

Apache

Ngày 14/6/2025

Lê Quang Tân, Trần Quốc Việt



Nội dung

Giới thiệu Apache

- Giới thiệu và cấu hình Apache

Apache khi so sánh với nginx

- So sánh qua các phương diện kiến trúc, hiệu suất, cấu hình, module

Demo

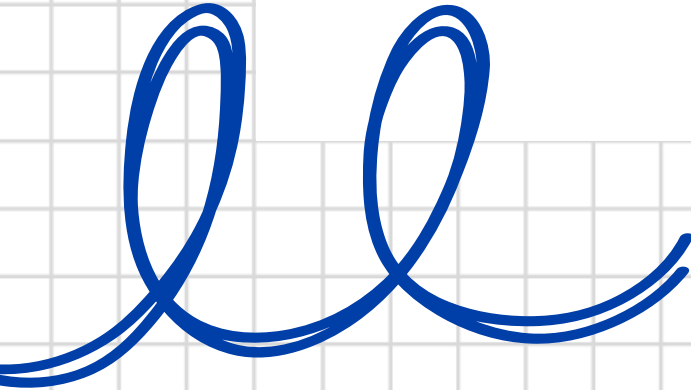
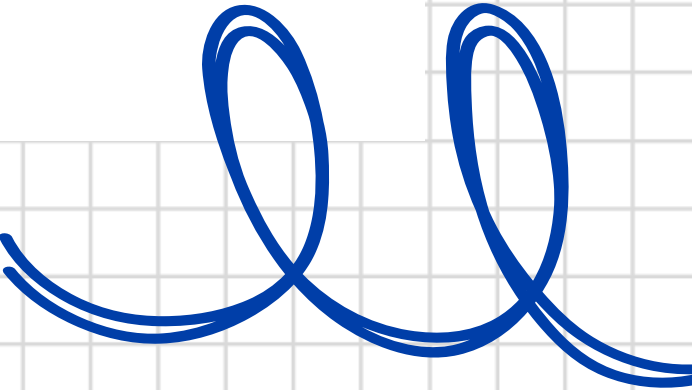
- Cách cài đặt và cấu hình





Apache là gì?

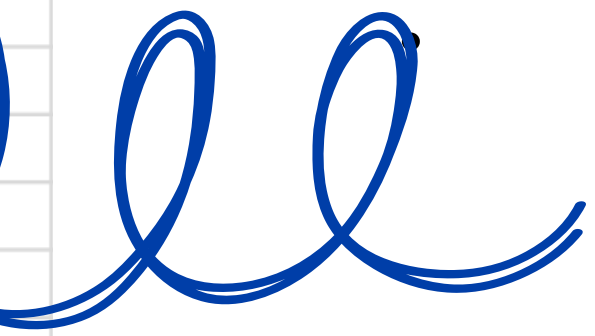


- **Apache HTTP Server: Phần mềm máy chủ web mã nguồn mở, phát triển bởi Apache Software Foundation từ 1995.**
 - **Thống kê: Theo W3Techs (2025), Apache chiếm ~30% thị phần máy chủ web, đứng sau Nginx (~35%).**
- 
- 



Kiến trúc sử dụng Multi-Processing Modules (MPM):

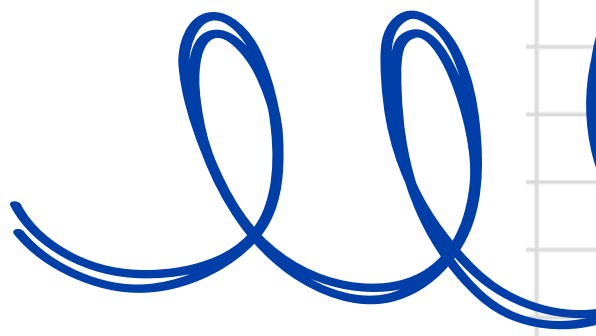
Prefork

- Mỗi yêu cầu tạo một tiến trình (process), phù hợp với PHP.
- 

Worker

- Kết hợp tiến trình và luồng (thread), hiệu quả hơn.

Event

- Tối ưu cho lưu lượng cao, tương tự mô hình sự kiện của Nginx.
- 

Cài đặt Apache

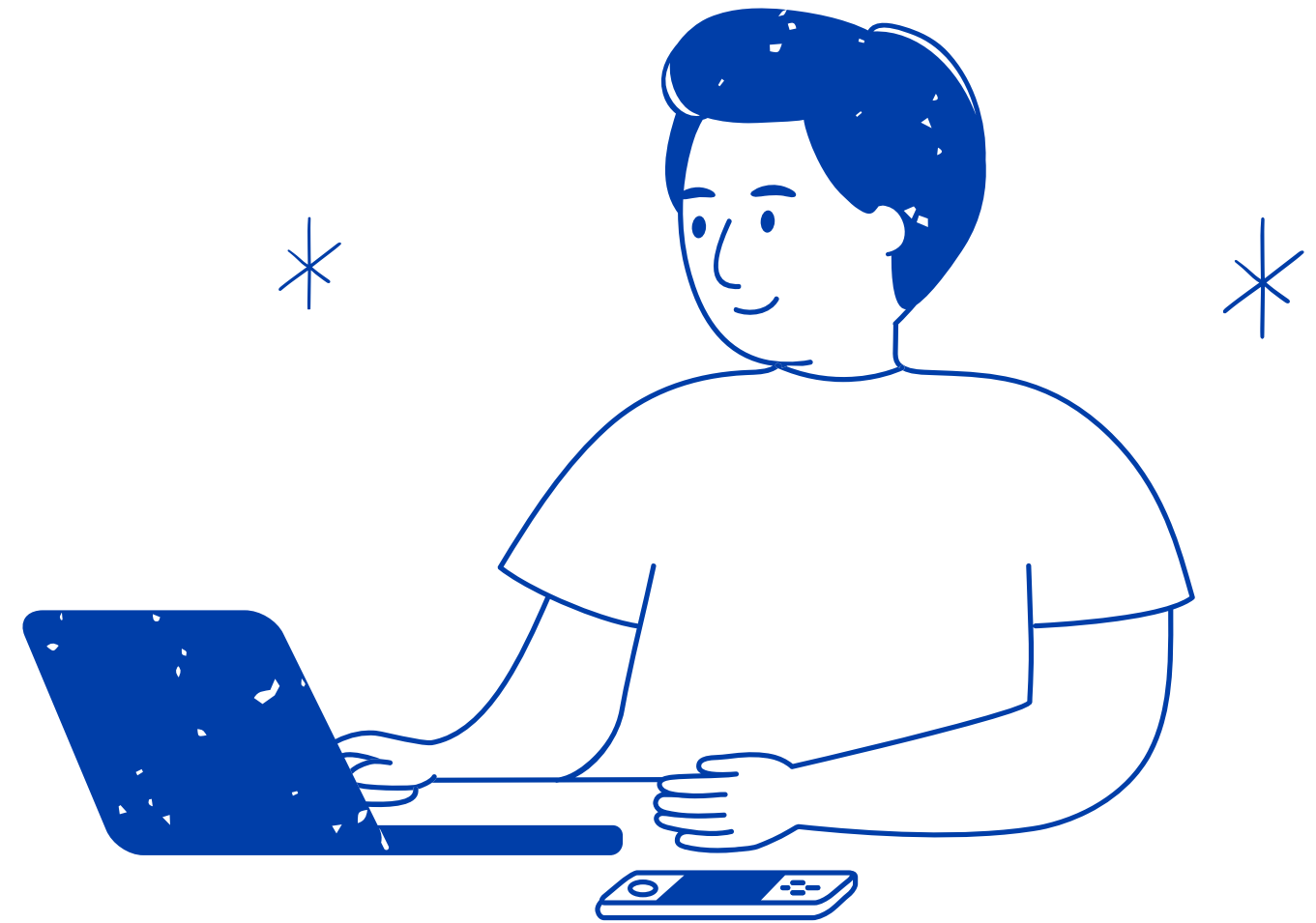
- Tập cấu hình chính: `/etc/apache2/apache2.conf` (Ubuntu) hoặc `/etc/httpd/conf/httpd.conf` (CentOS).
- Thư mục cấu hình:
 - `/etc/apache2/sites-available/`: Chứa cấu hình virtual hosts.
 - `/etc/apache2/sites-enabled/`: Liên kết đến các site được kích hoạt.
 - `/etc/apache2/mods-available/`: Chứa module có thể bật/tắt.
- Công cụ quản lý:
 - `a2enmod`, `a2dismod`: Bật/tắt module.
 - `a2ensite`, `a2dissite`: Kích hoạt/vô hiệu hóa site.
 - `apachectl -t`: Kiểm tra cú pháp cấu hình.

```
bash
```

```
sudo apt update  
sudo apt install apache2  
sudo systemctl start apache2  
sudo systemctl enable apache2
```

Tính năng nổi bật

- Hỗ trợ virtual hosts, SSL/TLS, URL rewriting.
- Tập .htaccess cho phép cấu hình cấp thư mục.
- Tích hợp tốt với các ngôn ngữ lập trình (PHP, Python, Perl).
- Logging chi tiết và khả năng tùy chỉnh.



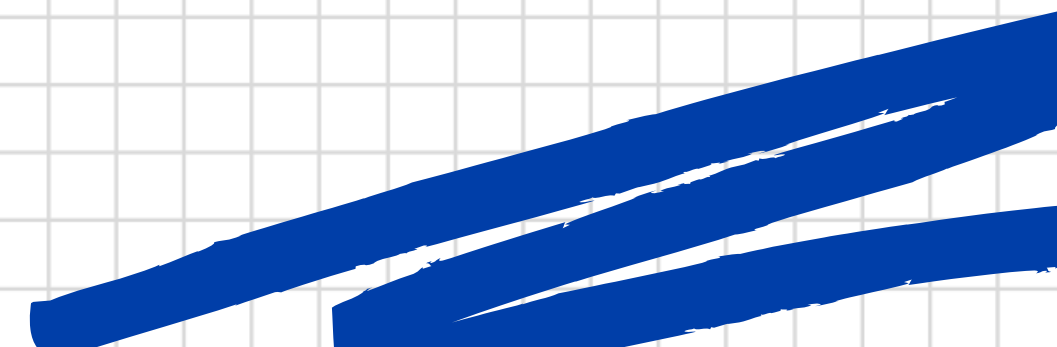
So sánh Apache với Nginx

Tiêu chí	Apache	Nginx
Kiến trúc	Process-based (prefork) hoặc hybrid (worker/event). Mỗi kết nối có thể tiêu tốn nhiều bộ nhớ.	Event-driven, xử lý hàng nghìn kết nối đồng thời với ít tài nguyên hơn.
Hiệu suất	Tốt cho ứng dụng động, nhưng chậm hơn với nội dung tĩnh hoặc lưu lượng lớn (>10K kết nối).	Vượt trội với nội dung tĩnh, reverse proxy, và xử lý lưu lượng cao.
Cấu hình	Dễ dàng với .htaccess, không cần khởi động lại server khi thay đổi cấu hình cấp thư mục.	Cấu hình tập trung, yêu cầu chỉnh sửa tệp nginx.conf và khởi động lại server.
Module	Hỗ trợ module động, dễ cài đặt qua apt hoặc yum. Ví dụ: mod_wsgi, mod_security.	Module tĩnh, cần biên dịch lại nếu thêm module mới, ít linh hoạt hơn.
Tích hợp PHP	Tích hợp chặt chẽ qua mod_php, dễ triển khai nhưng tiêu tốn tài nguyên.	Dùng PHP-FPM, hiệu quả hơn nhưng cần cấu hình bổ sung.
Khả năng proxy	Hỗ trợ reverse proxy (mod_proxy), nhưng không tối ưu bằng.	Ưu thế làm reverse proxy và load balancer.
Khả năng bảo mật	Module như mod_security cung cấp WAF mạnh mẽ.	Cần thêm phần mềm bên thứ ba (Fail2Ban, WAF) để đạt mức bảo mật tương tự.

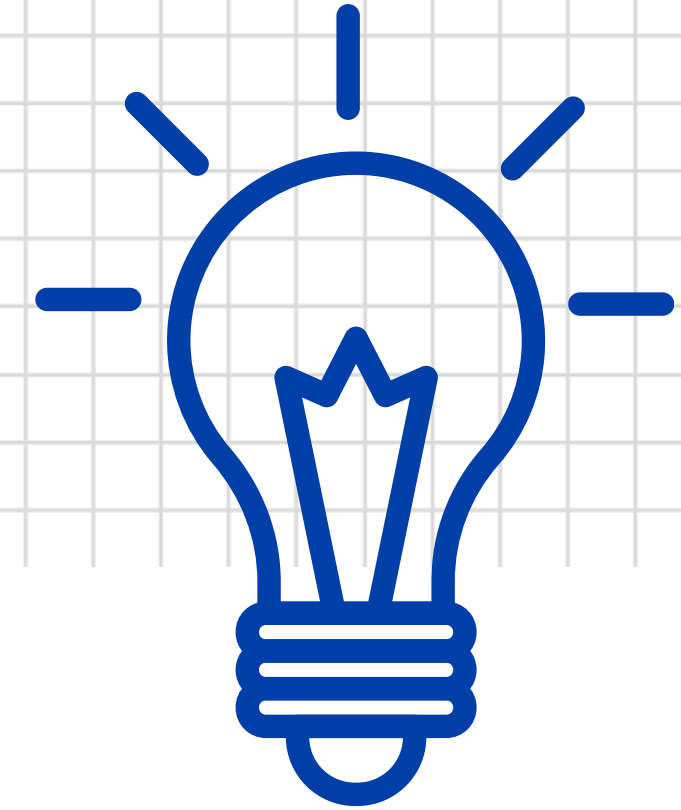


Kiến trúc

- Apache tạo ra một tiến trình (process) hoặc luồng (thread) cho mỗi kết nối. Khi có nhiều người dùng truy cập cùng lúc, hệ thống phải tạo nhiều tiến trình nên tốn RAM và CPU, nhất là với lưu lượng lớn.
- Nginx dùng kiến trúc event-driven non-blocking. Một worker có thể xử lý hàng nghìn kết nối nhờ việc không chặn (non-blocking I/O), dùng ít tài nguyên hơn.



Hiệu suất

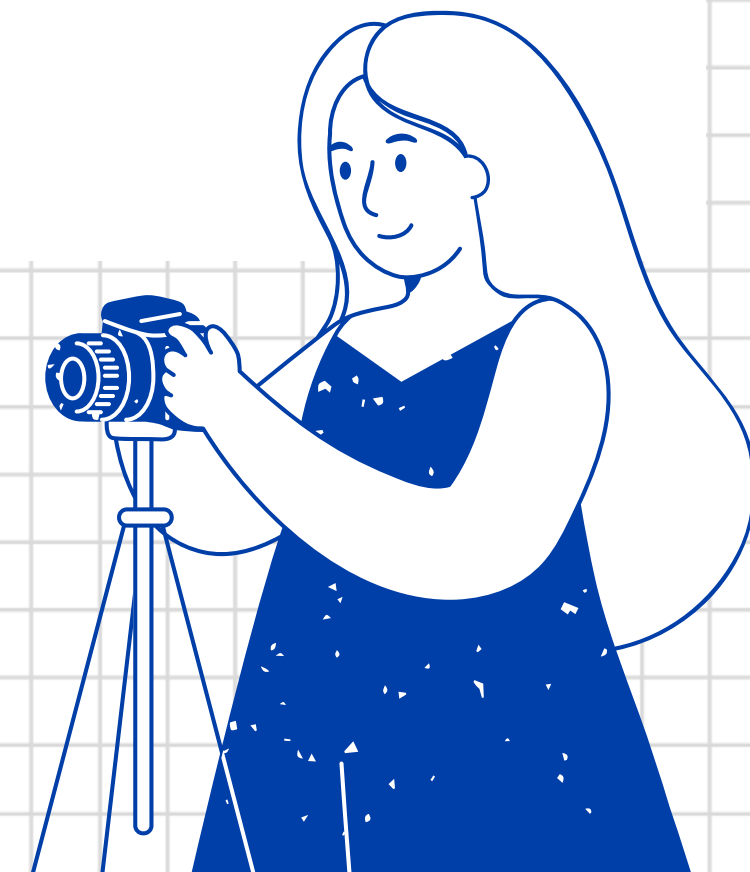
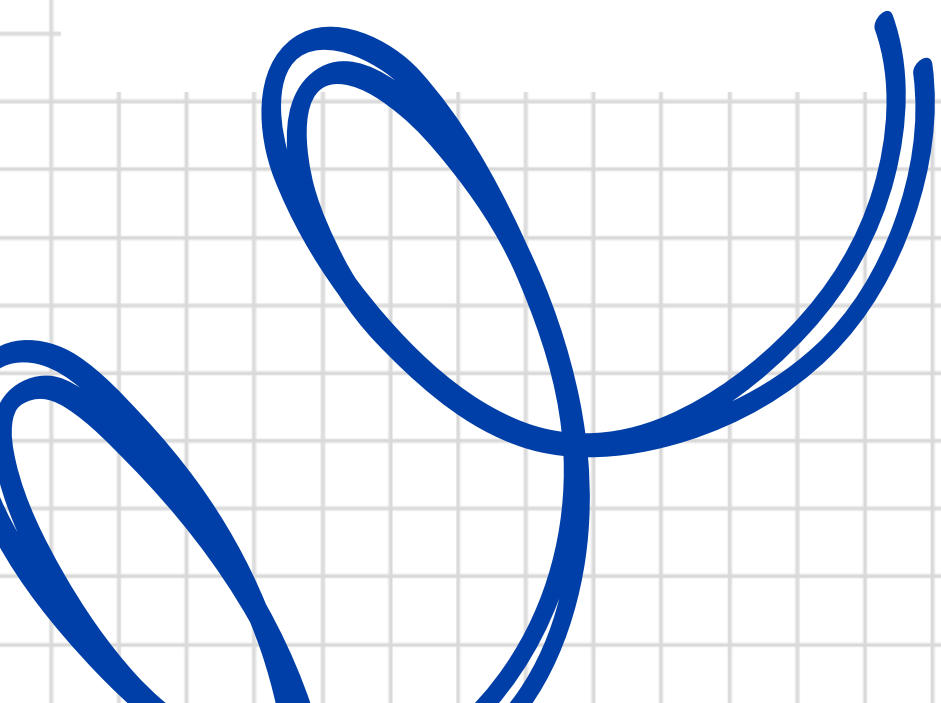


- Apache hoạt động tốt khi xử lý các ứng dụng web động (như PHP thông qua mod_php) nhờ tích hợp sâu. Nhưng với nội dung tĩnh (ảnh, HTML), do phải tạo tiến trình/luồng, Apache kém hiệu quả hơn khi xử lý hàng nghìn request.
- Nginx tối ưu cho nội dung tĩnh, truyền tải file nhanh, dùng ít CPU, RAM → phù hợp với website nhiều truy cập hoặc làm reverse proxy.



Cấu hình

- Apache cho phép đặt file .htaccess ở mỗi thư mục, tiện lợi cho việc cấu hình phân quyền, rewrite, v.v. mà không cần restart.
- Nginx thì chỉ cấu hình trong 1 hoặc vài file cấu hình trung tâm (nginx.conf). Việc thay đổi cấu hình thường yêu cầu reload/restart, nhưng đổi lại là bảo mật và hiệu năng tốt hơn.





Module

- Apache hỗ trợ Dynamic Shared Object (DSO) – module có thể nạp hoặc gỡ lúc chạy mà không cần biên dịch lại server → rất linh hoạt.
- Nginx thường phải biên dịch lại từ đầu nếu muốn thêm module (trừ khi dùng bản thương mại hoặc openresty) → ít linh hoạt, nhưng nhanh hơn vì ít overload.



Tích hợp PHP

- Apache dùng mod_php để chạy PHP, rất đơn giản để thiết lập. Nhưng vì mod_php chạy trong cùng process, nó tiêu tốn tài nguyên kể cả khi không cần PHP.
- Nginx không hỗ trợ mod_php; nó gửi yêu cầu PHP tới PHP-FPM, là một tiến trình độc lập. Cách này tách biệt rõ ràng, dễ scale và tiết kiệm tài nguyên, nhưng cần thêm bước cấu hình.



Khả năng Proxy

- Apache có mod_proxy nhưng thiết kế ban đầu không dành cho proxy nên xử lý kém tối ưu.
- Nginx được thiết kế ngay từ đầu để làm reverse proxy, nên xử lý connection pooling, load balancing, caching rất tốt → thường dùng đứng trước các app khác như Node.js, PHP, Python backend.



Khả năng bảo mật

- Apache có mod_security, là một WAF (Web Application Firewall) rất mạnh, dễ tích hợp.
- Nginx không tích hợp WAF gốc, nhưng có thể dùng ModSecurity với Nginx (khó hơn) hoặc kết hợp với Fail2Ban, hoặc để Nginx xử lý frontend, sau đó chuyển sang Apache có mod_security.



Tại Sao Apache Mất Thị Phần Web?

Hiệu suất kém hơn Nginx

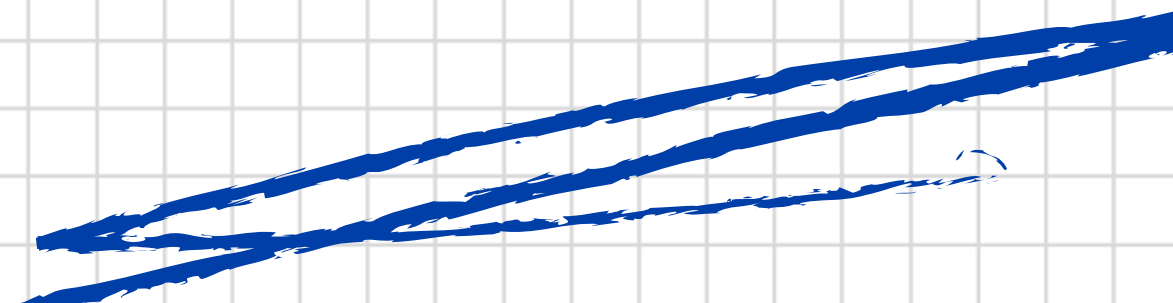
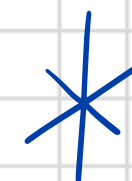
Tích hợp PHP

Nhu cầu nội dung tĩnh tăng

Đối thủ mới: Caddy, LiteSpeed

Xu Hướng Cloud và Containerization

Cộng Đồng và Hệ Sinh Thái có tốc độ đổi mới chậm





Khi nào chọn Apache?

- Ứng dụng cần tích hợp sâu với PHP hoặc các ngôn ngữ backend khác.
- Dự án yêu cầu cấu hình cấp thư mục qua .htaccess (ví dụ: WordPress).
- Đội ngũ quen thuộc với Apache hoặc cần sử dụng module đặc thù.

Khi nào chọn Nginx?

- Phục vụ nội dung tĩnh hoặc xử lý lưu lượng truy cập cao.
- Cần reverse proxy để cân bằng tải hoặc tích hợp microservices.
- Ưu tiên tiết kiệm tài nguyên server.





Demo 1: HTML

- Cài đặt Apache
 - *sudo apt update*
 - *sudo apt install apache2*
 - *sudo systemctl start apache2*
 - *sudo systemctl enable apache2*
- Tạo */var/www/html/index.html* (có thể build từ frontend có sẵn)
- Kiểm tra *http://localhost*



Demo 2: PHP

- Cài đặt Apache
 - *sudo apt install php libapache2-mod-php*
 - *sudo systemctl restart apache2*
- Cấu hình Apache */etc/apache2/sites-available/php.conf*
- Tạo thư mục */var/www/php*
- Đặt frontend vào thư mục */var/www/php*, backend đưa vào */var/www/php/api*
- Kích hoạt:
 - *sudo a2ensite php.conf*
 - *sudo systemctl restart apache2*
- Kiểm tra: web bằng địa chỉ trong cấu hình



Demo 3: PythonFlask

- Cài đặt:
 - *sudo apt install python3-pip libapache2-mod-wsgi-py3 pymysql*
 - *pip3 install flask*
- Tạo thư mục */var/www/flask_app/* và đưa frontend vào
- Tạo thư mục */var/www/flask_app/api* cho backend
- Tạo file backend *index.py* và *wsgi.py* trong thư mục *api*
- Cấu hình Apache */etc/apache2/sites-available/flask_app.conf*:
- Kích hoạt:
 - *sudo a2ensite flask_app.conf*
 - *sudo systemctl restart apache2*
- Kiểm tra web bằng địa chỉ trong cấu hình