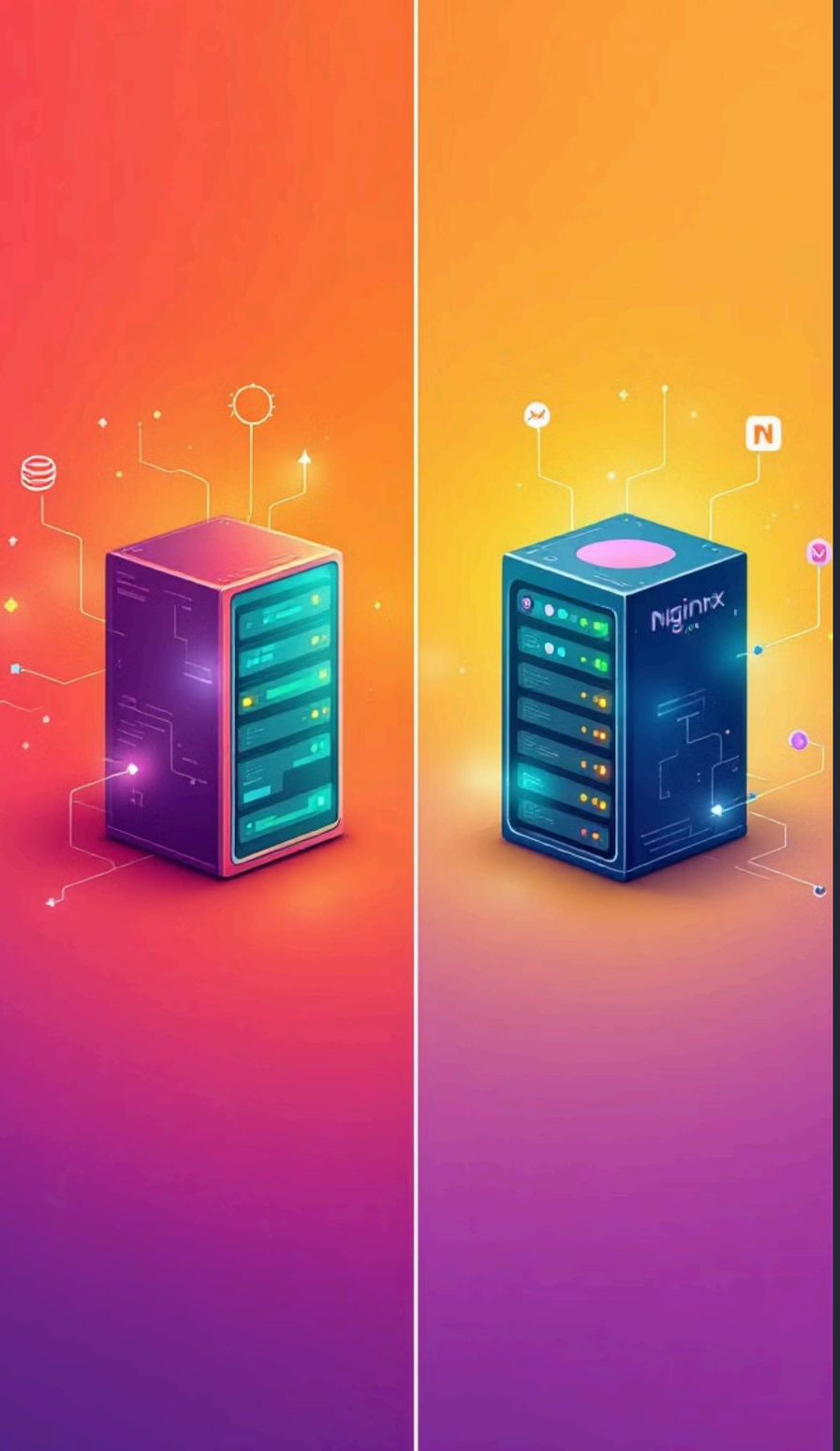
Web Server Configuration - Apache & Nginx

Module 1: Introduction to Web Server Configuration

DK **by Dan K**



Overview of Web Servers: Apache and Nginx

1 What is a Web Server?

Software that delivers web content to users via HTTP or HTTPS protocols. Acts as a bridge between client requests and server-side resources.

2 Introduction to Apache

Developed by the Apache Software Foundation. Known for its flexibility, module-based architecture, and widespread usage. Supports dynamic content and multiple processing modes (e.g., MPM prefork, MPM worker).

3 Introduction to Nginx ("engine x")

Originally designed as a reverse proxy and load balancer. Lightweight, high-performance web server with event-driven architecture. Ideal for handling high traffic with low resource usage.

Role of Web Servers in Hosting Environments

Hosting Websites

Serves static and dynamic content. Handles client requests and delivers appropriate responses.

Reverse Proxy

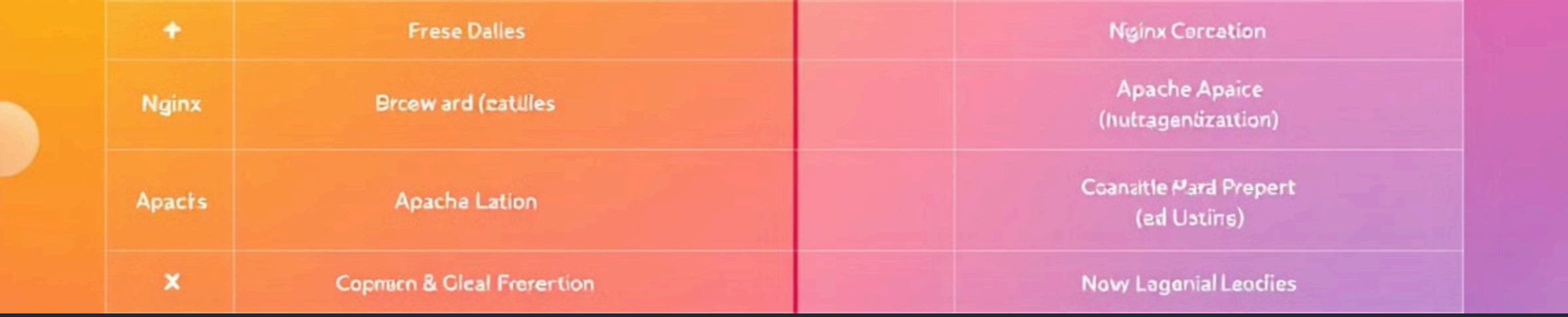
Balances traffic between backend servers. Enhances security by hiding backend infrastructure.

Caching

Improves website performance by storing frequently accessed data.

Logging and Monitoring

Tracks server activity for troubleshooting and optimization.



Key Differences Between Apache and Nginx

Feature	Apache	Nginx
Architecture	Process-based	Event-driven
Performance	Suitable for dynamic content	Excels at serving static content
Configuration	.htaccess for per-directory configs	Centralized configuration
Resource Usage	Higher	Lower
Use Case	Flexible, dynamic content	High-traffic static content, reverse proxy

Installation and Initial Setup of Apache and Nginx

Using AlmaLinux

1

Update System Packages

sudo dnf update -y

2

Install Apache

sudo dnf install httpd -y

3

Enable and Start Apache

sudo systemctl enable httpd
sudo systemctl start httpd
sudo systemctl status httpd

4

Install Nginx

sudo dnf install nginx -y

5

Enable and Start Nginx

sudo systemctl enable nginx
sudo systemctl start nginx
sudo systemctl status nginx

Installation and Initial Setup (Continued)

Using Ubuntu/Debian

1

Update System Packages

sudo apt update && sudo apt upgrade -y

2

Install Apache

sudo apt install apache2 -y

3

Enable and Start Apache

sudo systemctl enable apache2

sudo systemctl start apache2

systemctl status apache2 or service apache2 status

4

Install Nginx

sudo apt install nginx -y

5

Enable and Start Nginx

sudo systemctl enable nginx

sudo systemctl start nginx

sudo systemctl status nginx

Tools and Utilities for Managing Web Servers

Apache Utilities

- apachectl: Controls Apache services.
- htpasswd: Manages basic HTTP authentication.

Nginx Utilities

- nginx -t: Tests Nginx configurations.
- nginx -s reload: Reloads configurations without downtime.

Log Monitoring Tools

Command Line Tools

tail, grep, and awk for analyzing log files.

Real-time Analysis

Tools like GoAccess for real-time log analysis.

Performance Monitoring Tools

top

Command-line tool for real-time system monitoring.

htop

Interactive process viewer and system monitor.

Conclusion: Web Server Configuration Essentials

1 Choose the Right Web Server

Consider Apache for flexibility and Nginx for high-performance static content delivery.

2 Master Installation and Setup

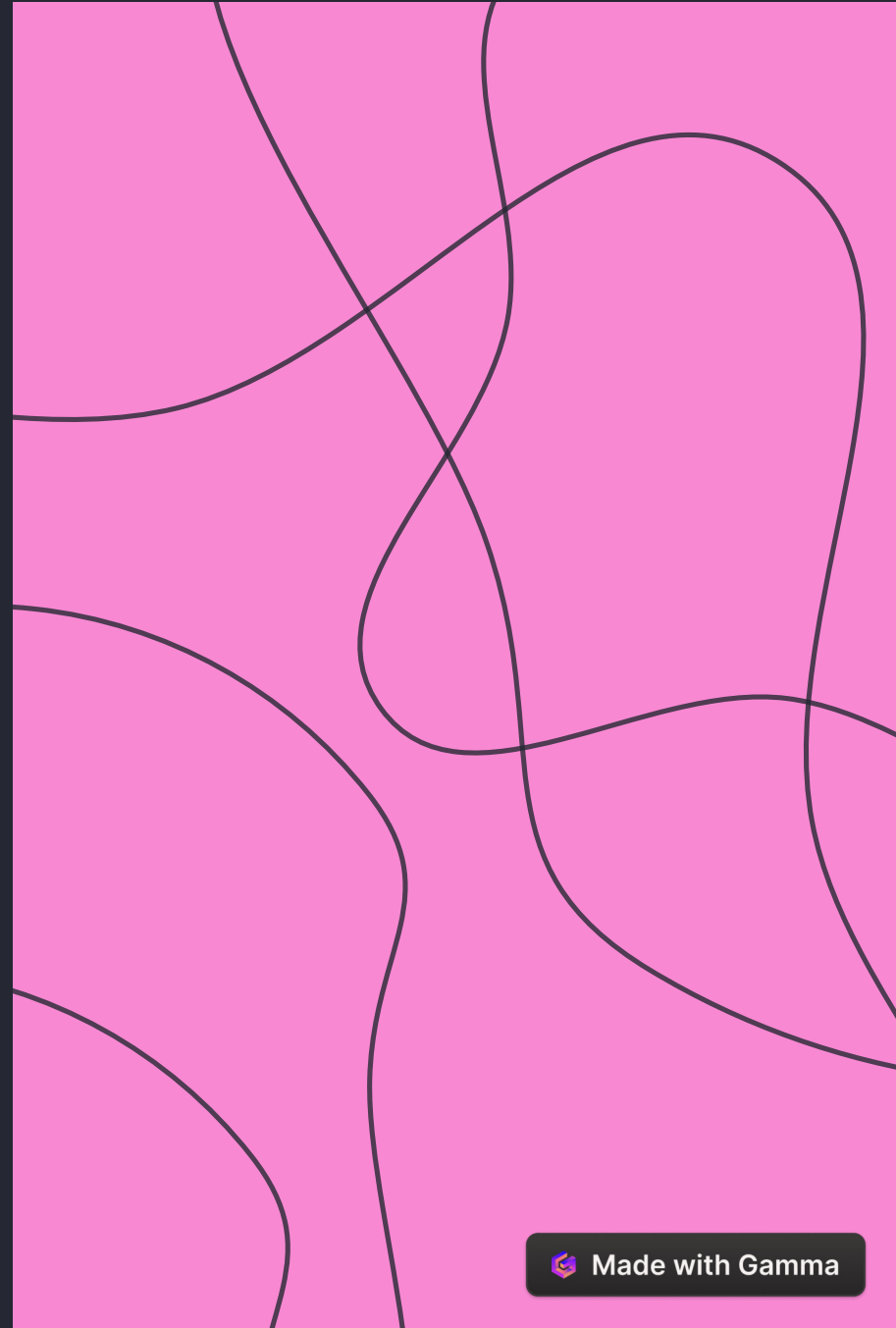
Familiarize yourself with installation processes on different operating systems.

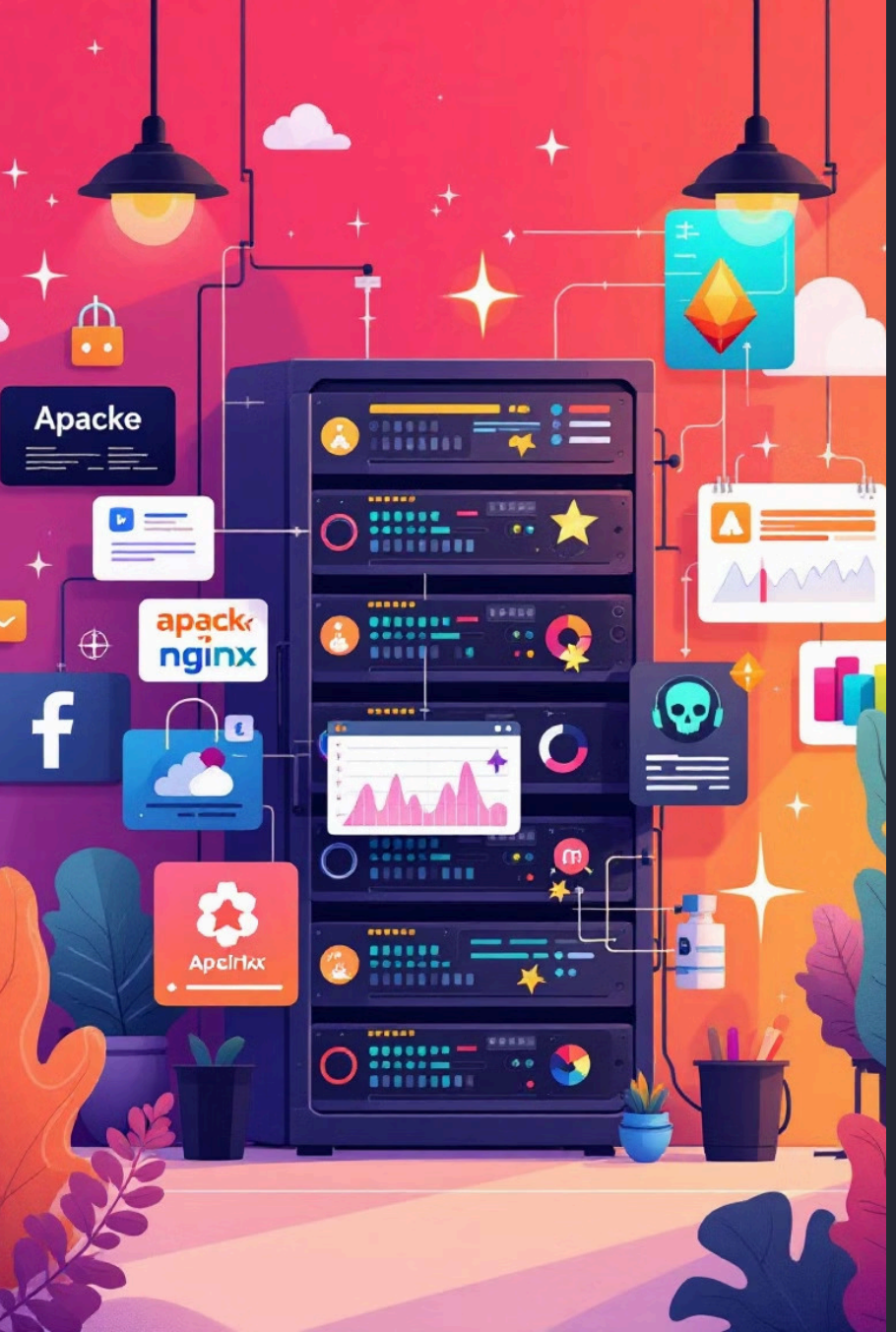
3 Utilize Management Tools

Learn to use utilities specific to Apache and Nginx for efficient server management.

4 Implement Monitoring Practices

Regularly monitor logs and performance to ensure optimal server operation.





Tools and Utilities for Managing Web Servers

This presentation provides a detailed overview of essential tools and utilities for managing web servers, focusing on Apache and Nginx, as well as various monitoring and performance testing tools.

DK **by Dan K**

Apache Utilities

apachectl

Controls Apache services.

htpasswd

Manages basic HTTP authentication.

Apache Control Tool (apachectl)

1

apachectl start

Starts the Apache server

2

apachectl stop

Stops the Apache server

3

apachectl restart

Restarts the Apache server

4

apachectl graceful

Restarts without disconnecting current clients

5

apachectl configtest

Tests the configuration file syntax

Useful for quickly applying changes or troubleshooting Apache configurations.

htpasswd Utility

Creating a New User

```
htpasswd -c /etc/apache2/.htpasswd username
```

Adding/Updating a User

```
htpasswd /etc/apache2/.htpasswd username
```

Protect directories or files using .htpasswd with the .htaccess file.

mod_status Module

Real-time Monitoring

A module to monitor Apache activity and performance in real-time.

Access URL

http://your-server-ip/server-status

Configuration

Requires enabling the module and configuring access control.

Check Module Status

Run: apachectl -M | grep status

Nginx-Specific Utilities

nginx -t

Tests Nginx configuration files for syntax errors before reloading or restarting.

nginx -s reload

Reloads the configuration without downtime.

Example: sudo nginx -t

Helps avoid downtime caused by faulty configurations.

Nginx Control Commands

1. nginx -s stop

   Stops the server

2. nginx -s reload

   Reloads the configuration without downtime

3. nginx -s quit

   Gracefully shuts down the server

Log Directories:

- Access logs: /var/log/nginx/access.log
- Error logs: /var/log/nginx/error.log

Command for monitoring: tail -f /var/log/nginx/access.log

Log Monitoring Tools

tail, grep, awk

- tail: Monitors logs in real-time.
- grep: Searches for specific patterns in logs.
- awk: Processes and formats log data.

GoAccess

Real-time log analyzer for web servers.

- Visualizes data in a terminal or web interface.
- Provides metrics like visitor counts, popular URLs, and HTTP status codes.

Performance Testing Tools

ApacheBench (ab)

A command-line tool to benchmark web servers.

Example: ab -n 100 -c 10 http://your-server-ip/

Tests 100 requests with 10 concurrent users.

wrk

A modern and powerful load testing tool for web servers.

Example: wrk -t12 -c400 -d30s http://your-server-ip/

Simulates high traffic with 12 threads, 400 connections, for 30 seconds.

Configuration Editors and SSL

vim and nano

Text editors for modifying configuration files.

Example:

```
sudo vim /etc/apache2/apache2.conf
sudo nano /etc/nginx/nginx.conf
```

Certbot

Automates the installation of SSL certificates from Let's Encrypt.