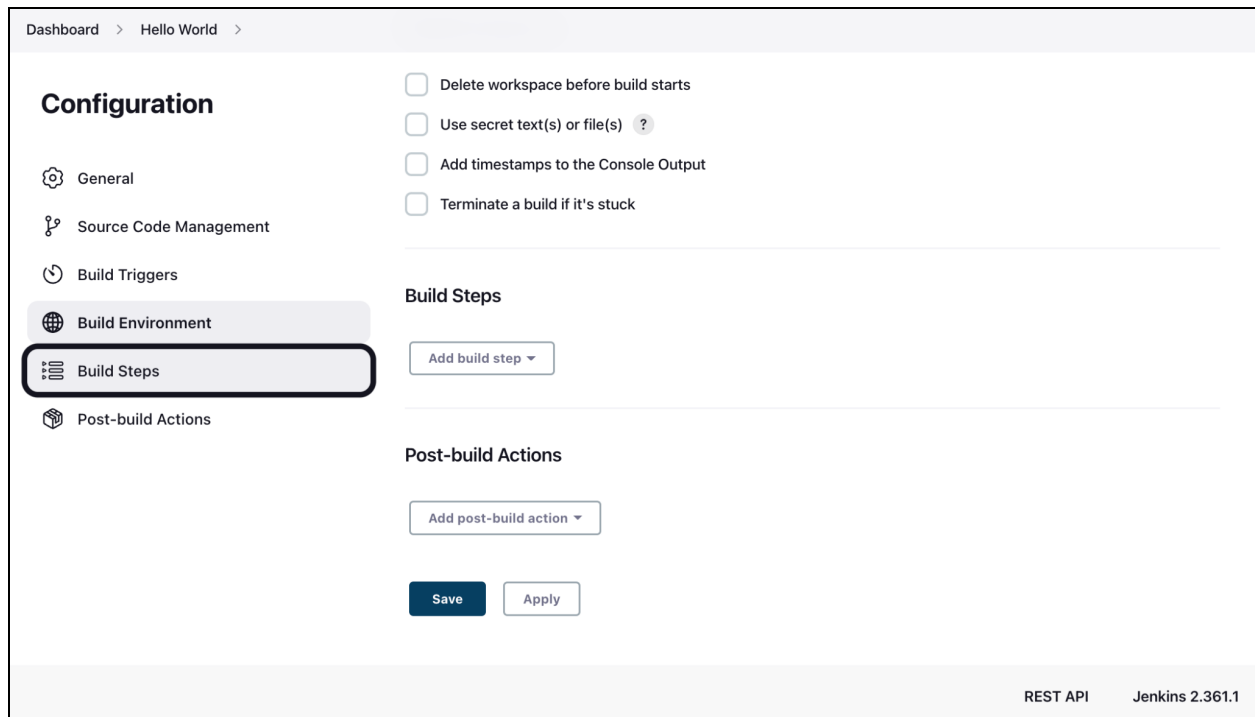
☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

[Advanced...](#)

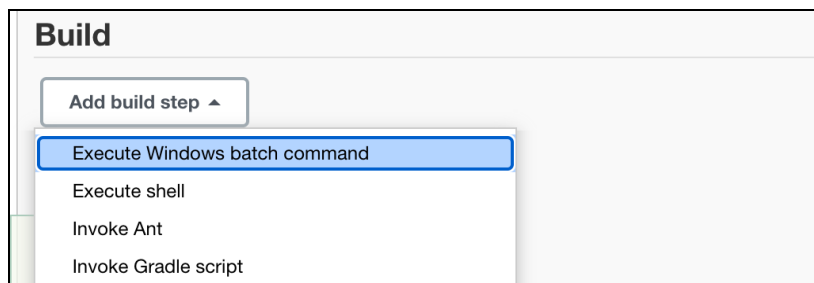
Save **Apply**

6. คลิกที่เมนูซ้ายมือ Build Steps

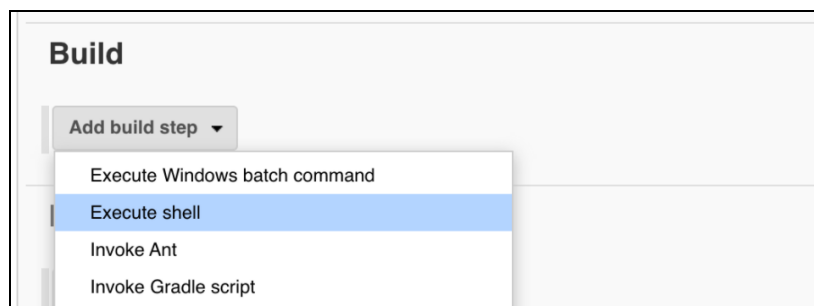


7. Click “Add build step”

- **Windows**: choose “Execute Windows batch command”

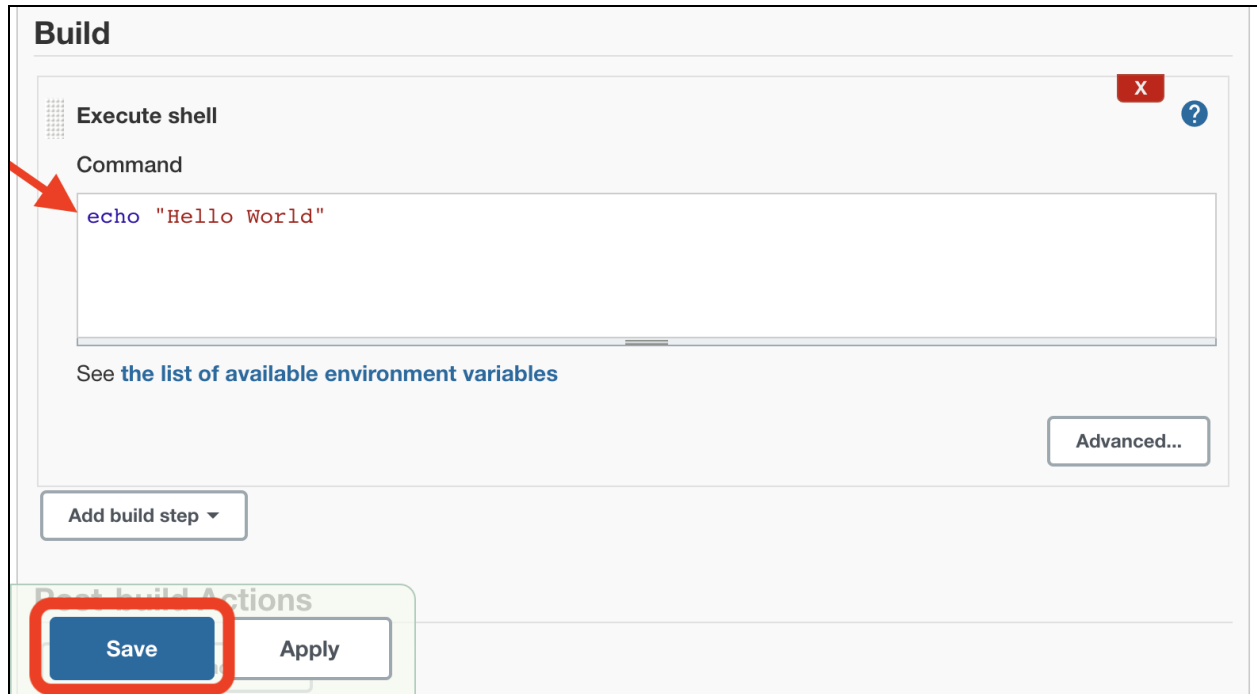


- **MacOS and Linux**: choose “Execute shell”

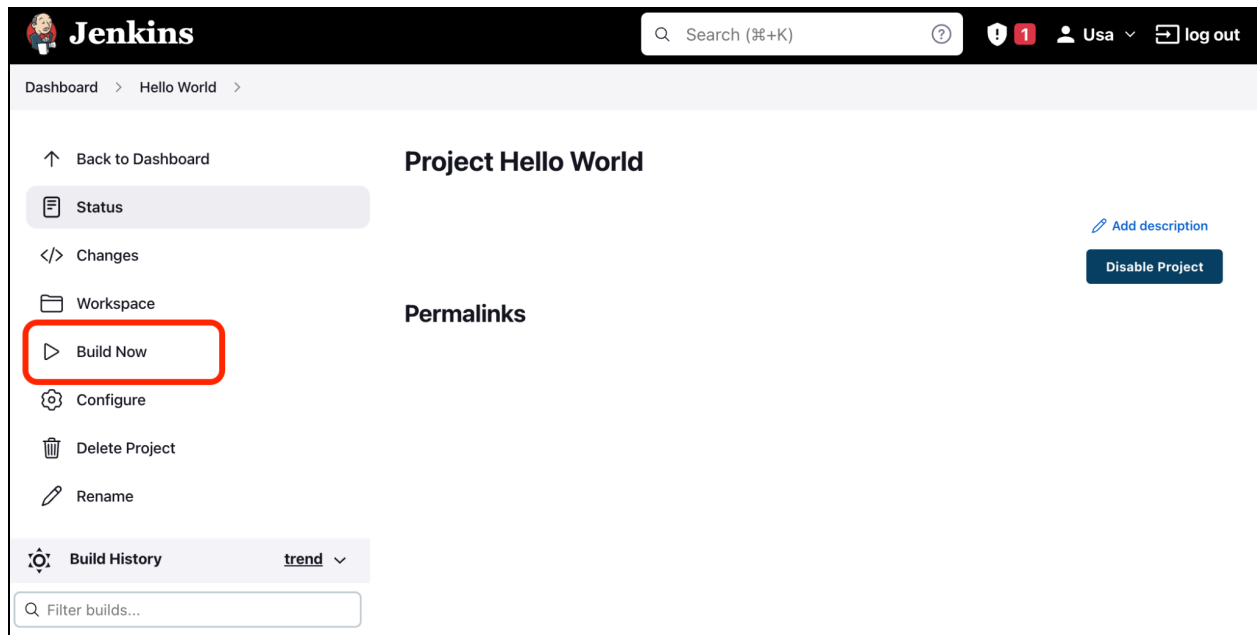


8. Type in the following command, which will print out “Hello World” to the console. Then, click “Save”

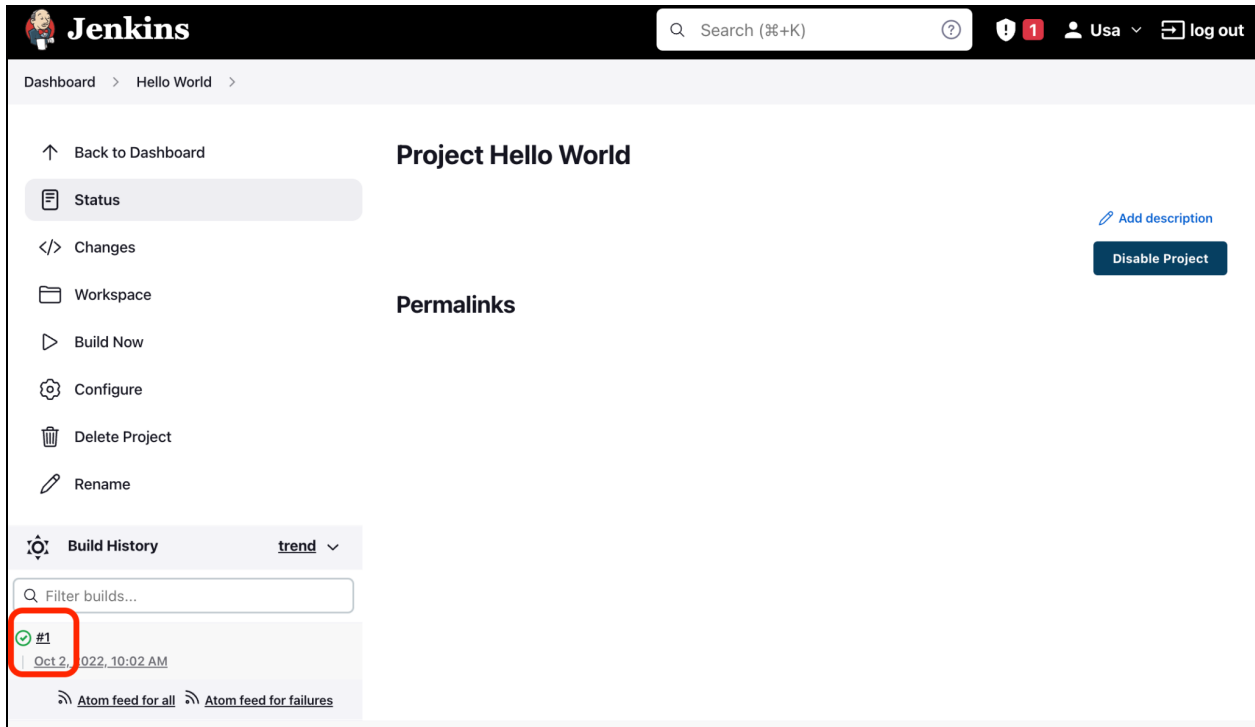
```
echo "Hello World"
```



9. Click “Build Now” to tell Jenkins to execute Windows batch command / Shell command



10. Click at Build History #1 to see the run result



The screenshot shows the Jenkins web interface for a project named "Hello World". The top navigation bar includes the Jenkins logo, a search bar, a help icon, a notification badge with the number "1", the user "Usa", and a "log out" button. The breadcrumb trail shows "Dashboard > Hello World >".

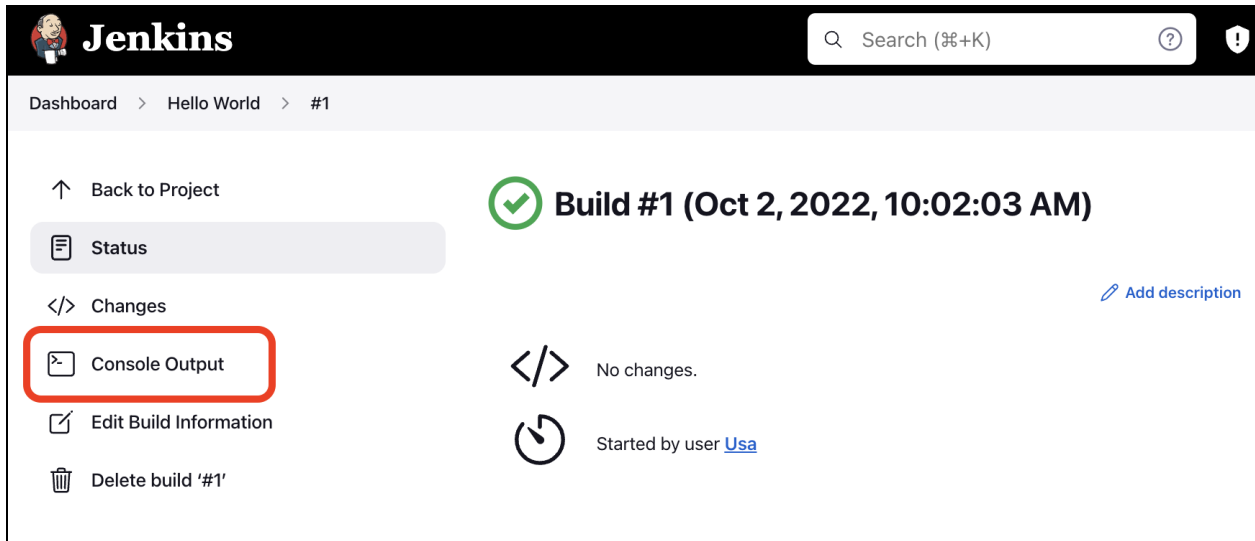
On the left sidebar, there are several menu items: "Back to Dashboard", "Status" (highlighted), "Changes", "Workspace", "Build Now", "Configure", "Delete Project", and "Rename".

The main content area is titled "Project Hello World" and contains a "Permalinks" section. On the right side of this section, there are two buttons: "Add description" and "Disable Project".

Below the "Status" menu item, there is a "Build History" section. It features a "trend" dropdown menu and a search bar labeled "Filter builds...". A table of build history is displayed below the search bar. The first build, labeled "#1", is highlighted with a red box. It shows a green checkmark icon, indicating a successful build, and the timestamp "Oct 2, 2022, 10:02 AM". At the bottom of the build history section, there are two links: "Atom feed for all" and "Atom feed for failures".

Build	Status	Timestamp
#1	Success	Oct 2, 2022, 10:02 AM

11. Click “Console Output” to see if we get “Hello World”



The screenshot shows the Jenkins interface for Build #1. The left sidebar contains a list of actions: 'Back to Project', 'Status', 'Changes', 'Console Output' (highlighted with a red rectangle), 'Edit Build Information', and 'Delete build '#1''. The main area displays 'Build #1 (Oct 2, 2022, 10:02:03 AM)' with a green checkmark icon. Below this, it shows 'No changes.' and 'Started by user Usa'. A blue link 'Add description' is visible on the right.

Dashboard > Hello World > #1

↑ Back to Project

📄 Status

</> Changes

🖨️ Console Output

📝 Edit Build Information

🗑️ Delete build '#1'

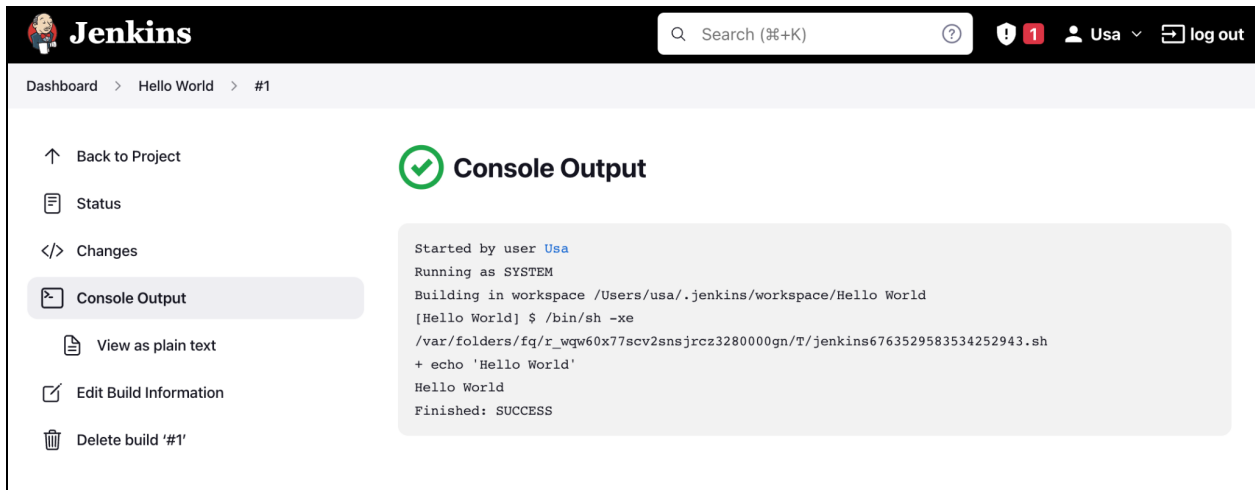
✅ Build #1 (Oct 2, 2022, 10:02:03 AM)

🔗 Add description

</> No changes.

🕒 Started by user [Usa](#)

12. We see our “Hello World”



The screenshot shows the Jenkins interface for Build #1, specifically the 'Console Output' view. The left sidebar is the same as in the previous screenshot, but 'Console Output' is now selected. The main area displays the console output text, which includes the build's execution details and the successful output of 'Hello World'.

Dashboard > Hello World > #1

↑ Back to Project

📄 Status

</> Changes

🖨️ Console Output

📄 View as plain text

📝 Edit Build Information

🗑️ Delete build '#1'

✅ Console Output

```
Started by user Usa
Running as SYSTEM
Building in workspace /Users/usa/.jenkins/workspace/Hello World
[Hello World] $ /bin/sh -xe
/var/folders/fq/r_wqw60x77scv2snsjrcz3280000gn/T/jenkins6763529583534252943.sh
+ echo 'Hello World'
Hello World
Finished: SUCCESS
```

II. Pipeline

Pipeline helps you divide a job into stages: build, test, deploy, etc... Pipeline is defined in a script called “Jenkinsfile”.

เราจะใช้โปรแกรมร้านหนังสือ Wisdom Book โดยนิสิตสามารถใช้ repo ของอาจารย์ได้เลย หรือ fork repo ให้อยู่ในบัญชีของนิสิต (ถ้าเคย fork มาแล้ว สามารถใช้ของนิสิตได้ แต่ต้องตรวจสอบชื่อไฟล์ jar และ port ให้ดี)


<https://github.com/ladyusa/wisdom-book>

- เครื่องนิสิตต้องติดตั้งโปรแกรม mvn จากลิงก์นี้ <https://maven.apache.org/> เสียก่อน
- สำหรับ Windows ต้อง set JAVA_HOME environment variable ตามลิงก์นี้ <https://java2blog.com/how-to-set-java-path-windows-10/>


1. Create a new job item. Choose “Pipeline”. (Pipeline is a build script.)

Enter an item name


» Required field

**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

2. คลิกที่เมนู Pipeline

The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, and user information. Below the header is a breadcrumb trail: Dashboard > Wisdom Book >. The main content area is titled 'Configuration' and 'General'. On the left, there is a sidebar with three menu items: 'General' (with a gear icon), 'Advanced Project Options' (with a wrench icon), and 'Pipeline' (with a pipeline icon). The 'Pipeline' menu item is highlighted with a red rectangular border. To the right of the sidebar, the 'General' tab is active. It features a 'Description' text area, a '[Plain text] Preview' link, and a list of checkboxes: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', and 'Pipeline uses Maven build system'. At the bottom of the configuration area are 'Save' and 'Apply' buttons. The 'Enabled' toggle switch is turned on.

Jenkins

Search (⌘+K)

Dashboard > Wisdom Book >

Configuration

General

Enabled

General

Advanced Project Options

Pipeline

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☐ GitHub project

☐ Pipeline uses Maven build system ?

Save Apply

3. เราจะสั่งให้ Jenkins ช่วย build test และรันโปรแกรม wisdom book ผ่าน script ที่เรียกว่า “Jenkinsfile”

ตัวอย่างด้านล่าง คือ “Jenkinsfile” สำหรับระบุขั้นตอนต่าง ๆ ในการ build / test

- มีการแบ่งเป็น 4 ขั้นตอน ทำให้เราแยกได้ง่ายว่า มีขั้นตอนใดผ่าน/ไม่ผ่านบ้าง
- **Windows:** ใช้ bat ใน steps (ถ้า fork repo ให้ปรับ **USERNAME** ให้เป็นของนิสิต)

```
pipeline {
  agent any
  stages {
    stage('Source') {
      steps {
        git branch: 'main',
            url: 'https://github.com/ladyusa/wisdom-book.git'
      }
    }
    stage('Build') {
      steps {
        bat 'mvn package -DskipTests'
      }
    }
    stage('Test') {
      steps {
        echo 'testing...'
        //bat 'mvn test'
      }
    }
    stage('Deploy') {
      steps {
        bat 'java -jar ./target/book-1.0.jar'
      }
    }
  }
}
```

- **MacOS and Linux:** ใช้ sh ใน steps (ถ้า fork repo ให้ปรับ **USERNAME** ให้เป็นของนิสิต)

```
pipeline {
  agent any
  stages {
    stage('Source') {
      steps {
        git branch: 'main',
            url: 'https://github.com/ladyusa/wisdom-book.git'
      }
    }
    stage('Build') {
      steps {
        sh 'mvn package -DskipTests'
      }
    }
    stage('Test') {
      steps {
        echo 'testing...'
        //sh 'mvn test'
      }
    }
    stage('Deploy') {
      steps {
        sh 'java -jar ./target/book-1.0.jar'
      }
    }
  }
}
```

4. ให้ copy script จาก Jenkinsfile ด้านบน และวางใน pipeline script form (ถ้า fork repo ให้ปรับ **USERNAME** ให้เป็นของนิสิต) คลิก “Save”

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3   stages {
4     stage('Source') {
5       steps {
6         git branch: 'main',
7           url: 'https://github.com/ladyusa/wisdom-book.git'
8       }
9     }
10    stage('Build') {
11      steps {
12        sh 'mvn package -DskipTests'
13      }
14    }
15    stage('Test') {
16      steps {
```

☒ Use Groovy Sandbox ?

Save

Apply

5. Click the menu “Build Now”. You will see the status of each stage.

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename


Pipeline Syntax

Build Historytrend ^

find x

#1 Oct 6, 2021 12:52 PM

Pipeline Wisdom Book

 Recent Changes

Stage View

Average stage times:

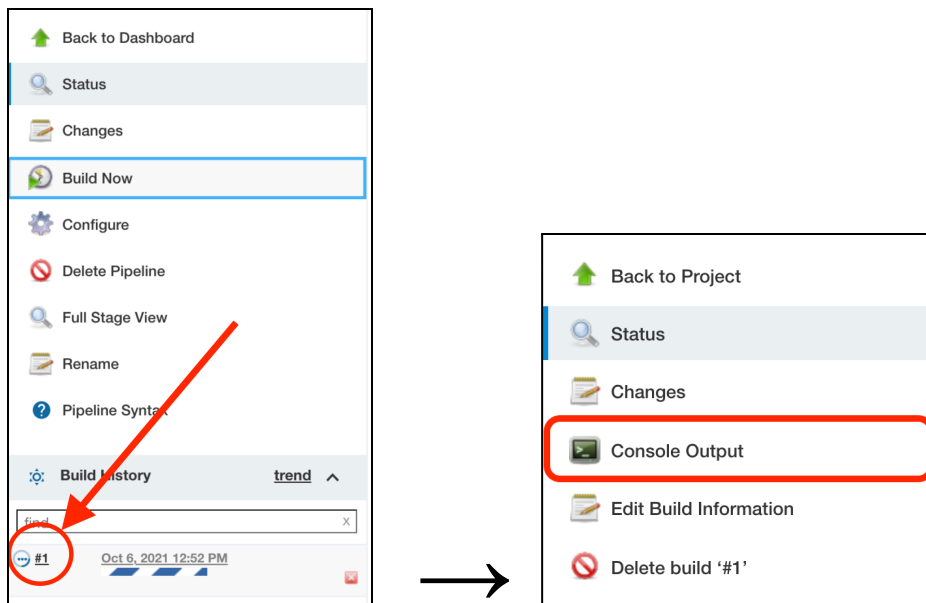
Source	Build	Test	Deploy
1s	5s	77ms	19s

#1 Oct 06 12:52 No Changes

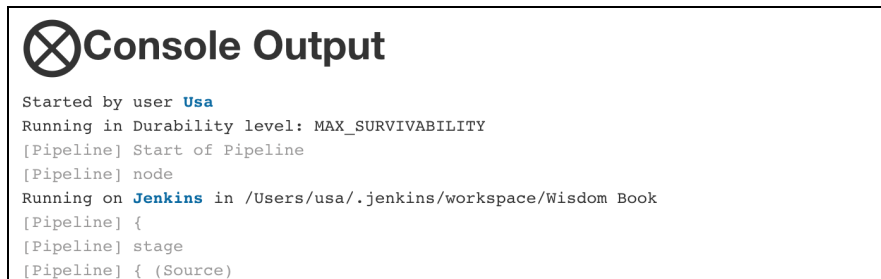
1s	5s	77ms	
----	----	------	--

Permalinks

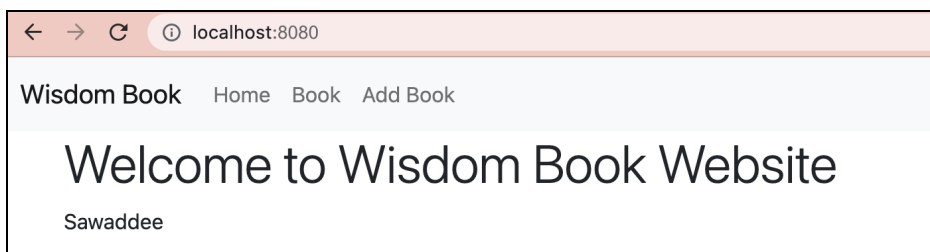
4. Click menu Build History “#..” → Console Output



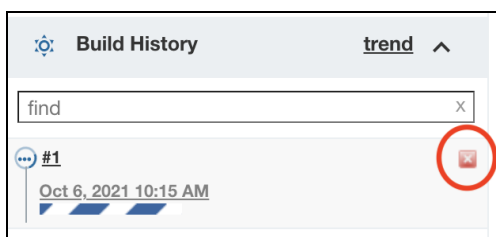
- You will see your server is running.



5. Goto <http://localhost:8090/> You will see a wisdom book website.



6. Stop the build



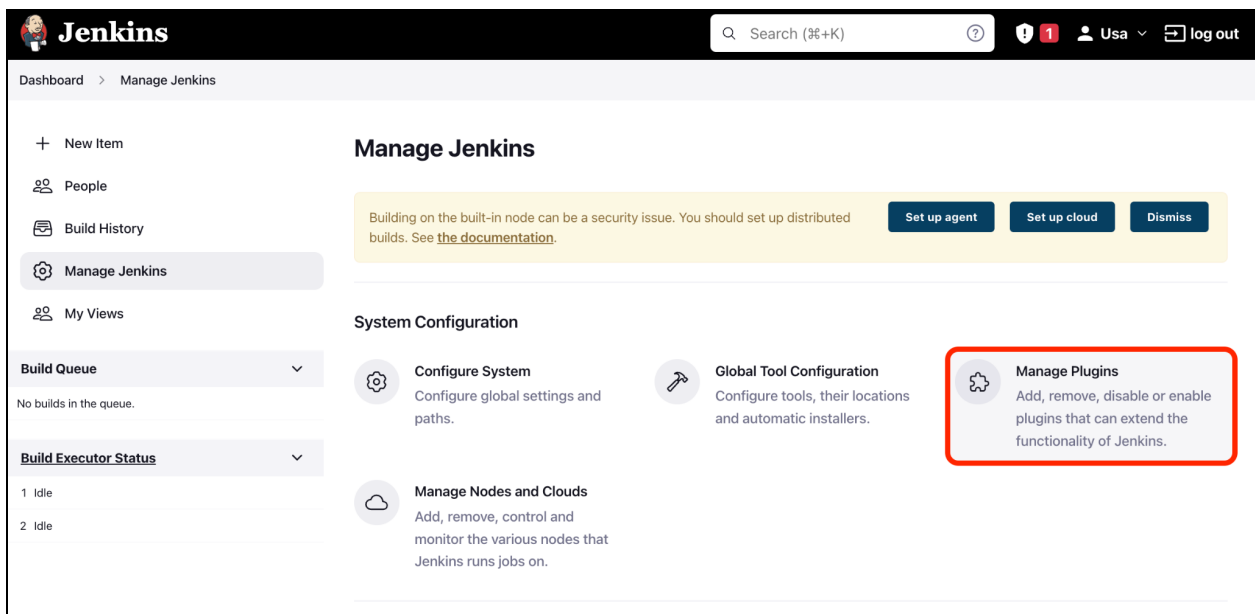
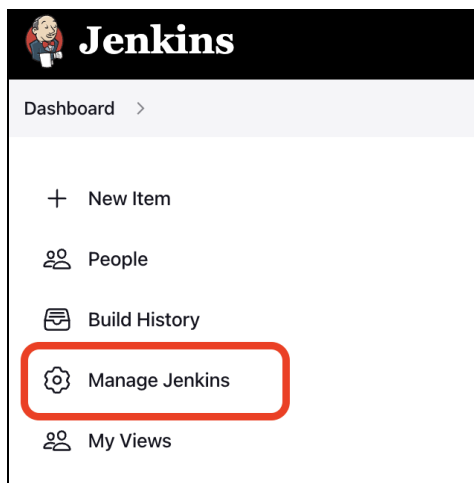
II. Pipeline and Docker

ในหัวข้อที่แล้ว เราเรียนรู้ Wisdom Book บนเครื่องที่รัน Jenkins ไว้ (คือ เครื่อง agent master) ซึ่งไม่ควรทำปกติเราควรใช้เครื่อง slave ซึ่งต้องเป็น server อันหรือบน cloud ในการรันโปรแกรมที่เราต้องการ deploy ในแลปนี้ เราจะใช้ docker container แทนเครื่อง slave

- นอกจากนั้น หากเราใช้เครื่องที่รัน Jenkins เราจะต้อง install mvn ด้วย ถ้าเราใช้ docker เราสามารถใช้ maven image ในการรันแทนเครื่องเราได้
- ดังนั้น ให้ติดตั้ง Docker บนเครื่องของเราเตรียมเอาไว้

1. ไปที่หน้า Dashboard ---> แล้วไปที่เมนู Manage Jenkins ---> Manage Plugin

- เพื่อลง Plugin Docker ต่าง ๆ



2. ไปที่ tab Available และพิมพ์ docker เพื่อค้นหา Docker plugin เลือก Docker Commons, Docker Pipeline, Docker, docker-build-step และกดปุ่ม Install without Restart

Search:

Updates Available Installed Advanced

Install ↑	Name	Version	Released
<input checked="" type="checkbox"/>	Docker Cloud Providers Cluster Management and Distributed Build docker This plugin integrates Jenkins with Docker	1.2.3	1 mo 17 days ago
<input checked="" type="checkbox"/>	Docker Commons api-plugin docker Library plugins (for use by other plugins) Provides the common shared functionality for various Docker-related plugins.	1.17	1 yr 3 mo ago
<input checked="" type="checkbox"/>	Docker Pipeline Deployment DevOps docker pipeline Build and use Docker containers from pipelines.	1.26	7 mo 13 days ago
<input type="checkbox"/>	Docker API api-plugin docker This plugin provides docker-java API for other plugins. <div>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</div>	3.1.5.2	1 yr 6 mo ago
<input checked="" type="checkbox"/>	docker-build-step Build Tools docker This plugin allows to add various docker commands to your job as build steps.	2.8	3 mo 9 days ago
	CloudBees Docker Build and Publish		

Update information obtained: 1 hr 7 min ago

—> อาจจะต้อง restart Jenkins เพื่อให้สามารถใช้ agent docker ได้ โดย

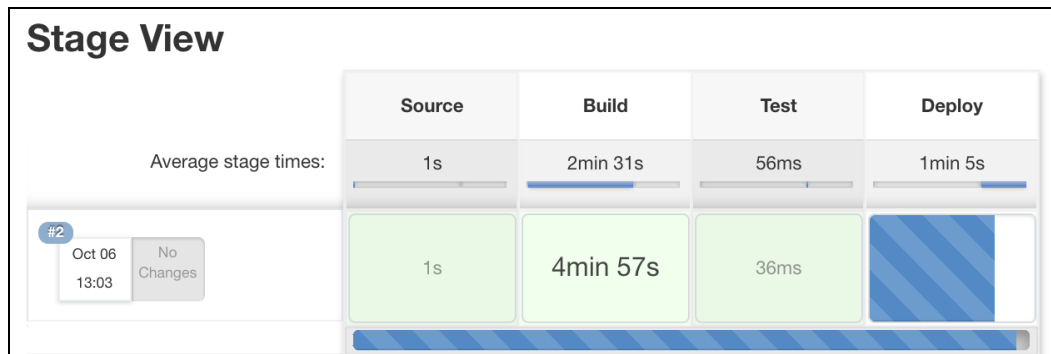
- กด Ctrl-C (ทั้งใน Windows และ MacOS) ที่ command line
- และรันคำสั่ง `java -jar jenkins.war --httpPort=9090` อีกครั้ง

3. ปรับแก้ Jenkinsfile script โดยเปลี่ยนให้ agent เป็น docker

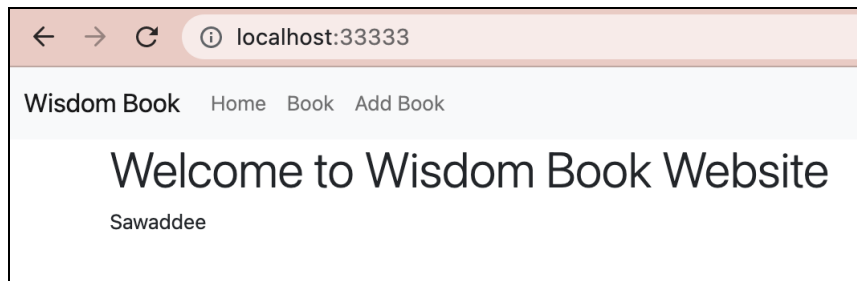
- จะเห็นว่า เราใช้ image ของ maven เวอร์ชัน 3-jdk-11 และรัน container ด้วย port 33333
 - นิสิตสามารถเปลี่ยนเวอร์ชันของ maven ให้ใช้ jdk อื่นได้ โดยดูจาก Maven Docker Hub
 - https://hub.docker.com/_/maven
- environment เป็นการระบุว่า HOME directory จะใช้ directory ที่อยู่ปัจจุบัน
- (สำหรับ **MacOS/Linux** ให้เปลี่ยนจาก bat ให้เป็น sh)
- (ถ้า fork repo ให้ปรับ **USERNAME** ให้เป็นของนิสิต)

```
pipeline {
    agent {
        docker {
            image 'maven:3-jdk-11'
            args '-p 33333:8090'
        }
    }
    environment {
        HOME = '.'
    }
    stages {
        stage('Source') {
            steps {
                git branch: 'main',
                    url: 'https://github.com/ladyusa/wisdom-book.git'
            }
        }
        stage('Build') {
            steps {
                bat 'mvn package -DskipTests'
            }
        }
        stage('Test') {
            steps {
                echo 'testing...'
            }
        }
        stage('Deploy') {
            steps {
                bat 'java -jar ./target/book-1.0.jar'
            }
        }
    }
}
```

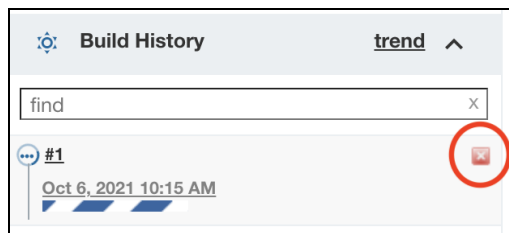
4. จากนั้น ให้ Save และ Build Now อีกครั้ง (จะใช้เวลาโหลดนานคะ อดทนรอนิดนึง)



5. แล้วไปที่หน้า <http://localhost:33333/> จะเห็น server ที่ Jenkins รันให้เรา



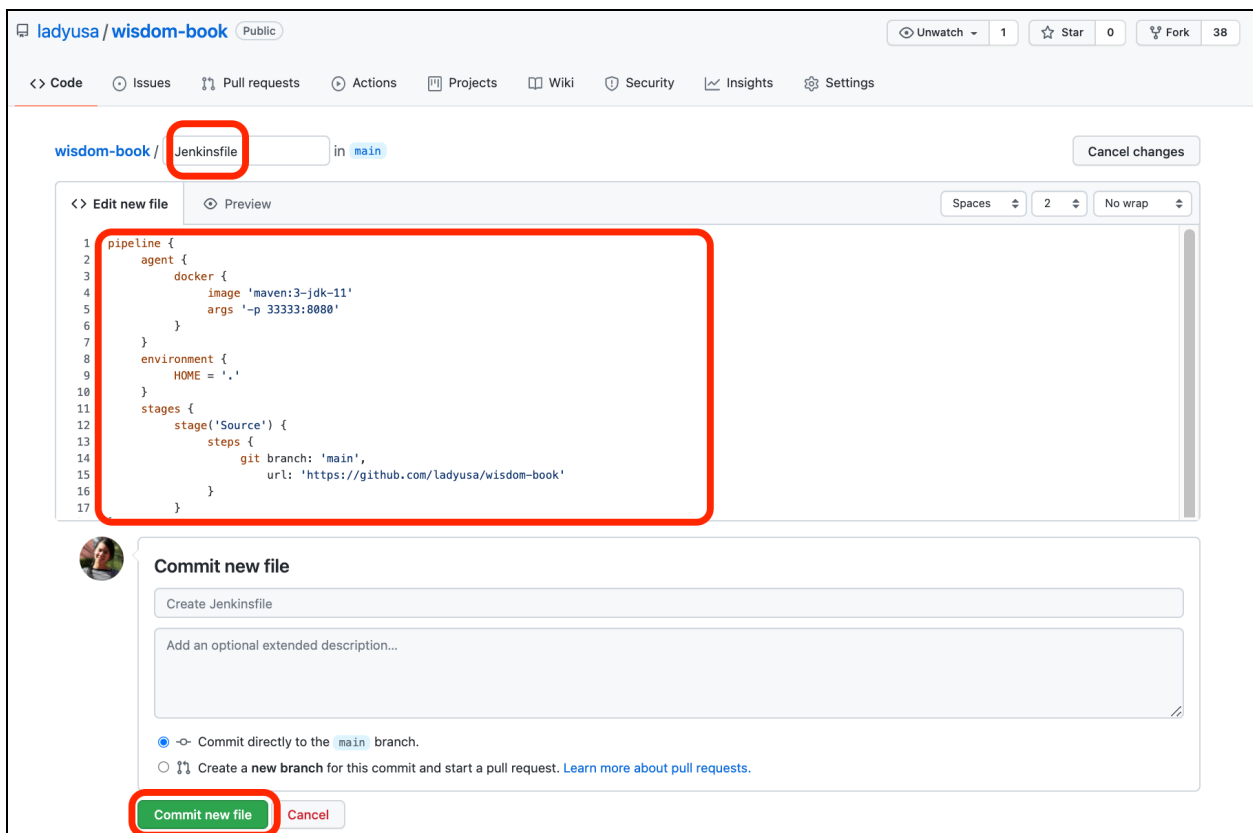
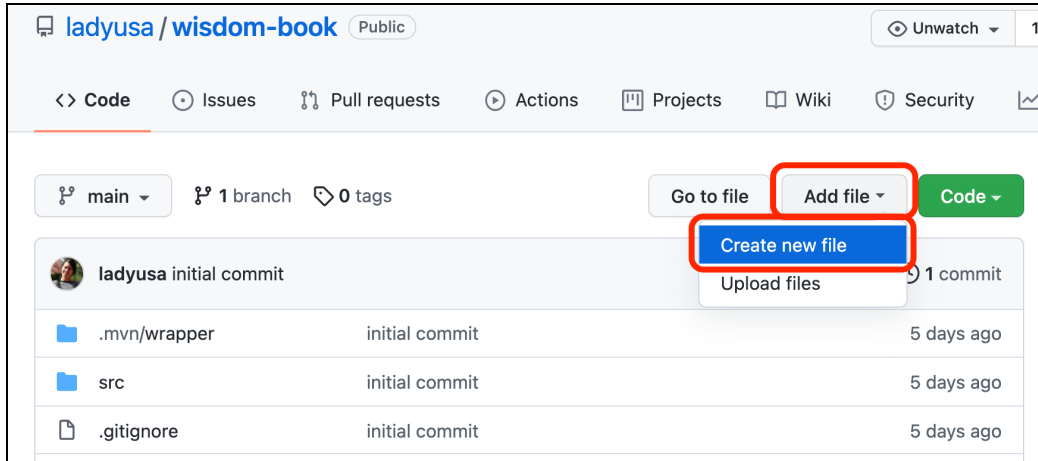
6. Stop the build



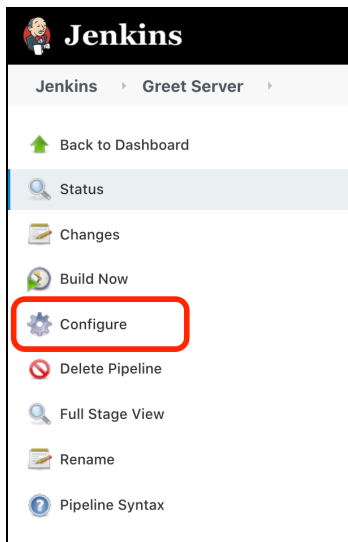
III. Pipeline and Jenkinsfile

Jenkinsfile is a script file that Jenkins can download from git and run it automatically.

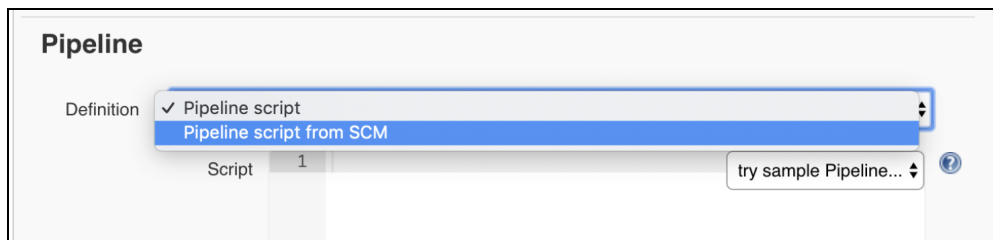
1. โดยให้เพิ่มไฟล์ชื่อ “Jenkinsfile” แล้วใส่ script ในหน้าที่แล้ว จากนั้น commit ขึ้น GitHub ของนิสิต



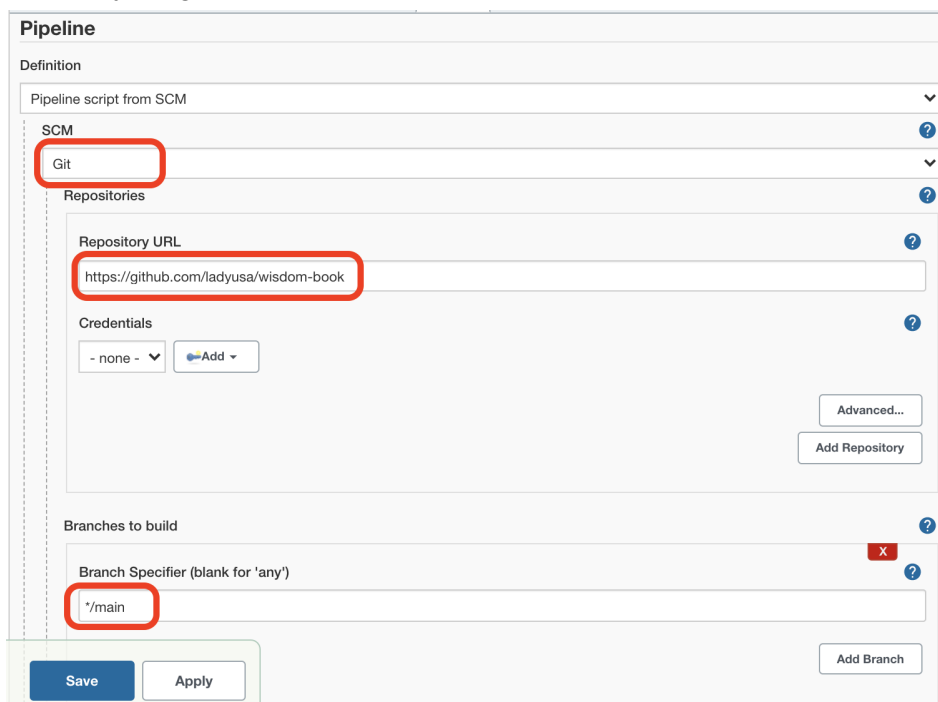
2. Click at menu Configure



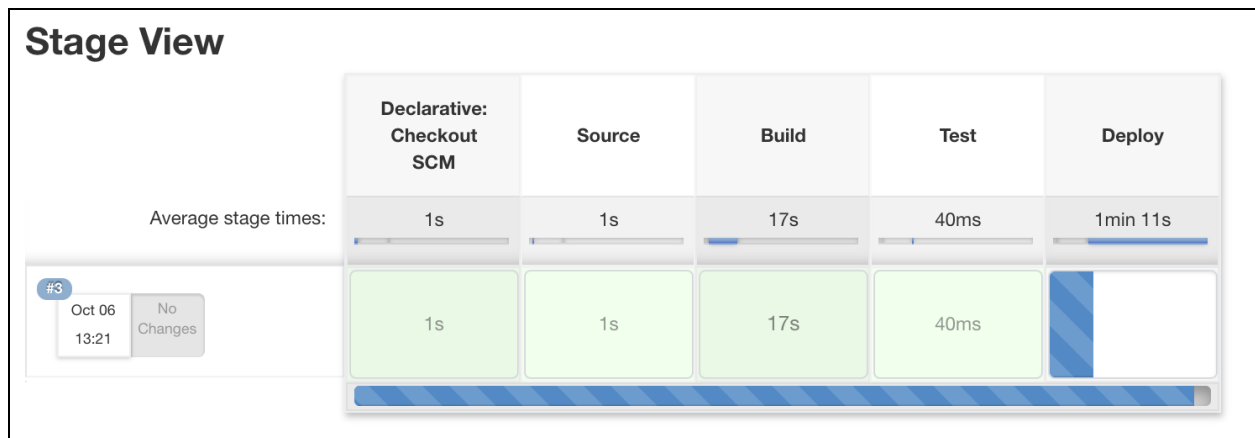
3. Scroll down to “Pipeline”. Choose “Pipeline script from SCM”



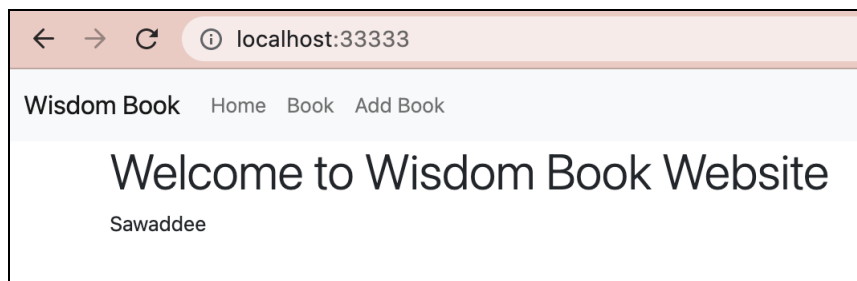
3. Enter your git URL that has Jenkinsfile.



4. Click the menu “Build Now”. You will see an additional stage (SCM)



5. แล้วไปที่หน้า <http://localhost:3333/> จะเห็น server ที่ Jenkins รันให้เรา



6. Stop the build

