NGINX Server Setup & Configuration on Ubuntu

NGINX is a high-performance web server and reverse proxy used for hosting websites, load balancing, and handling HTTP/HTTPS traffic efficiently. Below is a step-by-step guide to installing and configuring NGINX on an Ubuntu system.

---

1 Update System Packages

Before installing any software, update the package list to ensure you get the latest versions:

```
bash
CopyEdit
sudo apt update && sudo apt upgrade -y
```

---

2 Install NGINX

Install NGINX using the official Ubuntu package manager:

```
bash
CopyEdit
sudo apt install nginx -y
```

After installation, check if NGINX is running:

```
bash
CopyEdit
sudo systemctl status nginx
```

To start or restart the service manually:

```
bash
CopyEdit
sudo systemctl start nginx    # Start NGINX
sudo systemctl enable nginx   # Enable auto-start on boot
sudo systemctl restart nginx  # Restart after changes
```

---

3 Adjust Firewall Settings

Ensure your firewall allows HTTP (port 80) and HTTPS (port 443) traffic. If using `ufw` (Uncomplicated Firewall), run:

```
bash
CopyEdit
sudo ufw allow 'Nginx Full'
sudo ufw enable
sudo ufw status
```

---

4 Verify Installation

Check if NGINX is successfully installed by opening a web browser and entering:

```
arduino
CopyEdit
http://your-server-ip
```

You should see the **NGINX default welcome page**.

Alternatively, check via command line:

```
bash
CopyEdit
curl -I http://localhost
```

This should return an HTTP status code `200 OK`, confirming the server is running.

---

5 Manage NGINX Configuration

NGINX configuration files are located in:

- **Main configuration file:** `/etc/nginx/nginx.conf`
- **Site configuration files:** `/etc/nginx/sites-available/`
- **Enabled site symlinks:** `/etc/nginx/sites-enabled/`

To edit the main NGINX configuration file:

```
bash
CopyEdit
sudo nano /etc/nginx/nginx.conf
```

After making changes, check for syntax errors before restarting:

```
bash
CopyEdit
```

```
sudo nginx -t
sudo systemctl restart nginx
```

---

6 Set Up a Basic Website (Virtual Host)

1. **Create a new configuration file** for your domain:

   ```
   bash
   CopyEdit
   sudo nano /etc/nginx/sites-available/mywebsite
   ```

2. **Add the following configuration:**

   ```
   nginx
   CopyEdit
   server {
       listen 80;
       server_name mywebsite.com www.mywebsite.com;
       root /var/www/mywebsite;
       index index.html;

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

3. **Save and exit**, then create a symlink to enable the site:

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

4. **Create the website directory and test page:**

   ```
   bash
   CopyEdit
   sudo mkdir -p /var/www/mywebsite
   echo "<h1>Welcome to My Website</h1>" | sudo tee
   /var/www/mywebsite/index.html
   ```

5. **Set proper permissions:**

   ```
   bash
   CopyEdit
   sudo chown -R www-data:www-data /var/www/mywebsite
   sudo chmod -R 755 /var/www/mywebsite
   ```

6. **Restart NGINX to apply changes:**

```
bash
CopyEdit
sudo systemctl restart nginx
```

Now, visit `http://mywebsite.com` in a browser, and you should see your custom webpage.

---

7 Enable HTTPS with a Free SSL Certificate (Let's Encrypt)

For secure HTTPS, install **Certbot** and request an SSL certificate:

```
bash
CopyEdit
sudo apt install certbot python3-certbot-nginx -y
```

Run Certbot to automatically configure SSL for your domain:

```
bash
CopyEdit
sudo certbot --nginx -d mywebsite.com -d www.mywebsite.com
```

Follow the prompts to install the certificate. Certbot also sets up automatic renewal.

To manually test renewal:

```
bash
CopyEdit
sudo certbot renew --dry-run
```

---

8 Monitor & Manage NGINX

Check Server Logs

View access logs:

```
bash
CopyEdit
sudo tail -f /var/log/nginx/access.log
```

View error logs:

```
bash
CopyEdit
sudo tail -f /var/log/nginx/error.log
```

Restart & Reload Configuration

After making changes, restart or reload NGINX:

```
bash
CopyEdit
sudo systemctl restart nginx    # Full restart
sudo systemctl reload nginx     # Reload without downtime
```

Conclusion

By following these steps, you can successfully **install, configure, and secure NGINX on Ubuntu**. It serves as a powerful and efficient web server for hosting websites and handling requests effectively