Nginx vs Apache: Configuration Files and Paths

Introduction

When deploying web applications, choosing between Nginx and Apache is a crucial decision. Both are popular web servers, but they differ in performance, configuration, and use cases. This blog will explain what Nginx and Apache are, their configuration file locations, why we configure them, and how they work on Ubuntu and AWS Linux.

Why Do We Configure Web Servers?

Configuration is essential for web servers to:

- Define how they handle incoming requests
- Set up domain names and virtual hosts
- Enable security settings such as SSL and access controls
- Optimize performance and load balancing
- Log and monitor server activity

Proper configuration ensures that websites load quickly, securely, and efficiently, meeting user expectations and business requirements.

What is Apache?

Apache, officially known as Apache HTTP Server, is an open-source web server developed by the Apache Software Foundation. It is widely used for serving dynamic and static web content.

Apache Configuration File in Ubuntu & AWS Linux

- The main Apache configuration file is located at:
    - /etc/apache2/apache2.conf (Ubuntu)
    - /etc/httpd/conf/httpd.conf (AWS Linux)

- Virtual host configuration files are stored in:
    - /etc/apache2/sites-available/ (Ubuntu)
    - /etc/httpd/conf.d/ (AWS Linux)

- and enabled via symbolic links in:
    - /etc/apache2/sites-enabled/ (Ubuntu)

- Modules and additional configurations are found in:
    - /etc/apache2/mods-available/ (Ubuntu)
    - /etc/httpd/conf.modules.d/ (AWS Linux)

Basic Apache Configuration

To modify the default website configuration, edit the virtual host file:

- sudo nano /etc/apache2/sites-available/000-default.conf # Ubuntu
- sudo nano /etc/httpd/conf.d/vhost.conf # AWS Linux

A basic configuration may look like this:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

After making changes, restart Apache:

- sudo systemctl restart apache2 # Ubuntu
- sudo systemctl restart httpd # AWS Linux

What is Nginx?

Nginx is a high-performance, lightweight web server known for its reverse proxy and load balancing capabilities. It is more efficient than Apache when handling multiple concurrent connections.

Nginx Configuration File in Ubuntu & AWS Linux

- The main configuration file is located at:
  - /etc/nginx/nginx.conf

- Website configuration files are stored in:
  - /etc/nginx/sites-available/ (Ubuntu)
  - /etc/nginx/conf.d/ (AWS Linux)

- and enabled via symbolic links in:
  - /etc/nginx/sites-enabled/ (Ubuntu)

- Logs are stored in:
  - /var/log/nginx/

Basic Nginx Configuration

To modify the default server block, edit:

- sudo nano /etc/nginx/sites-available/default # Ubuntu
- sudo nano /etc/nginx/conf.d/default.conf # AWS Linux

A simple configuration file looks like this:

```
server {
    listen 80;

    server_name example.com;

    root /var/www/html;

    location / {        index index.html index.htm;

        }

}
```

After editing, restart Nginx:

- `sudo systemctl restart nginx`

Key Differences Between Apache and Nginx Configuration

Feature	Apache	Nginx
Configuration File	/etc/apache2/apache2.conf (Ubuntu), /etc/httpd/conf/httpd.conf (AWS Linux)	/etc/nginx/nginx.conf
Virtual Hosts	/etc/apache2/sites-available/ (Ubuntu), /etc/httpd/conf.d/ (AWS Linux)	/etc/nginx/sites-available/ (Ubuntu), /etc/nginx/conf.d/ (AWS Linux)
Modules	Uses .htaccess files	No .htaccess, direct config
Performance	Process-based, slower for many connections	Event-driven, efficient
Best For	Dynamic content (PHP, CGI)	Static content, reverse proxy

This article discusses the key differences between Nginx and Apache web servers, focusing on their configuration, performance, and use cases, specifically on Ubuntu and AWS Linux platforms. Apache is known for its dynamic content handling and robust module support, while Nginx excels in managing static content and handling multiple concurrent connections efficiently. The article provides insights into configuration file locations, basic setup instructions, and highlights the importance of proper server configuration to ensure optimal website performance and security.