Fiche mémo | Leçon 29 | 24 min

Le parc, ses sites et ses serveurs

L'écran Sites réunit tout le parc : chaque site avec son client, son projet, sa machine, son environnement et la personne qui s'en occupe. Vous apprenez à déclarer un site et sa machine, à ranger leurs accès, à faire agir l'agent sur un serveur sans risque, à lire l'inventaire et à préparer une Debian neuve avec le socle.

Objectifs

- Déclarer un site et sa machine dans le parc, rattachés au bon client, au bon projet et au bon environnement
- Faire lire ou modifier une machine par l'agent en nommant l'accès, l'environnement et les limites du geste
- Lire l'inventaire et les constats d'un site, puis décider de ce qui devient une tâche

À retenir

- Chaque site du parc porte son client, son projet, sa machine, son environnement et la personne qui s'en occupe.
- Les accès se rangent sur la machine, au coffre, avec leur environnement ; jamais dans les notes de la fiche.
- En production, une commande part sans confirmation : lire d'abord, puis un accord écrit et borné.
- L'inventaire n'est complet qu'avec un accès SSH et la racine web renseignée ; sinon, socle et PHP seulement.
- Un site qui ne sert plus s'archive ; le socle n'installe que sur une Debian vierge, après aperçu et confirmation.

Le parc, du client à la machine

Le client
Le négociant : sa fiche, son portail, les personnes qui le suivent.

Le projet
Le travail : tâches, Studio, dépôt, adresses de production et de préproduction.

Le site
Une adresse et un environnement, qui s'en occupe, ce qui y est installé.

La machine
Le serveur ou le mutualisé, son hébergeur et ses accès rangés au coffre.

Chaque site se lit de haut en bas : à qui il est, pour quoi, où il tourne.

La demande à recopier

Demande à recopier Lire l'ancien site sans y toucher

```
Sur l'ancien site du négociant ([adresse]), avec l'accès FTP de production de son hébergeur
mutualisé, en lecture seule :
- la version de WordPress (par WP-CLI s'il est présent, sinon dans wp-includes/version.php) ;
- la liste des extensions et leur version ;
- la taille du dossier du site et celle des médias.
Dis-moi d'abord l'accès et la voie employés (FTP ou SSH). Ne modifie, ne déplace et ne supprime
rien. Si une commande devait écrire, arrête-toi et demande-moi.
```

Le code à recopier

Lire le parc par l'API BASH

```bash
# la clé se crée dans Paramètres › API et MCP et agit avec vos droits ; elle reste dans une variable d'environnement
# les sites en panne du parc que vous voyez
curl "https://api.nexus-engine.eu/api/v1/sites?etat=DOWN" \
  -H "Authorization: Bearer $NEXUS_CLE"

# les machines que vous voyez, avec leurs sites
curl "https://api.nexus-engine.eu/api/v1/servers" \
  -H "Authorization: Bearer $NEXUS_CLE"
```

Sans clé, l'API répond 401 (vérifié le 15/11/2026). Un site archivé n'apparaît qu'avec le filtre des archives (paramètre archives).

Les erreurs fréquentes

L'erreur, et pourquoi elle coûte	La correction
Écrire le mot de passe FTP de l'hébergeur dans les notes de la fiche du site. Les notes ne sont pas chiffrées et un agent les lit : le secret circule en clair, sans trace de qui l'a vu.	Rangez-le au coffre comme accès de la machine, avec son environnement ; les notes gardent les consignes, jamais un secret.
Déclarer la préproduction et la production comme un seul site, sans environnement. L'agent choisit l'accès par l'environnement : il peut lancer en production ce qui devait être essayé en préproduction.	Un site par environnement, chacun avec ses accès ; nommez l'environnement dans chaque demande.
Supprimer le site d'un client qui part, pour « faire le ménage ». Ses analyses, ses sauvegardes et ses demandes disparaissent de la vue, alors qu'elles servent en cas de litige ou de retour du client.	Archivez le site : il sort de la liste courante et tout le reste demeure.
Rapatrier le dump d'une base de client dans le dépôt du projet. Les données personnelles des clients du client partent dans l'historique de git, chez GitLab ou GitHub, et y restent même après suppression.	Posez les dumps dans .nexus/donnees/ du dossier de la session, ignoré de git, et effacez-les quand la migration est faite.