



Expert Cloud Consulting

Enhance Optimise & Scale

ASCP GPUonCLOUD Pvt Ltd

“Expert Cloud Consulting” -

Documentation For Basic Nginx Server Setup with Custom 404 and Firewall Rules [Title,18, Arial]

14.April.2025 [Subtitle,14, Arial]

version 1.0

—

Contributed by Yogiraj Deshpande [Normal text,14, Arial]

Approved by Akshay Shinde(In Review)

Expert Cloud Consulting

Office #811, Gera Imperium Rise,

Hinjewadi Phase-II Rd, Pune, India – 411057

“Expert Cloud Consulting”

Documentation For Basic Nginx Server Setup with Custom 404 and Firewall Rules

1.0: Document Overview:

This document outlines how to configure a Linux-based virtual machine (VM) to serve a static Nginx welcome page, restrict access through firewall rules to only HTTP/HTTPS traffic, and display a custom 404 error page for missing content.

2.0: Objective:

- Set up Nginx to serve its default welcome page.
- Allow only HTTP and HTTPS traffic to enhance security.
- Customize the 404 error page for user-friendliness and branding.

3.0: Prerequisites:

The following practices/tools are required for this workflow:

- An AWS EC2 Linux instance (Amazon Linux 2 or Ubuntu).
- A security group with ports 22, 80, and 443 open (SSH optional for setup).
- SSH key pair (`.pem`) for initial instance access.
- Basic command line knowledge for server configuration.



4.0: Provision a Linux VM:

- Launch an EC2 instance (Amazon Linux 2 recommended).
- Attach a public IP for access.
- Assign a security group that initially allows:
 - Port 22 (SSH) — temporarily for configuration
 - Port 80 (HTTP)
 - Port 443 (HTTPS)

5.0: Steps:

5.1 Install Nginx

```
# For Amazon Linux / RHEL-based
sudo yum update -y
sudo yum install nginx -y

# For Ubuntu/Debian-based
# sudo apt update && sudo apt install nginx -y

# Start and enable Nginx
sudo systemctl start nginx
sudo systemctl enable nginx
```

- **Why this step?**
Installs and activates Nginx so it can handle HTTP requests immediately and automatically after reboots.

5.2 Create a Custom 404 Error Page:

```
echo "<h1>Oops! Page Not Found (404)</h1>" | sudo tee
/usr/share/nginx/html/404.html
```

- **Why this step?**
Improves user experience and avoids a plain error screen when users access invalid URLs.

5.3 Configure Nginx to Use the 404 Page:

Edit the main default config file:

```
sudo vim /etc/nginx/nginx.conf
```

Paste this and Make sure it includes this inside the **http** block or a **server** block::

```
server {  
    listen 80;  
    server_name _;  
  
    root /usr/share/nginx/html;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    error_page 404 /404.html;  
    location = /404.html {  
        internal;  
    }  
}
```

- **Why this step?**

This ensures that requests to missing resources trigger the custom 404 page.

5.4 Restart Nginx:

```
sudo nginx -t # Tests Nginx config syntax  
sudo systemctl restart nginx # Applies the new configuration  
sudo systemctl enable nginx # Applies the new configuration
```

- **Why this step?**

These commands ensure Nginx is configured correctly and will keep running after reboots.

6.0: Validation:

- Access: http://<EC2_PUBLIC_IP> → You should see the Nginx welcome page
- Try a missing path: http://<EC2_PUBLIC_IP>/xyz → You should see your custom 404 page