DevOps Shack

NGINX

3 REAL WORLD PROJECTS

By DevOps Shack

Click here for DevSecOps Course

Nginx Projects

Install Nginx On Ubuntu

sudo apt update

sudo apt install nginx -y

Project-1 | Serving Static Website Using Nginx

Project Repo: https://github.com/jaiswaladi246/static-site.git

Step 1: Place Files in Web Root

By default, Ubuntu NGINX serves from:

/var/www/html

You can replace or create your own site directory:

sudo mkdir -p /var/www/static-site

sudo cp -r * /var/www/static-site

Ensure permissions:

sudo chown -R www-data:www-data /var/www/static-site

Step 2: Create a Virtual Host File

sudo vi /etc/nginx/sites-available/static-site

```
server {
    listen 80;
    server_name skjptpp.in;

    root /var/www/static-site;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Breakdown:

Directive	Role
listen 80	Binds to port 80 (HTTP)
server_name	Domains this block will respond to
root	Document root directory
index	Default file served when / is requested
try_files	Attempts to resolve file path, else returns 404

Step 3: Enable Site and Reload

sudo ln -s /etc/nginx/sites-available/static-site /etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl reload nginx

Visit http://mywebsite.com (after DNS setup) — the static site is live!

🔸 2. Creating Custom Error Pages (404, 500)

❌ Default Behavior

NGINX uses built-in generic messages like:

404 Not Found

500 Internal Server Error

These are plain and unfriendly.

✅ Creating Custom Pages

1. Create error pages:

```
/var/www/mywebsite/
 ├──── 404.html
 ├──── 500.html
```

2. Update the server block:

```
server {
   ...
   error_page 404 /404.html;
   error_page 500 502 503 504 /500.html;

   location = /404.html {
```

```
        root /var/www/mywebsite;
        internal;
    }

    location = /500.html {
        root /var/www/mywebsite;
        internal;
    }
}
```

🧠 Explanation:

- error_page: Maps HTTP status to custom page.
- location =: Exact match.
- internal: These URLs can only be shown by NGINX errors — not accessed directly by users.

Project-2 | Nginx As a Reverse Proxy

Project Repo: https://github.com/jaiswaladi246/nginx-node-proxy.git

git clone https://github.com/jaiswaladi246/nginx-node-proxy.git

- sudo mkdir -p /var/www/frontend
- sudo cp -r frontend/* /var/www/frontend/
- sudo chown -R www-data:www-data /var/www/frontend

sudo vi /etc/nginx/sites-available/nginx-node-proxy

```
server {
    listen 80;
    server_name skjptpp.in;  # Use _ or your IP if no domain yet

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

    # Serve static files
    location / {
        try_files $uri $uri/ =404;
    }

    # Reverse proxy to Node.js API
    location /api/ {
        proxy_pass http://13.201.37.133:3000;
        proxy_http_version 1.1;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    # WebSocket support
    location /socket.io/ {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
    }
}
```

Goto Backend Folder --> Install Dependencies & start the Backend

```
cd /home/ubuntu/nginx-node-proxy/backend
npm init -y
npm install express cors socket.io
node index.js
```

Create Symbolic Link to Sites Enabled & Restart Nginx

```
sudo ln -s /etc/nginx/sites-available/nginx-node-proxy /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

Access on Your Domain like skjptpp.in

Project-3 | Nginx As Load Balancer

Project Repo: https://github.com/jaiswaladi246/nginx-loadbalancer.git

git clone https://github.com/jaiswaladi246/nginx-loadbalancer.git

- sudo mkdir -p /var/www/frontend
- sudo cp -r frontend/* /var/www/frontend/
- sudo chown -R www-data:www-data /var/www/frontend

sudo vi /etc/nginx/sites-available/nginx-loadbalancer

```
upstream backend_apis {
    least_conn;
    server 13.201.37.133:3001;
    server 13.201.37.133:3002;
}

server {
    listen 80;
    server_name skjptpp.in;

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

    # Serve frontend (HTML, CSS, JS, etc.)
    location / {
        try_files $uri $uri/ =404;
    }

    # Proxy API requests to backend Node.js apps
    location /api/ {
        proxy_pass http://backend_apis;
        proxy_http_version 1.1;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Goto Backend Folder --> Install Dependencies & start the Backend

```
cd /home/ubuntu/nginx-loadbalancer/
```

```
cd backend1 && npm init -y && npm install express   #in new Page
cd backend2 && npm init -y && npm install express   #in new Page
```

```
node index.js
```

Create Symbolic Link to Sites Enabled & Restart Nginx

```
sudo ln -s /etc/nginx/sites-available/nginx-loadbalancer /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

To simulate Traffic:

```
sudo apt install apache2-utils
ab -n 100 -c 10 http://skjptpp.in/api/
```