

VAGRANT CƠ BẢN

PHẦN 1: NHẬP MÔN VAGRANT

- 1.1. Hướng dẫn: Thêm một Vagrant Box Ubuntu (Bản mẫu máy ảo)
- 1.2. Hướng dẫn: Khởi tạo một máy ảo Ubuntu dùng một lần
- 1.3. Hướng dẫn: Tự động cài đặt VirtualBox Guest Additions

PHẦN 2: LÀM VIỆC VỚI CÁC NỀN TẢNG KHÁC NHAU

- 2.1. Hướng dẫn: Sử dụng máy ảo CentOS 7 với VMware
- 2.2. Hướng dẫn: Tùy chỉnh cấu hình máy ảo VMware (CPU, RAM)
- 2.3. Hướng dẫn: Tạo môi trường Vagrant đa nền tảng (VirtualBox & VMware)

PHẦN 3: TỰ ĐỘNG HÓA VỚI CÔNG CỤ PROVISIONING

- 3.1. Hướng dẫn: Dùng Ansible để tự động cài đặt Docker

PHẦN 1: NHẬP MÔN VAGRANT

1.1. Hướng dẫn: Thêm một Vagrant Box Ubuntu

Bài này sẽ giúp bạn tải về và lưu một "bản mẫu" (gọi là `box`) của hệ điều hành Ubuntu 16.04. Sau này, bạn có thể dùng bản mẫu này để tạo ra nhiều máy ảo giống hệt nhau một cách nhanh chóng mà không cần tải lại.

- Yêu cầu:
 - Đã cài đặt Vagrant.
 - Đã cài đặt một trình ảo hóa (hypervisor), ví dụ như VirtualBox.
 - Máy tính có kết nối Internet.
- Các bước thực hiện:
 1. Mở cửa sổ dòng lệnh (Terminal, PowerShell).
 2. Gõ lệnh sau để yêu cầu Vagrant tìm và tải về "box" có tên ``ubuntu/xenial64``.

```
vagrant box add ubuntu/xenial64
```

Vagrant sẽ hiển thị tiến trình tải file. Khi hoàn tất, bạn sẽ thấy một dòng thông báo thành công. "Bản mẫu" Ubuntu này đã sẵn sàng trong máy của bạn để sử dụng.

1.2. Hướng dẫn: Khởi tạo một máy ảo Ubuntu dùng một lần

- Mục tiêu: Bạn sẽ học cách tạo nhanh một máy ảo Ubuntu, truy cập vào bên trong nó để làm việc, và sau đó xóa đi để giải phóng tài nguyên máy tính.
- Yêu cầu:
 - Đã cài đặt Vagrant và VirtualBox.
 - Đã thêm box `ubuntu/xenial64` (theo hướng dẫn ở bài 1.1).
- Các bước thực hiện:
 1. Tạo một thư mục mới cho dự án và di chuyển vào trong đó:

```
mkdir vagrant_ubuntu_xenial_1 && cd $_
```

2. Tạo một file tên là `Vagrantfile`. Đây là file "công thức" để Vagrant biết phải làm gì.

```
Vagrant.configure("2") do |config|  
  config.vm.box = "ubuntu/xenial64"  
end
```

...

3. Bắt đầu khởi tạo và chạy máy ảo:

```
vagrant up
```

4. Truy cập vào bên trong máy ảo qua kết nối SSH:

```
vagrant ssh
```

...

5. Khi làm việc xong, thoát khỏi SSH (`exit`) và dùng lệnh này để xóa hoàn toàn máy ảo:

```
vagrant destroy
```

...

Lưu ý: Nếu bạn chỉ muốn tắt máy ảo để lần sau bật lại và giữ nguyên trạng thái, hãy dùng lệnh `vagrant halt`.

Một máy ảo Ubuntu hoàn chỉnh sẽ được tạo ra và bạn có thể truy cập qua SSH. Sau khi chạy `vagrant destroy`, máy ảo sẽ được xóa sạch.

1.3. Hướng dẫn: Tự động cài đặt VirtualBox Guest Additions

VirtualBox Guest Additions là một bộ công cụ giúp máy ảo chạy mượt hơn. Bài này hướng dẫn bạn cài một plugin để Vagrant tự động quản lý việc này.

- Yêu cầu:
 - Đã cài đặt Vagrant và VirtualBox.
- Các bước thực hiện:
 1. Cài đặt plugin có tên là `vagrant-vbguest`:

```
vagrant plugin install vagrant-vbguest
```

...

2. Mở file `Vagrantfile` và thêm đoạn code sau để Vagrant tự động xử lý plugin:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"

  # Chỉ áp dụng cấu hình nếu plugin "vagrant-vbguest" đã
  # được cài
  if Vagrant.has_plugin?("vagrant-vbguest") then
    config.vbguest.auto_update = false
  end
end
```

3. Khởi động máy ảo của bạn:

```
vagrant up
```

Khi bạn chạy `vagrant up`, bạn sẽ thấy các dòng thông báo về việc kiểm tra và cài đặt Guest Additions (nếu cần).

PHẦN 2: LÀM VIỆC VỚI CÁC NỀN TẢNG KHÁC NHAU

2.1. Hướng dẫn: Sử dụng máy ảo CentOS 7 với VMware

Khởi tạo một máy ảo CentOS 7.2, nhưng lần này sử dụng VMware (Workstation hoặc Fusion) làm trình ảo hóa thay vì VirtualBox.

- **Yêu cầu:**

- Đã cài đặt Vagrant.
- Đã cài đặt VMware Workstation (PC) hoặc VMware Fusion (Mac).
- Đã cài đặt plugin Vagrant cho VMware (tải từ trang chủ Vagrant).

- **Các bước thực hiện:**

1. Tạo một thư mục dự án mới.
2. Tạo file `Vagrantfile` với nội dung sau. Box `bento/centos-7.2` hỗ trợ nhiều trình ảo hóa khác nhau.

```
Vagrant.configure("2") do |config|  
  config.vm.box = "bento/centos-7.2"  
end
```

3. Khởi động máy ảo và chỉ định rõ trình ảo hóa là VMware:

```
# Nếu dùng VMware Fusion trên Mac  
vagrant up --provider=vmware_fusion
```

Một máy ảo CentOS 7.2 sẽ được tạo và chạy trên VMware.

2.2. Hướng dẫn: Tùy chỉnh cấu hình máy ảo VMware (CPU, RAM)

Tùy chỉnh các thông số phần cứng như CPU và RAM cho máy ảo VMware ngay trong file `Vagrantfile`.

- Yêu cầu:
 - Môi trường Vagrant và VMware đã hoạt động.
 - Một file `Vagrantfile` sử dụng box tương thích với VMware.
- Các bước thực hiện:
 1. Mở file `Vagrantfile`.
 2. Thêm khối cấu hình riêng cho provider VMware. Ví dụ này sẽ thiết lập 2 CPU và 1024MB RAM.

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/centos-7.2"

  # Cấu hình cho cả Fusion và Workstation
  ["vmware_fusion", "vmware_workstation"].each do |vmware|
    config.vm.provider vmware do |v|
      v.vmx["numvcpus"] = "2"      # Số lượng CPU
      v.vmx["memsize"] = "1024"   # Dung lượng RAM (MB)
    end
  end
end
```

3. Khởi động hoặc khởi động lại máy ảo để áp dụng thay đổi (`vagrant up` hoặc `vagrant reload`).

Máy ảo sẽ khởi động với cấu hình phần cứng bạn đã chỉ định.

2.3. Hướng dẫn: Tạo môi trường Vagrant đa nền tảng (VirtualBox & VMware)

Tạo ra một file `Vagrantfile` duy nhất có thể khởi tạo máy ảo trên cả VirtualBox và VMware, với các thiết lập phần cứng được quản lý tập trung.

- Yêu cầu:

- Đã cài đặt Vagrant, VirtualBox, và VMware (cùng plugin).
- Sử dụng một box hỗ trợ đa nền tảng như `bento/centos-7.2`.
- Các bước thực hiện:
 1. Tạo file `Vagrantfile` và khai báo các biến cho RAM và CPU để dễ quản lý.

```
vm_memory = 1024
vm_cpus = 2
```

2. Thêm các khối cấu hình cho từng provider, sử dụng các biến đã tạo:

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/centos-7.2"

  # Cấu hình cho VMware
  ["vmware_fusion", "vmware_workstation"].each do |vmware|
    config.vm.provider vmware do |v|
      v.vmx["numvcpus"] = vm_cpus
      v.vmx["memsize"] = vm_memory
    end
  end

  # Cấu hình cho VirtualBox
  config.vm.provider :virtualbox do |vb|
    vb.memory = vm_memory
    vb.cpus = vm_cpus
  end
end
```

Cùng một file `Vagrantfile`, bạn có thể khởi tạo máy ảo trên bất kỳ trình ảo hóa nào bạn chọn mà không cần sửa đổi file.

PHẦN 3: TỰ ĐỘNG HÓA VỚI CÔNG CỤ PROVISIONING

3.1. Hướng dẫn: Dùng Ansible để tự động cài đặt Docker

Sử dụng Vagrant để khởi tạo một máy ảo CentOS 7 và tự động cài đặt Docker lên đó bằng một script (playbook) của Ansible. Đây là một ví dụ về provisioning - tự động cài đặt và cấu hình phần mềm.

- Yêu cầu:
 - Đã cài đặt Vagrant và một trình ảo hóa (ví dụ: VirtualBox).
 - Đã cài đặt Ansible trên máy thật của bạn (cách dễ nhất: `pip install ansible`).
- Các bước thực hiện:

1. Tạo một thư mục dự án mới.
2. Bên trong thư mục, tạo một file script Ansible tên là `playbook.yml`:

```
---
- hosts: all
  become: yes
  tasks:
    - name: Install EPEL Repo
      yum: name=epel-release state=present
    - name: Install Docker
      yum: name=docker state=present
    - name: Enable and Start Docker Daemon
      service: name=docker state=started enabled=yes
    - name: Add vagrant user to Docker group
      user: name=vagrant groups=docker append=yes
```

3. Tạo file `Vagrantfile` để định nghĩa máy ảo và chỉ định việc chạy script Ansible:

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/centos-7.2"

  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "playbook.yml"
  end
end
```

4. Khởi động môi trường Vagrant:

```
vagrant up
```

5. Kết nối vào máy ảo và kiểm tra xem Docker đã được cài đặt và đang chạy chưa:

```
vagrant ssh
# Bên trong máy ảo, chạy các lệnh sau:
docker --version
docker run --rm hello-world
```

Vagrant sẽ tạo ra một máy ảo CentOS 7, sau đó Ansible sẽ tự động chạy và cài đặt Docker lên đó.