

# 09\_01 - la création de la VM

## le 260909

# VM Debian sur le NAS

## SasNexte — poste de

## l'univers Jux

*Procédure établie le 09/09/2026. Machine installée et opérationnelle le jour même. V2 du 09/09/2026 — redémarrage éprouvé, réservation DHCP posée, correction de l'erreur sur le gestionnaire de connexion.*

“ **Ce qui change par rapport à la V1.** Le §7 corrige une affirmation fausse : LightDM **est** installé, contrairement à ce que la V1 annonçait. Le §9 est neuf — la procédure de redémarrage et sa validation réelle. Le §4 insiste sur le démontage de l'ISO, qui avait été omis. Deux points du « reste à faire » sont clos : la réservation DHCP et la question ufw.

## 1. Raison d'être

Ce poste n'existe pas pour l'ubiquité, mais pour **l'étanchéité**. Il s'agit de cesser de mélanger deux univers sur une même machine.

|           | PC NEXTE (Windows)              | VM Debian (NAS)      |
|-----------|---------------------------------|----------------------|
| Univers   | Alteris / NEXTE — professionnel | Jux — personnel      |
| BookStack | bookstack.alteris.ovh           | bookstack.juxjux.ovh |

|                 | PC NEXTE (Windows)                            | VM Debian (NAS)                                       |
|-----------------|-----------------------------------------------|-------------------------------------------------------|
| Serveur d'appui | VPS Alteris 79.137.14.202                     | VPS Jux 51.77.141.54                                  |
| Outils          | QGIS, LibreOffice, PostGIS, pièces de mission | Firefox, Thunderbird, FileZilla, Calibre, Claude Code |

Un seul MCP BookStack par machine. C'est ce qui rend structurellement impossible de publier une procédure Alteris dans le BookStack perso, ou l'inverse — au lieu d'une discipline à tenir à chaque page.

“ Règle à ne pas relâcher. Pas de partage de mission NEXTE monté dans la VM, pas de coffre Alteris, pas de clé SSH vers 79.137.14.202. Un « juste un petit accès » suffit à rendre la séparation décorative.

## 2. Prérequis matériels — le seul point bloquant

Le DS220+ embarque un Celeron J4025 (2 cœurs, 2 threads). Le facteur limitant est la RAM.

|                  | Valeur constatée                                      |
|------------------|-------------------------------------------------------|
| RAM totale       | 5776 Mo (2 Go soudés + barrette 4 Go ajoutée en 2023) |
| Consommé par DSM | ~1160 Mo                                              |
| Volume1          | Btrfs — exigé par VMM — 1,4 To libres                 |
| DSM              | 7.4                                                   |

VMM réserve la mémoire statiquement : les 3 Go sont retirés à DSM en permanence tant que la VM tourne, qu'elle travaille ou non. Il n'y a pas de ballooning dynamique. Le plafond utile est d'environ 3,5 Go, DSM gardant un socle incompressible de 1,5 à 2 Go. Au-delà, c'est DSM qui swappe — et comme le disque virtuel est sur ce même NAS, tout ralentit ensemble.

Sur 2 Go alloués la VM tiendrait aussi : à l'usage réel, elle consomme 360 Mo au repos, et 473 Mo avec le bureau XFCE ouvert. Le dimensionnement se fait sur le pic, pas sur la somme des applications — elles ne tournent jamais ensemble.

# 3. Création de la machine virtuelle

Virtual Machine Manager n'était pas installé : Centre de paquets → **Virtual Machine Manager**, puis créer un stockage sur `volume1`.

À la première ouverture, DSM propose d'ouvrir les ports 30200-30300 dans le pare-feu. Accepter, **puis restreindre la règle** au sous-réseau local (Panneau de configuration → Sécurité → Pare-feu, source `192.168.1.0 / 255.255.255.0`). Ces ports n'ont aucune raison d'être joignables de l'extérieur.

## Paramètres retenus

| Écran               | Réglage                                                                   |
|---------------------|---------------------------------------------------------------------------|
| Nom                 | <code>debian-jux</code>                                                   |
| Processeurs         | 2                                                                         |
| Mémoire             | 3 Go                                                                      |
| Carte vidéo         | <code>vmvga</code> (seul choix)                                           |
| Type de machine     | <code>PC</code> (châssis i440fx, BIOS legacy)                             |
| Disque              | 40 Go, <b>contrôleur VirtIO SCSI</b> , réclamation d'espace <b>cochée</b> |
| Réseau              | Default VM Network, <b>VirtIO</b>                                         |
| Micrologiciel       | <b>Legacy BIOS</b> , pas UEFI                                             |
| Disposition clavier | <code>fr</code>                                                           |
| Port série          | <b>Désactiver</b> — inutile, et il a brouillé le diagnostic               |
| Contrôleur USB      | Désactivé                                                                 |
| Autostart           | <b>Oui</b>                                                                |
| Démarrer depuis     | <b>Disque virtuel</b>                                                     |

**Réclamation d'espace** : c'est le TRIM, à ne pas confondre avec l'allocation à la demande (que VMM applique déjà par défaut). Sans elle, le fichier disque croît au fil des `apt` et n'en redescend jamais.

**VirtIO SCSI** expose le disque en `/dev/sda`, pas `/dev/vda`.

**Adresse MAC.** VMM en génère une localement administrée — ici `02:11:32:25:c6:e8`, reconnaissable à son préfixe `02:`. Elle est stable tant que l'interface réseau existe, mais **supprimer et recréer la carte réseau dans VMM la change** — et rend caduque, en silence, la réservation DHCP du §8.

# 4. Installation de Debian 13

## L'image

Téléchargée directement par le NAS, ce qui évite de faire transiter 755 Mo par le PC :

```
cd '/volume1/Iso VM'
wget -c -O debian-13.6.0-amd64-netinst.iso \
    https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-13.6.0-amd64-netinst.iso
sha256sum debian-13.6.0-amd64-netinst.iso
curl -s https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/SHA256SUMS | grep netinst
```

Les deux sommes doivent être identiques.

## Le mode d'installation

**Prendre « Graphical install »**, pas « Install ». Le mode texte impose de saisir des numéros dans des listes qui défilent hors écran, et la console noVNC de VMM gère mal le clavier. Le mode graphique donne la souris : tous les choix se font au clic.

« Graphical install » ne désigne que le mode d'affichage — le premier écran reste le choix de la langue, où l'on prend « French — Français ».

## Les réponses qui comptent

| Écran                   | Réponse                                                                              |
|-------------------------|--------------------------------------------------------------------------------------|
| Nom de machine          | jux-debian                                                                           |
| Domaine                 | vide                                                                                 |
| Mot de passe root       | (voir remarque ci-dessous)                                                           |
| Partitionnement         | Assisté – disque entier → sda → <b>tout dans une seule partition</b>                 |
| Miroir                  | France, deb.debian.org, <b>pas de mandataire</b>                                     |
| Sélection des logiciels | <b>décocher tout</b> sauf <b>serveur SSH</b> et <b>utilitaires usuels du système</b> |
| GRUB                    | Oui → /dev/sda                                                                       |

Pas de LVM : sur un disque virtuel il n'apporte rien, on agrandit le disque dans VMM. Pas de chiffrement : la VM démarre automatiquement au boot du NAS et réclamerait une passphrase en console à chaque fois.

“ **Remarque sur le mot de passe root.** Le laisser vide fait installer `sudo` et place l'utilisateur dans le groupe. En définissant un mot de passe root — ce qui a été fait ici — `sudo` n'est **pas** installé, et le script de post-installation doit être lancé via `su -` avec le nom d'utilisateur en argument. `sudo` reste ensuite protégé par mot de passe : aucune automatisation à distance ne peut l'utiliser.

## ☐ Démontez l'ISO — l'étape qu'on oublie

Avant le redémarrage final : **démontez l'ISO** dans les paramètres de la VM (*Action → Modifier → Autres → Fichier ISO pour le démarrage → Démonter*).

**Ça n'a pas été fait le 09/09** — l'ISO est restée montée toute la journée sans qu'on s'en aperçoive, parce que *Démarrer depuis : Disque virtuel* masque le symptôme : la VM démarre sur le disque et tout paraît normal.

Le danger n'est pas immédiat, il est différé. Le jour où le disque devient non amorçable — GRUB cassé, disque plein — la VM ne s'arrête pas sur une erreur : **elle démarre l'installateur Debian**. Sur une machine sans écran, avec autostart activé, c'est le scénario où l'on formate sans le voir.

**Le vérifier fait partie de la mise en service, au même titre que le reste.** Le démontage est accepté à chaud, VM allumée.

## 5. LE PIÈGE — la console meurt au démarrage

**Symptôme.** Après l'installation, l'écran se fige systématiquement sur :

```
/dev/sda1: clean, 40622/2485504 files, 552845/9938432 blocks
[ 5.812838] systemd-ssh-generator[255]: Failed to query local AF_VSOCK CID
[ 5.958389] systemd[1]: Finished modprobe@efi_pstore.service
[ 6.941405] ACPI: bus type drm_connector registered
```

Plus rien ensuite. Aucune invite de connexion, `Ctrl+Alt+F2` sans effet, et la machine n'apparaît pas sur le réseau.

**Cause.** Trente millisecondes après `drm_connector registered`, le pilote de la carte vidéo `vmvga` prend la main sur l'écran et **tue la console**. Le système, lui, démarre parfaitement : SSH répond, le réseau monte, l'invite de connexion existe — elle n'est simplement plus affichée nulle part.

**Correctif** — dans `/etc/default/grub` :

```
GRUB_CMDLINE_LINUX_DEFAULT="nomodeset"
```

puis `update-grub`. `nomodeset` interdit le *kernel modesetting* et garde la console en VGA texte de bout en bout.

**Retirer `quiet` en même temps, et ne jamais le remettre.** C'est lui qui rend le diagnostic impossible : entre les messages du noyau et l'invite de connexion, l'écran reste muet, et rien ne distingue un démarrage figé d'un démarrage silencieux.

“ `nomodeset` **ne coûte rien à l'usage**. Vérifié le 09/09 : le bureau XFCE s'affiche normalement dans la console VMM malgré `nomodeset`. Seule l'accélération graphique est perdue, dont on n'a de toute façon pas l'emploi ici.

## Deux fausses pistes, pour ne pas les refaire

- **Le port série.** Plausible — on l'avait activé, et une console redirigée vers `ttyS0` produit exactement ce gel. Mais la ligne GRUB ne contenait **aucun** `console=ttyS0`. Vérifier avant de supposer.
- **Un blocage du démarrage.** Écarté par un détail : le compteur de fichiers affiché par `fsck` augmentait d'un démarrage à l'autre (40500 puis 40622). Un système qui ne démarre pas n'écrit rien.

## Comment atteindre GRUB

La console VMM n'existe que si la VM tourne — impossible de l'ouvrir « avant ». Il faut garder la fenêtre de console ouverte (bouton **Connecter**), démarrer la VM depuis DSM, cliquer **Connecter** aussitôt et **maintenir Maj** : GRUB affiche alors son menu et suspend le décompte ( `GRUB_TIMEOUT=5` ).

Sur la ligne « Debian GNU/Linux », appuyer sur `e` — pas Entrée — descendre jusqu'à la ligne `linux /boot/vmlinuz...`, aller en fin de ligne, ajouter `nomodeset` et supprimer `quiet`, puis `Ctrl+X`.

“ Une correction faite ainsi ne vaut **que pour ce démarrage**. Elle ne devient permanente qu'une fois écrite dans `/etc/default/grub` et suivie d'un `update-grub` — voir le §9, où cette distinction a failli passer inaperçue.

Les entrées « recovery mode » sont inutilisables si le compte root est verrouillé : le shell de secours refuse de s'ouvrir.

## 6. Post-installation

Script : `260909-postinstall_vm_debian_jux_v2.sh`. Idempotent, environ 20 minutes et 1,5 Go de paquets.

```
su -
bash /home/julien/260909-postinstall_vm_debian_jux_v2.sh julien 2>&1 \
| tee /home/julien/260909-postinstall.log
```

L'argument `julien` est nécessaire : sans `sudo`, le script ne peut pas deviner pour quel utilisateur il travaille.

| Étape | Contenu                                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------|
| 0     | GRUB : <code>nomodeset</code> ajouté, <code>quiet</code> retiré, fichier d'origine sauvegardé en <code>.avant-nomodeset-260909</code> |
| 1     | Locale <code>fr_FR.UTF-8</code> , clavier français, fuseau Europe/Paris                                                               |
| 2-3   | Mise à jour, socle ( <code>sudo</code> , <code>curl</code> , <code>git</code> , <code>ripgrep</code> ...)                             |
| 4     | XFCE                                                                                                                                  |
| 5     | xrdp + xorgxrdp, <code>~/.xsession</code> , règle polkit                                                                              |
| 6     | Firefox ESR, Thunderbird, FileZilla, Calibre (+ paquets de langue)                                                                    |
| 7     | Node.js, puis Claude Code via npm avec préfixe <code>~/.npm-global</code>                                                             |
| 8     | Syncthing, ufw (22, 3389, 22000, 21027)                                                                                               |

**L'interface web de Syncthing n'écoute qu'en local** (`127.0.0.1:8384`). L'exposer sur le réseau donnerait un accès sans mot de passe à tout le contenu synchronisé.

# 7. LightDM — ce qui est réellement installé

“ **Correction de la V1.** Elle affirmait « XFCE **sans gestionnaire de connexion** », en présentant comme un fait ce qui n'était qu'une intention de conception. **C'est faux.** »

Constaté le 09/09 après redémarrage :

```
lightdm.service – active (running), enabled
Paquets : lightdm, lightdm-gtk-greeter, light-locker
Cible systemd par défaut : graphical.target
```

LightDM a été tiré comme dépendance par le méta-paquet XFCE. L'étape 4 du script ne l'a jamais empêché.

## Décision retenue : on garde LightDM

Le raisonnement, contre-intuitif mais solide : **la console VMM de DSM est du noVNC** — c'est-à-dire un accès navigateur qui existe déjà, sans rien installer.

|                       | Ce que la console DSM affiche dans le navigateur       |
|-----------------------|--------------------------------------------------------|
| Avec LightDM (retenu) | écran de connexion, puis <b>le bureau XFCE complet</b> |
| Sans LightDM          | jux-debian login: — un terminal texte                  |

Garder LightDM donne donc « navigateur → bureau graphique » gratuitement. Coût : environ 150 Mo au repos, sans conséquence sur 3 Go.

## ⚠ La contrainte qui vient avec : une seule session à la fois

C'est exactement le conflit que la V1 disait vouloir éviter, et il est réel.

xrdp fonctionne ici en backend **Xorg**. Si une session XFCE est déjà ouverte sur `:0` (la console) et qu'on lance une session RDP avec le même utilisateur, on tombe sur « session déjà ouverte » : écran noir ou reprise erratique. `light-locker` ajoute sa part — la session RDP peut revenir verrouillée sans moyen commode de la déverrouiller.

**Console ou RDP, jamais les deux.** Se déconnecter proprement de XFCE (*Déconnexion*, pas seulement fermer la fenêtre VMM) avant de basculer d'un accès à l'autre.

## Si l'on change d'avis

Revenir au design initialement prévu tient en une commande, réversible :

```
sudo systemctl disable --now lightdm && sudo systemctl set-default multi-user.target
```

## 8. Accès

### Bureau à distance — RDP

```
mstsc /v:192.168.1.10
```

Accepter l'avertissement de certificat (auto-signé), puis sur l'écran xrdp : **Session =** `Xorg`, utilisateur `julien`. Depuis un mobile : *Microsoft Remote Desktop*, même adresse.

C'est l'accès le plus fluide pour travailler — nettement plus réactif que le noVNC de DSM.

### Bureau dans le navigateur — console DSM

`http://192.168.1.82:5000` → **Virtual Machine Manager** → `debian-jux` → **Connecter**.

Fonctionne depuis n'importe quel navigateur du réseau local, sans client à installer. Utile aussi quand le réseau de la VM est en panne : la console voit l'écran même si SSH et RDP ne répondent plus.

Ce n'est pas un lien direct — il faut passer par DSM. Pour un vrai signet vers le bureau, la piste est **Apache Guacamole** (passerelle RDP/VNC vers HTML5) sur le VPS Jux, mais **seulement après WireGuard** : exposé seul sur Internet, Guacamole devient la porte d'entrée de la VM.

# Réservation DHCP — faite le 09/09

L'adresse est **réservée sur la Freebox v8 (Ultra)**, passerelle `192.168.1.254` :

```
Paramètres de la Freebox → DHCP → Baux Statiques  
jux-debian 02:11:32:25:C6:E8 → 192.168.1.10
```

Éprouvée au redémarrage du §9 : la VM est revenue sur `.10`.

“ Si un jour la VM réapparaît sur une autre adresse sans raison, le premier réflexe est de comparer la MAC réelle (`ip link show ens3`) à celle du bail : la recreation de la carte réseau dans VMM en génère une nouvelle.

## RustDesk — écarté

L'infrastructure existe pourtant sur le VPS Jux (hbbs/hbbr, client enregistré, diode verte) mais n'a jamais donné de session exploitable — il échoue silencieusement, sans rien laisser d'exploitable pour diagnostiquer.

Pour le nomadisme, la voie retenue reste **WireGuard sur le VPS Jux** + RDP : le diagnostic y est binaire (`wg show` affiche un handshake, ou rien), les clients sont natifs partout, aucun port n'est à ouvrir sur la box — c'est la VM qui monte le tunnel — et le même tunnel sert pour SSH et pour DSM.

## Interface Syncthing

```
ssh -L 8385:127.0.0.1:8384 julien@192.168.1.10
```

puis `http://127.0.0.1:8385`.

Le port local est **8385** et non 8384 : le Syncthing du PC NEXTE occupe déjà 8384. Les deux interfaces sont identiques à l'écran — vérifier le nom de l'appareil en haut à droite (`jux-debian`) avant de déclarer un partage.

---

# 9. Redémarrage — procédure et validation

## Pourquoi ça ne va pas de soi

Le 09/09, la VM a tourné toute la journée sur un démarrage de 14:51 obtenu par **édition manuelle de GRUB** (`Ctrl+X`). Tout ce qui devait survivre à un redémarrage — GRUB persistant, ufw, configuration xrdp — a été écrit **après** ce démarrage, entre 15:02 et 15:12.

Autrement dit : la machine fonctionnait, mais **rien ne prouvait qu'elle refonctionnerait**. Une VM qui marche n'est pas une VM qui redémarre.

## La procédure

**Avant** — deux préalables, dans cet ordre :

1. **Réserver l'adresse dans le DHCP de la box** (§8). Sans ça, on teste le redémarrage en gardant le seul facteur capable de faire perdre la machine.
2. **Vérifier** `Action → Modifier → Autres` : *Autostart = Oui, Démarrer depuis = Disque virtuel*, et surtout **ISO démontée** (§4).

**Pendant :**

3. Ouvrir la **console** (bouton *Connecter*) et la garder à l'écran.
4. `Action → **Éteindre**` — c'est l'arrêt ACPI propre, l'équivalent d'un appui bref sur le bouton d'alimentation. **Pas** `Forcer l'arrêt`, qui est la coupure de courant brutale, à réserver aux machines qui ne répondent plus.
5. Attendre le statut `Arrêté`, puis `Action → **Mettre sous tension**`.

**Le point de contrôle à l'écran** : les messages du noyau doivent défiler **au-delà** de `ACPI: bus type drm_connector registered`. S'ils s'arrêtent là, `nomodeset` n'a pas tenu — voir §5.

“ Un instantané préalable (`Action → Prendre un instantané`) est possible et se supprime ensuite sans toucher la VM. Il n'a pas été jugé utile ici : la machine n'avait qu'un jour et tout son contenu est reproductible par le script du §6. Il prendra son sens quand elle contiendra des comptes Thunderbird et des données Syncthing.

# Le contrôle après redémarrage

Depuis le PC NEXTE, en PowerShell (voir la remarque sur `ssh.exe` au §11) :

```
who -b ; cat /proc/cmdline
for u in ssh xrdp xrdp-sesman syncthing@julien ufw; do \
    printf "%-18s %-10s %s\n" "$u" "$(systemctl is-active $u)" "$(systemctl is-enabled $u)";
done
systemctl --failed --no-pager
ip -4 addr show ens3 | grep inet
systemd-analyze
```

`is-enabled` compte autant que `is-active`. Un service `active` mais `disabled` fonctionne aujourd'hui et disparaît au prochain démarrage — c'est précisément le genre d'écart qu'un test de redémarrage existe pour révéler.

## Vérifier que le pare-feu filtre vraiment

`ufw status` exige root. On peut s'en passer : depuis le PC, comparer un port autorisé et un port qui ne l'est pas.

```
Test-NetConnection 192.168.1.10 -Port 22
Test-NetConnection 192.168.1.10 -Port 12345
```

- **refus immédiat** (quelques dizaines de ms) sur le port fermé = aucun filtrage, les paquets atteignent la pile réseau qui répond `RST`
- **expiration longue** (plus de 20 s) = politique `DROP` active, les paquets sont ignorés sans réponse

Mesuré le 09/09 : **16 ms** sur le port 22, **21 501 ms** sur le port 12345. Le pare-feu filtre.

## Résultat du test — 09/09/2026, 16:43

| Point de contrôle                                     | Constaté                                                                            |
|-------------------------------------------------------|-------------------------------------------------------------------------------------|
| <code>/proc/cmdline</code>                            | <code>ro nomodeset</code> , pas de <code>quiet</code> — la config persistante tient |
| <code>ssh, xrdp, xrdp-sesman, syncthing@julien</code> | <code>active</code> + <code>enabled</code>                                          |
| <code>ufw</code>                                      | <code>active</code> + <code>enabled</code> , et filtre réellement                   |
| Unités en échec                                       | aucune                                                                              |

| Point de contrôle               | Constaté                                         |
|---------------------------------|--------------------------------------------------|
| Erreurs prioritaires au journal | aucune                                           |
| Adresse                         | 192.168.1.10 — via la réservation                |
| Durée                           | 19,7 s (5,7 s noyau + 13,9 s espace utilisateur) |
| RAM au repos                    | 360 Mo sur 2978                                  |

**La question ufw laissée ouverte par la V1 est close** : il n'y avait aucun bug. Le service apparaissait `inactive` parce qu'il avait été installé après le démarrage en cours. Un redémarrage suffisait à le montrer.

## 10. Reste à faire

1. **MCP BookStack perso**, à déclarer **dans la VM** — jeton à créer sur `bookstack.juxjux.ovh` (profil → Jetons d'API ; le secret n'est affiché qu'une fois) :

```
claude mcp add -s user bookstack-jux \  
-e BOOKSTACK_BASE_URL=https://bookstack.juxjux.ovh/api \  
-e BOOKSTACK_API_TOKEN=ID:SECRET \  
-e MCP_TRANSPORT=stdio -- npx -y bookstack-mcp-server
```

Le suffixe `/api` est obligatoire, le jeton s'écrit `id:secret`. **Jamais sur le PC NEXTE** — c'est ce que la VM existe pour empêcher.

2. **Comptes Thunderbird.**
3. **WireGuard** sur le VPS Jux `51.77.141.54`.
4. (*différé*) **Apache Guacamole**, après WireGuard, si le lien navigateur direct se confirme utile.
5. **Publier cette procédure** dans `bookstack.juxjux.ovh`, depuis le Claude Code de la VM — donc après le point 1.

*Points clos depuis la V1 : réservation DHCP (§8) et vérification ufw (§9).*

## 11. À retenir pour toute future VM Linux sur ce NAS

- `nomodeset` **est obligatoire** avec la carte `vmvga` de VMM sous Debian 13. Sans lui, la machine paraît plantée alors qu'elle fonctionne parfaitement — et il ne coûte rien : le bureau graphique s'affiche quand même.
- **Ne jamais laisser `quiet`** : c'est la différence entre un diagnostic en deux minutes et une heure de tâtonnements.
- **Une correction passée dans GRUB au clavier ne survit pas au redémarrage.** Elle n'est acquise qu'écrite dans `/etc/default/grub` puis suivie d'`update-grub`.
- **Démonter l'ISO d'installation**, et le vérifier. Le symptôme est masqué par l'ordre de démarrage ; le danger ne se manifeste que le jour où le disque ne démarre plus.
- **Désactiver le port série** : inutile, et il oriente le diagnostic sur une fausse piste.
- **Mode graphique pour l'installateur**, à cause du clavier de la console noVNC.
- **VirtIO SCSI** → `/dev/sda`, jamais `/dev/vda`.
- **Redémarrer pour de vrai avant de déclarer une machine en service.** Tant qu'elle n'a pas subi un cycle complet, on ne sait pas si elle est configurée — seulement qu'elle marche.
- **Vérifier ce qui est installé, pas ce qu'on a voulu installer.** LightDM était là malgré une procédure qui affirmait le contraire. `systemctl status` et `dpkg -l` tranchent ; la documentation, non.
- **Depuis le PC NEXTE, SSH passe par `%WINDIR%\System32\OpenSSH\ssh.exe` en PowerShell**, jamais par le `ssh` de Git Bash qui ne parle pas au ssh-agent de Windows. Et pour tout script distant un peu long, l'encoder en base64 — PowerShell 5.1 détruit les guillemets internes.

---

Revision #4

Created 2026-09-09 14:08:04 UTC by Julien

Updated 2026-09-09 15:15:44 UTC by Julien