

Custom Nginx Docker Image Deployment Guide

Step 1: Prepare the Project Directory

Create a directory structure like this:

```
custom-nginx/  
├── Dockerfile  
├── default.conf  
└── html/  
    └── index.html
```

Step 2: Create Dockerfile

Dockerfile content:

```
FROM nginx:latest  
  
RUN rm /etc/nginx/conf.d/default.conf  
COPY default.conf /etc/nginx/conf.d/  
COPY html/ /usr/share/nginx/html/  
EXPOSE 80
```

Step 3: Create default.conf

default.conf content:

```
server {  
    listen 80;  
    server_name localhost;
```

```
location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}
}
```

Step 4: Create index.html

html/index.html content:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>Welcome to Custom Nginx!</title>
<style>
  body {
    margin: 0;
    padding: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(to right, #74ebd5, #9face6);
    color: #333;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    height: 100vh;
    text-align: center;
  }

  .container {
    background-color: #fff;
```

```

padding: 40px;
border-radius: 15px;
box-shadow: 0 10px 30px rgba(0,0,0,0.1);
max-width: 600px;
width: 90%;
animation: fadeIn 1s ease-in-out;
}

h1 {
font-size: 2.5rem;
color: #4A00E0;
}

p {
font-size: 1.2rem;
margin: 20px 0;
}

button {
background-color: #4A00E0;
color: white;
border: none;
padding: 12px 24px;
font-size: 1rem;
border-radius: 25px;
cursor: pointer;
transition: background-color 0.3s ease;
}

button:hover {
background-color: #8E2DE2;
}

@keyframes fadeIn {
from { opacity: 0; transform: translateY(30px); }
to { opacity: 1; transform: translateY(0); }
}
</style>
</head>
<body>
<div class="container">
<h1>🚀 Welcome to Your Custom Nginx!</h1>
<p>This containerized Nginx server is running smoothly and

```

```
securely using Docker Compose.</p>
  <button onclick="alert('Deployed with Docker Compose 🚀')">How
was it deployed?</button>
</div>
</body>
</html>
```

Step 5: Create docker-compose.yml

docker-compose.yml content (replace <your-dockerhub-username>):

```
services:
  web:
    image: <your-dockerhub-username>/custom-nginx:latest
    build: .
    ports:
      - "8080:80"
    volumes:
      - ./nginx-data:/var/opt/nginx
```

Step 6: Build and Deploy

1. Navigate to the project directory.
2. Build the image:
docker compose build

```
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker build -t suryapkh/custom-nginx:latest .
[+] Building 5.6s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 280B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load dockerignore
```

3. Run the container:

docker compose up -d

```
=> => naming to docker.io/suryapkh/custom-nginx:latest
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker compose up -d
WARN[0000] /home/ubuntu/docker-compose.yml: the attribute `version` is obsolete
[+] Running 2/2
 ✓ Network ubuntu_default Created
 ✓ Container ubuntu-web-1 Started
```

Step 7: Push to Docker Hub

1. Login to Docker Hub:

`docker login`

```
✓ Container ubuntu-web-1 Started
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker login

USING WEB-BASED LOGIN

Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: STFC-CGTD
Press ENTER to open your browser or submit your device code here: https://login.docker.com/acti

Waiting for authentication in the browser...

WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker tag custom-nginx suryapkh/custom-nginx:latest
```

2. Tag the image (if necessary):

`docker tag custom-nginx <your-dockerhub-username>/custom-nginx:latest`

```
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker tag custom-nginx suryapkh/custom-nginx:latest
Error response from daemon: No such image: custom-nginx:latest
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker push suryapkh/custom-nginx:latest
```

3. Push the image:

`docker push <your-dockerhub-username>/custom-nginx:latest`

```
ubuntu@ip-172-31-91-139:~/custom-nginx$ docker push suryapkh/custom-nginx:latest
The push refers to repository [docker.io/suryapkh/custom-nginx]
f50dec80bd9b: Pushed
835952073bf3: Pushed
3c8c10de6fb5: Pushed
8030dd26ec5d: Mounted from library/nginx
d84233433437: Mounted from library/nginx
f8455d4eb3ff: Mounted from library/nginx
286733b13b0f: Mounted from library/nginx
46a24b5c31d8: Mounted from library/nginx
84accda66bf0: Mounted from library/nginx
6c4c763d22d0: Mounted from library/nginx
latest: digest: sha256:f11d303bd56b1b5f8b3cee2b27fa373038f1b9ef4af786f17e05f59ae7
```

OUTPUT:

