

NOUVEAU

GUIDE PRATIQUE

VOLUME 2

XPRO**X**MO**X**



cloud-init

DEPLOYER UNE VM PROXMOX AVEC UN SCRIPT

CLOUD-INIT

L'acquisition des compétences par l'exemple

POINTS CLES DU TUTORIEL



Création du script d'automatisation de déploiement pas à pas



Modification des droits sur le script (exécution)



Lancement du déploiement automatisé de la machine virtuelle



Connexion à la session utilisateur et test de ping



Les différentes options figurant dans le menu personnalisé

Note :

Ce guide a été testé avec © Proxmox 9.1.1 et Debian 13.2 (« Trixie »).

L'objectif de ce guide est de déployer une machine virtuelle avec Cloud-Init mais, cette fois, en utilisant un **script qui automatise le déploiement d'une machine à partir d'un template**. **Pour cela, il faut avoir obligatoirement suivi et réalisé le 1^{er} volet de ces guides de déploiement.**

1 – PREPARATION DU SCRIPT DE DEPLOIEMENT AUTOMATISE

1^{ère} étape : création du script de déploiement sur Proxmox

- Connectez-vous à l'interface de gestion de votre hyperviseur © Proxmox (nous travaillerons avec l'utilisateur « root » pour simplifier les commandes)
- Dans le dossier personnel du « root », créez un fichier « **createvm-debian13-2.sh** » avec la commande suivante :

nano createvm-debian13-2.sh

Copiez et collez le contenu du script suivant (**attention, respectez la présentation !**) :

```
#!/bin/bash

# =====
# GESTION DES VMs DEBIAN 13 POUR SIO2 (GUI dialog)
# =====

TEMPLATE_ID=102      # Template Debian 13 fonctionnel
STORAGE="STOCKAGE-VMs"  # ZFS
BRIDGE="lanSIO2"      # Bridge Proxmox
SNIPPET_DIR="/var/lib/vz/snippets"

# VMs protégées (pfSense, infra, template...)
PROTECTED_VMS=(100 101 102)

# VLAN -> nom étudiant par défaut
declare -A STUDENTS_MAP=(
  [10]="alpha"
  [20]="ben"
  [30]="zac"
  [40]="kerian"
  [50]="paulc"
  [60]="hugo"
  [70]="paulr"
  [80]="malo"
  [100]="prof"
)

VLAN_ORDER=(10 20 30 40 50 60 70 80 100)

mkdir -p "$SNIPPET_DIR"

# Vérifier que dialog est installé
if ! command -v dialog >/dev/null 2>&1; then
  echo "Installation de dialog..."
```

```
apt update && apt install -y dialog
fi
```

----- Fonction utilitaire : VM protégée ? -----

```
is_protected() {
    local vmid="$1"
    for p in "${PROTECTED_VMS[@]"; do
        if [[ "$vmid" == "$p" ]]; then
            return 0
        fi
    done
    return 1
}
```

----- Choix VLAN via menu dialog -----

```
choose_vlan() {
    local menu_items=()
    for vlan in "${VLAN_ORDER[@]"; do
        menu_items+=("$vlan" "${STUDENTS_MAP[$vlan]})"
    done

    VLAN=$(dialog --clear --stdout \
        --title "Sélection du VLAN / élève" \
        --menu "Choisis un VLAN :" 18 60 10 \
        "${menu_items[@]})"

    echo "$VLAN"
}
```

----- Génération du snippet cloud-init -----

```
create_cloudinit_yaml() {
    local VMNAME="$1"
    local USER="$2"
    local PASS="$3"
    local SSHPORT="$4"

    local YAML="${SNIPPET_DIR}/${VMNAME}.yaml"

    cat > "$YAML" <<EOF
```

#cloud-config

```
hostname: $VMNAME
fqdn: $VMNAME.sio.local

users:
- name: $USER
  sudo: ALL=(ALL) NOPASSWD:ALL
  groups: sudo
  shell: /bin/bash
  lock_passwd: false
  passwd: $(openssl passwd -6 "$PASS")

chpasswd:
  expire: false
```

```
ssh_pwauth: true
```

```
write_files:
```

```
- path: /etc/ssh/sshd_config.d/port.conf
  content: |
    Port $SSHPORT
```

```
runcmd:
```

```
- systemctl restart ssh
```

```
EOF
```

```
echo "$YAML"
}
```

----- Création d'une VM -----

```
create_vm() {
```

```
  # VLAN
```

```
  VLAN=$(choose_vlan)
```

```
  [[ -z "$VLAN" ]] && return
```

```
  local default_user="${STUDENTS_MAP[$VLAN]}"
```

```
  # Nom utilisateur
```

```
  USER=$(dialog --clear --stdout \
```

```
    --title "Nom utilisateur" \
```

```
    --inputbox "Nom de l'utilisateur Linux :" 10 50 "$default_user")
```

```
  [[ -z "$USER" ]] && return
```

```
  # Mot de passe
```

```
  PASS=$(dialog --clear --stdout \
```

```
    --title "Mot de passe" \
```

```
    --insecure --passwordbox "Mot de passe de $USER :" 10 50)
```

```
  [[ -z "$PASS" ]] && return
```

```
  # Démarrer après création ?
```

```
  START_CHOICE=$(dialog --clear --stdout \
```

```
    --title "Démarrage" \
```

```
    --menu "Démarrer la VM après création ?" 10 50 2 \
```

```
    "yes" "Oui" \
```

```
    "no" "Non")
```

```
  [[ -z "$START_CHOICE" ]] && return
```

```
  VMNAME="debian13-${USER}"
```

```
  SSHPORT="22${VLAN}"
```

```
  # VMID auto
```

```
  VMID=$(pvesh get /cluster/nextid)
```

```
  dialog --clear --infobox "Clonage du template (VMID $VMID)..." 6 50
```

```
  qm clone "$TEMPLATE_ID" "$VMID" --name "$VMNAME" --full true --storage "$STORAGE"
```

```
  # Disque : redimensionner à 20G
```

```
  qm resize "$VMID" scsi0 20G >/dev/null 2>&1
```

```
  # Réseau
```

```
  qm set "$VMID" --net0 virtio,bridge="$BRIDGE",tag="$VLAN" >/dev/null 2>&1
```

Cloud-init de base

```
qm set "$VMID" --ide2 "$STORAGE":cloudinit >/dev/null 2>&1
qm set "$VMID" --agent enabled=1 >/dev/null 2>&1
qm set "$VMID" --ipconfig0 "ip=dhcp" >/dev/null 2>&1
```

Snippet cloud-init personnalisé

```
YAML_FILE=$(create_cloudinit_yaml "$VMNAME" "$USER" "$PASS" "$SSHPORT")
qm set "$VMID" --cicustom "user=local:snippets/${basename "$YAML_FILE"}" >/dev/null 2>&1
```

Démarrer ou non

```
if [[ "$START_CHOICE" == "yes" ]]; then
    qm start "$VMID" >/dev/null 2>&1
    START_TXT="Oui"
else
    START_TXT="Non"
fi
```

```
dialog --clear --msgbox "VM créée avec succès !\n\nID VM : $VMID\nNom : $VMNAME\nVLAN : $VLAN\nSSH : port $SSHPORT\nDémarrée : $START_TXT" 12 60
}
```

----- Supprimer une VM -----

```
delete_vm() {
    # Construire la liste des VMs
    local menu_items=()
    while read -r line; do
        [[ "$line" == VMID* ]] && continue
        local id name state icon label
        id=$(echo "$line" | awk '{print $1}')
        name=$(echo "$line" | awk '{print $2}')
        state=$(echo "$line" | awk '{print $3}')

        if is_protected "$id"; then
            icon="🔒"
        else
            icon=""
        fi

        label="$icon $name ($state)"
        menu_items+=("$id" "$label")
    done <<(qm list)

    if [[ ${#menu_items[@]} -eq 0 ]]; then
        dialog --clear --msgbox "Aucune VM QEMU trouvée." 8 50
        return
    fi

    SEL=$(dialog --clear --stdout \
        --title "Suppression de VM" \
        --menu "Sélectionne une VM à supprimer :\n🔒 = protégée" 20 70 10 \
        "${menu_items[@]}")

    [[ -z "$SEL" ]] && return

    if is_protected "$SEL"; then
```

```

    dialog --clear --msgbox "❌ La VM $SEL est protégée (🔒) et ne peut pas être supprimée." 8 70
    return
fi

```

```

NAME=$(qm config "$SEL" | awk '/^name:/{print $2}')

```

```

dialog --clear --yesno "Supprimer définitivement la VM $SEL ($NAME) ?" 8 60
[[ $? -ne 0 ]] && return

```

Arrêt propre puis forcé si besoin

```

STATE=$(qm status "$SEL" | awk '{print $2}')
if [[ "$STATE" == "running" ]]; then
    qm shutdown "$SEL" >/dev/null 2>&1
    for i in {1..10}; do
        sleep 1
        STATE2=$(qm status "$SEL" | awk '{print $2}')
        [[ "$STATE2" == "stopped" ]] && break
    done
    STATE2=$(qm status "$SEL" | awk '{print $2}')
    if [[ "$STATE2" == "running" ]]; then
        qm stop "$SEL" --force >/dev/null 2>&1
    fi
fi

```

```

qm destroy "$SEL" --purge >/dev/null 2>&1
rm -f "${SNIPPET_DIR}/${NAME}.yaml"

```

```

dialog --clear --msgbox "VM $SEL ($NAME) supprimée avec succès." 8 60
}

```

----- Lister les VMs -----

```

list_vms() {
    local out="Liste des VMs QEMU :\n\n"
    while read -r line; do
        [[ "$line" == VMID* ]] && continue
        local id name state icon
        id=$(echo "$line" | awk '{print $1}')
        name=$(echo "$line" | awk '{print $2}')
        state=$(echo "$line" | awk '{print $3}')
        if is_protected "$id"; then
            icon="🔒 "
        else
            icon=" "
        fi
        out+="$icon VMID $id → $name → $state\n"
    done <<(qm list)
}

```

```

dialog --clear --msgbox "$out" 25 80
}

```

===== MENU PRINCIPAL =====

```

while true; do
    CHOICE=$(dialog --clear --stdout \
        --title "CREATION VM DEBIAN 13.2 - SIO2" \
        --menu "Choisis une action :" 18 70 10 \
        "1" "Créer une VM élève" \

```

```
"2" "Supprimer une VM" \  
"3" "Lister les VMs" \  
"4" "Quitter")
```

```
case "$CHOICE" in  
  1) create_vm ;;  
  2) delete_vm ;;  
  3) list_vms ;;  
  4|*) clear; exit 0 ;;  
esac  
done
```

- Quittez et sauvegardez le script avec **CTRL + X + O** et pressez la touche **Entrée**

2^{ème} étape : donner les droits d'exécution sur le script

- Saisissez la commande suivante pour donner les droits d'exécution sur le script :

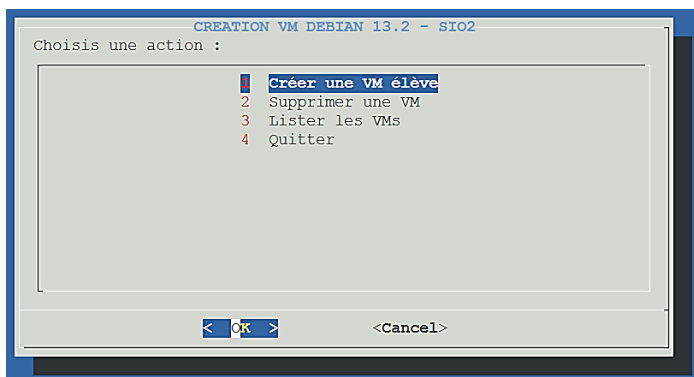
chmod +x createvm-debian13-2.sh

3^{ème} étape : lancement du script

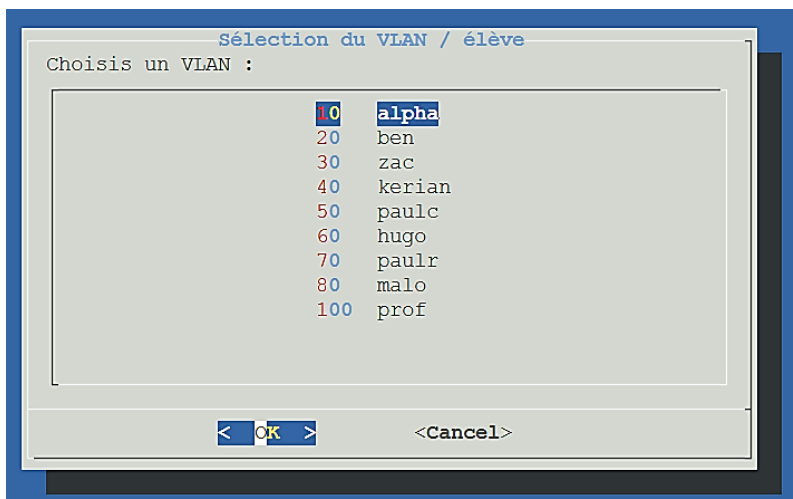
- Lancez l'exécution du script avec la commande suivante :

./createvm-debian13-2.sh

Le script se lance et affiche un menu :



- Sélectionnez l'option « **Créer une VM élève** » et pressez la touche Entrée
- Sélectionnez un élève dans la liste proposée et pressez la touche Entrée :



- Saisissez le nom d'utilisateur de la future machine Debian et pressez la touche Entrée :

Nom utilisateur

Nom de l'utilisateur Linux :

alpha

< OK > <Cancel>

- Saisissez un mot de passe pour l'utilisateur et pressez la touche Entrée :

Mot de passe

Mot de passe de alpha :

< OK > <Cancel>

- Sélectionnez « Oui » pour que la VM démarre automatiquement et pressez la touche Entrée :

Démarrage

Démarrer la VM après création ?

yes Oui

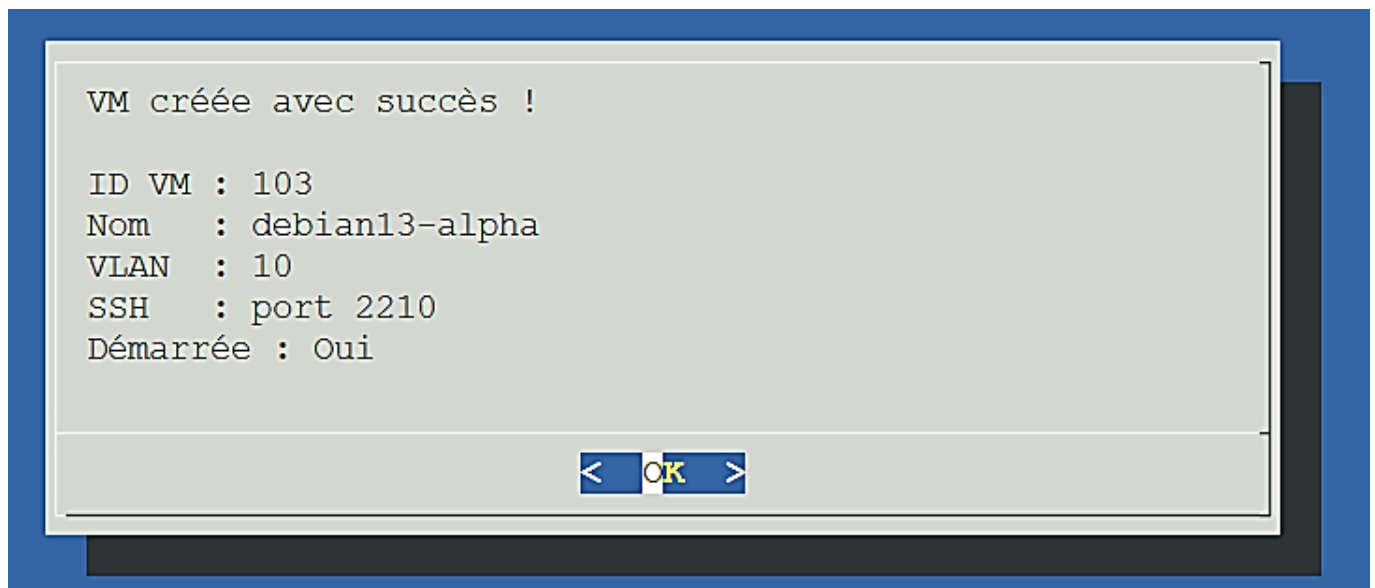
no Non

< OK > <Cancel>

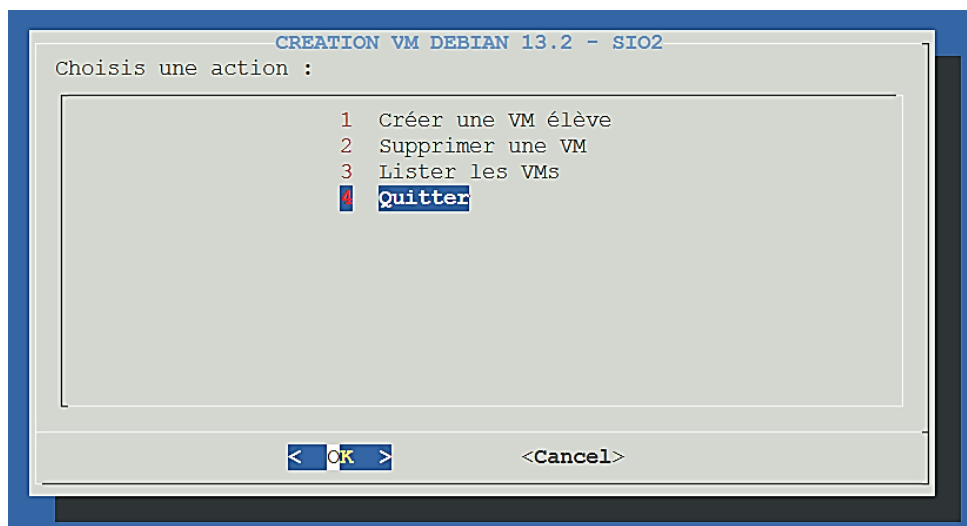
- Patientez pendant le déploiement de la VM (assez rapide) :

```
transferred 16.5 GiB of 20.0 GiB (82.48%)
transferred 16.7 GiB of 20.0 GiB (83.49%)
transferred 16.9 GiB of 20.0 GiB (84.49%)
transferred 17.1 GiB of 20.0 GiB (85.50%)
transferred 17.3 GiB of 20.0 GiB (86.50%)
transferred 17.5 GiB of 20.0 GiB (87.51%)
transferred 17.7 GiB of 20.0 GiB (88.52%)
transferred 17.9 GiB of 20.0 GiB (89.52%)
transferred 18.1 GiB of 20.0 GiB (90.53%)
transferred 18.3 GiB of 20.0 GiB (91.53%)
transferred 18.5 GiB of 20.0 GiB (92.54%)
transferred 18.7 GiB of 20.0 GiB (93.54%)
transferred 18.9 GiB of 20.0 GiB (94.55%)
transferred 19.1 GiB of 20.0 GiB (95.56%)
transferred 19.3 GiB of 20.0 GiB (96.56%)
transferred 19.5 GiB of 20.0 GiB (97.57%)
transferred 19.7 GiB of 20.0 GiB (98.57%)
transferred 19.9 GiB of 20.0 GiB (99.58%)
transferred 20.0 GiB of 20.0 GiB (100.00%)
```

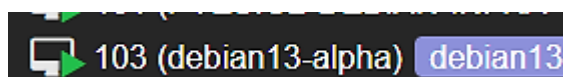
Une fois la VM déployée, vous obtenez le message suivant (ID VM, Nom, VLAN et génération automatique d'un numéro de port SSH qui correspond à 22 + le numéro du TAG VLAN) :



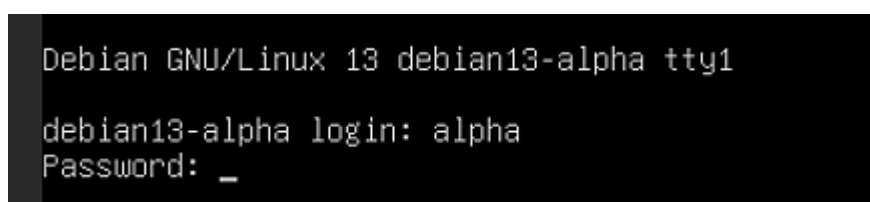
- Sélectionnez le menu « **Quitter** » et pressez la touche Entrée :



La machine virtuelle Debia 13 pour l'élève a été déployée rapidement et s'affiche dans l'interface de gestion de Proxmox :



- Saisissez l'identifiant et le mot de passe que vous avez indiqués lors du déploiement et pressez la touche Entrée pour vous connecter à la session :



- Saisissez la commande « **ip a** » pour vérifier que votre machine virtuelle a bien reçu une adresse dynamique de votre serveur DHCP pfSENSE :

```
Linux debian13-alpha 6.12.57+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.57-1 (2025-11-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
alpha@debian13-alpha:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:e6:57:1e brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    altname enxbc2411e6571e
    inet 10.0.10.108/24 brd 10.0.10.255 scope global dynamic noprefixroute ens18
        valid_lft 7125sec preferred_lft 6225sec
    inet6 fe80::e54:fc6b:a468:d79d/64 scope link
        valid_lft forever preferred_lft forever
alpha@debian13-alpha:~$
```

- Faites un test de ping pour vous assurer que la connexion Internet est fonctionnelle :

```
alpha@debian13-alpha:~$ ping -c5 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=9.67 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=8.63 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=9.45 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=55 time=9.43 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=55 time=8.77 ms

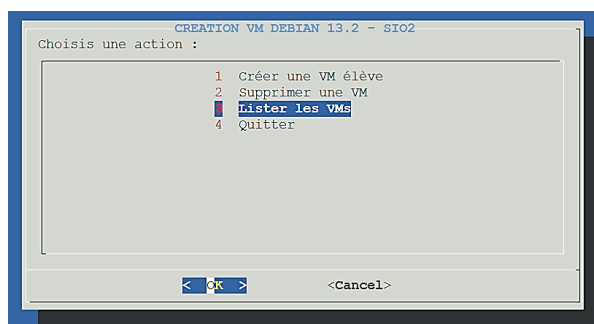
--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 8.630/9.188/9.670/0.411 ms
alpha@debian13-alpha:~$
```

La machine est parfaitement fonctionnelle !

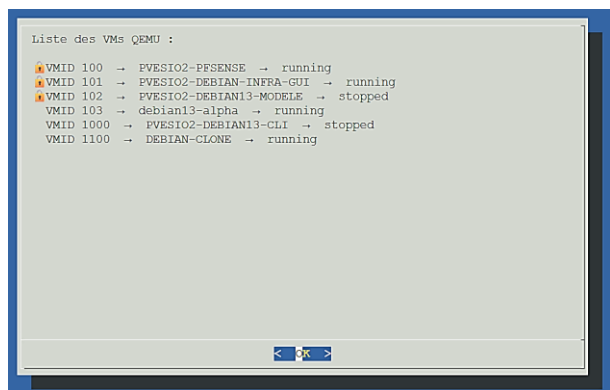
2 – LES POSSIBILITES OFFERTES PAR LE SCRIPT (personnalisation du menu)

LISTER LES VMs PRESENTENT SUR L'HYPERVISEUR

- Relancez le script avec la commande « **./createvm-debian13-2.sh** »
- Dans le menu, sélectionnez l'option 3 « **Lister les VMs** » et pressez la touche Entrée :

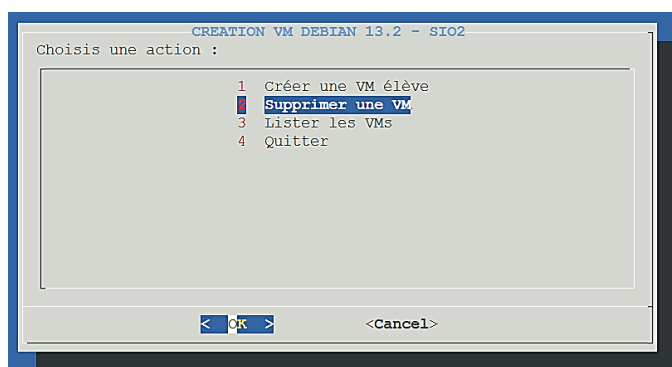


Les VMs s'affichent. On notera que certaines VMs sont protégées par une suppression accidentelle et elles apparaissent avec un petit cadenas :

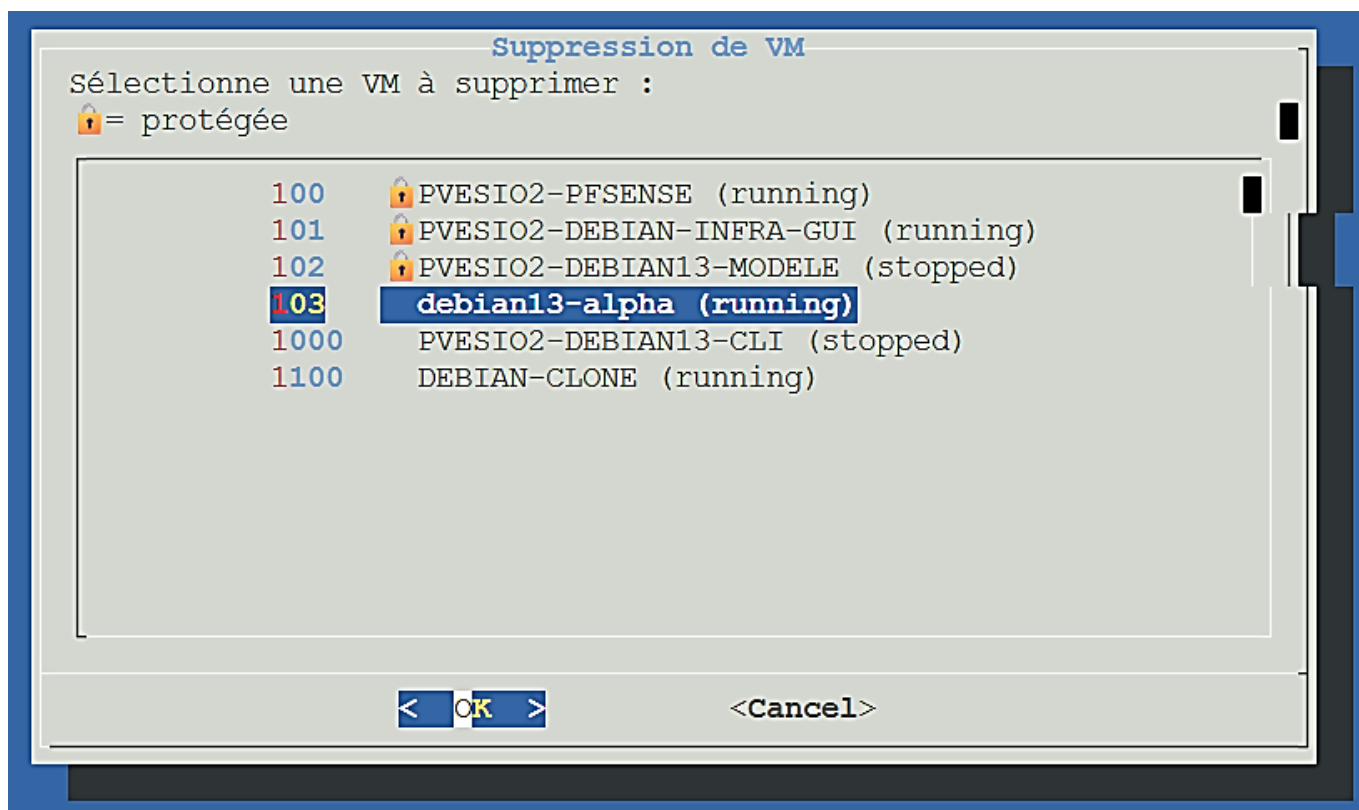


SUPPRIMER UNE VM

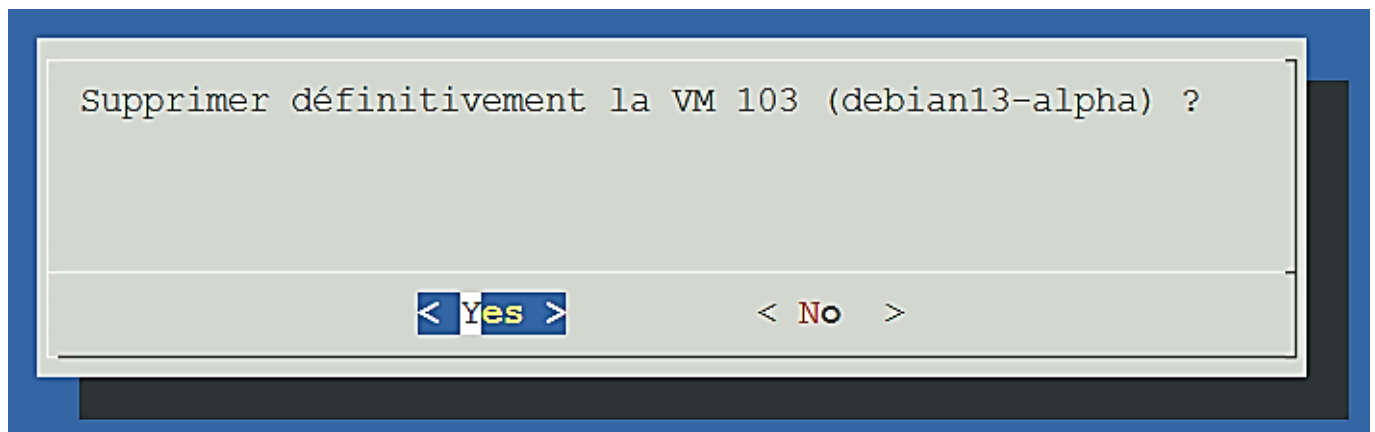
- Sélectionnez l'option 2 « **Supprimer une VM** » et pressez la touche Entrée :



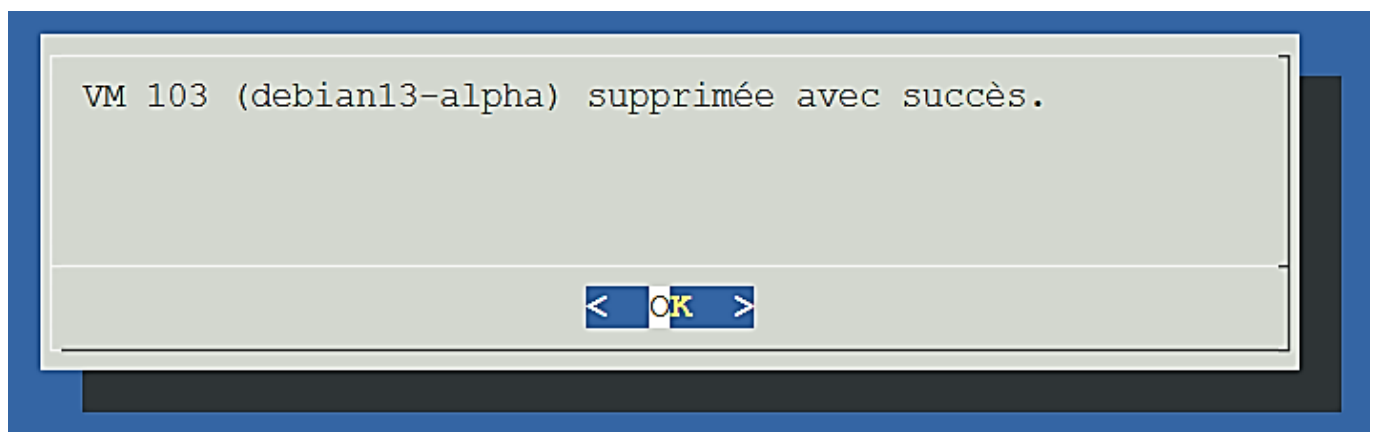
- Sélectionnez la VM à supprimer et pressez la touche Entrée :



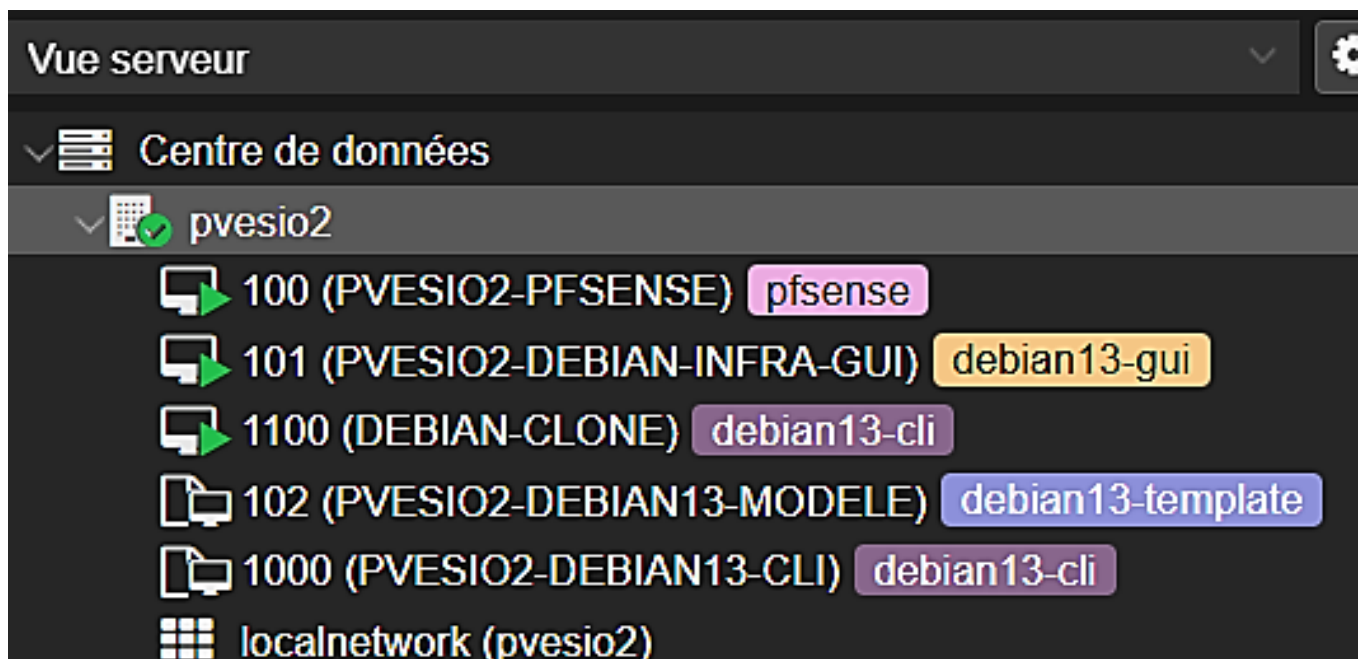
Un message demande la confirmation de suppression de la VM ; pressez la touche Entrée :



Un message indique que la VM a bien été supprimée ; pressez la touche Entrée :



La VM a été supprimée de l'hyperviseur :



Il ne vous reste plus qu'à poursuivre la personnalisation du menu et d'ajouter de nouvelles possibilités lors du déploiement comme la création de toutes les VMs en bloc, l'ajout d'un service dans la VM (Apache), etc.

Remarque : pensez à conserver une copie du script qui fonctionne avant de le modifier !