

Présentation et Initialisation

“ **Gitea** est un service de développement logiciel tout-en-un, facile à utiliser et auto-hébergé. Il inclut l'hébergement Git, la révision de code, la collaboration en équipe, un registre de paquets et des fonctionnalités d'intégration continue et de déploiement continu (CI/CD). Il est similaire à GitHub, Bitbucket et GitLab.

À l'origine, **Gitea** est issu d'un fork de *Gogs*, mais la quasi-totalité du code a été remaniée.

Objectifs - Déploiement Docker Compose

Résumé : Rédiger une stack Gitea auto-hébergée, pensée pour tourner derrière un Apache déjà en place sur un serveur, avec accès Git en SSH.

Prérequis :

- PostgreSQL comme base de données (conteneur dédié, réseau interne isolé)
- Le serveur SSH intégré de Gitea exposé sur le port hôte **2222** (le port 22 est déjà pris par le sshd système)
- Un accès HTTP publié uniquement sur `127.0.0.1:3000`, car **Apache** (natif sur l'hôte) fait déjà reverse-proxy/TLS pour les autres vhosts de ce serveur et gèrera aussi celui-ci
- Domaine cible : `git.mondomain.ovh` (email admin : placeholder `admin@mondomain.ovh`, à changer)
- Inscription publique désactivée (comptes créés par l'admin uniquement)
- Gitea Actions et mailer désactivés pour le moment (documentés en option, pas activés)
- Toute la configuration versionnable en clair, tous les secrets externalisés dans un `.env` non commité

Architecture

Internet

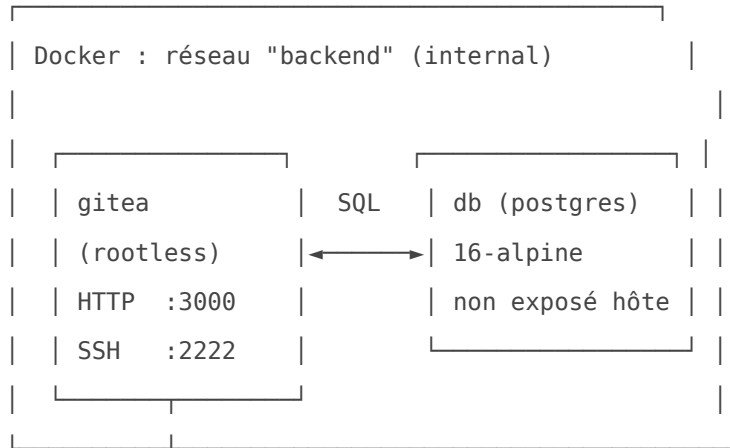
| 80/443 (TLS géré par Apache, certs Certbot existants)



```
| Apache (natif, sur l'hôte) |  
| vhost git.mondomain.ovh   |
```



| proxy_pass http://127.0.0.1:3000



| port 2222 publié sur toutes les interfaces



Clients Git (ssh://git@git.mondomain.ovh:2222/...)

Installation

1. Préparer les configuration

Nous proposons le docker-compose et le modèle app.ini par défaut.

Copier les deux fichiers dans votre dossier pour les configurer.

```
cd gitea/  
cp <from Wiki>.env.exemple .env  
cp <from Wiki>docker-compose.yaml  
cp <from Wiki>app.ini app.ini  
chmod 600 .env
```

Le template de la configuration App est accessible [ici](#), sans aucun secret.

Le template de la configuration docker-compose est accessible [ici](#), sans aucun secret.

Le template de la configuration .env est accessible [ici](#), sans aucun secret.

2. Configurer les variables & les secrets

Générer un mot de passe PostgreSQL fort :

```
openssl rand -base64 32
```

Générer les trois secrets internes de Gitea (aucun conteneur ne tourne encore, on utilise l'image pour exécuter juste la commande) :

```
docker run --rm gitea/gitea:1.27-rootless gitea generate secret SECRET_KEY
docker run --rm gitea/gitea:1.27-rootless gitea generate secret INTERNAL_TOKEN
docker run --rm gitea/gitea:1.27-rootless gitea generate secret JWT_SECRET
```

Reporter chaque valeur générée dans le `.env` ou directement dans le fichier `app.ini` (`POSTGRES_PASSWORD`, `GITEA_SECRET_KEY`, `GITEA_INTERNAL_TOKEN`, `GITEA_OAUTH2_JWT_SECRET`).

App.ini c'est le fichier dans lequel l'entrypoint Gitea écrit directement les secrets au démarrage (mot de passe DB, `SECRET_KEY`, `INTERNAL_TOKEN`, `JWT_SECRET`). Il permet également de configurer Gitea pour les logs et la gestion des inscriptions des comptes etc...

Avant le premier démarrage vérifier dans les 3 fichiers (`docker-compose.yml` / `.env` / `app.ini`) :

- Les secrets sont remplacés par ceux générés précédemment
- Le domaine `git.mondomain.ovh` et l'email admin placeholder (`admin@mondomain.ovh`) sont remplacés

3. Démarrer la stack

```
docker compose up -d db
docker compose ps          # attendre que "db" soit "healthy"
docker compose up -d gitea
docker compose logs -f gitea
```

4. Créer le premier compte administrateur

L'inscription publique étant désactivée, le premier compte se crée en ligne de commande :

```
docker compose exec gitea gitea admin user create \
  --username admin \
  --password 'UN_MOT_DE_PASSE_FORT' \
  --email admin@mondomain.ovh \
  --admin \
  --must-change-password=false
```

Vérifier en local avant de brancher Apache :

```
curl -I http://127.0.0.1:3000/
```

Revision #5

Created 2026-07-30 10:05:53 UTC by gpatruno

Updated 2026-07-31 15:32:23 UTC by gpatruno