o integrate a VM created in KVM into the Kickstart process for bare-metal deployment, you'll need to capture the VM's image, prepare it for deployment, and then incorporate it into your PXE/Kickstart environment. Here’s how you can achieve this:

1. Create and Prepare the VM in KVM

First, ensure that the VM you create in KVM is generalized, meaning it doesn’t contain hardware-specific information. You can use a base installation of Rocky Linux (or any OS) and configure it as follows:

a. Install the Base System in KVM

Create your VM with KVM:

bash

sudo virt-install \

--name rocky9.4-baremetal-template \

--ram 2048 \

--vcpus 2 \

--disk path=/sto/KVM/rocky9.4-baremetal-template.qcow2,size=20 \

--os-variant rocky9.0 \

--graphics none \

--cdrom /path/to/rocky9.4.iso \

--network network=default \

--console pty,target\_type=serial

b. Generalize the VM for Bare-Metal Deployment

After installation, perform the following steps inside the VM:

Remove Machine-Specific Identifiers:

bash

sudo truncate -s 0 /etc/machine-id

sudo rm -f /var/lib/dbus/machine-id

Clean Up Cloud-Init (if installed):

bash

sudo cloud-init clean

sudo rm -rf /var/lib/cloud/\*

Remove Persistent Network Rules:

bash

sudo rm -f /etc/udev/rules.d/70-persistent-net.rules

Shut Down the VM:

bash

sudo shutdown now

2. Capture the VM Image

Convert the KVM disk image to a raw format that is suitable for deployment on bare-metal systems:

bash

sudo qemu-img convert -O raw /sto/KVM/rocky9.4-baremetal-template.qcow2 /sto/KVM/rocky9.4-baremetal-template.raw

3. Integrate the VM Image into the PXE/Kickstart Environment

a. Store the Image on the PXE Server

Copy the raw image to the PXE server, so it’s available for deployment:

bash

sudo cp /sto/KVM/rocky9.4-baremetal-template.raw /var/www/html/images/

b. Modify the Kickstart File to Deploy the VM Image

Create or modify a Kickstart file to deploy the VM image to the bare-metal system. Here's an example Kickstart file (/var/www/html/kickstart/deploy-vm.ks):

bash

# Kickstart configuration for deploying a VM image to bare-metal

# System language

lang en\_US.UTF-8

# Keyboard layouts

keyboard us

# System timezone

timezone America/New\_York --isUtc

# Root password

rootpw --iscrypted $6$somehashedpassword

# Network information

network --bootproto=dhcp --device=eth0

# System authorization information

auth --useshadow --passalgo=sha512

# Do not configure X Window System

skipx

# SELinux configuration

selinux --enforcing

# Firewall configuration

firewall --enabled --ssh

# Partitioning scheme

zerombr

clearpart --all --initlabel

part /boot --fstype=ext4 --size=1024

part / --fstype=ext4 --size=1 --grow

# Use the raw image to populate the root partition

%pre

# Download the VM image and write it to disk

curl -o /tmp/image.raw http://192.168.1.10/images/rocky9.4-baremetal-template.raw

dd if=/tmp/image.raw of=/dev/sda bs=4M status=progress

%end

# Reboot after installation

reboot

%post

# Post-installation script

echo "Deployment completed on $(hostname)" > /root/deployment.log

%end

4. Update PXE Boot Configuration

Modify your PXE boot menu configuration to include the option for deploying the VM image:

Edit the PXE configuration file at /var/lib/tftpboot/pxelinux.cfg/default:

bash

DEFAULT menu.c32

PROMPT 0

TIMEOUT 100

ONTIMEOUT local

MENU TITLE PXE Boot Menu

LABEL deploy-vm

MENU LABEL Deploy Rocky Linux 9.4 VM to Bare Metal

KERNEL vmlinuz

APPEND initrd=initrd.img inst.ks=http://192.168.1.10/kickstart/deploy-vm.ks

Make sure that the kernel and initrd files from your Rocky Linux installation are available in the TFTP directory:

bash

sudo cp /var/www/html/rocky/images/pxeboot/vmlinuz /var/lib/tftpboot/

sudo cp /var/www/html/rocky/images/pxeboot/initrd.img /var/lib/tftpboot/

5. Deploy the VM Image to Bare-Metal Systems

Boot the bare-metal system from the network.

Select the "Deploy Rocky Linux 9.4 VM to Bare Metal" option from the PXE boot menu.

The system will automatically deploy the VM image onto the bare-metal hardware, configuring the partitions as specified in the Kickstart file.

6. Handle Different Hardware Configurations

The %pre script in the Kickstart file is key to making this adaptable to different hardware. The image will be deployed as-is, and the script can adjust the partitioning based on the detected disk size, similar to how you would do in a traditional Kickstart file.

7. Finalization and Customization

After the image is deployed, any post-installation scripts you include in the Kickstart file can handle final configuration, such as setting up networking, installing additional packages, or applying updates.

Summary

This setup allows you to deploy a pre-configured VM image to any bare-metal system via PXE and Kickstart. The flexibility in the Kickstart file allows you to accommodate different hardware configurations, making the deployment process efficient and consistent across multiple systems.