

Cloud Init Templates in Proxmox VE

Die einfachste und effizienteste Art und Weise Linux VMs in Proxmox VE zu deployen ist der Einsatz von sogenannten Cloud-Init Images. Diese Images sind speziell angefertigte Installationsmedien, die von nahezu jeder bekannteren Linux-Distribution wie z.B. Debian, Ubuntu oder auch CentOS angeboten werden. Proxmox VE unterstützt Cloud-Init Deployments bereits seit einigen Versionen.

Vorteile gegenüber .ISO

Vorteile im Vergleich zu ISO Images sind:

- Das Betriebssystem muss nicht manuell installiert werden, sondern fährt nach dem Deployen automatisch hoch.
- Über die Proxmox VE Oberfläche können diverse Parameter je Template-Klon mitgegeben werden.
- Vor dem Deployment kann man mittels diverser Tools wie den libguestfs-tools Anpassungen an den Images vornehmen:
 - Integration von Software-Paketen über den Paketmanager
 - Änderung von Dateien nach eigenem Wunsch vor Deployment
 - Setzen eines Root-Passworts oder das Anlegen eines neuen Users

Cloud Init Parameter innerhalb PVE

Folgende Parameter sind einsehbar innerhalb des angelegten Cloud-Init-Drives in der Proxmox VE Oberfläche:

- User (es wird ein User angelegt)
- Password (es wird ein Passwort gesetzt)
- DNS domain (DNS Domain Override zum PVE-Host Default)
- DNS servers (DNS Server Override zum PVE-Host Default)
- SSH public key (ein eigener persönlicher SSH-Pub Key)
- Upgrade packages yes (das Image wird beim Erst-Start komplett aktualisiert)
- IP Config (net0) (IP-Konfiguration für das erste Netzwerk-Device, Statisch oder DHCP)

Cloud Init Images vorbereiten

Es folgt eine Anlage eines Templates mit der ID 9500, welches die von uns empfohlenen Best-Practices bezüglich einer VM-Anlage in PVE beinhaltet. Einige Parameter müssen noch an Ihre Infrastruktur angepasst werden, zum Beispiel der Pfad, indem das Image heruntergeladen werden soll.

Debian 12 (Bookworm) Cloud-Image

```
apt update  
apt install libguestfs-tools
```

```
cd ~/tmp/  
wget https://cloud.debian.org/images/cloud/bookworm/latest/debian-12-  
generic-amd64.qcow2  
virt-customize -a ~/tmp/debian-12-generic-amd64.qcow2 --install qemu-guest-  
agent  
virt-customize -a ~/tmp/debian-12-generic-amd64.qcow2 --run-command "echo -n  
> /etc/machine-id"
```

Cloud Init Template erzeugen

Erstellen eines Bash Scriptes

```
vim ~/tmp/createDebian12CloudInit_VMtemplate.sh
```

```
# Set the VM ID to operate on  
VMID=900  
# Choose a name for the VM  
TEMPLATE_NAME=Debian12CloudInit  
# Choose the disk image to import  
DISKIMAGE=debian-12-genericcloud-amd64.qcow2  
# Select Host disk  
HOST_DISK=Data  
  
# fetch cloud-init image  
wget https://cloud.debian.org/images/cloud/bookworm/latest/debian-12-  
genericcloud-amd64.qcow2  
  
# Delete old template  
qm destroy $VMID  
# Create the VM  
qm create $VMID --name $TEMPLATE_NAME --net0 virtio,bridge=vmbr60 --pool  
Templates  
# Set the OSType to Linux Kernel 6.x  
qm set $VMID --ostype l26  
# Import the disk  
qm importdisk $VMID $DISKIMAGE $HOST_DISK  
# Attach disk to scsi bus  
qm set $VMID --scsihw virtio-scsi-pci --scsi0 $HOST_DISK:vm-$VMID-  
disk-0,cache=writeback,ssd=1  
# Set scsi disk as boot device  
qm set $VMID --boot c --bootdisk scsi0  
# Create and attach cloudinit drive  
qm set $VMID --ide2 $HOST_DISK:cloudinit  
# Set serial console, which is needed by OpenStack/Proxmox
```

```
qm set $VMID --serial0 socket --vga serial0
# Enable Qemu Guest Agent
qm set $VMID --agent enabled=1 # optional but recommended
# Convert the VM into a template
qm template $VMID
```

Skript ausführen

```
cd ~/tmp/
./createDebian12CloudInit_VMtemplate.sh
```

From:

<https://www.cooltux.net/> - TuxNet DokuWiki

Permanent link:

https://www.cooltux.net/doku.php?id=it-wiki:proxmox:templating_cloudinit&rev=1738168212

Last update: **2025/01/29 16:30**

