Configurer une clé SSH

Ce guide explique comment configurer une clé SSH pour cloner, pousser et gérer tous vos dépôts Gitea sans mot de passe.

1. Une seule clé suffit pour tous vos dépôts

En SSH, la clé publique est rattachée à votre **compte utilisateur Gitea**, pas à un dépôt en particulier. Une fois ajoutée une seule fois dans **Paramètres → Clés SSH/GPG**, elle vous donne accès en SSH à :

- tous les dépôts dont vous êtes propriétaire,
- tous les dépôts d'organisation où vous avez des droits (lecture/écriture selon vos permissions),
- avec les mêmes droits que ceux définis dans l'interface web (une clé SSH ne donne pas plus de droits qu'un compte n'en a déjà).

Inutile donc de créer une clé par dépôt.

2. Générer une clé SSH (si vous n'en avez pas déjà une)

```
ssh-keygen -t ed25519 -C "votre_email@example.com" -f ~/.ssh/id_ed25519_gitea
```

- `ed25519` : algorithme moderne, recommandé (plus court et au moins aussi sûr que RSA).
- Une **passphrase** est demandée : renseignez-en une (voir §6, bonnes pratiques).
- Si vous avez déjà une clé SSH que vous souhaitez réutiliser, passez directement à l'étape 3 avec `~/.ssh/id_ed25519` (ou le nom de votre clé existante) à la place de `~/.ssh/id_ed25519_gitea` dans la suite de ce guide.

3. Ajouter la clé publique à votre compte Gitea

Afficher la clé publique à copier :

```
cat ~/.ssh/id_ed25519_gitea.pub
```

Dans l'interface web de Gitea :

1. **Paramètres du compte** (icône profil, en haut à droite) → **Clés SSH/GPG**

2. **Ajouter une clé** → coller le contenu de `id_ed25519_gitea.pub`
3. Donner un nom explicite à la clé (ex : `laptop-perso`, `poste-bureau`) — utile pour la révoquer précisément plus tard sans casser l'accès de vos autres machines
4. Valider

4. Configurer le client SSH (`~/.ssh/config`)

Le serveur SSH intégré de Gitea écoute sur le port **2222** (et non 22). Un alias dans `~/.ssh/config` évite d'avoir à le préciser à chaque commande :

```
Host git.mondomain.ovh
    HostName git.mondomain.ovh
    Port 2222
    User git
    IdentityFile ~/.ssh/id_ed25519_gitea
    IdentitiesOnly yes
```

```
chmod 600 ~/.ssh/config
```

5. Tester la connexion

```
ssh -T git@git.mondomain.ovh
```

Réponse attendue :

```
Hi <votre_nom_utilisateur>! You've successfully authenticated, but Gitea does not provide
shell access.
```

C'est normal : Gitea confirme l'authentification mais ne donne volontairement pas de shell.

Sans l'alias `~/.ssh/config` configuré à l'étape 4 :

```
ssh -T -p 2222 git@git.mondomain.ovh
```

6. Cloner, pousser, committer

Cloner un nouveau dépôt (avec l'alias configuré) :

```
git clone git@git.mondomain.ovh:USER/REPO.git
```

Sans alias :

```
git clone ssh://git@git.mondomain.ovh:2222/USER/REPO.git
```

Basculer un dépôt existant (cloné en HTTP) vers SSH :

```
git remote set-url origin git@git.mondomain.ovh:USER/REPO.git
```

Ensuite, `git add`, `git commit`, `git push`, `git pull` fonctionnent normalement : l'authentification est entièrement gérée par la clé SSH, plus besoin de nom d'utilisateur ni de mot de passe/token.

7. Bonnes pratiques (DevSecOps)

- **Une clé par appareil**, pas une clé unique copiée sur toutes vos machines : en cas de perte ou de compromission d'un appareil, vous pouvez révoquer uniquement sa clé sans couper l'accès des autres.
- **Toujours une passphrase** sur la clé privée, combinée à `ssh-agent` pour ne pas la retaper à chaque commande :

  ```
  eval "$(ssh-agent -s)"
  ssh-add ~/.ssh/id_ed25519_gitea
  ```

- La clé **privée** (`id_ed25519_gitea`, sans `.pub`) ne doit jamais quitter votre machine : vérifier `chmod 600 ~/.ssh/id_ed25519_gitea` et `chmod 700 ~/.ssh`.
- Nommez explicitement chaque clé ajoutée dans Gitea (étape 3) : l'interface affiche aussi la date de dernière utilisation, utile pour repérer une clé oubliée/inutilisée à révoquer.
- **Révoquer immédiatement** une clé en cas de perte/vol de l'appareil correspondant : Paramètres → Clés SSH/GPG → supprimer la clé concernée.

8. Dépannage

Symptôme	Cause probable	Solution
`Permission denied (publickey)`	Clé non ajoutée dans Gitea, ou mauvaise clé utilisée	`ssh -vT git@git.mondomain.ovh` pour voir quelle clé est proposée ; vérifier qu'elle correspond bien à celle ajoutée dans Gitea
`Permission denied (publickey)` malgré une clé correcte	Plusieurs clés dans l'agent SSH, la mauvaise est essayée en premier	Ajouter `IdentitiesOnly yes` dans `~/.ssh/config` (voir §4)
`ssh: connect to host ... port 22: Connection refused`	Le port `2222` n'est pas utilisé (alias non configuré ou oublié dans l'URL)	Utiliser l'alias `~/.ssh/config`, ou `ssh://git@git.mondomain.ovh:2222/...`
Timeout / `Connection refused` sur le port 2222	Port fermé côté pare-feu ou conteneur non démarré	`sudo ufw status` côté serveur ; `docker compose ps` (le port `2222` doit apparaître)
`Bad permissions` / clé ignorée par SSH	Droits trop ouverts sur `~/.ssh` ou la clé privée	`chmod 700 ~/.ssh && chmod 600 ~/.ssh/id_ed25519_gitea`

Symptôme	Cause probable	Solution
`WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!` qui **revient à chaque redémarrage** du conteneur	Clés hôte SSH du **serveur** (pas de votre poste) illisibles par le conteneur — voir §9	Corriger les permissions de `data/data/ssh/` côté serveur (voir §9)

Clé refusé après redémarrage du service

Côté serveur : la clé hôte change à chaque redémarrage du conteneur

Symptôme : l'avertissement "*REMOTE HOST IDENTIFICATION HAS CHANGED*" réapparaît après chaque redémarrage de `gitea`, même en supprimant l'entrée dans `known_hosts` et en faisant confiance à la nouvelle clé à chaque fois — le fingerprint proposé change encore la fois suivante.

Cause : les fichiers de clés hôte SSH persistées (`data/data/ssh/gitea.rsa`, `ssh_host_ecdsa_key`, `ssh_host_ed25519_key`...) appartiennent à `root:root` en mode `600` au lieu de `1000:1000` (l'UID `git` du conteneur rootless). Gitea échoue silencieusement à charger ces clés persistées — visible dans les logs :

```
docker compose logs gitea | grep -i -A1 "Adding SSH host key"
# [I] Adding SSH host key: /var/lib/gitea/data/ssh/gitea.rsa
# [E] Failed to set Host Key. open /var/lib/gitea/data/ssh/gitea.rsa: permission denied
```

Le serveur SSH démarre quand même, mais avec **une clé hôte générée en mémoire, différente à chaque redémarrage du conteneur** — d'où l'avertissement qui revient sans cesse, même après avoir fait confiance à la clé précédente : elle n'est jamais la même deux fois.

Ce cas survient typiquement après une restauration/migration manuelle où le dossier `ssh/` a été copié avec `sudo` séparément du reste des données (voir aussi `BACKUP.md`, §7, sur la correspondance UID à respecter côté serveur).

Correction, côté serveur :

```
sudo chown -R 1000:1000 data/data/ssh
docker compose restart gitea

# Vérifier que le chargement réussit désormais (plus de ligne [E])
docker compose logs gitea | grep -i -A1 "Adding SSH host key"
```

Puis côté client, une dernière fois (la clé hôte restera stable à partir de maintenant) :

```
ssh-keygen -f ~/.ssh/known_hosts -R "[git.mondomain.ovh]:2222"
ssh -T -p 2222 git@git.mondomain.ovh
```

Revision #2
Created 31 July 2026 15:26:41 by gpatruno

Updated 31 July 2026 15:28:46 by gpatruno