

# Auto-hébergeur OpenSign™ avec Docker sur Linux/macOS/Windows

Lien :

- <https://docs.docker.com/engine/install/debian/>
- <https://docs.opensignlabs.com/docs/self-host/docker/run-locally>

## Installer Docker

- créer un conteneur LXC et installer Docker : <https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/docker/installationlinux>
- renseigner les dépôts

```
# Add Docker's official GPG key:
apt update
apt install -y ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF

apt update
```

- Installer les packages Docker

```
apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

## Installer Opensign

- s'authentifier au préalable avec son compte Docker

```
docker login
```

\* exécuter localement OpenSign

```
curl --remote-name-all https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/docker-compose.yml
https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/Caddyfile
https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/.env.local_dev && mv .env.local_dev
.env.prod && docker compose up --force-recreate
```

- télécharger les fichiers **docker-compose.yml**, **Caddyfile** et **.env.local\_dev**
- renommer **.env.local\_dev** en **.env.prod**

## Installer mimio - S3

## Objectif

Avoir un service S3-compatible prêt à l'emploi (local/on-prem), accessible via <https://s3.example.fr> avec console d'admin, comptes IAM (users), buckets, politiques, et sauvegardes.

## Prérequis

- conteneur Debian 13
  - CPU : 2 vCPU (4+ si charge), RAM : 4-8 Go, Disque : 100+ Go selon besoins
  - Réseau : IP fixe + DNS correct
  - Stockage : idéalement un disque dédié (virtio-scsi, cache write-back) pour des performances stables
- Ports à ouvrir (Firewall Proxmox / VM / routeur) :
  - Mono-instance : 9000/tcp (API S3), 9001/tcp (console) — ou proxifiés derrière 80/443
  - Distribué : ajouter 9000-900X pour chaque nœud et ports internes (TCP)
- Nom de domaine (optionnel mais recommandé) :
  - s3.example.fr pour l'endpoint S3
  - console.s3.example.fr (ou /console) pour la console

Astuce : Dans Proxmox, attacher un disque virtuel distinct pour /data MinIO (facile à migrer/monitorer).

```
docker run -p 9000:9000 -p 9001:9001 \  
-v /minio/data:/data \  
-e "MINIO_ROOT_USER=admin" \  
-e "MINIO_ROOT_PASSWORD=motdepasse" \  
quay.io/minio/minio server /data --console-address ":9001"
```

- Console MinIO : <http://IP:9001>
- Endpoint S3 : <http://IP:9000>

From:

[/ - Les cours du BTS SIO](#)

Permanent link:

[/doku.php/systeme/opensign/autohebergement?rev=1770152113](https://doku.php/systeme/opensign/autohebergement?rev=1770152113)

Last update: **2026/02/03 21:55**

