Praxis Beispiele

- Proxmox Debian VM mit Ansible und Cloudinit Provisionieren

Proxmox Debian VM mit Ansible und Cloudinit Provisionieren

Beschreibung:

Ein Playbook zum erstellen einer Debian 11 VM per Ansible.

```
---
- name: Create a new VM on Proxmox
  hosts: proxmox
  gather_facts: no
  tasks:
    - name: Ensure the VM disk image is downloaded
      get_url:
        url: https://cloud.debian.org/images/cloud/bullseye/latest/debian-11-generic-amd64.qcow2
        dest: /var/lib/vz/images/{{ vmid }}/vm-{{ vmid }}-disk-1.qcow2
      register: downloaded_image

    - name: Create VM
      community.general.proxmox_kvm:
        api_user: "{{ prox_user }}"
        api_password: "{{ prox_password }}"
        api_host: "{{ prox_host }}"
        node: "{{ prox_node }}"
        name: "{{ vmvar_hostname }}"
        vmid: "{{ vmid }}"
        memory: "{{ vmvar_memory }}"
        cores: "{{ vmvar_cores }}"
        sockets: 1
        cpuunits: 1000
        net: '{"net0":"virtio,bridge={{ vmvar_vmbr }},firewall=1"}'
        virtio: '{"virtio0":"local:{{ vmid }}/vm-{{ vmid }}-disk-1.qcow2,cache=unsafe,discard=on,size={{
```

```
vmvar_disksize }}G"}'
        ostype: "{{ prox_ostype }}"
        kvm: yes
        acpi: yes
        autostart: no
        boot: cnd
        bootdisk: virtio0
        onboot: yes
        scsihw: virtio-scsi-pci
        description: "{{ vmvar_description }}"
        force: yes
      register: created_vm

    - name: Configure VM cloud-init
      community.general.proxmox_cloudinit:
        api_user: "{{ prox_user }}"
        api_password: "{{ prox_password }}"
        api_host: "{{ prox_host }}"
        node: "{{ prox_node }}"
        vmid: "{{ vmid }}"
        searchdomain: "{{ vmvar_searchdomain }}"
        nameserver: "{{ vmvar_nameserver }}"
        gateway: "{{ vmvar_gateway }}"
        ipconfig: "{{ vmvar_ipconfig }}"
        hostname: "{{ vmvar_hostname }}"
        city: "{{ vmvar_city }}"
        country: "{{ vmvar_country }}"
        timezone: "{{ vmvar_timezone }}"
        user: root
        sshkeys: "{{ lookup('file', vmvar_public_key) }}"
        force: yes
      when: created_vm.changed

    - name: Start VM
      community.general.proxmox:
        api_user: "{{ prox_user }}"
        api_password: "{{ prox_password }}"
        api_host: "{{ prox_host }}"
        node: "{{ prox_node }}"
        vmid: "{{ vmid }}"
```

```
      state: started
    register: started_vm

  - name: Wait for SSH to become available
    ansible.builtin.wait_for:
      host: "{{ vmvar_ipconfig.split('/')[0] }}"
      port: 22
      search_regex: OpenSSH
      delay: 10
      timeout: 600
    when: started_vm.changed

  - name: Wait for preseed_complete file
    ansible.builtin.command: "ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -i {{
vmvar_private_key }} root@{{ vmvar_ipconfig.split('/')[0] }} 'test -f /root/preseed_complete'"
    register: preseed_complete
    until: preseed_complete.rc == 0
    retries: 60
    delay: 10
    when: started_vm.changed

  - name: Remove the downloaded QCOW2 image
    ansible.builtin.file:
      path: /var/lib/vz/images/{{ vmid }}/vm-{{ vmid }}-disk-1.qcow2
      state: absent
    when: downloaded_image.changed and started_vm.changed and preseed_complete.rc == 0
```

Nun ein neues Verzeichnis template erstellen.
Darin eine neue Datei mit dem namen preeseed.cfg.j2 erstellen
Und diesen Inhalt einfügen

```
# Preseed configuration

d-i debian-installer/locale string en_US
d-i keyboard-configuration/xkb-keymap select {{ vmvar_keyboard_layout }}
d-i time/zone string {{ vmvar_timezone }}

# Partitioning
d-i partman-auto/method string regular
d-i partman-auto/disk string /dev/vda
```

```
d-i partman-auto/expert_recipe string                     \
    boot-root ::                                          \
        512 512 512 ext4                                  \
            $primary{ } $bootable{ }                      \
            method{ format } format{ }                    \
            use_filesystem{ } filesystem{ ext4 }          \
            mountpoint{ /boot }                           \
        .                                                 \
        {{ vmvar_swap_partsize }} {{ vmvar_swap_partsize }} {{ vmvar_swap_partsize }} linux-swap \
            $primary{ }                                   \
            method{ swap } format{ }                      \
        .                                                 \
        10000 10000 -1 ext4                               \
            $primary{ }                                   \
            method{ format } format{ }                    \
            use_filesystem{ } filesystem{ ext4 }          \
            mountpoint{ / }                               \
        .

d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# Network configuration
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string {{ vmvar_hostname }}
d-i netcfg/get_domain string {{ vmvar_domain }}

# Account setup
d-i passwd/user-fullname string {{ vmvar_fullname }}
d-i passwd/username string {{ vmvar_username }}
d-i passwd/user-password password {{ vmvar_password }}
d-i passwd/user-password-again password {{ vmvar_password }}
d-i user-setup/encrypt-home boolean false

# Package selection
tasksel tasksel/first multiselect standard
d-i pkgsel/include string openssh-server
d-i pkgsel/upgrade select full-upgrade
```

```
# Finishing the installation
d-i finish-install/reboot_in_progress note

# Preseed complete marker
d-i preseed/late_command string echo "preseed_complete" >> /var/log/installer/syslog
d-i preseed/late_command string in-target touch /root/preseed_complete
```

Die Hilfe zu den Parametern

```
# help.txt

Beispiel für die Verwendung des Ansible-Playbooks zum Erstellen einer VM:

ansible-playbook -i inventory.ini create_vm.yml

Stellen Sie sicher, dass Sie die Werte in der inventory.ini-Datei an Ihre Umgebung anpassen. Die wichtigsten Parameter, die Sie anpassen sollten, sind:

1. Proxmox-Host-Parameter:
- prox_api_user: Der Benutzer, der die Proxmox-API verwendet (normalerweise "root@pam").
- prox_api_password: Das Passwort für den API-Benutzer.
- prox_api_host: Die IP-Adresse oder der Hostname Ihres Proxmox-Servers.
- prox_node: Der Name des Proxmox-Knotens, auf dem die VM erstellt werden soll.

2. VM-Parameter:
- vmvar_ostype: Der OS-Typ (z. B. l26 für Linux 2.6/3.x/4.x/5.x Kernel).
- vmvar_disk_size: Die Größe der Festplatte in Gigabyte (z. B. 64G).
- vmvar_cores: Die Anzahl der CPU-Kerne.
- vmvar_memory: Der Arbeitsspeicher in Megabyte.
- vmvar_vmbr: Die Netzwerkbrücke (z. B. vmbr0).
- vmvar_ip_address: Die IP-Adresse der VM.
- vmvar_swap_partsize: Die Größe der Swap-Partition (z. B. 32G).

Die VM wird mit folgendem Partitionslayout erstellt:
- Erste Partition: Boot-Partition mit einer Größe von 512 MB (fest).
- Zweite Partition: Swap-Partition mit einer Größe, die durch den Parameter vmvar_swap_partsize fest
```

Liste der OS-Typen:

- l26: Linux 2.6/3.x/4.x/5.x Kernel
- other: Anderes OS
- wxp: Windows XP
- w2k: Windows 2000
- w2k3: Windows 2003
- w2k8: Windows 2008
- wvista: Windows Vista
- win7: Windows 7
- win8: Windows 8/2012
- win10: Windows 10/2016/2019

Beispiel einer Inventory Datei

```
[proxmox]
#wenn das skript direkt auf dem proxmox host ausgeführt wird
myvm ansible_connection=localhost
#oder remote von einem laptop oder so
myvm ansible_host=192.168.178.120 ansible_user=root

[proxmox:vars]
prox_api_user=root@pam
prox_api_password=12345678
prox_api_host=192.168.178.120
prox_node=my-node

vmvar_gw=192.168.178.1
vmvar_root_password=mysecretpassword
vmvar_private_key=/path/to/your/private_key
vmvar_public_key="ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD3F6..."

vmvar_dns=8.8.8.8
vmvar_searchdomain=mydomain.local
vmvar_timezone=Europe/Berlin
vmvar_keyboard_layout=de
```

```
[myvm:vars]
vmvar_ostype=l26
vmvar_disk_size=64G
vmvar_swap_partsize=32G
vmvar_cores=2
vmvar_memory=2048
vmvar_vmbr=vmbr0
vmvar_ip_address=192.168.178.200
```