How to host website with AWS for LAMP, LEMP, MEAN, MERN Stacks?

Introduction

- **Overview of cloud hosting with AWS**

Amazon Web Services (AWS) is a leading cloud platform offering flexible, scalable, and reliable infrastructure for hosting modern web applications. AWS allows you to quickly deploy virtual servers, manage databases, configure secure networks, and leverage managed services, making it suitable for any technology stack.

- **What are LAMP, LEMP, MEAN, MERN stacks?**

Stack	Components	Uses
LAMP	Linux, Apache, MySQL, PHP	Dynamic websites, CMS, PHP apps
LEMP	Linux, Nginx, MySQL/MariaDB, PHP	Dynamic sites with high performance
MEAN	MongoDB, Express.js, Angular, Node.js	Single-page apps, APIs, JavaScript apps
MERN	MongoDB, Express.js, React, Node.js	Single-page apps, modern JS front-ends

When to Choose LAMP vs LEMP

Choose LAMP when:

- Building traditional web applications or CMS sites
- Working with WordPress, Drupal, or similar platforms
- Team has extensive Apache experience
- Development or learning environment

Choose LEMP when:

- Expecting high traffic volumes
- Building API-heavy applications

- Optimizing for performance and resource efficiency
- Implementing micro services architecture
- Need advanced load balancing features

Prerequisites

- **AWS account creation and setup.**

Sign up at AWS and verify your email and billing information.

Set up multi-factor authentication (MFA) for account security.

Create and configure an AWS IAM user with administrator permissions for deployments.

- **Basic knowledge requirements.**

Understanding of SSH for server access.

Basic concepts of Virtual Private Cloud (VPC).

Fundamentals of web servers and networking (domain names, HTTP ports).

Core Linux commands.

Step-by-Step Deployment Guide

Separate detailed sections for each stack (LAMP, LEMP, MEAN, MERN)

⚙️ LAMP Stack Setup – Amazon Linux LAMP.sh

```
# Update and upgrade system packages
sudo yum update -y
sudo yum upgrade -y
# Install Apache HTTP Server, In Amazon Linux Apache2 Server's Name Is "httpd"
sudo yum install httpd -y
# Install MariaDB Server(MySql)
sudo yum install mariadb105-server -y
# Install PHP
sudo yum install php -y
```

```
# Start services
sudo service httpd start
sudo service mariadb start
sudo service php-fpm start

# Navigate to Apache web root
cd /var/www/html

# Create test HTML and PHP files
sudo touch index.html index.php
echo "Welcome to LAMP Server" | sudo tee index.html
echo "<?php phpinfo(); ?>" | sudo tee index.php
```

Start Service Status With Efficient Way, Start services with feedback.

```
sudo service httpd start    && echo "httpd started"    || echo "httpd not started"
sudo service mariadb start  && echo "mariadb started"  || echo "mariadb not started"
sudo service php-fpm start  && echo "php-fpm started"  || echo "php-fpm not started"
```

⚙️ LEMP Stack Setup – Amazon Linux LEMP.sh

```
# Update system
sudo yum update -y
sudo yum upgrade -y
# Install Nginx Web Server
sudo yum install nginx -y
# Install MariaDB
sudo yum install mariadb105-server -y
# Install PHP and PHP-FPM
sudo yum install php php-fpm -y
# Start all core services
sudo service nginx start
sudo service mariadb start
sudo service php-fpm start
# Navigate to Nginx default web directory
cd /usr/share/nginx/html
# Remove default Nginx welcome page
sudo rm -f index.html
# Create new test HTML and PHP files
sudo touch index.html index.php
# Add content to test files
```

```
echo "LEMP stack is running on Amazon Linux!" | sudo tee index.html
echo "<?php phpinfo(); ?>" | sudo tee index.php
```

⚙️ LAMP Stack Setup – Ubuntu LAMP.sh

```
# Update and upgrade the system

sudo apt-get update -y
sudo apt-get upgrade -y

# Install Apache2

sudo apt-get install apache2 -y

# Install MySQL Server

sudo apt-get install mysql-server -y

# Install PHP 8.3 CLI and PHP-FPM

sudo apt-get install php8.3 -y
sudo apt-get install php8.3-fpm -y

# Enable necessary Apache modules for PHP-FPM integration , Restart Apache to apply changes

sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php8.3-fpm

sudo systemctl restart apache2

# Start PHP-FPM and MySQL services

sudo systemctl start php8.3-fpm
sudo systemctl start mysql

# Optional: Enable services to start at boot

sudo systemctl enable apache2
sudo systemctl enable php8.3-fpm
sudo systemctl enable mysql

# Navigate to Apache web root

cd /var/www/html || { echo "Web root not found"; exit 1; }

# Remove default index page (if exists)

sudo rm -f index.html

# Create test web pages

sudo touch index.html index.php
echo "Apache + PHP8.3 FPM LAMP server is running!" | sudo tee index.html
echo "<?php phpinfo(); ?>" | sudo tee index.php

# Test web server (optional)

curl http://localhost
curl
http://localhost/index.php
```

⚙️ LEMP Stack Setup – Ubuntu LEMP.sh

```
# Update and upgrade packages
sudo apt-get update -y
sudo apt-get upgrade -y
# Install Nginx
sudo apt-get install nginx -y
# Install MySQL Server
sudo apt-get install mysql-server -y
# Install PHP-FPM (version 8.3)
sudo apt-get install php8.3-fpm -y
# Start core services
sudo systemctl start nginx
sudo systemctl start mysql
sudo systemctl start php8.3-fpm
# Enable services to start at boot
sudo systemctl enable nginx
sudo systemctl enable mysql
sudo systemctl enable php8.3-fpm
# Move to Nginx web root
cd /var/www/html
# Remove default index if exists
sudo rm -f index.html
# Create fresh test files
sudo touch index.html index.php
echo "LEMP on Ubuntu is live!" | sudo tee index.html
echo "<?php phpinfo(); ?>" | sudo tee index.php
```

⚠️ Enable PHP in Nginx

By default, Nginx does **not** process PHP. To enable PHP:

1. Open the Nginx default config file:

```
bashsudo nano /etc/nginx/sites-enabled/default
```

2. Locate and uncomment this block:

```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;

# IMPORTANT: Update PHP-FPM socket based on your installed version
    fastcgi_pass unix:/run/php/php8.3-fpm.sock;
# Change php version
```

```
    # Alternative for TCP (not recommended for local):
    # fastcgi_pass 127.0.0.1:9000;

}
```

3. Save and exit the file.

🔁 Restart Services to Apply Changes:

```
sudo systemctl restart nginx
sudo systemctl restart php8.3-fpm
```

✅ Final Verification

You can verify it's working by visiting:

```
texthttp://<your-EC2-public-IP>/index.php
```

Or test from the CLI:

```
curl http://localhost
curl http://localhost/index.php
```

- Secure access using AWS Security Groups and configure port access (22, 80,/443).