



# Introduction to NGINX with Docker

Learn how to set up and deploy **NGINX** using Docker for web server development and deployment.



## What is NGINX?

- High-performance web server and reverse proxy
- Handles static files, load balancing, and caching
- Commonly used with Docker for containerized web apps



## Why Use NGINX with Docker?

- Easy to configure and deploy
- Isolate environments
- Automate setup and testing
- Integrate with apps (PHP, Node.js, etc.)



## Prerequisites

- Install **Docker** and optionally **Docker Compose**
- Basic knowledge of command line
- Create a working project directory





## Project Structure

Example project folder:

```
project/  
├── index.html  
└── Dockerfile
```

**index.html** is your static website entry point.





## Sample index.html

```
<!DOCTYPE html>
<html>
<head><title>Hello NGINX</title></head>
<body>
  <h1>Welcome to NGINX running in Docker!</h1>
</body>
</html>
```



## Create a Dockerfile

```
FROM nginx:alpine
COPY index.html /usr/share/nginx/html/index.html
```

- Uses a lightweight NGINX image
- Replaces default index.html with your own



## Build and Run

```
docker build -t my-nginx .  
docker run -d -p 8080:80 my-nginx
```

- Access at: <http://localhost:8080>





## Updating Content

Edit `index.html`, then rebuild:

```
docker build -t my-nginx .  
docker restart <container_id>
```

Or use **bind mounts** in `docker run` for live development



## Custom NGINX Config (Optional)

Add your own `nginx.conf` :

```
server {  
    listen 80;  
    location / {  
        root /usr/share/nginx/html;  
        index index.html;  
    }  
}
```

Update Dockerfile:

```
COPY nginx.conf /etc/nginx/conf.d/default.conf
```



## With Docker Compose (Optional)

**docker-compose.yml**

```
version: '3'
services:
  web:
    image: nginx:alpine
    volumes:
      - ./index.html:/usr/share/nginx/html/index.html
    ports:
      - "8080:80"
```

Run with:

```
docker-compose up -d
```



## Deployment Tips

- Use a reverse proxy to serve backend APIs (PHP, Node, etc.)
- Enable HTTPS with Let's Encrypt + Certbot
- Log and monitor traffic with NGINX logs





## Summary

- NGINX is fast and easy to deploy with Docker
- Serves static content or reverse-proxies to apps
- Combine with Docker Compose for larger apps



## You're Ready

Deploy your own NGINX server in minutes using Docker!