on the job

Red Hat is working to ease the process of creating a Kickstart-based installation server. For more information, see the Cobbler project at https://fedorahosted.org/cobbler/.

Sample Kickstart File

I've based this section on the anaconda-ks.cfg file created when I installed RHEL 6 on a KVM-based VM. I've added a number of comments. While you're welcome to use it as a sample file, be sure to customize it for your hardware and network. This section just scratches the surface on what you can do with a Kickstart file; your version of this file may vary.

exam watch

Unlike what's available for many other Red Hat packages, available Kickstart documentation within an installed RHEL 6 system is somewhat sparse. In other words, you can't really rely on man pages or files in the /usr/share/doc directory for help during an exam. If you're uncertain about specific commands to include in the Kickstart file, the Kickstart Configurator described later in this chapter can help.

While most of the options are self-explanatory, I've interspersed my explanation of each command within the file. This file illustrates just a small portion of available commands. For more information on each command (and options) in this file, read the latest RHEL 6 Installation Guide, which is available online at http://docs.redhat.com/docs/en-US.

Follow these ground rules and guidelines when setting up a Kickstart file:

- In general, retain the order of the directives. However, some variation is allowable depending on whether the installation is from local media or over a network.
- You do not need to use all the options.
- If you leave out a required option, the user will be prompted for the answer.
- Don't be afraid to make a change; for example, partition-related directives are commented out by default.

on the job

If you leave out an option, the installation process will stop at that point. This is an easy way to see if a Kickstart file is properly configured. But as some Kickstart options change the partitions on a hard drive, even tests can be dangerous. So it's best to test a Kickstart file on a test system, or even better, an experimental VM.

The following is the code from one of my anaconda-ks.cfg files. The first two lines are comments that tell me that this file was created during the installation process for RHEL 6:

```
# Kickstart file automatically generated by anaconda.
#version=RHEL6
```

The first command is simple; it starts the installation process. It defaults to the first available local media; in this case, the first RHEL installation DVD/CD or USB key.

```
install
```

The next step is to specify the source of the installation files. To use RHEL 6 DVDs, enter **cdrom**. To install from an NFS server, specify the URI as follows. If there's a reliable DNS server on the local network, you can substitute the hostname for the IP address.

```
nfs --server=192.168.122.1 --dir=/inst
```

You can also configure a connection to an FTP or HTTP server by substituting one of the commands shown here. The directories I specify are based on the FTP and HTTP installation servers created in Chapter 1:

```
url --url http://192.168.122.1/inst
```

or

```
url --url ftp://192.168.122.1/pub/inst
```

If the ISO file that represents the RHEL 6 DVD exists on a local hard drive partition, you can specify that as well. For example, the following directive points to ISO CDs or DVDs on the /dev/sda10 partition:

```
harddrive --partition=/dev/sda10 --dir=/home/michael/
```

The **lang** command specifies the language to use during the installation process. It matters if the installation stops due to a missing command in this file. The **keyboard** command is self-explanatory, as it specifies the keyboard to configure on this computer.

```
lang en_US.UTF-8
keyboard us
```

The required **network** command is simplest if there's a DHCP server for the local network: **network --device eth0 --bootproto dhcp**. In contrast, the following line configures static IP address information, with the noted network mask (**--netmask**), gateway address (**--gateway**), DNS server (**--nameserver**), and computer name (**--hostname**).

```
network --device eth0 --bootproto static --ip 192.168.122.150 --netmask
255.255.255.0 --gateway 192.168.122.1 --nameserver 192.168.122.1 --hostname
tester1.example.com
```

Please note that all options for the **network** command *must* be on *one* line. Line wrapping, if the options exceed the space in a text editor, is acceptable. If you're setting up this file for a different system, don't forget to change the IP address and hostname information accordingly. Be aware, if you did not configure networking during the installation process, it won't be written to the subject anaconda-ks.cfg file. Given the complexity of the network directive, you could either use the Kickstart Configurator to help set up that directive, or configure networking after installation is complete.

As the password for the root user is part of the RHEL 6 installation process, the Kickstart configuration file can specify that password in encrypted format. While encryption is not required, it can at least delay a cracker who might break into a system after installation is complete. Since the associated cryptographic hash function is the same as is used for the /etc/shadow file, you can copy the desired password from that file.

```
rootpw --iscrypted $6$5UrLfXTk$CsCW0nQytrUuvycuLT317/
```

As for security, the **firewall** directive suggests that it's enabled. When coupled with **--service=ssh**, it specifies the service port number that's allowed through the firewall, based on how it's defined in the /etc/services file.

```
firewall --service=ssh
```

Next, the **authconfig** command sets up the Shadow Password Suite (**--enableshadow**), the SHA 512 bit encryption algorithm for password encryption (**--passalgo=sha512**), and authentication with any existing fingerprint reader.

An password encrypted to the SHA512 algorithm starts with a **$6**, like the root administrative password just shown.

```
authconfig --enableshadow --passalgo=sha512
--enablefingerprint
```

The **selinux** directive can be set to **--enforcing**, **--permissive**, or **--disabled**.

```
selinux --enforcing
```

The **timezone** command is associated with a long list of time zones. They're documented in the tzdata package. For a full list, run the **rpm -ql tzdata** command. By default Red Hat sets the hardware clock to the equivalent of Greenwich Mean Time with the **--utc** switch. That setting supports automated changes for daylight saving time. The following setting can be found as a subdirectory and file in the /usr/share/zoneinfo directory.

```
timezone America/Los_Angeles
```

The default bootloader is GRUB. It should normally be installed on the Master Boot Record (MBR) of a hard drive. You can include a **--driveorder** switch to specify the drive with the bootloader and an **--append** switch to specify commands for the kernel. While the given **crashkernel=auto** option should automatically select available memory upon a crash, early reports on RHEL 6 suggest that you may need to replace it with a specific memory location such as **crashkernel=128M@16M**.

```
bootloader --location=mbr --driveorder=vda
--append="crashkernel=auto rhgb quiet"
```

As suggested by the comments that follow, it's first important to clear some existing set of partitions. First, the **clearpart --drives=vda --all --initlabel** directive clears all volumes on the vda virtual hard drive. If it hasn't been used before, **--initlabel** initializes that drive. Of course, before such a command takes effect, any existing comment character (#) must be removed. The **ignoredisk** directive that follows specifies volumes only on the noted vda drive. Of course, this works only if there is an specified virtual drive on the target VM. (It's possible to specify PATA or SCSI drives on such VMs, which would conflict with these directives.)

```
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
```

```
clearpart --drives=vda --all --initlabel
ignoredisk --only-use=vda
```

If you're planning to use this Kickstart file with the **virt-install** command described earlier, you'll need to substitute the sda device for vda, as the **virt-install** command does not normally use that virtual hard drive device file.

Changes are required in the partition (**part**) directives that follow. They should specify the directory, filesystem format (**--fstype**), and **--size** in MB.

```
part /boot --fstype=ext4 --size=500
part / --fstype=ext4 --size=8000
part swap --size=1000
part /home --fstype=ext4 --size=1000
```

Be aware, your version of an anaconda-ks.cfg file may include an **--onpart** directive that specifies partition device files such as /dev/vda1. That would lead to an error unless the noted partitions already exist. So if you see those **--onpart** directives, it's simplest to delete them. Otherwise, you'd have to create those partitions before starting the installation process, and that can be tricky.

The default version of the Kickstart file may contain a **repo** directive. It would point to the FTP network installation source from Chapter 1, Lab 2, and should be deleted from or commented out of the Kickstart file as follows:

```
#repo --name="Red Hat Enterprise Linux"
--baseurl=ftp://192.168.122.1/pub/ --cost=100
```

To make sure the system actually completes the installation process, this is the place to include a directive such as **reboot**, **shutdown**, **halt**, or **poweroff**. I personally prefer the **shutdown** directive; if you want to avoid the First Boot process described earlier, you can also include the **firstboot --disabled** directive. As there's no way to set up a Kickstart file with answers to the First Boot process, that **--disabled** directive helps automate the Kickstart process.

```
shutdown
firstboot --disabled
```

If you're reusing an existing KVM-based VM, it may be necessary to shut off the system to change the boot media from the CD/DVD to the hard drive. So instead, you may prefer to substitute the following directive:

```
shutdown
```

While other options are available with respect to setting up RAID arrays and logical volumes, the focus of the Red Hat exams is to set up such volumes after installation is complete. What follows is a list of package groups that are installed through this Kickstart configuration file. These names correspond to the names that you can find in the *-comps-rhel6-Server.xml file in the RHEL 6 DVD /repodata directory described in Chapter 1. Since the list is long, the following is just excerpts of package groups (which start with the @) and package names:

```
%packages
@ base
@ console-internet
...
nss-pam-ldapd
perl-DBD-SQLite
%end
```

After the package groups are installed, you can specify post-installation commands after the following directive. For example, you could set up custom configuration files. But the %post directive and anything that follows is not required.

```
%post
```

EXERCISE 2-3

Create and Use a Sample Kickstart File

In this exercise, you will use the anaconda-ks.cfg file to duplicate the installation from one computer to another with identical hardware. This exercise installs all the exact same packages with the same partition configuration on the second computer. This exercise even configures the SELinux context for that Kickstart file.

As the objective is to install the same packages as the current installation, no changes are required to packages or package groups from the default anaconda-ks.cfg file in the /root directory. This assumes access to a network installation source such as that created in Chapter 1 Lab 2.

The steps in this exercise assume sufficient space and resources for at least two different KVM-based VMs, as discussed in Chapter 1.

1. Review the /root/anaconda-ks.cfg file. Copy it to ks.cfg.
2. If there's an existing **network** directive in the file, modify it to point to an IP address of 192.168.122.150, with a hostname of tester1.example.com. It is okay if such a directive doesn't already exist; networking can be configured after installation is complete, using the techniques discussed in Chapter 3.