OpenSign - Installation et configuration

---

1. Créer une VM Linux (Ubuntu/Debian) :

1.1. Dans Docker, en lui attribuant un numéro de port et en mappant les ports requis :

```
sudo docker run -it --name opensign-vm \
  -p <port_hôte>:<port_docker> \
  -p 80:80 \
  -p 443:443 \
  -p 3001:3001 \
  ubuntu:24.04
```

Attention! L'adresse IP n'est pas modifiable, elle correspond à celle du conteneur Docker.

1.2. Sur Proxmox :

1.2.1. Installation de la VM :

Cf. Documentation Proxmox

1.2.2. Attribution d'une adresse IP :

Si utilisation de **/etc/network** :

```
sudo nano /etc/network/interfaces
```

Compléter comme suit :

```
iface eth0 inet static
address <ip_address>
netmask <netmask_address>
gateway <gateway_address>
dns-nameservers <dns_servers_addresses>
```

ou si utilisation de **/etc/netplan** (le cas échéant) :

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Compléter comme suit :

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: false
      dhcp6: false
      addresses:
      - <ip_address>/CIDR
      routes:
      - to: default
        via: <gateway_address>
      nameservers:
       addresses: [<dns_servers_addresses>]
```

1.2.3. Ouverture des ports :

```
# Vérifier le statut actuel du pare-feu :
sudo ufw status
# Activer le pare-feu :
sudo ufw enable
# Afficher l'état actuel des règles :
sudo ufw status verbose
# Ouvrir le port 80 en TCP :
sudo ufw allow 80/tcp
# Ouvrir le port 443 en TCP :
sudo ufw allow 443/tcp
# Ouvrir le port 3001 en TCP (Caddy) :
sudo ufw allow 3001/tcp
# Refuser le trafic entrant suivant les règles par défaut :
sudo ufw default deny
```

1.3. Sur un VPS (AWS) :

1.3.1. Installation de la VM :

1. Se connecter à **AWS**
2. Sélectionner **Amazon Lightsail**
3. Sur la page d'accueil de Amazon Lightsail, cliquer sur **Create instance**
4. Dans la fenêtre **Create an instance** :
    1. Modifier, le cas échéant, **Instance location**
    2. Dans le paragraphe **Pick your instance image**
        1. Dans **Select a platform**, cliquer sur **Linux/Unix**
        2. Dans **Select a blueprint**, cliquer sur **Operating System (OS) only**, puis sélectionner l'OS souhaité (Debian ou Ubuntu)

3. Dans le paragraphe **Choose your instance plan** :
    1. Dans **Select a network type**, cliquer sur **Dual-stack** (IPv4 et IPv6)
    2. Dans **Select a size**, cliquer sur le plan adapté aux caractéristiques techniques requises (RAM, CPU, stockage)
    4. Dans le paragraphe **Identify your instance**, inscrire le nom souhaité dans **Instance name**, ici **OpenSign**
5. Pour finaliser l'installation de la VM, cliquer sur **Create Instance**

1.3.2. Attribution d'un adresse IP statique :

Depuis **Amazon Lightsail** :

1. Dans le menu de gauche, cliquer sur **Networking**
2. Cliquer sur **Create static IP**
3. Dans la fenêtre **Create a static IP address** :
    1. Dans le paragraphe **Attach to an instance**, sélectionner **OpenSign**
    2. Dans le paragraphe **Identify your static IP**, inscrire **StaticIp-<nom_instance>**, ici **StaticIp-OpenSign**
    3. Pour finaliser la création de l'adresse IP statique, cliquer sur **Create**
4. Une page de synthèse s'affiche à l'écran, reprenant les informations relatives à l'adresse IP statique publique créer.

1.3.3. Ouverture des ports :

Depuis **Amazon Lightsail** :

1. Dans le menu de gauche, cliquer sur **Instances**
2. Dans la fenêtre **Instances**, cliquer sur **OpenSign**
3. Dans la fenêtre **OpenSign**, cliquer sur l'onglet **Networking** :
    1. Dans le paragraphe **IPv4 Firewall**, cliquer sur **Add rule**
    2. Renseigner comme suit :
Application	Protocol	Port or range
Custom	TCP	80 , 443 , 3001
    3. Cocher la case **Duplicate rule for IPv6**

2. Installer Docker et Compose :

```
sudo apt update && sudo apt install docker.io -y
sudo mkdir -p ~/.docker/cli-plugins/
sudo curl -SL
https://github.com/docker/compose/releases/download/v2.24.5/docker-compose-
linux-x86_64 -o ~/.docker/cli-plugins/docker-compose
```

```
sudo chmod +x ~/.docker/cli-plugins/docker-compose
sudo systemctl enable docker
```

En cas de problème avec Docker Compose non reconnu :

```
sudo apt update
sudo snap install docker
sudo apt install docker-compose
```

3. Configurer le projet et les volumes :

3.1. Créer le répertoire de travail :

```
sudo mkdir opensign && cd opensign
```

3.2. Télécharger les fichiers :

```
export HOST_URL=https://opensign.domain_name.com
sudo curl -O
https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/docker-
compose.yml
sudo curl -O
https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/Caddyfile
sudo curl -O
https://raw.githubusercontent.com/OpenSignLabs/OpenSign/main/.env.local_dev
sudo mv .env.local_dev .env.prod
```

4. Configurer le fichier .env.prod :

Éditer le fichier **.env.prod** :

```
sudo nano .env.prod
```

Configurer les paramètres, comme suit :

```
PUBLIC_URL=https://opensign.<domain_name>.com
HOST_URL=https://opensign.<domain_name>.com
MASTER_KEY=<vault_bitwarden>
SERVER_URL=http://<>server_address:8080/app

# Digital ocean space name or AWS S3 bucket name for uploading documents
DO_SPACE=opensign-<domaine_name>
# Digital ocean spaces endpoint or AWS S3 endpoint for uploading documents
DO_ENDPOINT=https://s3.amazonaws.com
# Digital ocean baseurl or AWS S3 base URL
```

```
DO_BASEURL=https://opensign-<domain_name>.s3.eu-west-3.amazonaws.com
# Digital ocean spaces access key ID or AWS s3 Access key ID for uploading
the docs
DO_ACCESS_KEY_ID=<DO_ACCESS_KEY_ID>
# Digital ocean spaces secret access key or AWS s3 secret access key for
uploading the docs
DO_SECRET_ACCESS_KEY=<DO_SECRET_ACCESS_KEY>
# Digital ocean spaces region or AWS s3 region
DO_REGION=eu-west-3
# local storage
USE_LOCAL=false

#MAILGUN_API_KEY=
#MAILGUN_DOMAIN=
#MAILGUN_SENDER=
SMTP_ENABLE=true
SMTP_HOST=in-v3.mailjet.com
SMTP_PORT=587
SMTP_USER=<Clé_API_MailJet>
SMTP_PASS=<Clé_API_MailJet>   # if your password includes spaces then write
password in single quotes ('asdf pasd asdf bgds').
SMTP_FROM_MAIL=OpenSign
SMTP_FROM_EMAIL=noreply@opensign.fr
```

5. Configurer le fichier docker-compose.yml :

Éditer le fichier **docker-compose.yml** :

```
sudo nano docker-compose.yml
```

Configurer les paramètres comme suit :

```
services:
  server:
    image: opensign/opensignserver:main
    container_name: OpenSignServer-container
    volumes:
      - opensign-files:/usr/src/app/files
    ports:
      - "8080:8080"
    depends_on:
      - mongo
    env_file: .env.prod
    networks:
      - app-network
  mongo:
    image: mongo:latest
    container_name: mongo-container
```

```
    volumes:
      - data-volume:/data/db
    ports:
      - "27018:27017"
    networks:
      - app-network
  client:
    image: opensign/opensign:main
    container_name: OpenSign-container
    depends_on:
      - server
    environment:
      - NEXT_PUBLIC_BACKEND_URL=https://opensign.<domain_name>.com
      - MASTER_KEY=<cf._fichier_.env.prod>
    ports:
      - "3000:3000"
    networks:
      - app-network
  caddy:
    image: caddy:latest
    container_name: caddy-container
    ports:
      - "3001:3001"
      - "80:80"
      - "443:443"
      - "443:443/udp"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
      - caddy_data:/data
      - caddy_config:/config
    networks:
      - app-network
    environment:
      - HOST_URL=opensign.<domain_name>.com
networks:
  app-network:
# Configuration réseau via VM locale :
    driver: bridge
# Configuration réseau via VPS :
    ipam:
      driver: default
      config:
        - subnet: <netmask_address>/CIDR
          gateway: <gateway_address>
volumes:
  data-volume:
  web-root:
  caddy_data:
```

```
  caddy_config:
  opensign-files:
```

6. Démarrer le conteneur avec Compose :

```
sudo docker compose up -d --force-recreate
```

Ou (le cas échéant)

```
sudo docker-compose up -d --force-recreate
```

7. Vérifier l'ensemble :

Éléments vérifiés :

- La connexion à l'adresse https://opensign.domain_name.com charge l'application avec HTTPS ;
- Caddy obtient automatiquement les certificats SSL ;
- Le backend est accessible par /api/* ;
- MongoDB conserve les données ;
- Les fichiers téléchargés persistent dans le volume **opensign-files**.
  Pour tester, exécuter la commande :

```
sudo docker exec -it OpenSignServer-container ls /usr/src/app/files
sudo docker volume inspect opensign-files
```

8. Configurer le redémarrage automatique :

```
sudo crontab -e
```

Ajouter la ligne suivante :

```
@reboot cd /home/<admin_nbame>/opensign && docker compose up -d
```

9. Configurer la sauvegarde des volumes :

```
# Backup MongoDB volume
sudo docker run --rm -v opensign_data-volume:/data -v $(pwd):/backup ubuntu
tar czvf /backup/mongo-backup.tar.gz /data

# Backup OpenSign files
```

```
sudo docker run --rm -v opensign_opensign-files:/data -v $(pwd):/backup
ubuntu tar czvf /backup/files-backup.tar.gz /data
```

10. Ajouter un enregistrement sur le DNS :

10.1. Dans le fichier docker-compose.yml :

Éditer le fichier **docker-compose.yml** et y ajouter les lignes suivantes :

```
dns:
  - <dns_server_address> # DNS local - Hébergement sur un serveur local
  - <dns_server_address> # DNS secondaire
dns_search:
  - <domain_name>.com
```

10.2. Sur Gandi :

1. Se connecter à Gandi
2. Dans le menu de gauche, sélectionner **NOM DE DOMAINE**
3. Dans la fenêtre **Noms de domaine**, sélectionner **domain_name.com**
    1. Cliquer sur l'onglet **Enregistrements DNS**
    2. Cliquer sur **Ajouter un enregistrement**
    3. Dans la fenêtre **Ajouter un enregistrement DNS**
    4. Renseigner l'adresse IPv4 avec l'adresse IP statique publique précédemment créée dans Amazon Lightsail
    5. Pour finaliser la création de la l'enregistrement DNS, cliquer sur **Créer**

11. En cas de modification du fichier docker-compose.yml :

```
sudo docker-compose down
sudo docker system prune -a
sudo docker volume prune
sudo docker-compose up -d --force-recreate
```

12. Générer le certificat auto-signé :

Exécuter les commandes suivantes et utiliser la phrase de passe :

```
openssl genrsa -des3 -out ./local_dev.key 2048
openssl req -key ./local_dev.key -new -x509 -days 365 -out ./local_dev.crt
openssl pkcs12 -inkey ./local_dev.key -in ./local_dev.crt -export -out
./local_dev.pfx
openssl base64 -in ./local_dev.pfx -out ./base64_pfx
```

13. Mettre à jour les données dans le fichier .env.prod :

Copier les données du fichier **base64_pfx** et les coller dans le fichier **.env.prod** à ce niveau :

```
# Base64 encoded PFX or p12 document signing certificate file
**************************************************************************
*******
PFX_BASE64='<données_du_fichier_base64_pfx>'
```

14. Se connecter à l'interface Web d'Opensign:

- Depuis un navigateur, aller à l'adresse :

```
https///opensign.domain_name.com/addadmin
```

Remarques : L'adresse doit être accessible.

- Suivre les instructions pour la création du compte administrateur.