1. Set Up the PXE Server

Install Required Packages

First, you'll need to install the necessary services on your PXE server, including DHCP, TFTP, and HTTP (or NFS) for serving the Kickstart files.

bash

sudo dnf install dhcp-server tftp-server httpd syslinux xinetd

Configure the DHCP Server

Configure the DHCP server to handle PXE boot requests. Edit /etc/dhcp/dhcpd.conf:

bash

default-lease-time 600;

max-lease-time 7200;

log-facility local7;

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.100 192.168.1.200;

option routers 192.168.1.1;

option broadcast-address 192.168.1.255;

next-server 192.168.1.10; # IP of the PXE server

filename "pxelinux.0"; # PXE bootloader

}

Configure the TFTP Server

Edit the TFTP configuration in /etc/xinetd.d/tftp:

bash

service tftp

{

socket\_type = dgram

protocol = udp

wait = yes

user = root

server = /usr/sbin/in.tftpd

server\_args = -s /var/lib/tftpboot

disable = no

}

Create the necessary PXE boot directory structure and copy the bootloader files:

bash

sudo mkdir -p /var/lib/tftpboot/pxelinux.cfg

sudo cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

sudo cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot/

sudo cp /usr/share/syslinux/memdisk /var/lib/tftpboot/

sudo cp /usr/share/syslinux/mboot.c32 /var/lib/tftpboot/

sudo cp /usr/share/syslinux/chain.c32 /var/lib/tftpboot/

Start the DHCP and TFTP Services

bash

sudo systemctl enable --now dhcpd

sudo systemctl enable --now xinetd

2. Prepare the Kickstart File

Kickstart files automate the installation process. Create a generic Kickstart file that can handle varying hardware configurations. Here’s an example:

Create a directory to host the Kickstart file via HTTP:

bash

sudo mkdir -p /var/www/html/kickstart

Example Kickstart file (/var/www/html/kickstart/generic.ks):

bash

# Generic Kickstart configuration

# 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

# Use text mode install

text

# Firewall configuration

firewall --enabled --http --ssh

# SELinux configuration

selinux --enforcing

# Do not configure X Window System

skipx

# Installation source

url --url="http://192.168.1.10/rocky"

# Partitioning scheme

autopart --type=lvm

clearpart --all --initlabel

# Reboot after installation

reboot

# Bootloader configuration

bootloader --location=mbr --append="rhgb quiet"

# Packages selection

%packages

@core

@^minimal-environment

chrony

%end

%post

# Post installation script

echo "Post installation script" > /root/post\_install.log

%end

3. Prepare the Installation Media

You need to host the Rocky Linux installation files on your PXE server. You can use HTTP or NFS to share the installation files. Here’s how to use HTTP:

Copy the Rocky Linux Installation Files

Mount the Rocky Linux ISO and copy its contents to your HTTP server:

bash

sudo mkdir -p /var/www/html/rocky

sudo mount -o loop /path/to/rocky.iso /mnt

sudo cp -r /mnt/\* /var/www/html/rocky/

sudo umount /mnt

4. Configure PXE Boot Menu

Create the PXE boot menu 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 linux

MENU LABEL Install Rocky Linux

KERNEL vmlinuz

APPEND initrd=initrd.img inst.repo=http://192.168.1.10/rocky ks=http://192.168.1.10/kickstart/generic.ks

Copy the kernel and initrd images from the Rocky Linux installation files to the TFTP boot 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. Test the PXE Boot and Kickstart

Boot your bare-metal system from the network. Ensure the system is set to boot from the network in BIOS/UEFI settings.

The system should receive an IP address via DHCP and load the PXE boot menu.

Select the option to install Rocky Linux.

The installation will proceed automatically using the Kickstart file.

6. Handling Different Hardware Configurations

The Kickstart file is already designed to automatically handle different hardware configurations using the autopart --type=lvm directive, which will automatically create LVM partitions based on available disk space.

You can further customize the Kickstart file to:

Detect available memory and allocate swap space accordingly.

Automatically configure network settings based on detected interfaces.

Use %pre and %post scripts within the Kickstart file to dynamically adapt to hardware differences (e.g., adjusting partitioning schemes or installing different packages based on hardware capabilities).

Example of Dynamic Partitioning:

In the %pre section of the Kickstart file, you could add logic to detect disk size and create custom partitions:

bash

%pre

if [ $(lsblk -b -d -n -o SIZE /dev/sda) -gt 100000000000 ]; then

echo 'clearpart --all --initlabel' > /tmp/partitioning.ks

echo 'part /boot --fstype="ext4" --size=1024' >> /tmp/partitioning.ks

echo 'part pv.01 --size=1 --grow' >> /tmp/partitioning.ks

echo 'volgroup VolGroup --pesize=4096 pv.01' >> /tmp/partitioning.ks

echo 'logvol / --fstype="ext4" --size=10240 --name=root --vgname=VolGroup' >> /tmp/partitioning.ks

echo 'logvol swap --fstype="swap" --size=2048 --name=swap --vgname=VolGroup' >> /tmp/partitioning.ks

else

echo 'clearpart --all --initlabel' > /tmp/partitioning.ks

echo 'part /boot --fstype="ext4" --size=512' >> /tmp/partitioning.ks

echo 'part pv.01 --size=1 --grow' >> /tmp/partitioning.ks

echo 'volgroup VolGroup --pesize=4096 pv.01' >> /tmp/partitioning.ks

echo 'logvol / --fstype="ext4" --size=5120 --name=root --vgname=VolGroup' >> /tmp/partitioning.ks

echo 'logvol swap --fstype="swap" --size=1024 --name=swap --vgname=VolGroup' >> /tmp/partitioning.ks

fi

%end

%include /tmp/partitioning.ks

This setup allows your PXE boot and Kickstart configuration to be highly adaptable, ensuring compatibility across various hardware models and configurations.