



Automated installation with the Cobbler provisioning tool

NEW SHOES

Cobbler helps you install new systems in a hurry. We'll show you how to use this nifty shoemaker to deploy Xen and VMware virtual machines. **BY DAVID NALLEY**

If you try to install or upgrade several computers at once, you will soon discover that manual installation is a huge time sink. Even if you are working with a checklist, it is often difficult to get everything installed the same way every time. For this reason, most systems administrators understand the importance of an automated install system.

It's no surprise that virtually every operating system has the ability to automate installations. What is curious, given the necessity of such systems, is that configuring automated installation typically requires so much time and effort. In the past, automated installation required a fair amount of knowledge about networking, Pre-eXecutable Environment (PXE), and, of course, the operating system itself. Those prerequisites represented a significant barrier to entry.

Automated provisioning techniques have taken a number of forms over the years. The approach familiar to most people is taking a disk image of a system with the use of *dd*, PING (Partimage is not Ghost) [1], or another tool, then deploying that disk image anytime you need it. Unfortunately, this solution doesn't scale well because it relies on a

uniform environment and does not take into account differences in hardware or function. Also, the image gets out of date over time – with the constant appearance of updates, managing the images can become a full-time job. The more robust way of provisioning is by automatically installing the operating system each time, rather than relying on a predefined disk image. This method is generally considered more challenging.

The Cobbler project [2], born at Red Hat and lead by Michael DeHaan, significantly lowers the barrier for provisioning in Linux. Cobbler is an all-encompassing framework for automated provisioning systems that provides a “soup-to-nuts” solution, bringing together provisioning prerequisites like DHCP and TFTP, providing management utilities for controlling software repositories, and offering up a framework to enable both a command-line and web interface for easy management (Figure 1).

In addition to handling hardware-based installs, Cobbler is proficient at configuring virtual machines: You can provision a collection of Xen or KVM virtual machines as easily as a new hardware-based server or workstation. In this

article, I show you how to set up a Cobbler provisioning system and take a look at the provisioning of virtual machines with Cobbler's helper application Koan.

Cobbler provides a framework for configuring and managing:

- *dhcpcd* – the DHCP daemon
- *dnsmasq* – masquerading DNS server
- *bind* – DNS server
- *tftpd* – trivial file transfer protocol daemon

By managing all the configurations for all services, Cobbler literally can run an entire provisioning network from a single box. Of course, you don't have to use Cobbler to configure all these services because you can easily use existing DNS

Table 1: Required Ports and Services

Port	Service
TCP 53	DNS
UDP 53	DNS
UDP 68	DHCP
UDP 69	TFTP
TCP 80	HTTP
UDP 25150	Remote syslog
TCP 25151	XMLRPC
TCP 25152	XMLRPC

and DHCP servers if they are already present. In the event they aren't, you can use Cobbler to manage the daemons and their configurations.

Aside from letting you manage daemons, Cobbler provides a framework to mirror the installation framework from a DVD or an Internet repository. Additionally, Cobbler can mirror and keep current with third-party repositories. In the event that PXE booting isn't an option, Cobbler can also generate a bootable ISO that will present the user with the same installation options that are presented to a machine that uses PXE.

A single Cobbler command will download an operating system from the Internet or optical media, configure Syslinux, and make the system ready for deployment. Another command grabs all of the packages from an update repository.

Most discussions of Cobbler focus on its ease of use, but equally important is its power and flexibility. For instance, Cobbler allows you to use snippets of code across many configuration files. If a snippet of code needs to be in multiple Kickstart files, you can place it in one file. The classic use for this feature is with SSH keys. No one wants to manually push an SSH key to each host. Rather than being limited to simple includes, Cobbler has the power to dynamically insert code snippets so you can programmatically change the configuration on the basis of the type of machine, function of machine, IP address, or a host of other factors. This approach allows you to have a single Kickstart file with embedded logic that deploys differ-

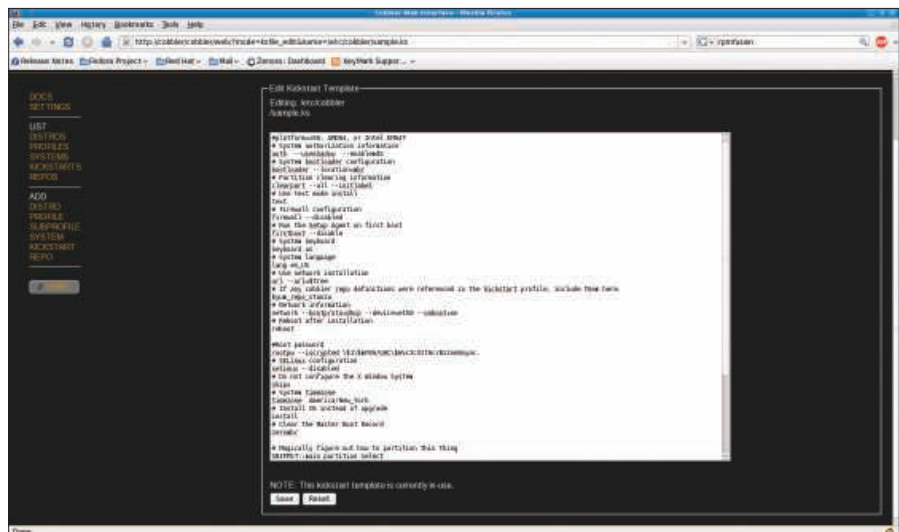


Figure 2: Editing a Kickstart template.

ent sets of packages and pushes differing configuration files based on that logic. For example, Cobbler allows you to push an SSH key up programmatically, so your development machines might get one set of SSH keys and your mail servers get a different set.

This example is just a simple demonstration of the type of flexibility that Cobbler has built in. Another feature that is getting a lot of attention is Cobbler's integration with configuration management systems such as Puppet.

Installing Cobbler

Although packages are available for Cobbler in Debian and Ubuntu, they are still very much in development. At the time of this writing, a machine with Fedora 8 or 9 or with RHEL/CentOS 4 or 5 would be a better candidate for Cobbler. Once

the operating system is installed, you can set up Cobbler and its dependencies. For the remainder of this article, I'll assume you plan to use Cobbler as an all-in-one provisioning system and that it will be managing DHCP and DNS.

Cobbler is available in the standard repository set with Fedora, but with RHEL and CentOS, you'll need to use the EPEL repositories. To install Cobbler, enter:

```
yum install cobbler httpd
dhcpd bind tftp-server
```

If you have SELinux enabled, you will need to configure it so that Cobbler can serve up the dynamically configured Kickstart files:

```
setsebool -P
httpd_can_network_connect true
```

If you change the `/etc/cobbler/settings` file to contain the following settings, Cobbler will manage configuration and startup of DHCP and DNS on the installation client. The `server` and `next_server` settings relate to PXE booting and are used in configuring DHCP and defining where machines should look to get boot information. In this case, the cobbler server has the address `192.168.1.5`:

```
manage_dhcp: 1
manage_dns: 1
next_server: '192.168.1.5'
server: '192.168.1.5'
```

If you use an address other than `192.168.1.5`, modify `/etc/cobbler/dhcp`.

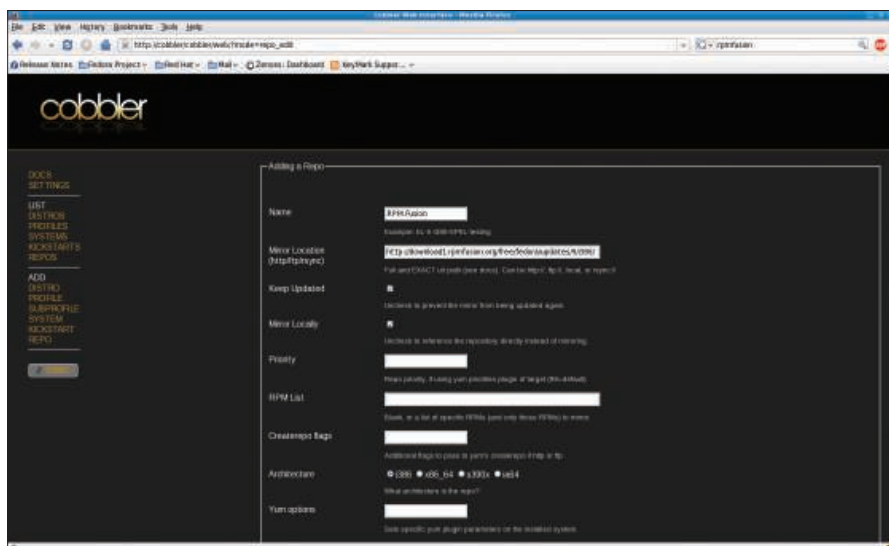


Figure 1: Cobbler comes with an optional web-based interface.

template appropriately to match the network you have chosen. If you don't do this, machines will unsuccessfully attempt to PXE boot to the 192.168.1.5 address, and machines booted with a Cobbler-generated network install ISO will vainly attempt to reach installation repositories on 192.168.1.5.

In */etc/cobbler/named.template*, you'll want to set the *listen_on* variable to match the IP address of the Cobbler server. This step defines one of the few information items Cobbler needs to configure bind to supply DNS, and of course, it instructs the bind configuration to listen on port 53, which is the standard port for DNS:

```
listen-on port 53 {
  192.168.1.5; }
```

If you are using a firewall, make sure you open the ports shown in Table 1.

Getting the Bits

Now it is time to import the first distribution. Cobbler makes this an easy, one-command step, with only two arguments: the name you assign to the distribution and the location of the installation data. The installation files can be pulled from install media or from the Internet.

If you have DVD install media, the command is as easy as:

```
cobbler import --name=F9 --mirror=/media/dvd
```

To use an Internet mirror, enter:

```
cobbler import --name=F9 --mirror=
rsync://mirror.anl.gov/fedora/linux/releases/9/
Fedora/i386/
```

Listing 2: Starting Services

```
01 /sbin/chkconfig httpd on
02 /sbin/chkconfig dhcpd on
03 /sbin/chkconfig xinetd on
04 /sbin/chkconfig tftpd on
05 /sbin/chkconfig cobblerd on
06 /sbin/service httpd start
07 /sbin/service dhcpd start
08 /sbin/service xinetd start
09 /sbin/service cobblerd start
```

Listing 1: Cobbler's Configuration Check

```
01 [root@nalleyt61 ~]# cobbler check
02 The following potential problems
03 were detected:
04 #0: service cobblerd is not running
05 #1: service httpd is not running
06 #2: since iptables may be running,
07 ensure 69, 80, 25150, and 25151 are
08 unblocked
09 #3: One or more kickstart
10 templates references default
11 password 'cobbler' and should
12 be changed for security reasons
13 : /etc/cobbler/sample.ks,
14 /etc/cobbler/legacy.ks,
15 /etc/cobbler/sample_end.ks
```

One of the interesting features of Cobbler is that it provisions systems dynamically based on the latest packages. This means that if Cobbler knows about an update repository, it will install the latest updated packages in that repository. This can save lots of time by eliminating post-install updates, particularly if it has been a long while since the distribution was released. Also, you are not limited to update repositories; you could just as easily use a repository for Dell's OpenManage or HP's System Management packages or some other third-party repository.

To define an update repository, use the *cobbler repo add* command, assign the repository a name, and then specify the location:

```
cobbler repo add --name=F9-updates
--mirror=http://mirror.anl.gov/fe
dora/linux/updates/9/i386/
```

With a repository defined by the preceding command, the actual content is synced from the mirror to the local Cobbler server with the following command.

To keep this local repository in sync, you need to run this command regularly or automate it with a cron job. Note that, if you have defined multiple repositories, this command will sync them all:

```
cobbler reposync
```

For a home setup or installfest, you might disregard the following setting, but if Cobbler will be running in a large lab or in a corporate environment, you will want to point your installed machines at the repositories Cobbler is now keeping locally. This allows you to maintain a local update repository and update all of your machines based on that repository. You can change this setting in */etc/cobbler/settings* to match:

```
yum_post_install_mirror: 1
```

Cobbler uses the concept of a *profile* to define a collection of common configuration settings. A Cobbler profile consists of a distribution, optionally one or more additional repositories, and the installation configuration file. This flexible technique allows you to use the same distribution in multiple profiles – each with different installation configuration files – or even different repositories. In addition to providing flexibility, the profile approach also minimizes the amount of storage used over a traditional image-based provisioning system. A profile for installing Fedora 9 with the latest updates would look like:

```
cobbler profile add --name=
Fedora9 --distro=F9-i386
```

What if I Use a Non-RH-like Distro?

It's no surprise that Cobbler is great at deploying Red Hat and Fedora; after all, Cobbler's primary author, Michael DeHaan, is a Red Hat employee and Fedora contributor, and Cobbler originated (and still lives) within Red Hat's Emerging Technologies division. However, Cobbler will also install SUSE and Debian-based distributions. Some of the other interesting options that are already available include PXE booting any Live CD and updating firmware for network hardware from HP and Dell. As of this writing, development work is occurring to provide provisioning of Windows, Solaris, and hard drive images that are operating system agnostic.

For those who don't want to use Fedora or RHEL for their provisioning system, Debian and Ubuntu community members are working to get it packaged and functional for Debian-based distributions.

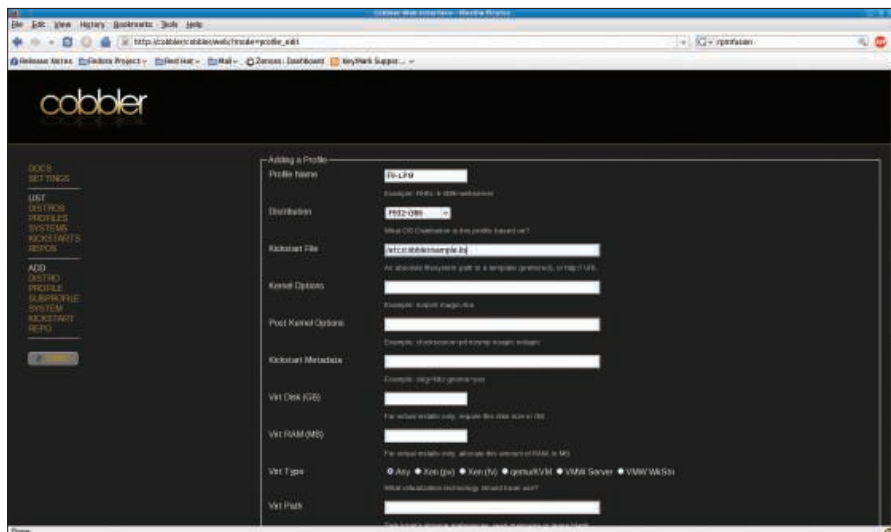


Figure 3: A profile associates a distribution with a Kickstart file and other settings.

```
--updates=F9-updates
--kickstart=?
/etc/cobbler/sample.ks
```

Note that I listed *sample.ks* as the Kickstart file. That particular file will install a minimal system. It is a good idea to add some packages and possibly make some configuration changes in the Kickstart file. This *sample.ks* is a good base to start with, though because it certainly has the minimum requirements, and you can add from there.

After you make any changes to */etc/cobbler/settings* (or use any of the cobbler commands), you'll want to run *cobbler sync* so that the changes are reflected by the Cobbler daemon. Now is a good time to do so:

```
cobbler sync
```

Cobbler also comes with a configuration-checking mechanism that looks for things that might be awry, such as a firewall blocking traffic, SELinux prohibiting

Cobbler from functioning, or one of the prerequisite services not running. You can invoke that mechanism with the following command:

```
cobbler check
```

At this point, Cobbler should return the results (Listing 1).

Problem numbers 0 and 1 (*service cobblerd is not running* and *service httpd is not running*, respectively) will be rectified shortly. Problem number 2 (*since iptables may be running, ensure 69, 80, 25150, and 25151 are unblocked*) will throw an alert if iptables is running, so as long as the ports listed above are opened, this problem can be ignored. Problem number 3 (*One or more kickstart templates references default password 'cobbler' and should be changed for security reasons*) will show up because all of the sample Kickstarts have a root password that is *cobbler*. This situation is only a problem if those Kickstarts are to be used. The root password in each of those files can be changed easily. Because I assume that a user-created Kickstart file will be used, wherein a unique root password will be set, you can disregard this problem as well.

The services that Cobbler uses must be started and configured to start on boot (see Listing 2). At this point, I should indicate that 90 percent of Cobbler's configuration is also available in the web interface. By changing the *[authentication]* portion of */etc/cobbler/modules.conf*, you can enable access to the web interface:

```
[authentication]
module = authn_configfile
```

This change will allow connections to *http://192.168.1.5/cobbler/web* with a username of *cobbler* and a password of *cobbler*. Please be sure to change the password to something other than the default as follows:

```
htdigest
/etc/cobbler/users.digest
"Cobbler" cobbler
```

From within the web interface, system administrators can add, modify, and delete distributions, repositories, and profiles.

At this point, you can PXE boot or boot to a Cobbler-generated network install ISO and begin a network install that's completely automated.

Burning and booting the created ISO file will present the user with the same options and experience as if you had used PXE. The resultant CD already has the connection information to point back to Cobbler and the list of available profiles.

Virtual Provisioning with Koan

Koan, Cobbler's client-side helper application, makes it easy to deploy virtual machines. With a Cobbler server in place, you can use Koan to configure your virtual disk, network, and other functions automatically; load the OS automatically; and start the virtual machine. Moreover, Koan works with Xen, KVM, and certain versions of VMware.

Currently, only VMware Workstation and VMware Server are supported, but

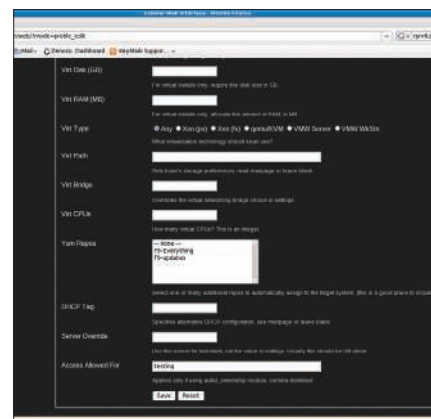


Figure 4: The full range of profile options offers a flexible framework for customizing the virtual system.

Table 2: Koan Override Options

Option	Description
--virt-name	Name of virtual machine to create
--virt-type	Forces use of <i>qemu/xen/vmware</i>
--virt-ram	Size in megabytes
--virt-bridge	Name of bridge device
--virt-path	Use this block device, directory
--no-gfx	Do not use VNC graphics (Xen only)

support for ESX is in progress. With Koan installed, a single command will take care of provisioning a virtual machine:

```
koan --server=2
cobber.example.org 2
--virt --virt-type=qemu 2
--profile=Fedora9
```

The preceding command will contact Cobbler to determine the amount of RAM and disk space, write the configuration file, start the virtual machine, and install the operating system. Sometimes, you might want to change default values such as the amount of memory. Table 2 shows some override options available to Koan.

VMware-based virtual machines require a bit more work – but only one more line. Essentially, you need to define a system record in Cobbler for the new virtual machine's MAC address and assign a profile:

```
cobbler system add --name=foo 2
--mac=00:50:56:00:00:00 2
#on the cobbler server
koan --server=2
cobber.example.org 2
--name=foo --virt 2
--virt-type=vmware 2
#on the vmware host
```

One cautionary note: VMware requires that the MAC address of the virtual machine be between 00:50:56:00:00:00 and 00:50:56:3F:FF:FF for networking to function properly.

Conclusion

Cobbler is a great enterprise-capable tool that removes a lot of the traditional headaches associated with setting up a provisioning system. This makes it great not just for the users at large data centers and clusters, but also for a LUG that wants to set up a network install server for an installfest. ■

What if I Don't Have Control of DHCP?

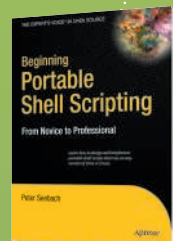
Not having control over DHCP is a common enough problem, especially in larger organizations. In some cases, networks don't even use DHCP, and some machines that use network cards don't support PXE booting. To enable PXE booting, Cobbler needs to have DHCP push out several options. One of those is the next-server option, which identifies what server is running tftpd and that the machines can load boot information for PXE. The other field is the file name that PXE will load from the tftp server. If you aren't able to configure a DHCP server to provide these settings, Cobbler can build a small ISO that contains essentially the same information that PXE booting would yield. For a Cobbler-generated network install ISO, enter:

```
cobbler buildiso
```

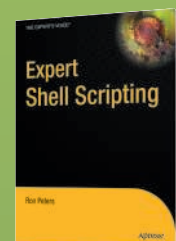
INFO

- [1] Partimage is not Ghost: <http://ping.windowsdream.com/>
- [2] Cobbler project: <https://fedorahosted.org/cobbler>
- [3] Kickstart documentation: <http://fedoraproject.org/wiki/Anaconda/Options>

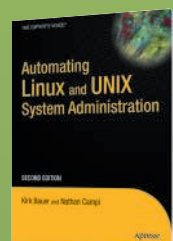
GET YOUR HANDS ON SOME HOT NEW BOOKS FROM APRESS



Peter Seebach
978-1-4302-1043-6
\$34.99 | 376 pp | November 2008



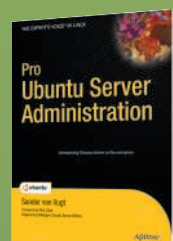
Ron Peters
978-1-4302-1841-8
\$39.99 | 330 pp | December 2008



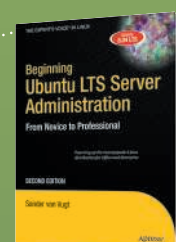
Kirk Bauer and Nathan Campi
978-1-4302-1059-7
\$49.99 | 425 pp | December 2008



Andy Channelle
978-1-4302-1590-5
\$39.99 | 500 pp | December 2008



Sander van Vugt
978-1-4302-1622-3
\$44.99 | 400 pp | December 2008



Sander van Vugt
978-1-4302-1082-5
\$39.99 | 424 pp | September 2008

For more information about Apress titles,
please visit www.apress.com

Don't want to wait for the printed book?
Order the eBook now at
<http://eBookshop.apress.com!>

Apress[®]
THE EXPERT'S VOICE™