

SSH refusé ? Le parcours rapide pour retrouver la bonne clé

Suivez ce parcours dans l'ordre pour identifier pourquoi le serveur refuse votre clé publique. Chaque étape isole une cause précise sans modifier la sécurité au hasard.

Au sommaire

- 1 Avant de commencer : ce que signifie l'erreur
- 2 Parcours de correction en 5 étapes
- 3 Repères de permissions à vérifier
- 4 Checklist finale avant de réessayer
- 5 Cas à ne pas confondre

Avant de commencer : ce que signifie l'erreur

« Permission denied (publickey) » est un refus d'authentification par le serveur, pas nécessairement un problème d'accès au réseau.

La clé privée reste exclusivement sur votre poste ; seule la clé publique est autorisée sur le serveur.

Les quatre contrôles prioritaires sont : compte distant, clé privée proposée, clé publique dans `authorized_keys` et permissions.



PERMISSIONS SSH LINUX RECOMMANDÉES



SÉCURISEZ VOS CLÉS SSH

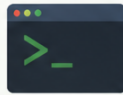
Des permissions correctes garantissent que SSH accepte vos clés et protège votre système contre les accès non autorisés.

ÉLÉMENT	CHEMIN (EXEMPLE)	PERMISSION RECOMMANDÉE	EXPLICATION
Dossier ~/.ssh Conteneur des clés SSH et fichiers associés.	~/.ssh	700 rwx-----	Le propriétaire peut lire, écrire et exécuter. Personne d'autre n'a aucun accès. Empêche les autres utilisateurs d'accéder à vos clés et métadonnées.
Clé privée Clé sensible utilisée pour vous authentifier.	~/.ssh/id_rsa ou ~/.ssh/id_ed25519	600 rw-----	Le propriétaire peut lire et écrire. Personne d'autre n'a aucun accès. Indispensable pour protéger votre clé privée contre toute lecture non autorisée.
Clé publique Clé publique partagée avec les serveurs distants.	~/.ssh/id_rsa.pub ou ~/.ssh/id_ed25519.pub	644 rw-r--r--	Le propriétaire peut lire et écrire. Le groupe et les autres peuvent lire. La clé publique peut être lue par tous : elle n'est pas sensible.

BONNES PRATIQUES

- ✓ Vérifiez et corrigez les permissions avant d'utiliser SSH.
- ✓ Ne stockez jamais votre clé privée dans un emplacement partagé ou synchronisé (ex: cloud).
- ✓ Utilisez une phrase de passe pour protéger votre clé privée (recommandé).
- ✓ Ne modifiez pas les permissions à des valeurs plus ouvertes.
- ✓ Contrôlez régulièrement les permissions avec la commande `ls -ld` ou `ls -l`.

```
$ chmod 700 ~/.ssh
$ chmod 600 ~/.ssh/id_rsa      # ou id_ed25519
$ chmod 644 ~/.ssh/id_rsa.pub  # ou id_ed25519.pub
```



RISQUES EN CAS DE MAUVAISES PERMISSIONS

- ✗ SSH peut refuser d'utiliser vos clés ("permissions too open").
- ✗ D'autres utilisateurs de la machine peuvent lire votre clé privée.
- ✗ Risque de compromission de votre compte et de vos serveurs.
- ✗ Non-conformité avec les bonnes pratiques de sécurité.



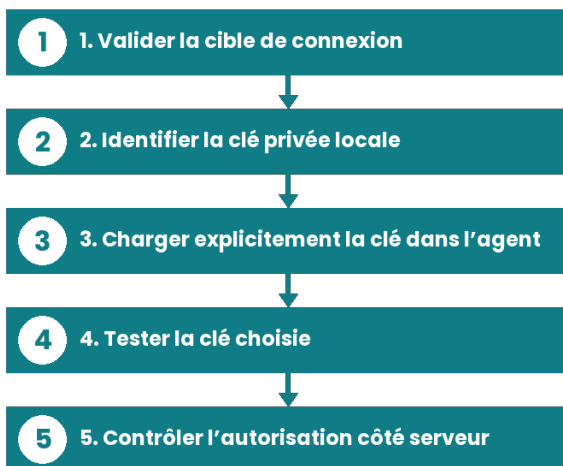
RAPPEL :

La sécurité de SSH repose autant sur la protection de votre clé privée que sur la robustesse de votre serveur.

🔒 SÉCURITÉ • CONFIDENTIALITÉ • INTÉGRITÉ 🔒

INFOGRAPHIE Schéma des permissions SSH Linux pour le dossier .ssh et les clés publiques ou privées.

Parcours de correction en 5 étapes



1. Valider la cible de connexion

Vérifiez l'utilisateur distant, le nom du serveur et, si nécessaire, le port communiqué. Dans « `ssh utilisateur@nom-du-serveur` », l'utilisateur est le compte Linux du serveur, pas votre nom local.

2. Identifier la clé privée locale

Listez les fichiers SSH avec « `ls -la ~/.ssh` ». Le fichier sans suffixe .pub est la clé privée ; le fichier .pub correspondant est la clé publique. Ne générez pas une nouvelle paire avant d'avoir vérifié les clés existantes.

3. Charger explicitement la clé dans l'agent

Démarrez l'agent avec « eval "\$(ssh-agent -s)" », puis ajoutez la clé avec « ssh-add ~/.ssh/id_ed25519 ». Contrôlez les identités chargées avec « ssh-add -l -E sha256 ».

4. Tester la clé choisie

Forcez temporairement la clé à utiliser : « ssh -i ~/.ssh/id_ed25519 utilisateur@nom-du-serveur ». Si ce test réussit alors que la commande simple échoue, la sélection automatique de clé ou l'agent est probablement en cause.

5. Contrôler l'autorisation côté serveur

Avec un accès d'administration alternatif, comparez la clé publique locale affichée par « cat ~/.ssh/id_ed25519.pub » avec une ligne du fichier « ~/.ssh/authorized_keys » du compte distant visé. La ligne doit être complète et rester sur une seule ligne.

 Erreur publickey



 Vérifier connexion



 Vérifier clé privée



 Charger agent SSH



 Vérifier authorized
keys



 Connexion validée

Repères de permissions à vérifier

Élément	Droit recommandé	Commande de vérification
Dossier ~/.ssh	700	ls -ld ~/.ssh
Clé privée	600	ls -l ~/.ssh/id_ed25519
Clé publique	644	ls -l ~/.ssh/id_ed25519.pub

Dossier ~/.ssh  **700**

Clé privée  **600**

Clé publique  **644**

Checklist finale avant de réessayer

- ☐ Le compte indiqué avant @ est bien le compte Linux distant attendu.
- ☐ Le port SSH est précisé avec -p si le serveur n'utilise pas le port habituel.
- ☐ La clé privée ciblée existe localement et n'a pas de suffixe .pub.
- ☐ La bonne clé est chargée dans ssh-agent ou testée avec l'option -i.
- ☐ La clé publique correspondante est présente dans authorized_keys du bon compte distant.
- ☐ Le dossier .ssh, les clés et leurs propriétaires respectent les droits attendus.
- ☐ Un test détaillé est lancé avec « ssh -v utilisateur@nom-du-serveur » pour voir la clé réellement proposée.

! Ne contournez pas le problème en ouvrant les droits

N'élargissez pas les permissions du dossier SSH ou des clés pour faire disparaître l'erreur : OpenSSH peut précisément refuser des fichiers trop exposés. Modifiez uniquement les fichiers appartenant à votre compte et ne communiquez jamais votre clé privée.

Cas à ne pas confondre

Un avertissement concernant l'empreinte ou l'identité du serveur ne correspond pas à « Permission denied (publickey) » : il intervient avant l'authentification. Pour GitHub, utilisez l'utilisateur distant « git » et testez la connexion avec « ssh -T git@github.com ».

Avant toute modification côté serveur, utilisez une voie d'administration autorisée et vérifiez que vous intervenez sur le bon compte distant.