

Tạo Cloud Image cho tất cả các loại Linux OS với shell script trên proxmox

Create Cloud Template Shell:

Nguồn từ bạn [meramsey](#), link:

<https://gist.github.com/meramsey/aa759614cb5e387d8b88a0adfe77cc1d>

create-cloud-template.sh

```
#!/bin/bash
set -o errexit

clear
printf "\n*** This script will download a cloud image and create a Proxmox VM template from it.

### HOW TO USE
### Pre-req:
### - run on a Proxmox 6 server
### - a dhcp server should be active on vmbri
###
### - fork the gist and adapt the defaults (especially SSHKEY) as needed
### - download latest version of the script:
### curl wget https://gist.githubusercontent.com/chriswayg/43fbea910e024cbe608d7dcb12cb8466/ra
### - (optionally) prepare a cloudinit user-config.yml in the working directory
### this could be copied and modified from the cloudinit user dump at the end of this script
### - run the script:
### $ create-cloud-template.sh
### - clone the finished template from the Proxmox GUI and test
###
### NOTES:
### - links to cloud images:
### Directory: https://docs.openstack.org/image-guide/obtain-images.html
### Debian http://cdimage.debian.org/cdimage/openstack/
### Ubuntu http://cloud-images.ubuntu.com/
### CentOS: http://cloud.centos.org/centos/7/images/
### CentOS: https://cloud.centos.org/centos/8/x86_64/images/
### AlmaLinux: https://repo.almaLinux.org/almaLinux/8/cloud/x86_64/images/
### Fedora: https://alt.fedoraproject.org/cloud/
### SUSE 15 SP1 JeOS: https://download.suse.com/Download?buildid=0E-3enq3uys~
### CirrOS http://download.cirros-cloud.net/
### CoreOS (EOL 05.2020): https://stable.release.core-os.net/amd64-usr/current/
### Flatcar (CoreOS fork): https://stable.release.flatcar-linux.net/amd64-usr/current/
```

```

### Gentoo: http://gentoo.osuosl.org/experimental/amd64/openstack
### Arch (also Gentoo): https://linuximages.de/openstack/arch/
### Alpine: https://github.com/chriswayg/packer-qemu-cloud/
### RancherOS: https://github.com/rancher/os/releases (also includes Proxmox iso version)
###
### - most links will download the latest current (stable) version of the OS
### - older cloud-init versions do not support hashed passwords

## TODO
## - verify authenticity of downloaded images using hash or GPG

printf "* Available templates to generate:\n 2) Debian 9\n 3) Debian 10\n 4) Ubuntu 18.04\n 5) U
read -p "* Enter number of distro to use: " OSNR

# defaults which are used for most templates
RESIZE=+30G
MEMORY=2048
BRIDGE=vibr1
USERCONFIG_DEFAULT=none # cloud-init-config.yml
CITYPE=nocloud
SNIPPETSPATH=/var/lib/vz/snippets
SSHKEY=~/.ssh/2019_id_rsa.pub # ~/.ssh/id_rsa.pub
NOTE=""

case $OSNR in

  2)
    OSNAME=debian9
    VMID_DEFAULT=51100
    read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
    VMID=${VMID:-$VMID_DEFAULT}
    VMIMAGE=debian-9-openstack-amd64.qcow2
    NOTE="\n## Default user is 'debian'\n## NOTE: Setting a password via cloud-config does not v
    printf "$NOTE\n"
    wget -P /tmp -N https://cdimage.debian.org/cdimage/openstack/current-9/$VMIMAGE
    ;;

  3)
    OSNAME=debian10
    VMID_DEFAULT=51200
    read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
    VMID=${VMID:-$VMID_DEFAULT}
    VMIMAGE=debian-10-openstack-amd64.qcow2
    NOTE="\n## Default user is 'debian'\n"
    printf "$NOTE\n"
    wget -P /tmp -N https://cdimage.debian.org/cdimage/openstack/current-10/$VMIMAGE
    ;;

  4)
    OSNAME=ubuntu1804
    VMID_DEFAULT=52000
    read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
    VMID=${VMID:-$VMID_DEFAULT}
    VMIMAGE=bionic-server-cloudimg-amd64.img

```

```
NOTE="\n## Default user is 'ubuntu'\n"
printf "$NOTE\n"
wget -P /tmp -N https://cloud-images.ubuntu.com/bionic/current/$VMIMAGE
;;
```

5)

```
OSNAME=ubuntu2004
VMID_DEFAULT=52004
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
VMIMAGE=focal-server-cloudimg-amd64.img
NOTE="\n## Default user is 'ubuntu'\n"
printf "$NOTE\n"
wget -P /tmp -N https://cloud-images.ubuntu.com/focal/current/$VMIMAGE
;;
```

6)

```
OSNAME=ubuntu2010
VMID_DEFAULT=52010
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
VMIMAGE=groovy-server-cloudimg-amd64.img
NOTE="\n## Default user is 'ubuntu'\n"
printf "$NOTE\n"
wget -P /tmp -N https://cloud-images.ubuntu.com/groovy/current/$VMIMAGE
;;
```

7)

```
OSNAME=ubuntu2104
VMID_DEFAULT=52104
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
VMIMAGE=hirsute-server-cloudimg-amd64.img
NOTE="\n## Default user is 'ubuntu'\n"
printf "$NOTE\n"
wget -P /tmp -N https://cloud-images.ubuntu.com/hirsute/current/$VMIMAGE
;;
```

8)

```
OSNAME=centos7
VMID_DEFAULT=53100
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
RESIZE=+24G
VMIMAGE=CentOS-7-x86_64-GenericCloud.qcow2
NOTE="\n## Default user is 'centos'\n## NOTE: CentOS ignores hostname config.\n# use 'hostnamectl' to set hostname"
printf "$NOTE\n"
wget -P /tmp -N http://cloud.centos.org/centos/7/images/$VMIMAGE
;;
```

9)

```
OSNAME=centos8
VMID_DEFAULT=53108
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
```

```
VMID=${VMID:-$VMID_DEFAULT}
RESIZE=+24G
VMIMAGE=CentOS-8-GenericCloud-8.4.2105-20210603.0.x86_64.qcow2
NOTE="\n## Default user is 'centos'\n## NOTE: CentOS ignores hostname config.\n# use 'hostnamectl' to set hostname\n"
printf "$NOTE\n"
wget -P /tmp -N https://cloud.centos.org/centos/8/x86_64/images/$VMIMAGE
;;
```

10)

```
OSNAME=almalinux8
VMID_DEFAULT=53208
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
RESIZE=+24G
VMIMAGE=AlmaLinux-8-GenericCloud-latest.x86_64.qcow2
NOTE="\n## Default user is 'almalinux'\n"
printf "$NOTE\n"
wget -P /tmp -N https://repo.almalinux.org/almalinux/8/cloud/x86_64/images/$VMIMAGE
;;
```

11)

```
# - Proxmox creates a configdrive with the option: 'manage_etc_hosts: true'
# which causes an error in 'user-configdrive.service':
# 'Failed to apply cloud-config: Invalid option to manage_etc_hosts'
# There is no problem, when supplying a compatible 'user-config.yml'.
# - CoreOS needs 'configdrive2'
# - CoreOS is End of Life in 05.2020, use Flatcar instead
# https://github.com/coreos/coreos-cloudinit/blob/master/Documentation/config-drive.md
#
# OSNAME=coreos
# VMID_DEFAULT=54600
# read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
# VMID=${VMID:-$VMID_DEFAULT}
# RESIZE=+24G
# VMIMAGE=coreos_production_qemu_image.img.bz2
# CITYPE=configdrive2
# NOTE="\n## Default user is 'core'\n## NOTE: In CoreOS, setting a password via cloud-config is not supported\n"
# printf "$NOTE\n"
# wget -P /tmp -N https://stable.release.core-os.net/amd64-usr/current/$VMIMAGE
OSNAME=flatcar
VMID_DEFAULT=54600
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
RESIZE=+24G
VMIMAGE=flatcar_production_qemu_image.img.bz2
CITYPE=configdrive2
NOTE="\n## Default user is 'coreos'\n## NOTE: Setting a password via cloud-config does not work\n"
printf "$NOTE\n"
wget -P /tmp -N https://stable.release.flatcar-linux.net/amd64-usr/current/$VMIMAGE
;;
```

12)

```
OSNAME=arch
VMID_DEFAULT=54200
```

```

read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
RESIZE=+29G
VMIMAGE=arch-openstack-LATEST-image-bootstrap.qcow2
NOTE="\n## Default user is 'arch'\n## NOTE: Setting a password via cloud-config does not work"
printf "$NOTE\n"
wget -P /tmp -N https://linuximages.de/openstack/arch/$VMIMAGE
;;

13)
OSNAME=alpine311
VMID_DEFAULT=54000
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
VMIMAGE=alpine-311-cloudimg-amd64.qcow2
NOTE="\n## Default user is 'alpine'\n## NOTE: Cloud-init on Alpine 3.11 is not able to apply changes"
printf "$NOTE\n"
wget -P /tmp -N https://github.com/chriswayg/packer-proxmox-templates/releases/download/v1.6.0/packer-proxmox-templates-v1.6.0.tar.gz
#cp -v /root/$VMIMAGE /tmp/ # for local testing
;;

14)
OSNAME=rancheros
VMID_DEFAULT=54400
read -p "Enter a VM ID for $OSNAME [$VMID_DEFAULT]: " VMID
VMID=${VMID:-$VMID_DEFAULT}
VMIMAGE=rancheros-openstack.img
CITYPE=configdrive2
NOTE="\n## Default user is 'rancher'\n## NOTE: Setting a password via cloud-config does not work"
printf "$NOTE\n"
wget -P /tmp -N https://github.com/rancher/os/releases/download/v1.5.5/$VMIMAGE
;;

*)
printf "\n** Unknown OS number. Please use one of the above!\n"
exit 0
;;
esac

[[ $VMIMAGE == *.bz2 ]] \
  && printf "\n** Uncompressing image (waiting to complete...)\n" \
  && bzip2 -d --force /tmp/$VMIMAGE \
  && VMIMAGE=$(echo "${VMIMAGE%.*}") # remove .bz2 file extension from file name

# TODO: could prompt for the VM name
printf "\n** Creating a VM with $MEMORY MB using network bridge $BRIDGE\n"
qm create $VMID --name $OSNAME-cloud --memory $MEMORY --net0 virtio,bridge=$BRIDGE

printf "\n** Importing the disk in qcow2 format (as 'Unused Disk 0')\n"
qm importdisk $VMID /tmp/$VMIMAGE local -format qcow2

printf "\n** Attaching the disk to the vm using VirtIO SCSI\n"
qm set $VMID --scsihw virtio-scsi-pci --scsi0 /var/lib/vz/images/$VMID/vm-$VMID-disk-0.qcow2

```

```

printf "\n** Setting boot and display settings with serial console\n"
qm set $VMID --boot c --bootdisk scsi0 --serial0 socket --vga serial0

printf "\n** Using a dhcp server on $BRIDGE (or change to static IP)\n"
qm set $VMID --ipconfig0 ip=dhcp
#This would work in a bridged setup, but a routed setup requires a route to be added in the guest
#qm set $VMID --ipconfig0 ip=10.10.10.222/24,gw=10.10.10.1

printf "\n** Creating a cloudinit drive managed by Proxmox\n"
qm set $VMID --ide2 local:cloudinit

printf "\n** Specifying the cloud-init configuration format\n"
qm set $VMID --citype $CITYPE

printf "#** Made with create-cloud-template.sh - https://gist.github.com/chriswayg/43fbea910e024

## TODO: Also ask for a network configuration. Or create a config with routing for a static IP
printf "\n*** The script can add a cloud-init configuration with users and SSH keys from a file
read -p "Supply the name of the cloud-init-config.yml (this will be skipped, if file not found)
USERCONFIG=${USERCONFIG:-$USERCONFIG_DEFAULT}
if [ -f $PWD/$USERCONFIG ]
then
    # The cloud-init user config file overrides the user settings done elsewhere
    printf "\n** Adding user configuration\n"
    cp -v $PWD/$USERCONFIG $SNIPPETSPATH/$VMID-$OSNAME-$USERCONFIG
    qm set $VMID --cicustom "user=local:snippets/$VMID-$OSNAME-$USERCONFIG"
    printf "#* cloud-config: $VMID-$OSNAME-$USERCONFIG\n" >> /etc/pve/nodes/proxmox/qemu-server/
else
    # The SSH key should be supplied either in the cloud-init config file or here
    printf "\n** Skipping config file, as none was found\n\n** Adding SSH key\n"
    qm set $VMID --sshkey $SSHKEY
    printf "\n"
    read -p "Supply an optional password for the default user (press Enter for none): " PASSWORD
    [ ! -z "$PASSWORD" ] \
        && printf "\n** Adding the password to the config\n" \
        && qm set $VMID --cipassword $PASSWORD \
        && printf "#* a password has been set for the default user\n" >> /etc/pve/nodes/proxmox/
    printf "#- cloud-config used: via Proxmox\n" >> /etc/pve/nodes/proxmox/qemu-server/$VMID.conf
fi

# The NOTE is added to the Summary section of the VM (TODO there seems to be no 'qm' command for
printf "#$NOTE\n" >> /etc/pve/nodes/proxmox/qemu-server/$VMID.conf

printf "\n** Increasing the disk size\n"
qm resize $VMID scsi0 $RESIZE

printf "\n*** The following cloud-init configuration will be used ***\n"
printf "\n----- User ----- \n"
qm cloudinit dump $VMID user
printf "\n----- Network ----- \n"
qm cloudinit dump $VMID network

# convert the vm into a template (TODO make this optional)
qm template $VMID

```

```
printf "\n** Removing previously downloaded image file\n\n"  
rm -v /tmp/$VMIMAGE  
  
printf "$NOTE\n\n"
```

Revision #2

Created 2024-06-22 13:17:30 UTC by Son.KyLuat

Updated 2024-06-22 15:41:56 UTC by Son.KyLuat