Template Proxmox

```
mkdir -p /root/provisioning/{ca,sshd}

# 1. Le fichier de déploiement des CAs (Yubikey + emergency concaténées)
scp ton-poste:~/mis-ssh-ca/mis-users-ca-deploy.pub
/root/provisioning/ca/mis-users-ca.pub

# 2. Le KRL initial vide (signé par ta CA Yubikey sur ton poste)
scp ton-poste:/tmp/mis-revoked-keys-empty /root/provisioning/ca/mis-revoked-
keys

# 3. Configuration sshd
cat > /root/provisioning/sshd/99-mis-ca.conf <<'EOF'
# === MIS SSH CA Configuration ===
TrustedUserCAKeys /etc/ssh/mis-users-ca.pub
RevokedKeys /etc/ssh/mis-revoked-keys
PasswordAuthentication no
KbdInteractiveAuthentication no
PubkeyAuthentication yes
PermitRootLogin no
EOF
```

```
#!/usr/bin/env bash
# /root/provisioning/build-debian13-mis-template.sh
# Construit la golden image MIS Debian 13 avec CA SSH intégrée
# Compatible avec tous types de storage Proxmox (local, lvm, lvm-thin, zfs,
etc.)
set -euo pipefail

# =====================
# Variables
# =====================
VMID=9013
DEBVER=13
DEBCODENAME="trixie"
STORAGE="local"                                # ← change ici si tu as un autre
storage
BRIDGE="vmbr0"
IMG="debian-${DEBVER}-genericcloud-amd64.qcow2"
IMG_URL="https://cloud.debian.org/images/cloud/${DEBCODENAME}/latest/${IMG}"
TEMPLATE_NAME="debian${DEBVER}-mis-template"
PROVISIONING_DIR="/root/provisioning"

# =====================
# Sanity checks
# =====================
echo ">>> Vérification des fichiers de provisioning"
for f in \
```

```
    "${PROVISIONING_DIR}/ca/mis-users-ca.pub" \
    "${PROVISIONING_DIR}/ca/mis-revoked-keys" \
    "${PROVISIONING_DIR}/sshd/99-mis-ca.conf"; do
    [[ -f "$f" ]] || { echo "ERREUR: fichier manquant $f"; exit 1; }
done

echo ">>> Vérification du storage ${STORAGE}"
if ! pvesm status --storage "${STORAGE}" &>/dev/null; then
    echo "ERREUR: storage '${STORAGE}' introuvable"
    echo "Storages disponibles :"
    pvesm status
    exit 1
fi

# Vérifier que le storage accepte les images de VM
STORAGE_CONTENT=$(pvesm status --storage "${STORAGE}" --content images -v
2>/dev/null || true)
if ! pvesm status -content images | awk 'NR>1 {print $1}' | grep -qx
"${STORAGE}"; then
    echo "ERREUR: le storage '${STORAGE}' n'accepte pas les images VM"
    echo "Active 'Disk image' dans Datacenter → Storage → ${STORAGE} → Edit"
    exit 1
fi

if qm status ${VMID} &>/dev/null; then
    echo "ATTENTION: VMID ${VMID} existe déjà."
    read -p "Supprimer la VM/template existant ? [y/N] " -n 1 -r
    echo
    [[ ! $REPLY =~ ^[Yy]$ ]] && exit 1
    qm destroy ${VMID} --purge
fi

# ======================
# Téléchargement de l'image cloud
# ======================
cd /tmp
[[ -f "${IMG}" ]] && rm "${IMG}"
echo ">>> Téléchargement ${IMG}"
wget -q --show-progress "${IMG_URL}"

# ======================
# Préparation libguestfs
# ======================
apt-get install -y libguestfs-tools >/dev/null

# ======================
# Customisation de l'image (golden image MIS)
# ======================
echo ">>> Customisation de l'image (golden image MIS)"

virt-customize -a "${IMG}" \
```

```
  --install qemu-guest-agent,fail2ban,vim,htop,chrony,unattended-
upgrades,curl,sudo,ca-certificates \
  --run-command 'systemctl enable qemu-guest-agent fail2ban chrony
unattended-upgrades ssh' \
  \
  --run-command 'useradd -m -s /bin/bash -G sudo mis-admin' \
  --run-command 'echo "mis-admin ALL=(ALL) NOPASSWD:ALL" >
/etc/sudoers.d/mis-admin' \
  --run-command 'chmod 0440 /etc/sudoers.d/mis-admin' \
  \
  --copy-in "${PROVISIONING_DIR}/ca/mis-users-ca.pub":/etc/ssh/ \
  --copy-in "${PROVISIONING_DIR}/ca/mis-revoked-keys":/etc/ssh/ \
  --copy-in "${PROVISIONING_DIR}/sshd/99-mis-
ca.conf":/etc/ssh/sshd_config.d/ \
  --run-command 'chown root:root /etc/ssh/mis-users-ca.pub /etc/ssh/mis-
revoked-keys /etc/ssh/sshd_config.d/99-mis-ca.conf' \
  --run-command 'chmod 0644 /etc/ssh/mis-users-ca.pub /etc/ssh/mis-revoked-
keys /etc/ssh/sshd_config.d/99-mis-ca.conf' \
  \
  --run-command 'sed -i
"s/^#\?PasswordAuthentication.*/PasswordAuthentication no/"
/etc/ssh/sshd_config' \
  --run-command 'sed -i "s/^#\?PermitRootLogin.*/PermitRootLogin no/"
/etc/ssh/sshd_config' \
  --run-command 'sed -i
"s/^#\?KbdInteractiveAuthentication.*/KbdInteractiveAuthentication no/"
/etc/ssh/sshd_config' \
  \
  --timezone Europe/Brussels \
  \
  --run-command 'apt-get clean && rm -rf /var/lib/apt/lists/*' \
  --run-command 'cloud-init clean --logs || true' \
  --truncate /etc/machine-id \
  --run-command 'rm -f /etc/ssh/ssh_host_*' \
  2>&1 | grep -v "random seed could not be set" || true

echo ">>> Image customisée prête"

# ====================
# Création de la VM template
# ====================
echo ">>> Création VM ${VMID}"

qm create ${VMID} \
  --name "${TEMPLATE_NAME}" \
  --memory 2048 --cores 2 \
  --net0 virtio,bridge="${BRIDGE}" \
  --scsihw virtio-scsi-single \
  --ostype l26 \
  --agent enabled=1,fstrim_cloned_disks=1 \
  --cpu host
```

```
# =====================
# Import du disque sur ${STORAGE} - méthode robuste tous types confondus
# =====================
echo ">>> Import du disque sur le storage '${STORAGE}'"

qm importdisk ${VMID} "${IMG}" "${STORAGE}"

# Récupération du volid réel tel que Proxmox l'a créé
# (la syntaxe diffère entre local, lvm, zfs, etc. donc on parse la config)
DISK_VOLID=$(qm config ${VMID} | awk -F': ' '/^unused[0-9]+:/ {print $2; exit}' | awk '{print $1}')

if [[ -z "${DISK_VOLID}" ]]; then
    echo "ERREUR: impossible de déterminer le volid du disque importé"
    qm config ${VMID}
    exit 1
fi

echo ">>> Disque importé : ${DISK_VOLID}"

# Attache le disque en scsi0
qm set ${VMID} --scsi0 "${DISK_VOLID},discard=on,ssd=1,iothread=1"

# =====================
# Disque cloud-init (sur le même storage que le disque principal)
# =====================
qm set ${VMID} --ide2 "${STORAGE}:cloudinit"

# =====================
# Boot order + console série
# =====================
qm set ${VMID} --boot order=scsi0
qm set ${VMID} --serial0 socket --vga serial0

# =====================
# Tags + description
# =====================
qm set ${VMID} --tags "template,debian13,mis,golden-image"

qm set ${VMID} --description "Golden image MIS - Debian ${DEBVER} (${DEBCODENAME})

Build date : $(date -u +%Y-%m-%d)
Storage    : ${STORAGE}

Includes :
  - mis-admin user (accès via cert SSH MIS CA)
  - MIS Users CA installée (/etc/ssh/mis-users-ca.pub)
  - KRL configuré (/etc/ssh/mis-revoked-keys)
  - qemu-guest-agent, fail2ban, chrony, unattended-upgrades
```

```
  - Hardening SSH (no password, no root, no kbd-interactive)
  - Timezone Europe/Brussels

Recovery (pas de break-glass dans la VM) :
  - Console : qm terminal <vmid>
  - Montage disque : guestmount -d <vmid> -i /mnt/rescue"

# =====================
# Conversion en template
# =====================
echo ">>> Conversion en template"
qm template ${VMID}

# =====================
# Cleanup
# =====================
rm -f "/tmp/${IMG}"

# =====================
# Résumé
# =====================
echo ""
echo "============================================="
echo "Template ${TEMPLATE_NAME} (VMID ${VMID}) prêt"
echo "Storage   : ${STORAGE}"
echo "Disque    : ${DISK_VOLID}"
echo "============================================="
echo ""
echo "Vérification de la config :"
qm config ${VMID}
echo ""
echo "Pour déployer une nouvelle VM cliente :"
echo "  qm clone ${VMID} <NEW_VMID> --name <client-srv01> --full"
echo "  qm set <NEW_VMID> --ipconfig0 ip=X.X.X.X/24,gw=Y.Y.Y.Y"
echo "  qm resize <NEW_VMID> scsi0 32G"
echo "  qm start <NEW_VMID>"
echo ""
```

```
virt-customize -a debian-12-genericcloud-amd64.qcow2 \
  --install qemu-guest-agent,fail2ban,vim,htop,chrony,unattended-
upgrades,curl \
  --run-command 'systemctl enable qemu-guest-agent fail2ban chrony
unattended-upgrades' \
  --run-command 'useradd -m -s /bin/bash -G sudo mis-admin' \
  --run-command 'echo "mis-admin ALL=(ALL) NOPASSWD:ALL" >
/etc/sudoers.d/mis-admin' \
  --ssh-inject mis-admin:file:/root/provisioning/keys/mis-admin.pub \
  --run-command 'sed -i
"s/^#\?PasswordAuthentication.*/PasswordAuthentication no/"
/etc/ssh/sshd_config' \
  --run-command 'sed -i "s/^#\?PermitRootLogin.*/PermitRootLogin no/"
```

```
/etc/ssh/sshd_config' \
  --timezone Europe/Brussels \
  --run-command 'apt-get clean' \
  --truncate /etc/machine-id

# Puis import dans Proxmox comme template
```