Setting Up a Web Server on Ubuntu Linux

May 15, 2025

Contents

1 Introduction

This guide provides a step-by-step process to set up a web server on Ubuntu Linux, host a website, configure phpMyAdmin for database management, and set up a domain name for public access. The setup uses a LAMP stack (Linux, Apache, MySQL, PHP) and includes security configurations.

2 Step 1: Set Up Ubuntu Server

- **Install Ubuntu Server**: Download Ubuntu Server 22.04 or 24.04 from https://ubuntu.com/download/server. Install on your hardware or virtual machine, setting up a user account and enabling OpenSSH for remote access.
- **Update System**:

```
sudo apt update
sudo apt upgrade -y
```

- **Set Static IP**: Edit `/etc/netplan/01-netcfg.yaml` to configure a static IP (e.g., 192.168.1.100). Example:

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.100/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

  Apply with:

```
sudo netplan apply
```

3 Step 2: Install LAMP Stack

3.1 Apache Web Server

- Install Apache:

```
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
```

- Test by visiting `http://<server-ip>` in a browser.

3.2 MySQL Database

- Install MySQL:

```
sudo apt install mysql-server -y
sudo mysql_secure_installation
```

- Set root password and secure the installation.

3.3 PHP

- Install PHP:

```
sudo apt install php libapache2-mod-php php-mysql -y
```

- Test with a PHP file:

```
<?php
phpinfo();
?>
```

  Save as `/var/www/html/info.php` and visit `http://<server-ip>/info.php`.

4 Step 3: Install and Configure phpMyAdmin

- Install phpMyAdmin:

```
sudo apt install phpmyadmin -y
```

  Select Apache2 and configure the database during installation.

- Enable configuration:

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
sudo a2enconf phpmyadmin
sudo systemctl reload apache2
```

- Access at `http://<server-ip>/phpmyadmin`. Create a new MySQL user:

```
CREATE USER 'phpmyadminuser'@'localhost' IDENTIFIED BY 'yourpassword';
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadminuser'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

- Secure with basic authentication:

```
AuthType Basic
AuthName "Restricted Access"
AuthUserFile /etc/phpmyadmin/.htpasswd
Require valid-user
```

  Add to `/etc/apache2/conf-available/phpmyadmin.conf` and create a password file:

```
sudo htpasswd -c /etc/phpmyadmin/.htpasswd admin
```

5 Step 4: Host Your Website

- Create website directory and index file:

```
sudo mkdir /var/www/yoursite
sudo nano /var/www/yoursite/index.html
```

Example index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>My Website</title>
</head>
<body>
    <h1>Welcome to My Website!</h1>
</body>
</html>
```

- Configure Apache virtual host:

```
<VirtualHost *:80>
    ServerName yoursite.com
    ServerAlias www.yoursite.com
    DocumentRoot /var/www/yoursite
    <Directory /var/www/yoursite>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/yoursite_error.log
    CustomLog ${APACHE_LOG_DIR}/yoursite_access.log combined
</VirtualHost>
```

Save as /etc/apache2/sites-available/yoursite.conf, then:

```
sudo a2ensite yoursite.conf
sudo systemctl reload apache2
```

- Set permissions:

```
sudo chown -R www-data:www-data /var/www/yoursite
sudo chmod -R 755 /var/www/yoursite
```

6 Step 5: Configure a Domain Name

- Purchase a domain (e.g., from Namecheap).
- Set DNS A record: Point yoursite.com to your servers public IP (find with curl ifconfig.me).
- (Optional) Use Dynamic DNS (e.g., No-IP) for dynamic IPs:

```
sudo apt install ddclient -y
```

- Update `yoursite.conf` with your domain and reload Apache.

7 Step 6: Secure with HTTPS

- Install Certbot:

```
sudo apt install certbot python3-certbot-apache -y
```

- Obtain SSL certificate:

```
sudo certbot --apache -d yoursite.com -d www.yoursite.com
```

- Verify at `https://yoursite.com`.

8 Step 7: Make Website Public

- **Port Forwarding**: Forward ports 80 and 443 to your servers IP (e.g., 192.168.1.100) in your router.
- **Firewall**:

```
sudo ufw allow 80
sudo ufw allow 443
sudo ufw enable
```

- Test public access via `https://yoursite.com` from an external network.

9 Step 8: Maintenance

- **Backups**:

```
tar -czvf /backup/yoursite_backup_$(date +%F).tar.gz /var/www/yoursite
mysqldump -u root -p --all-databases > /backup/mysql_backup_$(date +%F).sql
```

- **Monitoring**: Use `htop` or check logs:

```
sudo tail -f /var/log/apache2/error.log
```

- **Updates**:

```
sudo apt update && sudo apt upgrade -y
sudo certbot renew --dry-run
```

10 Optional: WordPress Setup

- Create a database:

```
CREATE DATABASE wordpress;
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost' IDENTIFIED BY 'wppassword';
```

```
FLUSH PRIVILEGES;
```

- Install WordPress:

```
wget https://wordpress.org/latest.tar.gz
tar -xzvf latest.tar.gz
sudo mv wordpress /var/www/yoursite
sudo chown -R www-data:www-data /var/www/yoursite
```

- Configure at `http://yoursite.com/wp-admin`.

11 Troubleshooting

- Website not loading: Check Apache status (`sudo systemctl status apache2`) or logs (`/var/log/apache2/error.log`).
- phpMyAdmin issues: Verify MySQL credentials and `/etc/phpmyadmin/apache.conf`.
- Domain not resolving: Check DNS with `dig yoursite.com`.
- Port forwarding issues: Confirm public IP (`curl ifconfig.me`) and router settings.