Red Hat Enterprise Linux (RHEL) is an Enterprise Linux distribution developed by Red Hat. This operating system is the dominating OS in the server market, challenged by Debian Linux, Suse Linux and MS Windows. Providing support to modernize your infrastructure and boost efficiency, RHEL provides the stability to take on today’s challenges and the flexibility to adapt to tomorrow’s demands. This practical guide will help you get to grips with RHEL 7 Server and help you automate its installation.

What this book will do for you...

- Set up and configure RHEL 7 Server
- Use NetworkManager to configure all aspects of your network
- Manage virtual environments using libvirt
- Set up software repositories
- Secure and monitor your RHEL environment
- Configure SELinux, and create and apply its policies
- Create kickstart scripts to automatically deploy RHEL 7 systems
- Use orchestration and configuration management tools to manage your environment

Inside the Cookbook...

- A straightforward and easy-to-follow format
- A selection of the most important tasks and problems
- Carefully organized instructions to solve problems efficiently
- Clear explanations of what you did
- Solutions that can be applied to solve real-world problems

Quick answers to common problems

Red Hat Enterprise Linux Server Cookbook

Over 60 recipes to help you build, configure, and orchestrate RHEL 7 Server to make your everyday administration experience seamless

William Leemans

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 2 'Deploying RHEL "En Masse"
- A synopsis of the book’s content
- More information on Red Hat Enterprise Linux Server Cookbook
William Leemans has over 20 years of experience in the IT industry in various positions and supporting several environments.

In 2005, he started his own consulting company, Critter BVBA, in the hope of offering open source solutions to his customers, who are mainly enterprises.

In 2010, William started supporting Red Hat products full time with the Federal Police, Belgium. Since then, he has moved on to support Red Hat products at Proximus and now Euroclear.

William is a strong open source supporter and contributes where he can. He has a couple of projects running at GitHub (https://github.com/bushvin). During the course of writing this book, William recertified himself as a Red Hat Certified Engineer, hoping to one day become a Red Hat Certified Architect.

When he’s not tapping away at the keyboard of his laptop, William likes to play around with his two young children, listen to rock music (Foo Fighters, AC/DC, and Queens of the Stone Age are some of his favorites), and devising complicated and intricate plots for the stories that he runs at his biweekly roleplaying sessions with his friends.
Preface

Gnu/Linux is the most important OS in the data center but how do you leverage it? How do you maintain and contain it? Many Gnu/Linux distributions try to answer these questions, but not all succeed. Red Hat Enterprise Linux is one that does answer these questions.

The next question is how do you, as a system administrator, manage a RHEL infrastructure? How do you deploy not just one system, but many? How do you make sure that it is secure and up to date? How can you monitor system components?

It may seem odd to you, but as a Red Hat Certified Engineer, I prefer the "lazy" approach—not as in "I can't be bothered," but as in "I like to do something once and do it good the first time and spend the rest of my time doing fun stuff."

In this book, I try to show you how to set up and configure systems, mainly by providing useful information to automate the setup, configuration, and management. This also explains the lack of the use of a GUI in this book. I'll be honest with you; I couldn't live without one on my laptop or desktop, but I do not believe servers should have a GUI. GUI-based applications tend not to have command-line counterparts, and I solemnly believe that if you cannot install, configure, manage, and maintain a piece of software through a script, it does not belong on a server.

This book does not pretend to be the de facto answer to all questions (that would be 42), but I do hope that you will learn something new and that, in turn, you will put this knowledge to good use. Remember, with great power, comes great responsibility!

What this book covers

Chapter 1, Working with KVM Guests, will not start by installing a basic RHEL system. It will start by introducing you to KVM if you don't already know it. You'll learn how to install and configure the KVM host and manage your KVM guests (the VMs). It will discuss the basics of adding resources on the fly, moving disks, and even moving the entire guest to another KVM host.
Chapter 2, *Deploying RHEL "En Masse"*, will explore the ways of installing a RHEL system, introducing you to kickstart deployments, which are used to streamline automated system installs. If you want to orchestrate your environment, this chapter will lay out the basics for you to build on.

Chapter 3, *Configuring Your Network*, will explore NetworkManager tools to manage your network configuration, including advanced topics such as VLANs, link aggregation, and bridges. It will show you how to leverage its command-line tools to automate your system's network configuration during its deployment or afterwards, when all is installed.

Chapter 4, *Configuring Your New System*, will explain how to configure the basics, such as log retention, time, and your boot environment. It will also introduce you to the new systemd, which is SysVinit's replacement, and to monitoring and managing your services.

Chapter 5, *Using SELinux*, will give you an overview, but a brief one, on how to manage and troubleshoot SELinux on your system. SELinux is becoming more and more important in today's world because of its security implementation, and it's better to know about it than to just turn it off because you can't handle it.

Chapter 6, *Orchestrating with Ansible*, will tell you all about Ansible, which was recently bought by Red Hat. It will show you how to create simple playbooks that easily deploy new systems and how to manage your system's configuration.

Chapter 7, *Puppet Configuration Management*, will show you how to set up and configure Puppet. It will also give you a peek at its configuration management capacities.

Chapter 8, *Yum and Repositories*, will take a look at yum repositories, how you can create your own mirrors of the existing (Red Hat) repositories, and how to leverage it to keep your RHEL environment up to date without breaking a sweat.

Chapter 9, *Securing RHEL 7*, will take security configuration and auditing problems a bit further. We'll explore how to configure setting up centralized secure authentication and privilege escalation. It will show you how you can operate a system that appears to be "hung" and trace the root cause of the event.

Chapter 10, *Monitoring and Performance Tuning*, will show you the basics of easy performance tuning and how to monitor your system's resources.
In this chapter, the following recipes are provided:

- Creating a kickstart file
- Publishing your kickstart file using httpd
- Deploying a system using pxe
- Deploying a system using a custom boot ISO file

**Introduction**

In this chapter, you will find the answer to deploying multiple systems with the same basic setup. We will first look at creating an answer file, the kickstart file that will drive the unattended installation. Then, we’ll take a look at a possible way to make this kickstart file accessible through the Apache web server. Finally, we’ll discuss two common ways to install physical and virtual machines.

This chapter assumes that you have a working knowledge of system network configuration components, such as DNS, DNS search, IP addresses, and so on, and yum repositories.

**Creating a kickstart file**

A kickstart file is essentially a file containing all the necessary answers to questions that are asked during a typical install. It was created by Red Hat in response to the need for automated installs. Using kickstart, an admin can create one file or template containing all the instructions.
There are three ways to create a kickstart file:

- By hand
- Using the GUI's `system-config-kickstart` tool
- Using the standard Red Hat installation program Anaconda

In this recipe, I will cover a combination of the first two.

### Getting ready

Before we can get down to the nitty-gritty of generating our base kickstart file or template, we need to install `system-config-kickstart`. Run the following command:

```bash
-# yum install -y system-config-kickstart
```

### How to do it...

First, let's create a base template for our kickstart file(s) through the following steps:

1. First, launch **Kickstart Configurator** from the menu.
2. Select your system's basic configuration from the **Kickstart Configurator** GUI.

   The following screenshot shows the options you can set in the **Basic Configuration** view:
3. Now, select the installation method from the Kickstart Configurator GUI. The following screenshot shows the options that you can set in the Installation method view:

![Kickstart Configurator](image)

4. Next, substitute the values for **HTTP Server** and **HTTP Directory** with your own repositories.
5. Ensure that the correct settings are applied for **Boot Loader**.

The following screenshot shows the options that you can set in the **Boot Loader options** view:
6. Configure your disk and partition information. Simply create a `/boot` partition and be done with it! We'll edit the file manually for better customization.

The following screenshot shows the options you can set in the **Partition Information** view:

![Partition Information](image)

- **Master Boot Record**
  - Clear Master Boot Record
  - Do not clear Master Boot Record

- **Partitions**
  - Remove all existing partitions
  - Remove existing Linux partitions
  - Preserve existing partitions

- **Disk label**
  - Initialize the disk label
  - Do not initialize the disk label

- **Layout**
  - Device/Partition Number
  - Mount Point/Type
  - Format
  - Size (MB)

- **Hard Drives**
  - Boot: xfs: Yes: 512
7. Configure your network. You need to know the name of your device if you want to correctly configure your network.

The following screenshot shows the **Network Device** information that you can edit in the **Network Configuration** view:
8. Now, disable **Installing a graphical environment**.

We want as few packages as possible. The following screenshot shows the options that you can set in the **Display Configuration** view:
9. Next, perform any preinstallation and/or postinstallation tasks you deem necessary. I always try to make root accessible through SSH and keys.

The following screenshot shows the options that you can set in the Post-Installation Script view:

10. Save the kickstart file.

11. Open the file using your favorite editor and add the following to your partition section:

```bash
part pv.01 --size=1 --ondisk=sda --grow
volgroup vgl pv.01
logvol / --vgname=vgl --size=2048 --name=root
logvol /usr --vgname=vgl --size=2048 --name=usr
logvol /var --vgname=vgl --size=2048 --name=var
logvol /var/log --vgname=vgl --size=1024 --name=var
logvol /home --vgname=vgl --size=512 --name=home
logvol swap --vgname=vgl --recommended --name=swap -fstype=swap
```
12. Now, add the following script to your network line:
   
   --hostname=rhel7

13. Add the following script before %post:
   
   %packages --nobase
   @core --nodefaults
   %end

14. Create a password hash for use in the next step, as follows:
   
   $1$meIlXKN$6VRdaRkevJw9nngcMtRl0.

15. Save the resulting file. You should have something similar to this:

   #platform=x86, AMD64, or Intel EM64T
   #version=DEVEL
   # Install OS instead of upgrade
   install
   # Keyboard layouts
   keyboard 'be-latin1'
   # Halt after installation
   halt
   # Root password
   rootpw --iscrypted $1$meIlXKN$6VRdaRkevJw9nngcMtRl0.
   # System timezone
   timezone Europe/Brussels
   # Use network installation
   url -url="http://repo.example.com/rhel/7/os/x86_64/"
   # System language
   lang en_US
   # Firewall configuration
   firewall --disabled
   # Network information
   network --bootproto=static --device=enol --
   gateway=192.168.0.254 --ip=192.168.0.1 --nameserver=192.168.0.253
   --netmask=255.255.255.0 --hostname=rhel7
   # System authorization information
   auth --useshadow --passalgo=sha512
   # Use text mode install
   text
   # SELinux configuration
   selinux --enforcing
   # Do not configure the X Window System
   skipx
Deploying RHEL "En Masse"

```bash
# System bootloader configuration
bootloader --location=none
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="xfs" --ondisk=sda --size=512
part pv.01 --size=1 --ondisk=sda --grow
volgroup vg1 pv.01
logvol / --vgname=vg1 --size=2048 --name=root --fstype=xfs
logvol /usr --vgname=vg1 --size=2048 --name=usr --fstype=xfs
logvol /var --vgname=vg1 --size=2048 --name=var --fstype=xfs
logvol /var/log --vgname=vg1 --size=1024 --name=var --fstype=xfs
logvol /home --vgname=vg1 --size=512 --name=home --fstype=xfs
logvol swap --vgname=vg1 --recommended --name=swap --fstype=swap

%packages --nobase
@core --nodefaults
%end

%post
mkdir -p ~/.ssh
chmod 700 ~/.ssh
# Let's download my authorized keyfile from my key server...
curl -o ~/.ssh/authorized_keys
https://keys.example.com/authorized_keys
chmod 600 ~/.ssh/authorized_keys
%end
```

How it works...

The `system-config-kickstart` is used to generate a minimal install as any addition would be more complex than the tool can handle and we need to be able to add them manually/dynamically afterwards. The fewer the number of packages the better as you'll need to apply bug and security fixes for every package installed.

Although the GUI allows us to configure the brunt of the options we need, I prefer tweaking some portions of them manually as they are not as straightforward through the GUI.

Step 9 adds the necessary information to use the rest of the disk as an LVM physical volume and partitions it so that big filesystems can easily be extended if necessary.
The **--recommended** argument for the SWAP partition creates a swap partition as per the swap size recommendations set by Red Hat.

Step 10 adds a hostname for your host. If you do not specify this, the system will attempt to resolve the IP address and use this hostname. If it cannot determine any hostname, it will use `localhost.localdomain` as fqdn.

Step 11 ensures that only the core system is installed and nothing more, so you can build from here.

If you want to know exactly which packages are installed in the core group, run the following command on an RHEL 7 system:

```
-# yum groupinfo core
```

### There’s more...

I didn’t cover one option that I mentioned in the *Getting Ready* section as it is automatically generated when you install a system manually. The file can be found after installation at `/root/anaconda-ks.cfg`. Instead of using the `system-config-kickstart` tool to generate a kickstart file, you can use this file to get started.

Starting with RHEL 7, kickstart deployments support add-ons. These add-ons can expand the standard kickstart installation in many ways. To use kickstart add-ons, just add the `%addon` `addon_name` option followed by `%end`, as with the `%pre` and `%post` sections. Anaconda comes with the `kdump` add-on, which you can use to install and configure `kdump` during the installation by providing the following section in your kickstart file:

```
%addon com_redhat_kdump --enable --reserve-mb=auto
%end
```

### See also

For more detailed information about kickstart files, refer to the website [https://github.com/rhinstaller/pykickstart/blob/master/docs/kickstart-docs.rst](https://github.com/rhinstaller/pykickstart/blob/master/docs/kickstart-docs.rst).

Publishing your kickstart file using httpd

You can save your kickstart file to a USB stick (or any other medium), but this becomes a bit cumbersome if you need to install multiple systems in different locations.

Loading kickstart files over the network from the kernel line during an install only supports NFS, HTTP, and FTP.

In this recipe, I choose HTTP as it is a common technology within companies and easy to secure.

How to do it...

Let’s start by installing Apache httpd, as follows:

1. Install Apache httpd through the following command:
   ```bash
   ~ ]# yum install -y httpd
   ```

2. Enable and start the httpd daemon, as follows:
   ```bash
   ~ ]# systemctl enable httpd
   ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/
   system/multi-user.target.wants/httpd.service'
   ~ ]# systemctl start httpd
   ```

3. Create a directory to contain the kickstart file(s) by running the following command:
   ```bash
   ~ ]# mkdir -p /var/www/html/kickstart
   ~ ]# chown apache:apache /var/www/html/kickstart
   ~ ]# chmod 750 /var/www/html/kickstart
   ```

4. Copy your kickstart file to this new location:
   ```bash
   ~ ]# cp kickstart.ks /var/www/html/kickstart/
   ```
5. In a browser, browse to the kickstart directory on your web server, as shown in the following screenshot:

![Index of /kickstart](image)

**Index of /kickstart**

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>kickstart ks</td>
<td>2015-06-27 11:46</td>
<td>1.5K</td>
<td></td>
</tr>
</tbody>
</table>

**There's more...**

In this way, you can create multiple kickstart files, which will be available from anywhere in your network.

Additionally, you could use CGI-BIN, PHP, or any other technology that has an Apache module to dynamically create kickstart files based on the arguments that you specify in the URL.

An alternative to creating your own solution for dynamic kickstart files is Cobbler.

**See also**

For more info on Cobbler, go to [http://cobbler.github.io/](http://cobbler.github.io/).
Deploying RHEL "En Masse"

Deploying a system using PXE

PXE, or Preboot eXecution Environment, allows you to instruct computers to boot using network resources. This allows you to control a single source to install servers without the need to physically insert cumbersome DVDs or USB sticks.

Getting ready

For this recipe, you will need a fully working RHEL 7 repository.

How to do it...

With this recipe, we'll install and configure PXE boots from the RHEL 7 installation media, as follows:

1. Install the necessary packages using the following command:
   ```bash
   #!# yum install -y dnsmasq syslinux tftp-server
   ```
2. Configure the DNSMASQ server by editing `/etc/dnsmasq.conf`, as follows:
   ```plaintext
   # interfaces to bind to
   interface=en01,lo
   # the domain for this DNS server
domain=rhel7.lan
   # DHCP lease range
dhcp-range=en01,192.168.0.3,192.168.0.103,255.255.255.0,1h
   # PXE - the address of the PXE server
dhcp-boot=pxelinux.0,pxeserver,192.168.0.1
   # Gateway
dhcp-option=3,192.168.0.254
   # DNS servers for DHCP clients (your internal DNS servers, and one of Google's DNS servers)
dhcp-option=6,192.168.1.1, 8.8.8.8
   # DNS server to forward DNS queries to
   server=8.8.4.4
   # Broadcast Address
dhcp-option=28,192.168.0.255
   pxe-prompt="Press F1 for menu.", 60
   pxe-service=x86_64PC, "Install RHEL 7 from network",
   pxelinux
   enable-tftp
   tftp-root=/var/lib/tftpboot
   ```
3. Enable and start dnsmasq using the following:
   ```bash
   ~]# systemctl enable dnsmasq
   ~]# systemctl start dnsmasq
   ```

4. Now, enable and start the xinet daemon by running the following:
   ```bash
   ~]# systemctl enable xinetd
   ~]# systemctl start xinetd
   ```

5. Enable the tftp server's xinet daemon, as follows:
   ```bash
   ~]# sed -i '/disable/ s/yes/no/' /etc/xinetd.d/tftp
   ```

6. Copy the syslinux boot loaders to the tftp server's boot directory by executing the following command:
   ```bash
   ~]# cp -r /usr/share/syslinux/* /var/lib/tftpboot
   ```

7. Next, create the PXE configuration directory using this command:
   ```bash
   ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
   ```

8. Then, create the PXE configuration file, as follows:
   ```bash
   default menu.c32
   prompt 0
   timeout 300
   ONTIMEOUT local
   menu title PXE Boot Menu
   label 1
       menu label ^1 - Install RHEL 7 x64 with Local http Repo
       kernel rhel7/vmlinuz
       append initrd=rhel7/initrd.img method=http://repo.critter.be/rhel7/os/x86_64/
devfs=nomount ks=http://kickstart.critter.be/kickstart.ks
   label 2
       menu label ^2 - Boot from local media
   ```

9. Copy initrd and kernel from the RHEL 7 installation media to /var/lib/tftpboot/rhel7/, and run the following commands:
   ```bash
   ~]# mkdir /var/lib/tftpboot/rhel7/
   ~]# mount -o loop /dev/cdrom /mnt
   ~]# cp /mnt/images/pxeboot/{initrd.img,vmlinuz} /var/lib/tftpboot/rhel7/
   ~]# umount /mnt
   ```
10. Open the firewall on your server using these commands (however, this may not be necessary):

```bash
~]# firewall-cmd --add-service=dns --permanent
~]# firewall-cmd --add-service=dhcp --permanent
~]# firewall-cmd --add-service=tftp --permanent
~]# firewall-cmd --reload
```

11. Finally, launch your client, configure it to boot from the network, and select the first option shown in the following figure:

![PXE Boot Menu](image)

**How it works...**

DNMSAQ takes care of pointing booting systems to the `tftp` server by providing the `enable-tftp` option in the `dnsmasq` configuration file.

Syslinux is needed to provide the necessary binaries to boot from the network.

The `tftp` server itself provides access to the `syslinux` files, RHEL 7 kernel, and `initrd` for the system to boot from.

The PXE configuration file provides the necessary configuration to boot a system, including a kickstart file that automatically installs your system.
There's more...

This recipe's base premise is that you do not have a DHCP server installed. In most companies, you already have DHCP services available.

If you have an ISC-DHCP server in place, this is what you need to add to the subnet definition(s) you want to allow in PXE:

```
next-server <ip address of TFTP server>;
filename "pxelinux.0";
```

See also

Check out Chapter 8, Yum and Repositories to set up an RHEL 7 repository from the installation media.

Deploying a system using a custom boot ISO file

PXE is a widely used way to deploy systems, and so are ISO's. PXE may not always be at hand because of security, hardware availability, and so on.

Many hardware manufacturers provide remote access to their systems without an OS installed. HP has iLO, while Dell has RIB. The advantage of these "remote" control solutions is that they also allow you to mount "virtual" media in the form of an ISO.

How to do it...

Red Hat provides boot media as ISO images, which you can use to boot your systems from. We will create a custom ISO image, which will allow us to boot a system in a similar way.

Let's create an ISO that you can mount as virtual media, write a CD-ROM, or even use `dd` to write the contents on a USB stick/disk through the following steps:

1. Install the required packages to create ISO9660 images, as follows:
   ```bash
   $ sudo yum install -y genisoimage
   ```
2. Mount the RHEL 7 DVD's ISO image by executing the following command:
   ```bash
   $ sudo mount -o loop /path/to/rhel-server-7.0-x86_64-dvd.iso /mnt
   ```
3. Copy the required files for the custom ISO from the RHEL 7 media via the following commands:
   ```bash
   ~# mkdir -p /root/iso
   ~# cp -r /mnt/isolinux /root/iso
   ~# umount /mnt
   ```

4. Now, unmount the RHEL 7 DVD's ISO image by running the following:
   ```bash
   ~# umount /mnt
   ```

5. Next, remove the `isolinux.cfg` file using the following command:
   ```bash
   ~# rm -f /root/iso/isolinux/isolinux.cfg
   ```

6. Create a new `isolinux.cfg` file, as follows:
   ```bash
   default vesamenu.c32
   timeout 600
   display boot.msg
   menu clear
   menu background splash.png
   menu title Red Hat Enterprise Linux 7.0
   menu vshift 8
   menu rows 18
   menu margin 8
   menu helpmsgrow 15
   menu tabmsgrow 13
   menu color sel 0 #ffffff #00000000 none
   menu color title 0 #ffcc000000 #00000000 none
   menu color tabmsg 0 #84cc0000 #00000000 none
   menu color hotsel 0 #84cc0000 #00000000 none
   menu color hotkey 0 #ffffff #00000000 none
   menu color cmdmark 0 #84b8ffff #00000000 none
   menu color cmdline 0 #ffffff #00000000 none
   label linux
      menu label ^Install Red Hat Enterprise Linux 7.0
      kernel vmlinuz
      append initrd=initrd.img ks=http://kickstart.critter.be/kickstart.ks text
   
   label local
      menu label Boot from ^local drive
      localboot 0xffff
   ```
7. Now, create the ISO by executing the following command:

```bash
# cd /root/iso
~/iso# mkisofs -o ../boot.iso -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table
-J -r .
```

More information on the options used with the `mkisofs` command can be found in the man pages for `mkisofs(1)`.

The following image shows the progress on creating a custom ISO:

8. Then, use the ISO to install a guest on a KVM server, as shown in the following commands:

```bash
# virsh vol-create-as --pool localfs-vm --name rhel7_guest-da.
qucows2 --format qcows2 --capacity 10G

# virt-install \
--hvm \
--name rhel7_guest \
```
Deploying RHEL "En Masse"

```
--memory 2G,maxmemory=4G \n--vcpus 2,max=4 \n--os-type linux \n--os-variant rhel7 \n--boot hd,cdrom,network,menu=on \n--controller type=scsi,model=virtio-scsi \n--disk device=cdrom,vol=iso/boot.iso,readonly=on,bus=scsi \n--disk device=disk,vol=localfs-vm/rhel7_guest-vda.qcow2,cache=none,bus=scsi \n--network network=bridge-eth0,model=virtio \n--graphics vnc \n--graphics spice \n--noautoconsole \n--memballoon virtio
```

The following screenshot shows the console when booted with the custom ISO image:
How it works...

Using the RHEL 7 installation media, we created a new boot ISO that allows us to install a new system. The ISO can be used to either burn a CD, with the `dd` tool to be copied on a USB stick, or to mount as virtual media. The way to mount this ISO as virtual media is different on each hardware platform, so this recipe shows you how to install it using KVM.
Where to buy this book

You can buy Red Hat Enterprise Linux Server Cookbook from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.