Complete Ubuntu Server Setup Guide for Beginners

📋 Step 1: Update Your Ubuntu System

First, always update your system packages:

Command:

sudo apt update && sudo apt upgrade -y

What this does: Updates package lists and upgrades all installed packages to latest versions.

🔒 Step 2: Configure UFW Firewall

2.1 Enable UFW (if not already enabled)

Command:

sudo ufw enable

Type 'y' when prompted to confirm

2.2 Allow SSH (Port 22) - IMPORTANT!

Commands:

sudo ufw allow ssh

OR

sudo ufw allow 22

⚠️ **WARNING:** Always allow SSH before enabling UFW, or you might lock yourself out!

2.3 Check UFW status

Command:

sudo ufw status

Expected output:

Status: active

To Action From

-- ------ ----

22/tcp ALLOW Anywhere

22/tcp (v6) ALLOW Anywhere (v6)

2.4 Reload UFW to apply changes

Command:

sudo ufw reload

🌐 Step 3: Install and Configure Nginx

3.1 Install Nginx

Command:

sudo apt install nginx -y

3.2 Start Nginx and enable it to start on boot

Commands:

sudo systemctl start nginx

sudo systemctl enable nginx

3.3 Check if Nginx is running

Command:

sudo systemctl status nginx

3.4 Allow Nginx through firewall

Command:

sudo ufw allow 'Nginx Full'

3.5 Verify firewall rules

Command:

sudo ufw status

Expected output should now include:

Nginx Full ALLOW Anywhere

📦 Step 4: Install Git

4.1 Install Git

Command:

sudo apt install git -y

4.2 Verify Git installation

Command:

git --version

🧪 Step 5: Test Nginx Default Page

Open your web browser and go to:

- `http://your-server-ip`
- If testing locally: `http://localhost`

You should see the Nginx welcome page.

🚀 Step 6: Deploy Your Static Website

6.1 Navigate to Nginx web directory

Command:

cd /var/www/html

6.2 Remove default Nginx page

Command:

sudo rm index.nginx-debian.html

6.3 Clone your static website from Git

Command:

sudo git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY.git .

Replace with your actual repository URL. Example:

sudo git clone https://github.com/john/my-website.git .

Alternative method - if you want to clone into a subdirectory: Commands:

sudo git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY.git website

sudo cp -r website/* .

sudo rm -rf website

6.4 Set proper permissions

Commands:

sudo chown -R www-data:www-data /var/www/html

sudo chmod -R 755 /var/www/html

📁 Step 7: Important Nginx File Locations & Commands

7.1 Key Nginx Directories and Files:

- 📂 **Web files location:** `/var/www/html/`
- ⚙️ **Nginx config:** `/etc/nginx/nginx.conf`
- 📄 **Site configs:** `/etc/nginx/sites-available/`
- ✅ **Enabled sites:** `/etc/nginx/sites-enabled/`
- 📝 **Log files:** `/var/log/nginx/`

7.2 Essential Nginx Commands:

Start Nginx:

sudo systemctl start nginx

Stop Nginx:

sudo systemctl stop nginx

Restart Nginx:

sudo systemctl restart nginx

Reload Nginx (apply config changes without stopping):

sudo systemctl reload nginx

Check Nginx status:

sudo systemctl status nginx

Test Nginx configuration:

sudo nginx -t

View error logs:

sudo tail -f /var/log/nginx/error.log

View access logs:

sudo tail -f /var/log/nginx/access.log

⚙️ Step 8: Create a Simple Custom Nginx Configuration (Optional)

8.1 Create new site config

Command:

sudo nano /etc/nginx/sites-available/mywebsite

8.2 Add this basic configuration:

```
server {
    listen 80;
```

```
    server_name your-domain.com www.your-domain.com;

    root /var/www/html;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }

    # Optional: Security headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
}
```

8.3 Enable the site

Command:

```
sudo ln -s /etc/nginx/sites-available/mywebsite /etc/nginx/sites-enabled/
```

8.4 Test and reload

Commands:

```
sudo nginx -t
sudo systemctl reload nginx
```

🔄 Step 9: Update Your Website

9.1 To update your website with new changes from Git:

Commands:

cd /var/www/html

sudo git pull origin main

sudo systemctl reload nginx

🛠️ Step 10: Troubleshooting Common Issues

10.1 If website doesn't load:

Check if Nginx is running:

sudo systemctl status nginx

Check firewall:

sudo ufw status

Check error logs:

sudo tail -f /var/log/nginx/error.log

10.2 If you get permission errors:

Commands:

sudo chown -R www-data:www-data /var/www/html

sudo chmod -R 755 /var/www/html

10.3 If SSH connection fails:

Ensure UFW allows SSH:

sudo ufw allow ssh

Check SSH service:

sudo systemctl status ssh

Summary of What We Accomplished

✅ Updated Ubuntu system
✅ Configured UFW firewall with SSH access
✅ Installed and configured Nginx web server
✅ Installed Git
✅ Deployed static website from Git repository
✅ Set proper file permissions
✅ Learned essential Nginx commands and file locations

Your static website should now be accessible via your server's IP address or domain name!