

# BÀI TẬP THỰC HÀNH CHI TIẾT

**Tiêu đề:** Xây dựng Môi trường Phát triển LAMP tự động với Vagrant và Shell Script

**Mục tiêu:** Xây dựng một dự án hoàn chỉnh sử dụng Vagrant để tự động tạo ra một máy ảo Ubuntu, cài đặt web server Apache, và cấu hình để chạy một trang web đơn giản từ mã nguồn trên máy thật.

## Phần A: Thiết lập Cấu trúc Dự án

### 1. Tạo thư mục gốc cho dự án:

Trên máy cá nhân (host machine), mở Terminal (hoặc PowerShell/CMD trên Windows) và tạo một thư mục cho dự án.

```
mkdir my-lamp-project  
cd my-lamp-project
```

### 2. Khởi tạo Git Repository:

Đây là bước cực kỳ quan trọng để quản lý phiên bản.

```
git init
```

### 3. Tạo cấu trúc thư mục con:

Chúng ta sẽ tổ chức dự án một cách khoa học ngay từ đầu.

```
mkdir scripts  
mkdir src
```

- `scripts/`: Sẽ chứa các Script tự động hóa (provisioning scripts).
- `src/`: Sẽ chứa mã nguồn trang web của chúng ta.

### 4. Khởi tạo Vagrantfile:

```
vagrant init ubuntu/focal64
```

Lệnh này sẽ tạo ra một file tên là `Vagrantfile`.

### 5. Commit lần đầu:

Lưu lại trạng thái ban đầu của dự án.

```
git add .  
git commit -m "Initial project structure and Vagrantfile"
```

## Phần B: Cấu hình Vagrantfile và Khởi động Máy ảo

### 1. Chỉnh sửa

Mở file Vagrantfile bằng một trình soạn thảo văn bản. Xóa hầu hết các dòng comment và chỉnh sửa file để có nội dung như sau. Đọc kỹ các comment để hiểu ý nghĩa từng dòng.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  # 1. Cấu hình Box: Sử dụng Ubuntu 20.04 (Focal Fossa) 64-bit.
  config.vm.box = "ubuntu/focal64"

  # 2. Cấu hình Mạng:
  # Chuyển tiếp (forward) cổng 8080 từ máy thật sang cổng 80 của máy ảo (cổng HTTP mặc định).
  # Giúp ta truy cập web server trên máy ảo từ trình duyệt máy thật qua địa chỉ http://localhost:8080
  config.vm.network "forwarded_port", guest: 80, host: 8080

  # 3. Cấu hình Thư mục Đồng bộ (Synced Folder):
  # Đồng bộ thư mục 'src' trên máy thật với thư mục '/var/www/project' trên máy ảo.
  # Bất kỳ thay đổi nào trong thư mục 'src' sẽ ngay lập tức được phản ánh trong máy ảo.
  config.vm.synced_folder "./src", "/var/www/project"
end
```

### 2. Khởi động máy ảo:

Chạy lệnh sau và quan sát quá trình Vagrant tải box về (nếu là lần đầu) và khởi tạo máy ảo.

```
vagrant up
```

### 3. Kết nối vào máy ảo:

Sử dụng SSH để truy cập vào dòng lệnh của máy ảo Ubuntu.

```
vagrant ssh
```

*Lưu ý: Giờ đang ở bên trong máy ảo. Dấu nhắc lệnh sẽ thay đổi, ví dụ  
vagrant@ubuntu-focal:~\$.*

### 4. Kiểm tra thư mục đồng bộ:

Bên trong máy ảo, hãy kiểm tra xem thư mục đã được đồng bộ đúng chưa.

```
ls -l /var/www/
```

Sẽ thấy một thư mục tên là `project`.

### 5. Thoát khỏi máy ảo:

Gõ lệnh `exit` để quay trở lại máy thật.

### 6. Commit các thay đổi:

```
git add Vagrantfile
git commit -m "Feat: Configure network and synced folder in Vagrantfile"
```

## Phần C: Viết Script Tự động hóa Cài đặt (Provisioning Script)

### 1. Tạo file Script:

Trên máy thật, tạo một file mới trong thư mục `scripts`.

```
touch scripts/bootstrap.sh
```

### 2. Viết nội dung cho `bootstrap.sh`:

Mở file `scripts/bootstrap.sh` và thêm vào nội dung sau. Đây là phần cốt lõi của bài thực hành.

```
#!/bin/bash

# -- Cập nhật hệ thống --
echo "Updating package list..."
apt-get update -y
echo "System updated."

# -- Cài đặt Apache2 --
echo "Installing Apache2..."
apt-get install -y apache2
echo "Apache2 installed."

# -- Cấu hình Apache --
# Tắt trang web mặc định của Apache
echo "Disabling default Apache site..."
a2dissite 000-default.conf

# Tạo một file cấu hình Virtual Host mới cho dự án của chúng ta
echo "Creating new Virtual Host config for the project..."
cat > /etc/apache2/sites-available/project.conf <<EOF
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/project
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
EOF

# Kích hoạt trang web của dự án
echo "Enabling project site..."
a2ensite project.conf

# Khởi động lại Apache để áp dụng thay đổi
echo "Restarting Apache..."
systemctl restart apache2

echo "Provisioning finished!"
```

### 3. Kết nối Script vào `vagrantfile`:

Mở lại `Vagrantfile` và thêm dòng `config.vm.provision` vào bên trong khối `config`.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  # 1. Cấu hình Box
  config.vm.box = "ubuntu/focal64"

  # 2. Cấu hình Mạng
  config.vm.network "forwarded_port", guest: 80, host: 8080

  # 3. Cấu hình Thư mục Đồng bộ
  config.vm.synced_folder "./src", "/var/www/project"

  # 4. Cấu hình Provisioning:
  # Chạy kịch bản bootstrap.sh để tự động cài đặt Apache.
  config.vm.provision "shell", path: "scripts/bootstrap.sh"
end
```

#### 4. Hủy và tạo lại môi trường để kiểm tra Script:

Cách tốt nhất để kiểm tra provisioning là tạo lại máy ảo từ đầu.

```
vagrant destroy -f  
vagrant up
```

Lần này, sẽ thấy các dòng `echo` từ Script `bootstrap.sh` được in ra màn hình trong quá trình `vagrant up`.

#### 5. Commit các thay đổi:

```
git add Vagrantfile scripts/bootstrap.sh  
git commit -m "Feat: Create and integrate shell provisioning script for Apache"
```

## Phần D: Triển khai Website và Xác thực

#### 1. Tạo trang web đơn giản:

Trên máy thật, tạo một file `index.html` bên trong thư mục `src/`.

```
touch src/index.html
```

Mở file `src/index.html` và thêm nội dung HTML đơn giản.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>My LAMP Project</title>  
  <style>  
    body { font-family: sans-serif; background-color: #f0f0f0; color: #333; text-align: center; margin-top: 5em; }  
    h1 { color: #007bff; }  
  </style>  
</head>  
<body>  
  <h1>Welcome to My Automated Web Server!</h1>  
  <p>This page is served by Apache on an Ubuntu VM, managed by Vagrant.</p>  
  <p>The source code is located on the host machine in the <code>src</code> directory.</p>  
</body>  
</html>
```

#### 2. Truy cập trang web:

Mở trình duyệt trên máy thật và truy cập vào địa chỉ: <http://localhost:8080>

Kết quả mong đợi: Sẽ thấy trang web mình vừa tạo.

#### 3. Kiểm tra và gỡ lỗi (Troubleshooting):

SSH vào máy ảo để kiểm tra.

```
vagrant ssh
```

Bên trong máy ảo, thực hiện các lệnh sau:

- Kiểm tra trạng thái dịch vụ Apache:

```
systemctl status apache2
```

- Kiểm tra access log:

```
tail -f /var/log/apache2/access.log
```

(Tải lại trang web trên trình duyệt và quan sát log).

- Kiểm tra error log:

```
cat /var/log/apache2/error.log
```

#### 4. Thoát khỏi máy ảo:

Gõ `exit`.

#### 5. Commit lần cuối:

```
git add src/index.html  
git commit -m "Feat: Add sample index.html and verify deployment"
```

### Phần E: Cleanup

#### 1. Tạm dừng máy ảo:

Nếu muốn tạm dừng công việc và quay lại sau.

```
vagrant halt
```

#### 2. Xóa hoàn toàn máy ảo:

Khi đã hoàn thành và muốn giải phóng tài nguyên.

```
vagrant destroy -f
```

