TRƯỜNG ĐẠI HỌC HỌC VĂN LANG

**KHOA CÔNG NGHỆ THÔNG TIN**

A red and black shield with white logo

Description automatically generated

**TÀI LIỆU THỰC HÀNH**

**SPRING BOOT**

**CHƯƠNG 10: HOSTING**

Giảng viên biên soạn: Ths. Nguyễn Minh Tân

2024

**BUỔI 10: HOSTING**

1. **MỤC TIÊU**

* Hướng dẫn tạo project Spring Boot
* Hướng dẫn triển khai trên Docker
* Hướng dẫn cài Docker Desktop
* Chạy dự án trên JDK 17 nhé. Hạ version xuống nha các em.

1. **NỘI DUNG THỰC HÀNH:**

Đầu tiên, chúng ta cần chuẩn bị môi trường: Intellij IDEA 2024, Maven , Jdk 17). Tiếp theo, ở đây chúng ta sẽ sử dụng lại dự án `webbanhang` luôn nha.

**Cài đặt Maven**

Down load tại <https://maven.apache.org/download.cgi>

A red circle with blue text

Description automatically generated

Chọn file download tại mục “Files” -> “Binary zip archive“

* Extract file và để vào nơi bạn muốn. Ví dụ “C:\Program Files\apache-maven-3.9.8”.

A screenshot of a computer

Description automatically generated

#### **Setup biến môi trường**

Cũng giống với cách setup biến môi trường cho Java. Đầu tiên chúng ta mở Environment Variables như sau:

* All Control Panel Items > System > Advanced system settings > Environment Variables
* Click vào nút “**New”** trên “**System variables”,**nhập nội dung như sau
  + Variable name: “MAVEN\_HOME”
  + Variable value: Nơi ban extract maven file ở step 1

“C:\Program Files\apache-maven-3.9.8”

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

### **Kiểm tra đã được cài đặt**

Mở “**Run**” -> nhập “**cmd**” để mở command prompt lên, và nhập “**mvn -version**” rồi nhấn “**Enter**”

A screenshot of a computer

Description automatically generated

With a correctly configured Maven file, we can then create an executable jar file:

mvn clean package

Next, we’ll start up the Spring Boot application:

### **3. Hướng dẫn cài Docker Desktop**

# **Cài Docker Desktop**

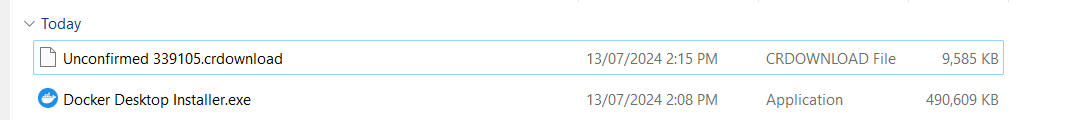
Trên Windows, cài Docker Desktop Installer theo link này:

<https://docs.docker.com/desktop/install/windows-install/>

A screenshot of a computer

Description automatically generated

Sau khi đã tải Docker, hãy chọn Docker Desktop Installer để tiến hành cài đặt.



Tại đây, Docker sẽ hỏi bạn có muốn tạo shortcut trên Desktop và sử dụng WSL 2 thay cho Hyper-V, hãy đánh dấu vào ô chọn để tiếp tục quá trình cài đặt.

A screenshot of a computer

Description automatically generated

Quá trình cài đặt đang diễn ra

A screenshot of a computer

Description automatically generated

Sau khi cài đặt xong, bạn cần khởi động lại máy tính để hoàn tất quá trình cài đặt.

A screenshot of a computer

Description automatically generated

Bạn cần tạo 1 tài khoản và đăng nhập vào docker

A screenshot of a computer

Description automatically generated

Sau khi khởi động lại Docker, bạn sẽ thấy icon của Docker Desktop ở góc dưới màn hình, cùng với trạng thái hoạt động của nó. Có 3 trạng thái: Stopping, Restarting và Running. Chọn Dashboard để vào giao diện chính của Docker.

A screenshot of a computer

Description automatically generated

Có 2 cách để vào Settings của Docker:

* Click chuột phải trên icon của Docker và chọn vào mục Settings
* Trên giao diện chính chọn icon bánh răng trên góc phải của giao diện

Có 2 trường hợp chọn backend engine cho Docker:

* Chọn Windows Container làm backend engine
* Chọn WSL 2 làm backend engine

**Lưu ý:** Cả hai phương pháp đều có ưu và nhược điểm riêng. Nếu bạn sử dụng Windows Container, việc cài đặt sẽ đơn giản hơn vì Windows 10 đã hỗ trợ đầy đủ. Tuy nhiên, hiệu suất chạy của nó không mượt mà bằng WSL 2. Ngược lại, nếu bạn chọn sử dụng WSL 2, hiệu suất chạy sẽ tốt hơn so với phương pháp đầu tiên, nhưng quá trình cài đặt sẽ phức tạp hơn.

### Cài đặt Windows Container

Trên thanh tìm kiếm của Windows, bạn hãy nhập Turn Windows features on or off. Tại đây, hãy đánh dấu vào hai ô Container và Hyper-V.

A screenshot of a computer

Description automatically generated

Sau khi hoàn tất, hãy Restart lại máy tính.

A screenshot of a computer error

Description automatically generated

### Cài đặt WSL 2

Để cài đặt WSL 2 cho Docker, bạn cần bật tính năng Windows Subsystem for Linux trên Windows

Sử dụng CMD hoặc PowerShell (chạy với quyền quản trị viên – Administrator) để chạy câu lệnh sau:

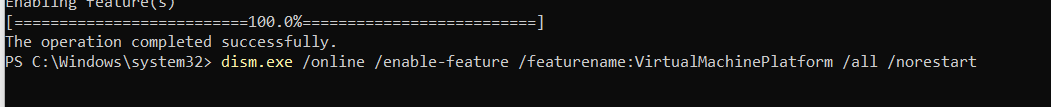
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

A screen shot of a computer

Description automatically generated

Tiếp theo, Bật tính năng Virtual Machine (máy ảo)

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart



Tải về gói update Linux kernel tại đây

Để đặt WSL 2 làm mặc định khi cài đặt Linux distribution

wsl --set-default-version 2

A black background with white text

Description automatically generated

Về cơ bản làm xong bước trên là oke rồi nhưng nếu bạn muốn sử dụng Ubuntu, Kali hoặc Debian thì bạn có thể tải tại đây hoặc sử dụng câu lệnh sau:

* Để xem các bản Linux distribution

wsl --list --online

A screenshot of a computer

Description automatically generated

* Để cài đặt Linux distribution, bạn cần thay <DistroName> bằng tên của bản Linux mà bạn muốn cài đặt. Ví dụ, nếu bạn muốn cài đặt Ubuntu, bạn sẽ thay <DistroName> bằng Ubuntu.

wsl --install -d <DistroName>

A screen shot of a computer

Description automatically generated

Sau đó Linux sẽ đòi hỏi bạn tạo Username, Password các kiểu nhưng chỉ một lần thôi

A screenshot of a computer

Description automatically generated

* User: tan
* Pass: 123

### **4. Hướng dẫn triển khai trên Docker**

Thực hiện các bước trên không có khó khăn gì với các thím chứ? Nếu không có vấn đề gì, chúng ta đi tiếp vào phần quan trọng nhất của bài viết này thôi nào các bạn ơiiiii Source code project Spring Boot, trong đó có file Dockerfile

FROM eclipse-temurin:17-jdk-focal

WORKDIR /app

COPY .mvn/ .mvn

COPY mvnw pom.xml ./

RUN ./mvnw dependency:go-offline

COPY src ./src

CMD ["./mvnw", "spring-boot:run"]

*Trong đó,*

* FROM eclipse-temurin:17-jdk-focal : dùng thư viện jdk 17, Eclipse Temurin cung cấp code và quy trình hỗ trợ xây dựng các tệp nhị phân time runing.
* WORKDIR /app: tạo một thư mục để chứa code ứng dụng của hình ảnh project.
* COPY .mvn/ ./mvn: Copy thư viện Maven từ máy local đến container image .
* COPY mvnw pom.xml ./ Copy file pom từ máy local đến container image.
* RUN ./mvnw dependency:go-offline RUN kích hoạt tất cả project.
* COPY src ./src Copy thư mục src từ local lên môi trường docker
* CMD ["./mvnw", "spring-boot:run"] Chạy câu lệnh Docker biên dịch và chạy app package

Tiếp theo, cần build lên Docker Image. Câu lệnh **docker-compose up -d**

**1: Docker là gì?**

* Định nghĩa

Docker là một công cụ được thiết kế để giúp tạo, triển khai và chạy các ứng dụng dễ dàng hơn bằng cách sử dụng các [container](https://opensource.com/resources/what-are-linux-containers?intcmp=7016000000127cYAAQ). Các [container](https://opensource.com/resources/what-are-linux-containers?intcmp=7016000000127cYAAQ) cho phép lập trình viên đóng gói một ứng dụng với tất cả các phần cần thiết, chẳng hạn như thư viện, database...vv và gửi tất cả ra dưới dạng một gói.

* Các thành phần chính

– Docker Engine : là thành phần chính của Docker, như một công cụ để đóng gói ứng dụng

– Docker Hub : là dịch vụ cloud để chia sẻ ứng dụng và tự động hóa chuỗi các công việc liên tục, có thể thao tác pull/push với các images

* **Các câu lệnh/chỉ thị trong Dockerfile**

**Dockerfile chứa một tập hợp các câu lệnh bao gồm cả của Docker và các câu lệnh của OS. Trước hết, cần tìm hiểu rõ các câu lệnh của Dockerfile.**

* **FROM:** Dựa trên một image có sẵn để tạo ra một image mới. Chỉ thị này phải được đặt ở đầu Dockerfile.
* **MAINTAINER:** (Tùy chọn) Điền thông tin của tác giả, người tạo ra image.
* **RUN:** Chỉ thị dùng để thực thi câu lệnh ở bên trong image
* **ADD:** Dùng để sao chép một file hoặc folder từ Host vào trong image. Có thể sử dụng một URL, Docker sẽ tải về thư mục đích bên trong image.
* **ENV:** Khởi tạo một biến môi trường bên trong image.
* **CMD:** Sử dụng để thực thi một câu lệnh khi tạo container được tạo từ image.
* **ENTRYPOINT:** Chỉ ra một câu lệnh được thực thi khi container chạy.
* **WORKDIR:** Chỉ ra thư mục làm việc khi tạo image hoặc khi khởi chạy container
* **USER:** Xác định user (UID) thực thi các câu lệnh ở các chỉ thị CMD, RUN, ENTRYPOINT,... được xác định ở phía sau nó.
* **VOLUME:** Cho phép truy cập/liên kết thư mục giữa container với host.
* **EXPOSE:** Khai báo các Port Container sử dụng.
* **ARG:** Khai báo sử dụng tham số khi build image sử dụng câu lệnh docker build với cờ--build-arg <varname>=<value>

**Dockerfile**

|  |
| --- |
| FROM openjdk:17-jdk-alpine  RUN apk update && apk add --no-cache maven  WORKDIR /app  COPY pom.xml .  RUN mvn dependency:go-offline  COPY src ./src  RUN mvn package -DskipTests  COPY target/\*.jar app.jar  EXPOSE 25565  CMD ["java", "-jar", "app.jar"] |

**docker-compose.yml**

|  |
| --- |
| services:  web:  image: mywebapp  build:  context: .  dockerfile: Dockerfile  ports:  - "25565:25565"  depends\_on:  - db  environment:  SPRING\_DATASOURCE\_URL: jdbc:mysql://db:3306/webbanhang  SPRING\_DATASOURCE\_USERNAME: root  SPRING\_DATASOURCE\_PASSWORD: 123456  SPRING\_JPA\_HIBERNATE\_DDL\_AUTO: update  SPRING\_JPA\_SHOW\_SQL: "true"  SPRING\_JPA\_PROPERTIES\_HIBERNATE\_DIALECT: org.hibernate.dialect.MySQLDialect  command: ["sh", "-c", "mvn clean package && java -jar target/\*.jar"]  db:  image: mysql:8  restart: always  ports:  - "3307:3306"  environment:  MYSQL\_ROOT\_PASSWORD: 123456  MYSQL\_DATABASE: webbanhang  volumes:  - db\_data:/var/lib/mysql  phpmyadmin:  image: phpmyadmin/phpmyadmin  restart: always  ports:  - "25560:80"  depends\_on:  - db  environment:  PMA\_HOST: db  MYSQL\_ROOT\_PASSWORD: 123456 volumes:  db\_data: |

./mvnw clean install

* **Build dự án lên docker**

**docker-compose up -d**

* **lệnh clear image trên docker**

docker images -a -q | % { docker image rm $\_ -f }

* **lệnh clear container trên docker**

docker rm -vf $(docker ps -aq)

Sau khi build lên Docker Image xong, kiểm tra trên Docker Desktop -> Images -> hiển thị Name "mywebapp" đúng như tên đã đặt

A screenshot of a computer

Description automatically generated

Ngoài ra, các bạn có thể đẩy trực tiếp hình ảnh của mình lên Docker Hub để dễ dàng chia sẻ. Action -> Push to Hub

A screenshot of a computer

Description automatically generated

Đặt tên Container Name -> Nhấn button "Run". Hiển thị thông tin **Containers**

A screenshot of a computer

Description automatically generated

Các bạn xem **Logs** chạy nhé

A screenshot of a computer

Description automatically generated

Docker Dashboard’s tab Stats, dễ dàng kiểm tra các thông tin CPU, Memory, Disk Read/Wirte, Network I/O

A graph of a graph

Description automatically generated with medium confidence

Kiểm tra database trong phpmyadmin

A screenshot of a computer

Description automatically generated

Chạy ứng dụng: <http://localhost:25565/login>

A screenshot of a computer

Description automatically generated