



My Personal Docker Dev & Cloud Stack

This document summarizes the installation and setup of my **local cloud and DevOps environment** using Docker.

It includes:

- **Docker** — Container platform
- **Portainer** — Docker management UI
- **Gitea** — Self-hosted Git server
- **MinIO** — Object storage (S3 alternative)
- **Nextcloud** — Personal cloud storage
- **Wiki.js** — Project documentation & knowledge base
- **n8n** — Workflow automation
- **Flowise** — AI app builder
- **MailHog** — Email testing server



Docker Installation

Install Docker on Linux (Debian/Ubuntu example):

```
sudo apt update  
sudo apt install -y docker.io docker-compose  
sudo systemctl enable docker --now  
sudo usermod -aG docker $USER
```

- Test installation: `docker --version` / `docker-compose --version`
- Log out and back in to apply group changes



Portainer (Docker management UI)

Purpose: Manage Docker containers via web interface.

Docker Run Command:

```
docker volume create portainer_data  
  
docker run -d  
-p 9000:9000  
-p 9443:9443  
--name portainer
```

```
--restart=always  
-v /var/run/docker.sock:/var/run/docker.sock  
-v portainer_data:/data  
portainer/portainer-ce:alpine
```

- **Web UI:** <http://localhost:9000> / <https://localhost:9443>
- Create admin user on first visit

Gitea (Self-hosted Git server)

Docker Run Command:

```
mkdir -p /srv/gitea  
  
docker run -d  
--name gitea  
-p 3000:3000  
-p 2222:22  
-v /srv/gitea:/data  
-v /etc/timezone:/etc/timezone:ro  
-v /etc/localtime:/etc/localtime:ro  
--restart always  
gitea/gitea:latest
```

- Web UI: <http://localhost:3000>
- SSH Git: <ssh://git@localhost:2222/username/repo.git>

MinIO (Object Storage)

```
mkdir -p /srv/minio/data  
  
docker run -d  
--name minio  
-p 9001:9000  
-e "MINIO_ROOT_USER=admin"  
-e "MINIO_ROOT_PASSWORD=admin123"  
-v /srv/minio/data:/data  
--restart=always  
minio/minio server /data
```

- Web UI: <http://localhost:9001>

Nextcloud (Personal Cloud)

```
mkdir -p /srv/nextcloud

docker run -d
--name nextcloud
-p 8080:80
-v /srv/nextcloud:/var/www/html
--restart=always
nextcloud:latest
```

- Web UI: <http://localhost:8080>
-

Wiki.js (Documentation)

PostgreSQL Database:

```
mkdir -p /srv/wikijs/db

docker run -d
--name wikijs-db
-e POSTGRES_DB=wiki
-e POSTGRES_USER=wikiuser
-e POSTGRES_PASSWORD=wikipass
-v /srv/wikijs/db:/var/lib/postgresql/data
--restart=always
postgres:13
```

Wiki.js Server:

```
mkdir -p /srv/wikijs/data

docker run -d
--name wikijs
-p 3001:3000
-e DB_TYPE=postgres
-e DB_HOST=wikijs-db
-e DB_PORT=5432
-e DB_USER=wikiuser
-e DB_PASS=wikipass
-e DB_NAME=wiki
```

```
-v /srv/wikijs/data:/wiki/data
--link wikijs-db
--restart=always
requarks/wiki:2
```

- Web UI: <http://localhost:3001>
-

n8n (Automation Workflows)

```
docker run -d
--name n8n
-p 5678:5678
-v /srv/n8n:/home/node/.n8n
--restart=always
n8nio/n8n
```

- Web UI: <http://localhost:5678>
-

Flowise (AI Apps / Chatbots)

```
docker run -d
--name flowise
-p 3002:3000
-v /srv/flowise:/usr/src/app/data
--restart=always
flowiseai/flowise
```

- Web UI: <http://localhost:3002>
 - Connect LLMs, vector DBs, or GPT endpoints
-

MailHog (Email Testing)

```
docker run -d
--name mailhog
-p 8025:8025
-p 1025:1025
mailhog/mailhog
```

- Web UI: <http://localhost:8025>

- SMTP port for testing: 1025
-

Ports Summary

App	Port	Purpose
Portainer	9000	Web UI
Portainer	9443	HTTPS UI
Gitea	3000	Web UI
Gitea	2222	SSH Git
MinIO	9001	Web UI / S3
Nextcloud	8080	Web UI
Wiki.js	3001	Web UI
n8n	5678	Web UI
Flowise	3002	Web UI
MailHog	8025	Web UI
MailHog	1025	SMTP

1 Notes & Tips

- Use persistent volumes for all apps (`/srv/...`) to keep data safe
- Use `--restart=always` for automatic container restarts
- All apps can be combined into a **docker-compose.yml** for easier management
- After setup, you have a **full DevOps + AI + Cloud Lab** locally