Templates

> Ci dessous les templates des 3 fichiers de configurations utile pour la stack Gitea en docker compose.

docker-compose.yaml

```
name: gitea

services:
  db:
    image: postgres:16-alpine
    container_name: gitea-db
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    cap_drop:
      - ALL
    cap_add:
      # Requis uniquement par postgres pour initdb / changement d'UID au démarrage.
      - CHOWN
      - DAC_OVERRIDE
      - SETUID
      - SETGID
    environment:
      POSTGRES_DB: ${POSTGRES_DB:-gitea}
      POSTGRES_USER: ${POSTGRES_USER:-gitea}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:?La variable POSTGRES_PASSWORD doit être définie
dans .env}
    volumes:
      - gitea_db_data:/var/lib/postgresql/data
    networks:
      - backend
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER:-gitea} -d ${POSTGRES_DB:-gitea}"]
      interval: 10s
```

```
      timeout: 5s
      retries: 5
      start_period: 10s
    logging:
      driver: json-file
      options:
        max-size: "10m"
        max-file: "3"
    deploy:
      resources:
        limits:
          cpus: "1.0"
          memory: 512M

  gitea:
    # Image "rootless" : le conteneur ne tourne jamais en root, pas de chown au démarrage.
    # ⚠ Vérifier régulièrement la dernière version stable sur
hub.docker.com/r/gitea/gitea/tags
    image: gitea/gitea:1.27-rootless
    container_name: gitea
    restart: unless-stopped
    # cap_drop:
    #   - ALL
    environment:
      # --- Non-secret, utile de garder ici plutôt que de dupliquer dans app.ini ---
      GITEA__database__DB_TYPE: postgres
      GITEA__database__HOST: db:5432
      GITEA__database__NAME: ${POSTGRES_DB:-gitea}
      GITEA__database__USER: ${POSTGRES_USER:-gitea}
      # --- Secrets injectés depuis .env, fusionnés dans app.ini par l'entrypoint Gitea ---
      # (mécanisme officiel "environment-to-ini", ne modifie pas le fichier monté sur disque)
      GITEA__database__PASSWD: "${POSTGRES_PASSWORD:?La variable POSTGRES_PASSWORD doit être
définie dans .env}"
      GITEA__security__SECRET_KEY: "${GITEA_SECRET_KEY:?Générez la valeur avec 'gitea generate
secret SECRET_KEY'}"
      GITEA__security__INTERNAL_TOKEN: "${GITEA_INTERNAL_TOKEN:?Générez la valeur avec 'gitea
generate secret INTERNAL_TOKEN'}"
      GITEA__oauth2__JWT_SECRET: "${GITEA_OAUTH2_JWT_SECRET:?Générez la valeur avec 'gitea
generate secret JWT_SECRET'}"
    volumes:
```

```
      - ./data:/var/lib/gitea
      # ⚠ Pas de ":ro" ici : l'entrypoint Gitea (environment-to-ini) écrit directement dans
ce
      # fichier au démarrage pour y fusionner les valeurs GITEA__* (dont les secrets). C'est
donc
      # config/app.ini (gitignoré, non versionné) qui reçoit le résultat — voir
config/app.ini.example
      # pour le template versionné sans secret.
      - ./config/app.ini:/etc/gitea/app.ini
    ports:
      - "3000:3000"
      - "2222:2222"
    networks:
      - backend
      - web
    depends_on:
      db:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "wget", "--no-verbose", "--tries=1", "--spider",
"http://127.0.0.1:3000/api/healthz"]
      interval: 30s
      timeout: 10s
      retries: 3
      start_period: 30s

networks:
  # Réseau isolé : "db" n'y a accès à rien d'autre, et n'est jamais routable vers l'extérieur.
  backend:
    internal: true
  # Réseau non-interne : nécessaire uniquement pour que les ports publiés de "gitea"
(3000/2222)
  # soient réellement mappés sur l'hôte (Docker ne publie pas de port pour un conteneur qui
  # n'est connecté qu'à un réseau "internal: true"). "db" n'y est volontairement pas connecté.
  web:

volumes:
  gitea_data:
  gitea_db_data:
```

App.ini

```
; ⚠ Fichier généré/complété au runtime — NE PAS VERSIONNER (voir .gitignore).
; Ce fichier est monté en écriture dans le conteneur : au démarrage, l'entrypoint Gitea
; (environment-to-ini) y écrit directement les valeurs GITEA__section__CLE définies dans
; docker-compose.yml/.env (mot de passe DB, SECRET_KEY, INTERNAL_TOKEN, JWT_SECRET...).
; Après le premier démarrage, ce fichier contient donc des secrets en clair.
; Le template versionné (sans secret) se trouve dans config/app.ini.example — c'est lui
; qu'il faut modifier pour changer la configuration fonctionnelle, puis recopier ici.
APP_NAME = Gitea
RUN_MODE = prod
RUN_USER = git
WORK_PATH = /var/lib/gitea

[repository]
ROOT = /var/lib/gitea/git/repositories
DEFAULT_PRIVATE = private
DEFAULT_BRANCH = main
DISABLE_HTTP_GIT = false

[server]
PROTOCOL = http
DOMAIN = git.mondomain.ovh
ROOT_URL = https://git.mondomain.ovh/
HTTP_ADDR = 0.0.0.0
HTTP_PORT = 3000
; Serveur SSH intégré à Gitea (pas de passthrough OpenSSH de l'hôte).
START_SSH_SERVER = true
SSH_DOMAIN = git.mondomain.ovh
; Port annoncé aux utilisateurs (clone URLs) et port d'écoute interne du conteneur.
; Correspond au mapping "2222:2222" du docker-compose (le port hôte 22 reste au sshd système).
SSH_PORT = 2222
SSH_LISTEN_PORT = 2222
LFS_START_SERVER = true
OFFLINE_MODE = true
LFS_JWT_SECRET =

[database]
DB_TYPE = postgres
HOST = db:5432
```

```
NAME = gitea
USER = gitea
PASSWD = CHANGE_ME

; NAME / USER / PASSWD sont fournis par les variables d'environnement GITEA__database__*
; (voir docker-compose.yml). Ne pas les redéfinir ici pour éviter toute divergence.
[security]
; Empêche l'assistant d'installation web de s'afficher : la config est déjà complète.
INSTALL_LOCK = true
; SECRET_KEY / INTERNAL_TOKEN fournis via l'environnement (GITEA__security__*).
COOKIE_SECURE = true
REVERSE_PROXY_LIMIT = 1
; Apache tourne nativement sur le même hôte : seul 127.0.0.1 est une source de confiance
; pour les en-têtes X-Forwarded-*.
REVERSE_PROXY_TRUSTED_PROXIES = 127.0.0.1
; Un admin de dépôt ne peut pas éditer les hooks Git côté serveur (surface de RCE réduite).
DISABLE_GIT_HOOKS = true
DISABLE_WEBHOOKS = false
MIN_PASSWORD_LENGTH = 12
PASSWORD_COMPLEXITY = lower,upper,digit,spec
INTERNAL_TOKEN =
SECRET_KEY =

[service]
; Comptes créés uniquement par un administrateur (voir README) : pas d'auto-inscription.
DISABLE_REGISTRATION = true
REQUIRE_SIGNIN_VIEW = true
REGISTER_EMAIL_CONFIRM = false
ENABLE_NOTIFY_MAIL = false
DEFAULT_KEEP_EMAIL_PRIVATE = true
DEFAULT_ALLOW_CREATE_ORGANIZATION = true
NO_REPLY_ADDRESS = noreply.git.mondomain.ovh

[openid]
ENABLE_OPENID_SIGNIN = false
ENABLE_OPENID_SIGNUP = false

[session]
PROVIDER = db
COOKIE_NAME = gitea_session
```

```
COOKIE_SECURE = true
SESSION_LIFE_TIME = 86400

[log]
MODE = console, file
LEVEL = Info
ROOT_PATH = /var/lib/gitea/log

[api]
; Réduit la surface d'information exposée publiquement ; réactivable si besoin d'intégrations API.
ENABLE_SWAGGER = false
MAX_RESPONSE_ITEMS = 50

[other]
SHOW_FOOTER_VERSION = false
SHOW_FOOTER_TEMPLATE_LOAD_TIME = false

[mailer]
; Désactivé pour le moment (pas de réinitialisation de mot de passe par email).
; Pour l'activer : ENABLED = true puis ajouter PROTOCOL/SMTP_ADDR/SMTP_PORT/USER
; et injecter GITEA__mailer__PASSWD via l'environnement (jamais en clair ici).
ENABLED = false

[actions]
; Gitea Actions (CI/CD) désactivé pour cette installation minimale.
ENABLED = false

[metrics]
; Désactivé par défaut : à activer uniquement derrière une authentification/IP restreinte.
ENABLED = false

[oauth2]
JWT_SECRET =
```

.env.exemple

```
# Copier ce fichier vers .env puis renseigner de vraies valeurs :
#   cp .env.example .env && chmod 600 .env
#
```

```
# ⚠ .env ne doit JAMAIS être commité (voir .gitignore). Toutes les valeurs ci-dessous
# sont des secrets ou quasi-secrets.

# --- PostgreSQL ---
POSTGRES_DB=gitea
POSTGRES_USER=gitea
# Générer un mot de passe fort, par exemple : openssl rand -base64 32
POSTGRES_PASSWORD=CHANGE_ME

# --- Secrets internes Gitea ---
# Générer chaque valeur avec le binaire Gitea, par exemple :
#   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
GITEA_SECRET_KEY=CHANGE_ME
GITEA_INTERNAL_TOKEN=CHANGE_ME
GITEA_OAUTH2_JWT_SECRET=CHANGE_ME
```

Revision #2
Created 31 July 2026 15:07:17 by gpatruno
Updated 31 July 2026 15:13:06 by gpatruno