**23 - R710 Proxmox Hashicorp Packer and Ansible - FULL STACK**

These notes cover installing Packer and its use with Ansible to build new template from existing one and have the new template provisioned with Ansible before it’s saved.

The work is done on the run3 host and user on prox3 hypervisor

This document builds upon the previous documents that have setup the run3 host.

A new cloud starting image will be created.

You will need to adjust the IP addresses used here and in the accompanying repo to suit your environement.

# Install Hashicorp’s Packer &make:

1. Log into rhys@run3
2. The following steps are from:  
   <https://learn.hashicorp.com/tutorials/packer/get-started-install-cli>
3. Do:  
   **curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -**  
     
   **sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb\_release -cs) main"**  
     
   **sudo apt-get update && sudo apt-get install packer**
4. To verify the installation, do:  
   packer  
     
   To see something like:  
   *Usage: packer [--version] [--help] <command> [<args>]*

*Available commands are:*

*build build image(s) from template*

*console creates a console for testing variable interpolation*

*fix fixes templates from old versions of packer*

*fmt Rewrites HCL2 config files to canonical format*

*hcl2\_upgrade transform a JSON template into an HCL2 configuration*

*init Install missing plugins or upgrade plugins*

*inspect see components of a template*

*validate check that a template is valid*

*version Prints the Packer version*

1. ‘make’ will be required, so install it with:  
   **sudo apt install make**

# Cloud-Init template creation:

1. Follow the same named section as this in document: “17 - R710 Proxmox Ubuntu cloud-init image - Terraform - Ansible”, but instead of creating file “**create-template.sh**”, do the following:

**touch c1.sh**  
**chmod +x c.sh**

in this file put (and adjust user and root passwords to your values):  
  
#!/usr/bin/bash

# run this with sudo

# change 'template\_storage' to local-lvm when needed

set -Eeuxo pipefail

image\_name="focal-server-cloudimg-disk-amd64.img"

master\_image\_name="focal-server-cloudimg-amd64.img"

# remove previous changed image, if present

[ -e ${image\_name} ] && rm ${image\_name}

echo "Copying original image to apply modifications against"

cp original/${master\_image\_name} ${image\_name}

# add net-tools and midnight commander:

virt-customize -a ${image\_name} --install net-tools

virt-customize -a ${image\_name} --install mc

# remove not needed:

virt-customize -a ${image\_name} --uninstall chef

virt-customize -a ${image\_name} --uninstall puppet

# inject the SSH keys into the cloud image itself before turning it into a template and VM.

# You need to create a user first and the necessary folders:

virt-customize -a ${image\_name} --run-command 'adduser rhys'

virt-customize -a ${image\_name} --password rhys:password:**userblafixme**

virt-customize -a ${image\_name} --run-command 'mkdir -p /home/rhys/.ssh'

virt-customize -a ${image\_name} --ssh-inject rhys:file:/home/rhys/.ssh/id\_rsa.pub

virt-customize -a ${image\_name} --run-command 'chown -R rhys:rhys /home/rhys'

virt-customize -a ${image\_name} --run-command 'chmod 700 /home/rhys/.ssh'

virt-customize -a ${image\_name} --run-command 'usermod -aG sudo rhys'

virt-customize -a ${image\_name} --root-password password:**blafixme**

# Ubuntu cloud img doesn't include qemu-guest-agent required for packer to get IP details from proxmox

virt-customize --install qemu-guest-agent -a ${image\_name}

# Create Proxmox VM using modified image

template\_name="ubuntu2004-cloud"

vmid="9100"

# NOTE: memory, cores and disk\_size will get over-ridden by Terraform in the final VM deployment

memory="2048"

cores="2"

disk\_size="6G"

template\_storage="Data2"

qm create ${vmid} --name ${template\_name} --memory ${memory} --cores=${cores} --net0 virtio,bridge=vmbr0

qm importdisk ${vmid} $image\_name ${template\_storage}

qm set ${vmid} --scsihw virtio-scsi-pci --scsi0 ${template\_storage}:vm-${vmid}-disk-0

qm set ${vmid} --boot c --bootdisk scsi0

qm set ${vmid} --ide2 ${template\_storage}:cloudinit

qm set ${vmid} --serial0 socket --vga serial0

qm set ${vmid} --agent enabled=1

# Packer will discover the IP of the template

qm set ${vmid} --ipconfig0 ip=**192.168.124.151/24,gw=192.168.124.1**

# Resize the primary boot disk (otherwise it will be around 2G by default)

qm resize ${vmid} scsi0 ${disk\_size}

# Convert VM to a template

qm template ${vmid}

# Remove modified master image

rm $image\_name

1. To create the template VM from template, do:

**sudo ./create-template.sh**The above must be run as root to have access to the ‘**qm**’ commands in the script.  
  
You now have VMID 9100 template.

1. Follow the other sections in document: “17 - R710 Proxmox Ubuntu cloud-init image - Terraform - Ansible” for othe parts of setup like getting the Proxmox API Token ID and secret to use later.

# Getting FULL-STACK code and adjustments for your Proxmox:

1. On run3 host, user rhys’s public folder, do:  
   **git clone https://github.com/redhug1/Proxmox-Cloud-Init-Packer-Ansible-Terraform-FULL-STACK.git**
2. Apply suitable values for your Proxmox system in the file**: packer/proxmox.pkvars.hcl**

1. As per previous documents, sort your ‘**ssh\_key**’ in **terraform/modules-prox3/vars.tf**  
   And similarly for ‘**pm\_api\_token\_secret**’ in **terraform/modules-prox3/main.tf**

# Building Templates with Packer:

1. On run3, in directory ‘**packer**’, do:  
   **make base**  
     
   On my system, this completes in:   
   Wait completed after 2 minutes 19 seconds  
     
   Observe the new template in Proxmox named: “**9110 (9110-9100-base)**” created from template: “**9100 (ubuntu2004-cloud)**”
2. Then do:  
   **make bastion**  
     
   On my system, this completes in:  
   Wait completed after 42 seconds 694 milliseconds  
     
   Observe the new template in Proxmox named: “**9120 (9120-9110-bastion)**” created from template: “**9110 (9110-9100-base)**”

# Deploy VM’s using initial cloud-init template and the templates created with Packer:

1. In the terraform directory, do:  
   **terraform init**  
     
   **terraform plan**  
     
   **terraform apply**On my system, this takes about 2 mins to create the 4 VM’s in parallel.
2. Then do: **./clear-known-hosts.sh  
     
   ./ add-sshs.sh**
3. Observe 4 new hosts in Proxmox that can be accessed from run3 with:

**ssh rhys@man**  
  
**ssh rhys@cw1**  
  
**ssh rhys@cui**  
  
**ssh rhys@bast**

1. When you log into the new VM’s, you may see:  
   **\*\*\* System restart required \*\*\***  
     
   This will be dealt with in the next section.
2. Typing the following into each of the above hosts that you have logged into will show differing results indicating the different levels of configuration built up with Ansible during the Packer creation of the different templates:  
     
   **jq**  
     
   **ntpq -p**
3. NOTE: in the file **terraform/main.tf** the variable ‘**template\_name**’ selects the specific template from which to Terraform deploy a VM.

# Further provisioning after Terraform deployment:

1. After the above initial provisioning …  
   To demonstrate further provisioning, in the ansible directory, first do:  
   **ansible man -m shell -a 'df -h' --ask-become-pass**  
     
   to observe no directory called ‘**/work**’
2. Then do:  
   **ansible-playbook --ask-become-pass disk-setup.yml**
3. To check the changes, again do:  
   **ansible man -m shell -a 'df -h' --ask-become-pass**  
     
   To see the ‘**/work**’ directory on the second partition.
4. You can similarly inspect the other VM’s with:  
   **ansible cw1 -m shell -a 'df -h' --ask-become-pass**  
     
   **ansible cui -m shell -a 'df -h' --ask-become-pass**  
     
   **ansible bast -m shell -a 'df -h' --ask-become-pass**
5. When you logged into the new VM’s in the previous section and possibly saw:  
   \*\*\* System restart required \*\*\*  
     
   the following will restart all of the VM’s, in the Ansible directory do:  
   **ansible-playbook --ask-become-pass reboot.yml**  
     
   On my system this completes in less than 30 seconds