Consolidation — Gérer les accès d'une équipe de développement en entreprise

Contexte

Tu es administratrice système chez **WebForge**, une agence qui développe des sites web.

L'équipe vient de démarrer un projet Laravel nommé **boutique-en-ligne**, hébergé sur un serveur Linux que tu gères. Le projet doit être livrable, sécurisé, et accessible aux bonnes personnes.

Voici l'équipe telle qu'elle va exister sur ce projet :

Rôle	**Personne**	**Statut**
Administrateur système / toi	ton compte habituel	Accès total, gère le serveur, ne code pas sur le projet
Développeuse interne (lead dev)	**sophie**	Doit pouvoir tout faire sur le code du projet
Développeur interne	**marc**	Doit pouvoir tout faire sur le code du projet
Freelance externe	**paul**	Doit pouvoir travailler sur le code, **sauf** sur les fichiers sensibles (configuration, secrets)
Serveur web	**www-data**	Doit pouvoir écrire uniquement dans certains dossiers précis du projet, sans avoir la main sur le reste du code
Compte de déploiement	**deploy**	Compte technique utilisé uniquement pour livrer le code sur le serveur, jamais par un humain directement au clavier

Ce document reprend, dans un seul scénario continu et réaliste, tout ce que tu as manipulé dans les séances précédentes : chemins, permissions, utilisateurs et groupes, `setgid` et sticky bit. Comme d'habitude, aucune commande n'est donnée : à toi de la trouver.

Le rapport à me transmettre

Utilise la même méthode que pour les séances précédentes.

Phase 1 — Livrer le projet : le compte de déploiement

`deploy` est un compte technique : personne ne doit pouvoir s'y connecter avec un mot de passe, seulement par clé, et seulement depuis ta machine de travail (le futur poste qui déclenchera les mises en ligne).

1.1 — Créer le compte technique

Crée le compte `deploy`. Réfléchis à son groupe : doit-il appartenir à `webforge_boutique` ? Justifie ta décision avant d'agir.

1.2 — Authentification par clé uniquement

Génère une paire de clés dédiée à ce compte de déploiement. Fais en sorte que `deploy` accepte les connexions par cette clé.

1.3 — Fermer la porte du mot de passe pour ce compte précis

Configure le service SSH pour que le compte `deploy` ne puisse **jamais** se connecter par mot de passe, même si son mot de passe existe par ailleurs — alors que les autres comptes du serveur, eux, continuent à pouvoir utiliser un mot de passe normalement. (Indice de réflexion : la configuration SSH permet des règles différentes selon l'utilisateur concerné, pas seulement une règle globale pour tout le serveur.)

1.4 — Vérification

Essaie de te connecter en tant que `deploy` par mot de passe (doit échouer), puis par clé (doit réussir).

Phase 2 — Incident réaliste : fin de mission du freelance

Le contrat de Paul se termine aujourd'hui. Comme pour tout départ, l'accès doit être coupé proprement, pas juste "on espère qu'il ne revient pas".

2.1 — Couper l'accès, sans tout détruire

Retire l'accès de `paul` au projet dans son ensemble (il ne doit plus pouvoir rien lire ni écrire nulle part dans `/var/www/html`), sans supprimer son compte pour l'instant (l'entreprise veut garder une trace de son travail passé, notamment les fichiers déjà déposés).

2.2 — Décision finale

Une semaine plus tard (simulée), le compte de Paul doit être définitivement supprimé. Fais-le, et réfléchis avant d'agir à ce que deviennent les fichiers qu'il avait déposés dans `depot_livrables/` — vérifie concrètement ce résultat après suppression.