developerWorks 中国 技术主题 Linux 文档库

使用 Cobbler 自动化和管理系统安装

使用 Cobbler 轻松设置和管理网络安装环境

Cobbler 通过将设置和管理一个安装服务器所涉及的任务集中在一起，从而简化了系统配置。本文探讨 Cobbler 的一些特性，如何安装，以及如何创建一种适合自动安装多个客户端机器的配置。

XML error: Please enter a value for the author element's jobtitle attribute, or the company-name element, or both.

2013 年 8 月 26 日

设置一个网络环境可能涉及到许多步骤，才能为开始安装做好准备。您必须：

- 配置服务，比如 DHCP、TFTP、DNS、HTTP、FTP 和 NFS
- 在 DHCP 和 TFTP 配置文件中填入各个客户端机器的信息
- 创建自动部署文件（比如 kickstart 和 autoinst）
- 将安装媒介解压缩到 HTTP/FTP/NFS 存储库中。

这个过程并不简单，而且手动注册每个必须配置的客户端机器可能很麻烦。对配置一台机器的任何参数更改（比如要使用一个不同的操作系统），都需要对配置进行手动干预，并有可能对自动部署文件进行手动干预。当机器数量增加时，如果不高度重视文件组织的条理性，TFTP 目录等元素就可能变得混乱。

Cobbler 通过为机器配置的所有方面创建一个中央管理点，从而解决了这些不足。Cobbler 可重新配置服务，创建存储库，解压缩操作系统媒介，代理或集成一个配置管理系统，控制电源管理等。Cobbler 创建了一个抽象层，您可在其中运行 “add new repository” 或 “change client machine operating system” 等命令。Cobbler 负责处理所有事情：创建或更新配置文件，重新启动服务，或者将媒介解压到新创建的目录中。它的目的是隐藏所有与系统相关的问题，以便您可专注于任务本身。

本文介绍 Cobbler 是如何被设计的，其主要特性，以及使用这些特性快速且轻松地配置机器的示例。首先介绍 Cobbler 的特性。

该工具提供的功能

使用 Cobbler，您无需进行人工干预即可安装机器。Cobbler 设置一个 PXE 引导环境（它还可使用 yaboot 支持 PowerPC），并控制与安装相关的所有方面，比如网络引导服务（DHCP 和 TFTP）与存储库镜像。当希望安装一台新机器时，Cobbler 可以：

- 使用一个以前定义的模板来配置 DHCP 服务（如果启用了管理 DHCP）
- 将一个存储库（yum 或 rsync）建立镜像或解压缩一个媒介，以注册一个新操作系统
- 在 DHCP 配置文件中为需要安装的机器创建一个条目，并使用您指定的参数（IP 和 MAC 地址）
- 在 TFTFP 服务目录下创建适当的 PXE 文件
- 重新启动 DHCP 服务以反映更改

Tivoli Configuration Manager

IBM Tivoli® Configuration Manager 是一个集成的软件分发和资产管理套件，包括两个主要组件 Software Distribution 和 Inventory，以及各种服务。Tivoli Configuration Manager 控制一个多平台环境中的配置、分发、变更、版本和资产管理。

重新启动机器以开始安装（如果电源管理已启用）

Cobbler 支持众多的发行版：Red Hat、Fedora、CentOS、Debian、Ubuntu 和 SuSE。当添加一个操作系统（通常通过使用 ISO 文件）时，Cobbler 知道如何解压缩合适的文件并调整网络服务，以正确引导机器。

Cobbler 可使用 kickstart 模板。基于 Red Hat 或 Fedora 的系统使用 kickstart 文件来自动化安装流程。通过使用模板，您就会拥有基本的 kickstart 模板，然后定义如何针对一种配置文件或机器配置而替换其中的变量。例如，一个模板可能包含两个变量 *$domain* 和 *$machine_name*。在 Cobbler 配置中，一个配置文件指定 `domain=mydomain.com`，并且每台使用该配置文件的机器在 *machine_name* 变量中指定其名称。该配置文件中的所有机器都使用相同的 kickstart 安装且针对 `domain=mydomain.com` 进行配置，但每台机器拥有其自己的机器名称。您仍然可以使用 kickstart 模板在不同的域中安装其他机器并使用不同的机器名称。

为了协助管理系统，Cobbler 可通过 *fence scripts* 连接到各种电源管理环境。Cobbler 支持 apc_snmp、bladecenter、bullpap、drac、ether_wake、ilo、integrity、ipmilan、ipmitool、lpar、rsa、virsh 和 wti。要重新安装一台机器，可运行 `reboot system foo` 命令，而且 Cobbler 会使用必要的凭据和信息来为您运行恰当的 fence scripts（比如机器插槽数）。

除了这些特性，还可使用一个配置管理系统 (CMS)。您有两种选择：该工具内的一个内部系统，或者集成一个现有的外部 CMS，比如 Chef 或 Puppet。借助内部系统，您可以指定文件模板，这些模板会依据配置参数进行处理（与 kickstart 模板的处理方式一样），然后复制到您指定的位置。如果必须自动将配置文件部署到特定机器，那么此功能很有用。

使用 koan 客户端，Cobbler 可从客户端配置虚拟机并重新安装系统。我不会讨论配置管理和 koan 特性，因为它们不属于本文的介绍范畴。但是，它们是值得研究的有用特性。（有关更多信息，请参阅 参考资料。）

Cobbler 的设计方式

Cobbler 的配置结构基于一组注册的对象。每个对象表示一个与另一个实体相关联的实体（该对象指向另一个对象，或者另一个对象指向该对象）。当一个对象指向另一个对象时，它就继承了被指向对象的数据，并可覆盖或添加更多特定信息。以下对象类型的定义为：

发行版：表示一个操作系统。它承载了内核和 initrd 的信息，以及内核参数等其他数据。

配置文件：包含一个发行版、一个 kickstart 文件以及可能的存储库，还包含更多特定的内核参数等其他数据。

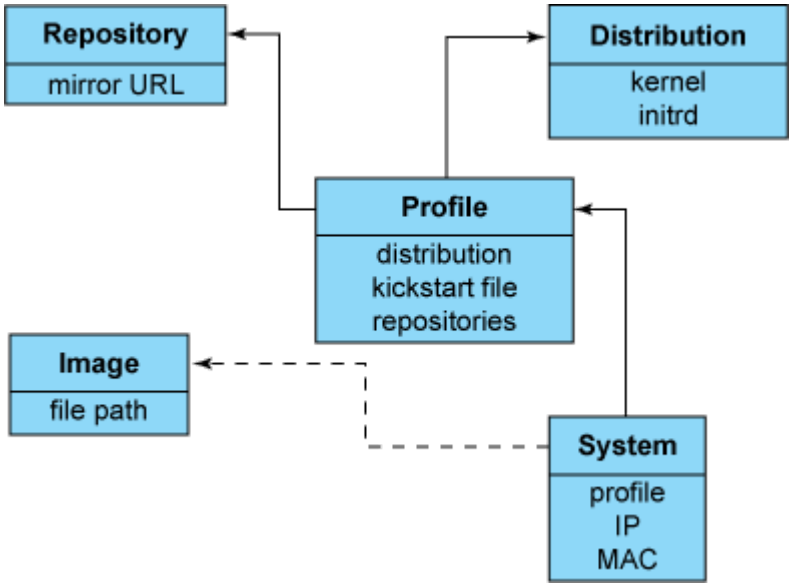
系统：表示要配给的机器。它包含一个配置文件或一个镜像，还包含 IP 和 MAC 地址、电源管理（地址、凭据、类型）以及更为专业的数据等信息。

存储库：保存一个 yum 或 rsync 存储库的镜像信息。

镜像：可替换一个包含不属于此类别的文件的发行版对象（例如，无法分为内核和 initrd 的对象）。

基于注册的对象以及各个对象之间的关联，Cobbler 知道如何更改文件系统以反映具体配置。因为系统配置的内部是抽象的，所以您可以仅关注想要执行的操作。

图 1. Cobbler 对象关系



使用 **Cobbler**

配置和使用 Cobbler 的方式包括命令行、API、XML-RPC 和 Web UI。我重点介绍命令行选项。

Cobbler 拥有 Fedora、RHEL、CentOS、Ubuntu 和 OpenSuse 发行版的程序包（参阅 EPEL 程序包）。我以一个 Fedora 系统上的指令为基础 — 如果愿意，可针对另一个系统轻松地调整这些指令。

首先，安装该工具和 fence 代理包，Cobbler 使用这个程序包执行电源管理活动。以 root 用户的身份，执行发行版的包管理器的 `install` 命令：

```
yum -y install cobbler fence-agents
```

安装好这些程序包后，启动必要的服务：

```
service cobblerd start
service httpd start
```

EPEL 程序包

RHEL 和 CentOS 的 Cobbler 程序包通过 EPEL (Extra Packages for Enterprise Linux) 分发，这是 Fedora 社区的一个特殊兴趣小组。EPEL 维护着一组针对 Enterprise Linux 的附加程序包，包括 Red Hat Enterprise Linux (RHEL)、CentOS 和 Scientific Linux (SL) 发行版。如需了解更多信息，请访问 Fedora Project。

要测试 Cobbler 是否在正常运行，可键入 `cobbler check`。这个命令显示可能需要调整的配置点。不要担心它们；该命令只是一个为用户提供帮助的指南。后续步骤将配置相关的各个方面（惟一的例外是 SELinux 警告，可按照来自 Cobbler 的指令调整它们）。如果您碰巧获得一条连接错误消息，则需要验证各种服务 — 以及 SELinux 日志（如果已启用）— 是否已正常启动。您可能需要将 SELinux 布尔值 `httpd_can_network_connect_cobbler` 设置为 true (`setsebool -P httpd_can_network_connect_cobbler 1`)。如果需要这么做，则请记住在完成后重新启动各项服务。

配置 **Cobbler**

主要的 Cobbler 配置文件是 /etc/cobbler/settings。使用文本编辑器打开这个文件，并设置以下选项：

manage_dhcp：1

manage_dns：1

manage_tftpd：1

restart_dhcp：1

restart_dns：1

pxe_just_once：1

next_server：<服务器的 IP 地址>

server：<服务器的 IP 地址>

选项 `manage_*` 和 `restart_*` 无需加以说明。选项 `next_server` 用在 DHCP 配置文件中，向机器告知提供引导文件的服务器地址。选项 `server` 在机器安装期间用于引用 Cobbler 服务器地址。最后，选项 `pxe_just_once` 预防将机器中的安装循环配置为始终从网络引导。激活此选项时，机器告诉 Cobbler 安装已完成。Cobbler 将系统对象的 netboot 标志更改为 false，这会强制机器从本地磁盘引导。本示例稍后

面将使用此选项。

您可能未激活所有选项（参阅 仅配置部分服务）。在本例中，配置 Cobbler 来管理所有服务，因为这是一种常见场景，并可展示如何进行配置。

现在，Cobbler 已知道要管理哪些服务，请告诉它要使用哪些程序。使用的选项为：

DHCP：ISC dhcpd 或 dnsmasq

DNS：BIND 或 dnsmasq

TFTP：in.tftpd 或 cobbler 的内部 TFTP

为 DHCP 和 DNS 使用 dnsmasq 是一个不错的主意，因为 dnsmasq 的配置过程很容易。可使用 in.tftpd，因为这是系统的默认选择。使用清单 1 中的设置编辑文件 /etc/cobbler/modules.conf：

仅配置部分服务

可以将 Cobbler 配置为仅管理某些服务来适应您的需要。例如，您可能管理一台文件服务器，但网络中您无法访问的另一台机器提供 DHCP。在这种情况下，DHCP 管理员设置 filename 选项来询问引导文件（x86 系统为 pxelinux.0，PowerPC 为 yaboot），设置 `next_server` 选项来指向您的文件服务器的 IP 地址。在网络中引导的机器会从您的服务器请求引导文件。然后，配置 Cobbler 来管理 TFTP 服务并在该工具中注册机器，以便它为这些机器提供合适的文件。

清单 1. 配置设置

```
[dns]
module = manage_dnsmasq

[dhcp]
module = manage_dnsmasq

[tftpd]
module = manage_in_tftpd
```

Cobbler 使用一个模板来创建服务的配置文件。需要编辑 /etc/cobbler/dnsmasq.template 上的 dnsmasq 模板来修改网络信息，比如要使用的网关地址和 IP 范围。假设运行 Cobbler 的服务器也是网关，而且我们的 IP 范围为 192.168.122.5-192.168.122.254，那么在文件中输入以下这行内容：

```
dhcp-range=192.168.122.5,192.168.122.254,255.255.255.0
```

通常，您希望阻止未注册的客户端从服务器引导。为此，添加参数 dhcp-ignore=tag:!known。（在以前的版本中，语法可能有所不同：dhcp-ignore=#known。如果有疑问，您可以同时插入两个版本。）文件内容类似于清单 2 中的代码：

清单 2. dnsmasq 模板文件

```
# Cobbler generated configuration file for dnsmasq
# $date
#

# resolve.conf .. ?
#no-poll
#enable-dbus
read-ethers
addn-hosts = /var/lib/cobbler/cobbler_hosts

dhcp-range=192.168.122.5,192.168.122.254,255.255.255.0
dhcp-ignore=tag:!known
dhcp-option=3,$next_server
dhcp-lease-max=1000
dhcp-authoritative
dhcp-boot=pxelinux.0
dhcp-boot=net:normalarch,pxelinux.0
dhcp-boot=net:ia64,$elilo

$insert_cobbler_system_definitions
```

Cobbler 基本上已可以使用了。重新启动服务，并将更改同步到文件系统以使它们生效。还要记住重新启动 xinetd 服务以提供 TFTP。运行以下命令：

```
service cobblerd restart
cobbler sync
service xinetd restart
```

您可添加发行版和存储库，创建配置文件，以及注册系统。请记得要验证您的防火墙配置是否允许网络服务 TFTP、DHCP 和 HTTP/HTTPS 使用端口上的流量。

安装 **Fedora 17** 系统

准备 Cobbler 以安装 Fedora 17 系统，有两个选项可用：Xfce 或 GNOME 桌面。要添加 Fedora 安装树，首先需要下载 ISO 媒介，然后运行以下命令来挂载媒介并解压缩其内容。（在一些系统中，Cobbler 可能

无法看到挂载的目录内容，而且无法导入媒介。如果遇到这种问题，可在执行挂载命令后重新启动 Cobbler 服务。）：

```
mount -o loop /Fedora-17-x86_64-DVD.iso /mnt/iso
cobbler import --arch=x86_64 --path=/mnt/iso --name=Fedora17
```

Cobbler 现在将媒介内容复制到文件系统。请耐心等待：该操作可能需要一段时间才能完成。命令 `cobbler import` 很方便，它会自动为您创建一个发行版和一个配置文件对象。您也可以将 Cobbler 直接指向一个网络存储库。结果类似于清单 3：

清单 3. `cobbler import` 命令的结果

```
cobbler distro report
Name                           : Fedora17-x86_64
Architecture                   : x86_64
TFTP Boot Files                : {}
Breed                          : redhat
Comment                        :
Fetchable Files                : {}
Initrd                         : /var/www/cobbler/ks_mirror/Fedora17-x86_64/images/
                                  pxeboot/initrd.img
Kernel                         : /var/www/cobbler/ks_mirror/Fedora17-x86_64/images/
                                  pxeboot/vmlinuz
Kernel Options                 : {}
Kernel Options (Post Install)  : {}
Kickstart Metadata             : {'tree': 'http://@@http_server@@/cblr/links/
                                  Fedora17-x86_64'}
Management Classes             : []
OS Version                     : generic26
Owners                         : ['admin']
Red Hat Management Key         : <<inherit>>
Red Hat Management Server      : <<inherit>>
Template Files                 : {}

cobbler profile report

Name                           : Fedora17-x86_64
TFTP Boot Files                : {}
Comment                        :
DHCP Tag                       : default
Distribution                   : Fedora17-x86_64
Enable gPXE?                   : 0
Enable PXE Menu?               : 1
Fetchable Files                : {}
Kernel Options                 : {}
Kernel Options (Post Install)  : {}
```

```
Kickstart                      :
Kickstart Metadata             : {}
Management Classes             : []
Management Parameters          : <<inherit>>
Name Servers                   : []
Name Servers Search Path       : []
Owners                         : ['admin']
Parent Profile                 :
Proxy                          :
Red Hat Management Key         : <<inherit>>
Red Hat Management Server      : <<inherit>>
Repos                          : []
Server Override                : <<inherit>>
Template Files                 : {}
Virt Auto Boot                 : 1
Virt Bridge                    : xenbr0
Virt CPUs                      : 1
Virt Disk Driver Type          : raw
Virt File Size(GB)             : 5
Virt Path                      :
Virt RAM (MB)                  : 512
Virt Type                      : qemu
```

这个配置文件是您将为希望安装的每个桌面创建的其他两个配置文件的父文件。

但是，在这么做之前，要考虑到您有一个 yum 存储库，这个存储库中包含更多要在安装中使用的程序包。为此，创建一个存储库对象：

```
cobbler repo add --arch=x86_64 --name=Flash-plugin \
  --mirror=http://linuxdownload.adobe.com/linux/x86_64/
cobbler reposync
cobbler repo report
```

对于 yum 存储库 URL，Cobbler 接受 http://、ftp://、rsync://、文件系统路径和 ssh 位置（通过使用基于私钥的身份验证）。`reposync` 操作很重要，因为它会从远程存储库中复制文件。如果创建了存储库对象但未运行 `reposync`，那么您的存储库将是空的，而且您的安装可能会失败。

要完成存储库激活，可将存储库与一个配置文件相关联。使用以下命令将其与 Fedora 配置文件相关联：

```
cobbler profile edit --name=Fedora17-x86_64 --repos=Flash-plugin
```

创建配置文件

下一步是创建配置文件。对于自动安装，使用 kickstart 模板特性指定一个 kickstart 文件。创建一个基于 /var/lib/cobbler/kickstarts 中的文件的简单 kickstart。然后，在 %packages 一节中，定义变量 *$desktop_pkg_group*，该变量在后面会被替换来确定安装哪些桌面包。清单 4 显示了 kickstart 文件的内容：

清单 4. kickstart 文件的内容

```
# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all --initlabel
# Run the Setup Agent on first boot
firstboot --disable
# Activate X
xconfig --startxonboot
# Use network installation
url --url=$tree
# additional repostories get added here
$yum_repo_stanza
# Reboot after installation
reboot
# System keyboard
keyboard us
# System language
lang en_US
# System timezone
timezone  America/New_York
# Root password
rootpw --iscrypted $default_password_crypted
# Install OS instead of upgrade
install
# Clear the Master Boot Record
zerombr
# Allow anaconda to partition the system as needed
autopart

%packages
@base
@base-x
firefox
flash-plugin
$desktop_pkg_group
%end

%post
# create a default user to log in X
useradd desktop-user
passwd -d desktop-user
```

```
# adds the yum repositories to the installed system
$yum_config_stanza
# cobbler final steps
$SNIPPET('kickstart_done')
%end
```

以 *$* 开头的变量替换为 Cheetah 程序（参阅 参考资料），Cobbler 使用该程序处理其模板。如果熟悉 Cheetah 模板，那么规则都是一样的。如需了解有关内部 Cobbler 变量的更多信息，比如 *$yum_config_stanza*，请查阅 /var/lib/cobbler/kickstarts 中的可用 kickstart。

创建该文件后，将其复制到 /var/lib/cobbler/kickstarts（如果不这么做，Cobbler 可能无法使用它）。Cobbler 知道 *$desktop_pkg_group* 的值，因为您在创建配置文件时已使用 `--ksmeta` 选项对其定义过。使用此选项，可确定在替换 kickstart 模板中的一个变量时要使用的值。清单 5 中的命令创建了 Xfce 和 GNOME 配置文件：

清单 5. 创建 Xfce 和 GNOME 配置文件的命令

```
cobbler profile add --name=Fedora17-xfce \
                    --ksmeta='desktop_pkg_group=@xfce-desktop' \
                    --kickstart=/var/lib/cobbler/kickstarts/example.ks \
                    --parent=Fedora17-x86_64
cobbler profile add --name=Fedora17-gnome \
                    --ksmeta='desktop_pkg_group=@gnome-desktop' \
                    --kickstart=/var/lib/cobbler/kickstarts/example.ks \
                    --parent=Fedora17-x86_64
cobbler profile report
```

`--parent` 参数告诉这些配置文件继承 Fedora 配置文件。这些配置文件使用 Fedora 发行版和额外的 Flash-plugin 存储库。要确保一切设置都是正确的，可在处理之后验证 kickstart 内容。结果类似于清单 6：

清单 6. 验证 kickstart 内容

```
cobbler profile getks --name=Fedora17-xfce

# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all --initlabel
# Run the Setup Agent on first boot
```

```
firstboot --disable
# Activate X
xconfig --startxonboot
# Use network installation
url --url=http://192.168.122.1/cblr/links/Fedora17-x86_64
# additional repostories get added here
repo --name=Flash-plugin --baseurl=http://192.168.122.1/cobbler/repo_mirror/Flash-plugin
repo --name=source-1 --baseurl=http://192.168.122.1/cobbler/ks_mirror/Fedora17-x86_64

# Reboot after installation
reboot
# System keyboard
keyboard us
# System language
lang en_US
# System timezone
timezone  America/New_York
# Root password
rootpw --iscrypted $1$mF86/UHC$WvcIcX2t6crBz2onWxyac.
# Install OS instead of upgrade
install
# Clear the Master Boot Record
zerombr
# Allow anaconda to partition the system as needed
autopart

%packages
@base
@base-x
firefox
flash-plugin
@xfce-desktop
%end

%post
# create a user we can use to log on X
useradd desktop-user
passwd -d desktop-user

# adds the yum repositories to the installed system
wget "http://192.168.122.1/cblr/svc/op/yum/profile/Fedora17-xfce" \
   --output-document=/etc/yum.repos.d/cobbler-config.repo

# cobbler final steps


wget "http://192.168.122.1/cblr/svc/op/ks/profile/Fedora17-xfce" -O /root/cobbler.ks
wget "http://192.168.122.1/cblr/svc/op/trig/mode/post/profile/Fedora17-xfce" -O /dev/null
%end
```

变量 *$dekstop_pkg_group* 已被正确替换为 *@xfce-desktop*，它告诉 Anaconda 安装程序安装 Xfce 桌面包

分组。

将机器与配置文件相关联

您基本上已准备好开始进行安装。最后一步是将机器（每个桌面一台机器）与您希望向其安装的配置文件相关联。使用的命令如清单 7 所示：

清单 7. 将机器与它们的配置文件相关联

```
cobbler system add --name=desktop-xfce-1 \
                   --profile=Fedora17-xfce \
                   --mac=52:54:00:b8:5e:8f \
                   --ip-address=192.168.122.10
cobbler system add --name=desktop-gnome-1 \
                   --profile=Fedora17-gnome \
                   --mac=52:54:00:88:f3:44 \
                   --ip-address=192.168.122.11
cobbler system report
```

Cobbler 中的电源管理特性可为您打开、关闭和重新引导机器。在有许多机器且必须组织电源管理信息时（比如每台机器的用户和密码，因为 Cobbler 在其数据库中注册了它们），此功能也很有用。假设机器 desktop-xfce-1 位于 Bay 2 的 IBM Bladecenter 中，并且 desktop-gnome-1 是一台由 RSA 委员会管理的机器。您可按照清单 8 中所示设置您的系统：

清单 8. 添加电源管理信息

```
cobbler system edit --name=desktop-xfce-1 \
                    --power-type=bladecenter \
                    --power-id=2 \
                    --power-user=admin_user \
                    --power-pass=admin_password \
                    --power-address=192.168.122.2
cobbler system edit --name=desktop-gnome-1 \
                    --power-type=rsa \
                    --power-user=rsa_user \
                    --power-pass=rsa_password \
                    --power-address=192.168.122.3
```

请记住将所有更改应用到文件系统：

```
cobbler sync
```

最后，您可以安装这些机器了。

开始安装

您已准备好引导机器并安装它们。这些机器必须配置为从网络引导 — 否则，它们可能从硬盘引导而且永远不会开始安装。如果激活了电源管理，那么 Cobbler 就可为您重新引导机器，这样您就可以使用一个简单命令开始安装：

```
cobbler system reboot --name=desktop-xfce-1
```

这个命令让 Cobbler 使用您指定的凭据连接到 Bladecenter，并向刀片服务器 2 发出一个重新引导命令。该刀片服务器通过网络引导重新启动，并从 Cobbler 中接收引导文件。安装过程会自动执行，而且该过程完成后将显示 Fedora 登录屏幕。

Cobbler 更简单：Web 界面

您可能希望轻松地可视化 Cobbler 对象，并为每天的重复任务重用对象值。Cobbler 提供了一个很有用的 Web 界面，您可以通过该界面实现此目的。要使用这个界面，首先需要安装它的程序包：

```
yum -y install cobbler-web
```

安装该程序包后，配置 Cobbler 授权和身份验证系统，以便您可以登录。配置位于文件 /etc/cobbler/modules.conf 中，类似于清单 9 中的代码：

清单 9. 默认的 Cobbler 授权和身份验证系统配置

```
# authentication:
# what users can log into the WebUI and Read-Write XMLRPC?
# choices:
#    authn_denyall    -- no one (default)
#    authn_configfile -- use /etc/cobbler/users.digest (for basic setups)
#    authn_passthru   -- ask Apache to handle it (used for kerberos)
#    authn_ldap       -- authenticate against LDAP
#    authn_spacewalk  -- ask Spacewalk/Satellite (experimental)
#    authn_pam        -- use PAM facilities
#    authn_testing    -- username/password is always testing/testing (debug)
```

```
#    (user supplied)  -- you may write your own module
# WARNING: this is a security setting,  do not choose an option blindly.
# for more information:
# https://github.com/cobbler/cobbler/wiki/Cobbler-web-interface
# https://github.com/cobbler/cobbler/wiki/Security-overview
# https://github.com/cobbler/cobbler/wiki/Kerberos
# https://github.com/cobbler/cobbler/wiki/Ldap

[authentication]

module = authn_denyall

# authorization:
# once a user has been cleared by the WebUI/XMLRPC, what can they do?
# choices:
#    authz_allowall   -- full access for all authneticated users (default)
#    authz_ownership  -- use users.conf, but add object ownership semantics
#    (user supplied)  -- you may write your own module
# WARNING: this is a security setting  do not choose an option blindly.
# If you want to further restrict cobbler with ACLs for various groups,
# pick authz_ownership.  authz_allowall does not support ACLs.  configfile
# does but does not support object ownership which is useful as an additional
# layer of control.

# for more information:
# https://github.com/cobbler/cobbler/wiki/Cobbler-web-interface
# https://github.com/cobbler/cobbler/wiki/Security-overview
# https://github.com/cobbler/cobbler/wiki/Web-authorization

[authorization]

module = authz_allowall
```

清单 9 中的帮助注释表明，可使用 LDAP、PAM 和配置文件等身份验证选项。因为 PAM 非常常见，所以使用它执行身份验证。在授权一节中，定义哪些用户拥有使用该工具的官方许可。将 module 值设置为 `authz_ownership`，以便您可在 users.conf 文件中指定谁能够访问 Web 界面。配置类似于清单 10 中的代码：

清单 10. Cobbler Web 界面的身份验证和授权配置

```
[authentication]

module = authn_pam

[authorization]
module = authz_ownership
```

保存该文件。接下来，您需要一个名为 myuser 的系统用户（如果没有，可使用 useradd myuser && passwd myuser 创建）。然后，打开文件 /etc/cobbler/users.conf 并将 myuser 添加到 admins 组（这个组拥有对象的完整访问权），如清单 11 所示：

清单 11. 将 myuser 添加到授权文件中的 admins 组

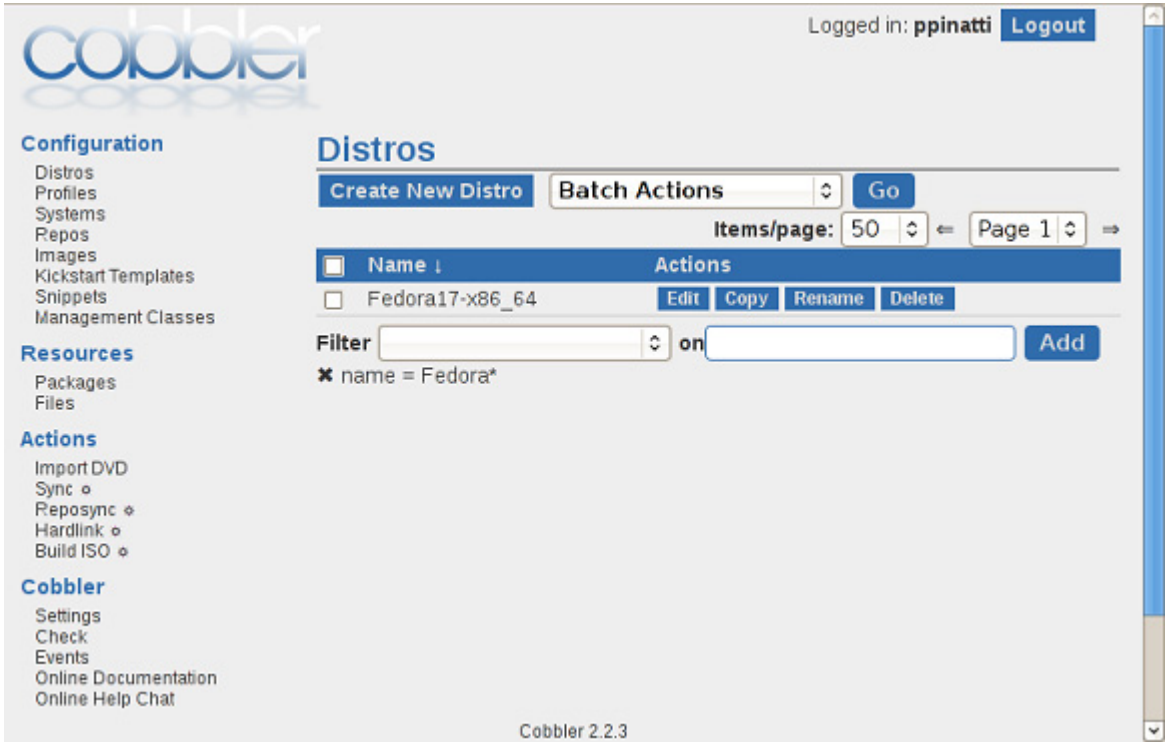
```
# Cobbler WebUI / Web Services authorization config file
#
# NOTICE:
# this file is only used when /etc/cobbler/modules.conf
# specifies an authorization mode of either:
#
#   (A) authz_configfile
#   (B) authz_ownership
#
# For (A), any user in this file, in any group, are allowed
# full access to any object in cobbler configuration.
#
# For (B), users in the "admins" group are allowed full access
# to any object, otherwise users can only edit an object if
# their username/group is listed as an owner of that object. If a
# user is not listed in this file they will have no access.
#
#     cobbler command line example:
#
#     cobbler system edit --name=server1 --owner=dbas,mac,pete,jack
#
# NOTE:  yes, you do need the equal sign after the names.
# don't remove that part.  It's reserved for future use.

[admins]

myuser = ""
```

配置已完成。现在，重新启动 Cobbler 和 Apache 服务以应用更改：

```
service cobblerd restart
service httpd restart
```

Web 界面很简单（参见图 2）：左侧的菜单显示了配置类（比如存储库、系统、发行版和配置文件）、资源（用于配置管理）和操作（导入、同步）。单击一个配置类，就会在屏幕右侧列出所有对象。可通过每一项旁边的按钮（Edit、Copy、Rename、Delete）应用列表过滤器和执行不同操作。

图 2. Cobbler Web 界面



结束语

Cobbler（一个自动化和简化系统安装的工具）使用网络引导来控制和启动安装。其他 Cobbler 特性包括存储库镜像、kickstart 模板和连接电源管理系统。

通过描述该工具的配置对象以及它们如何相互关联，演示了 Cobbler 的内部设计。Cobbler 的结构结构基于关联的对象以及它们之间的继承性，提供了良好的解压缩和重用能力，从而简化了针对系统安装而配置环境的任务。

您了解了如何安装和配置 Cobbler，以及如何使用其命令创建一种适合自动安装机器的配置。最后，您了解了如何安装和配置 Web 界面，可作为执行每日活动的实用替代方案。

Cobbler 可让系统管理员的工作变得更轻松。本文中未探讨的特性（包括 XML-RPC API、配置管理和

koan 客户端）使 Cobbler 变得更加强大，并为进一步探索提供了方向。

参考资料

学习

了解 koan 如何用于 Cobbler 的更多信息。

“Linux 初始 RAM 磁盘 (initrd) 概述”（M. Tim Jones，developerWorks，2006 年 7 月）：了解 initrd 结构剖析、创建和在 Linux 引导进程中的使用。

developerWorks Linux 专区 拥有面向 Linux 开发人员的更多资源，包括 不熟悉 Linux 的程序员。

Twitter 上的 developerWorks：立即加入并关注 developerWorks 推文。

developerWorks 播客：收听面向软件开发人员的有趣访谈和讨论。

developerWorks 演示中心：观看演示，从为初学者准备的产品安装，到为经验丰富的开发人员准备的高级功能。

获得产品和技术

Cobbler：访问这个 Linux 安装服务器，该服务器支持快速设置网络安装环境和自动化许多关联的 Linux 任务。

Cheetah：了解这个主要用于 Web 开发的开源模板引擎和代码生成工具的更多信息。

访问 IBM 试用软件（可下载或通过 DVD 获得），并使用专为开发人员提供的软件改进您的下一个开源开发项目。

下载 IBM 开发人员工具包：更新您的系统，并在这里获取最新的工具和技术。

讨论

developerWorks 社区：查看开发人员推动的博客、论坛、群组和维基，并与其他 developerWorks 用户交流。