# Proxmox: Working Flow to Clone a VM and Auto-Start **Any** Docker Compose App

## What this assumes

* Template **9000** = Debian 12 with Docker + QEMU Guest Agent installed, cloud-init enabled.
* Your **known-good** clone script exists at /root/clone\_from\_deb12\_docker\_base.sh (this sets ciuser, cipassword, sshkey and is why logins work).
* You will inject your app via **vendor-data** (not user-data) so Proxmox’s generated user-data (the login bits) stays intact.

## One-liner (works for any app)

# compose\_clone.sh <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>  
/root/compose\_clone.sh 950 kuma-01 ~/.ssh/id\_ed25519.pub 'YourStrongPass!' /path/to/docker-compose.yml

* Hostname becomes <new\_name>.
* Console + SSH logins work (because auth is from qm set, not overridden).
* Your compose file is written to /opt/app/docker-compose.yml and started on first boot.

## The script you need (drop-in)

Save as /root/compose\_clone.sh and chmod +x /root/compose\_clone.sh.

#!/usr/bin/env bash  
# Clone from a Docker-ready Debian 12 template and attach a compose file via vendor-data  
# without overwriting Proxmox user-data (so ciuser/cipassword/sshkey keep working).  
  
set -euo pipefail  
  
TEMPLATE\_VMID="${TEMPLATE\_VMID:-9000}"  
  
NEWID=${1:?Usage: $0 <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>}  
NEWNAME=${2:?Usage: $0 <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>}  
PUBKEY=${3:?Usage: $0 <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>}  
CIPASS=${4:?Usage: $0 <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>}  
COMPOSE\_SRC=${5:?Usage: $0 <new\_vmid> <new\_name> <pubkey\_path> <PasswordOr-> <compose.yml>}  
NEWDISK=${6:-} # optional: e.g. 40G; '-' to skip  
  
CIUSER=${CIUSER:-dockeruser}  
SNIPPETS\_DIR=/var/lib/vz/snippets  
VENDOR\_SNIP="$SNIPPETS\_DIR/${NEWNAME}-vendor.yaml"  
  
[[ -f "$PUBKEY" ]] || { echo "ERROR: pubkey not found: $PUBKEY"; exit 1; }  
[[ -f "$COMPOSE\_SRC" ]] || { echo "ERROR: compose file not found: $COMPOSE\_SRC"; exit 1; }  
mkdir -p "$SNIPPETS\_DIR"  
  
# Build a vendor-data snippet (DO NOT use as user-data)  
cat > "$VENDOR\_SNIP" <<'YAML'  
#cloud-config  
write\_files:  
 - path: /opt/app/docker-compose.yml  
 permissions: '0644'  
 content: |  
YAML  
  
# indent compose file by 6 spaces to fit under 'content: |'  
awk '{print " " $0}' "$COMPOSE\_SRC" >> "$VENDOR\_SNIP"  
  
cat >> "$VENDOR\_SNIP" <<'YAML'  
  
runcmd:  
 - 'command -v docker >/dev/null || (apt-get update && apt-get install -y docker.io qemu-guest-agent)'  
 - systemctl enable --now docker  
 - docker compose -f /opt/app/docker-compose.yml up -d  
YAML  
  
echo "[\*] Wrote vendor-data: $VENDOR\_SNIP"  
  
echo "[\*] Cloning template $TEMPLATE\_VMID -> $NEWID ($NEWNAME)..."  
qm clone "$TEMPLATE\_VMID" "$NEWID" --name "$NEWNAME" --full  
  
# Optional pre-boot resize  
if [[ -n "${NEWDISK}" && "${NEWDISK}" != "-" ]]; then  
 echo "[\*] Resizing disk to $NEWDISK..."  
 qm resize "$NEWID" scsi0 "$NEWDISK" || true  
fi  
  
echo "[\*] Injecting Proxmox user-data (ciuser/cipassword/sshkey) + DHCP..."  
if [[ "$CIPASS" != "-" ]]; then  
 qm set "$NEWID" --ciuser "$CIUSER" --cipassword "$CIPASS"  
else  
 qm set "$NEWID" --ciuser "$CIUSER"  
fi  
qm set "$NEWID" --sshkey "$PUBKEY"  
qm set "$NEWID" --ipconfig0 ip=dhcp  
  
echo "[\*] Binding vendor-data (compose) WITHOUT touching user-data..."  
qm set "$NEWID" --cicustom "vendor=local:snippets/${NEWNAME}-vendor.yaml"  
  
echo "[\*] Update cloud-init ISO and start (first boot will run compose)..."  
qm cloudinit update "$NEWID"  
qm start "$NEWID"  
  
echo "[✓] Clone $NEWID ($NEWNAME) created; vendor-data applied BEFORE first boot."  
echo " Compose file: $COMPOSE\_SRC"

## Why **vendor-data**?

* **user-data** is replaced if you point it to a file → breaks logins.
* **vendor-data** is merged in as additional → safe for app configs.

## Using it with any app

1. Place your docker-compose.yml somewhere accessible to the node.
2. Run the script:

/root/compose\_clone.sh 960 myapp-01 ~/.ssh/id\_ed25519.pub 'YourStrongPass!' /mnt/pve/pve-qnap/apps/myapp/docker-compose.yml

1. After boot:

docker ps

## Verification

* Host: qm cloudinit dump <id> vendor
* Guest: cloud-init status --long, docker ps