Cobbler Documentation

Release 3.0.0

Jörgen Maas

Mar 07, 2017

Contents

Contents:

Release Notes for Cobbler 3.0.0

Enhancements

- Use new dracut ip option for configuring static interfaces (koan).

- Add a whitelist of directories in order to persist a `cobbler sync`.

- Add proxy support for get-loaders, signature update and reposync.

- Add initial support for DJBDNS.

- Enable external YUM repo mirroring through a proxy server.

- DHCP configuration now also supports the per interface gateway setting.

- A new interface_type `BMC` was added which also can be managed with DHCP.

- Yaboot was updated to 1.3.17.

- Add ability to have per-profile/per-system `next_server` values (#1196).

- Add `--graphics` option to Koan.

- Improved input validation and error handling.

- Support `virtio26` for generic QEMU fallback in Koan.

- Debian network config: add support for tagged vlan only bonding interfaces.

- Documentation has been converted into rST and is now included with the source tree.

- Integrated pyflakes into the build system and resolved hundreds of issues.

- Integrated pep8 (coding style) into the build system and resolved thousands of issues.

- Add a new field to the system type `ipv6_prefix` (#203).

- Minor update to CSS; make better use of screen (tables) (cobbler-web).

- Add support for an empty system status.

- If `dns-name` is specified, set it as DHCP hostname in preference to the `hostname` field.

- Allow user to choose whether or not to delete item(s) recursively (cobbler-web).

- Set ksdevice kernel option to MAC address for ppc systems as bootif is not used by yaboot.

- Return to list of snippets/kickstarts when snippet/kickstart is saved (cobbler-web).

- Layout in snippet/kickstart edit form has been improved (cobbler-web).

- Better handling of copy/remove actions for subprofiles (API and cobbler-web).

- Make kickstart selectable from a pulldown list in cobbler-web (#991).

Bugfixes

- Changed Apache configuration directory in Ubuntu 14.04 (#1208).

- build_reporting no longer fails with an empty string in ignorelist (#1248).

- Kickstart repo statement, filter invalid values: `gpgcheck`, `gpgkey` and `enabled` (#323).

- Several improvements to Debian/Ubuntu packaging.

- Some class/method names have been changed to make the code more intuitive for developers.

- Remove `root=` argument in Koan when using grubby and replace-self to avoid booting the current OS.

- Exit with an error if the cobblerd executable can't be found (#1108, #1135).

- Fix cobbler sync bug by xmlrpclib returning NoneType object.

- Dont send the Puppet environment when system status is empty (#560).

- Cobbler-web kept only the most recent interface change (#687).

- Fix broken gitdate, gitstamp values in `/etc/cobbler/version`.

- Prevent disappearing profiles after cobblerd restart (#1030).

- Add missing icons to cobbler_web/content (#679).

- cobbler-ext-nodes was broken with `mgmt_classes` defined at the profile level (#790).

- Properly name the VLAN interface in the manual page.

- Fix wrong address of the Free Software Foundation.

- Remove legacy (EL5/6) cruft from the RPM specfile.

- Koan: use the print function instead of the print statement.

- Minor improvement to LDAP configuration (#217).

- Improvements to the unittest framework.

- Removed several unused functions from utils.

- List of authors is now automagically generated.

Upgrade notes

- Support for LDAP configuration through Koan has been removed.

- Support for redhat_management (Spacewalk/Satelite) has been moved to contrib. Users of this functionality should checkout contrib/redhat-management/README.

- Monit support has been removed; you really need to use a CMS to manage your services.

- Support for remote kickstart templates and files been removed (eg. kickstart=http://).

- All object names are now validated like that of the system object.

- The use of `parent` and `distro` on subprofiles are now mutually exclusive.

- Support for s390/s390x has been removed.

- Support for ia64 (Itanium) has been removed.

- Support for the MySQL backend has been removed.

- Support for deprecated fieldnames (`subnet`, `bonding_master`, `bonding`) has been removed.

- Cobbler now requires python 2.7 and Koan now requires python 2.6.

- Red Hat specific default kernel options have been removed from the settings file.

- Support for Func integration has been moved to contrib. Users of this functionality should checkout contrib/func/README.

- Deprecated Koan LiveCD: moved to contrib.

Install Guide

Setting up and running cobblerd is not a easy task. Knowledge in apache configuration (setting up ssl, virtual hosts, apache module and wsgi) is needed. Certificates and some server administration knowledge is required too.

Cobbler is available for installation in several different ways, through packaging systems for each distribution or directly from source.

Cobbler has both definite and optional prerequisites, based on the features you'd like to use. This section documents the definite prerequisites for both a basic installation and when building/installing from source.

Prerequisites

Packages

Please note that installing any of the packages here via a package manager (such as yum or apt) can and will require a large number of ancilary packages, which we do not document here. The package definition should automatically pull these packages in and install them along with Cobbler, however it is always best to verify these requirements have been met prior to installing cobbler or any of its components.

First and foremost, Cobbler requires Python. Any version over 2.7 should work. Cobbler also requires the installation of the following packages:

- createrepo
- httpd / apache2
- mkisofs / genisoimage
- mod_wsgi / libapache2-mod-wsgi
- mod_ssl / libapache2-mod-ssl
- python-cheetah
- python-netaddr
- python-simplejson

- python-urlgrabber
- PyYAML / python-yaml
- rsync
- syslinux
- tftp-server / atftpd
- yum-utils

Cobbler-web only has one other requirement besides Cobbler itself:

- Django / python-django

Koan can be installed apart from Cobbler, and has only the following requirement (besides python itself of course):

- python-simplejson

Source

Installation from source requires the following additional software:

- git
- make
- python-devel
- python-cheetah
- openssl

Installation

Cobbler is available for installation for many Linux variants through their native packaging systems. However, the Cobbler project also provides packages for all supported distributions which is the preferred method of installation.

Supported packages

Fedora

First, get the Cobbler repo file:

```
# cd /etc/yum.repos.d
# curl -o cobbler30.repo http://download.opensuse.org/repositories/home:/libertas-
↪ict:/cobbler30/Fedora_20/home:libertas-ict:cobbler30.repo
# yum update
```

Then install Cobbler:

```
# yum install cobbler cobbler-web
```

RHEL / CentOS

Make sure you have the EPEL repository enabled on your system:

```
# yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.mirror.triple-it.nl
 * epel: mirror.i3d.net
 * extras: centos.mirror.triple-it.nl
 * updates: mirror.amsiohosting.net
repo id                         repo name                                          ␣
↪     status
!base/7/x86_64                  CentOS-7 - Base                                     ␣
↪     8,465
!epel/x86_64                    Extra Packages for Enterprise Linux 7 - x86_64      ␣
↪     5,709
!extras/7/x86_64                CentOS-7 - Extras                                   ␣
↪     30
!updates/7/x86_64               CentOS-7 - Updates                                  ␣
↪     705
repolist: 14,909
```

Then, get the Cobbler repo file:

```
# cd /etc/yum.repos.d
# curl -o cobbler30.repo http://download.opensuse.org/repositories/home:/libertas-
↪ict:/cobbler30/CentOS_CentOS-7/home:libertas-ict:cobbler30.repo
# yum update
```

Then install Cobbler:

```
# yum install cobbler cobbler-web
```

openSUSE

TBD   http://download.opensuse.org/repositories/home:/libertas-ict:/cobbler30/openSUSE_13.1/home:libertas-ict:cobbler30.repo

Debian

TBD

http://download.opensuse.org/repositories/home:/libertas-ict:/cobbler30/Debian_7.0/

Ubuntu

TBD

http://download.opensuse.org/repositories/home:/libertas-ict:/cobbler30/xUbuntu_14.04/

Packages from source

It is also possible to build packages directly from the source tree.

RPM

```
$ make rpms
... (lots of output) ...
Wrote: /path/to/cobbler/rpm-build/cobbler-3.0.0-1.fc20.src.rpm
Wrote: /path/to/cobbler/rpm-build/cobbler-3.0.0-1.fc20.noarch.rpm
Wrote: /path/to/cobbler/rpm-build/koan-3.0.0-1.fc20.noarch.rpm
Wrote: /path/to/cobbler/rpm-build/cobbler-web-3.0.0-1.fc20.noarch.rpm
```

As you can see, an RPM is output for each component of Cobbler, as well as a source RPM. This command was run on a system running Fedora 20, hence the fc20 in the RPM name - this will be different based on the distribution you're running.

DEB

To install cobbler from source on Debian Squeeze, the following steps need to be made:

```
$ apt-get install make
$ apt-get install git
$ apt-get install python-yaml
$ apt-get install python-cheetah
$ apt-get install python-netaddr
$ apt-get install python-simplejson
$ apt-get install python-urlgrabber
$ apt-get install libapache2-mod-wsgi
$ apt-get install python-django
$ apt-get install atftpd

$ a2enmod proxy
$ a2enmod proxy_http
$ a2enmod rewrite

$ ln -s /usr/local/lib/python2.6/dist-packages/cobbler /usr/lib/python2.6/dist-
↪packages/
$ ln -s /srv/tftp /var/lib/tftpboot

$ chown www-data /var/lib/cobbler/webui_sessions
```

Change all `/var/www/cobbler` in `/etc/apache2/conf.d/cobbler.conf` to `/usr/share/cobbler/webroot/` Init script: - add Required-Stop line - path needs to be `/usr/local/...` or fix the install location

Source

The latest source code is available through git:

```
$ git clone https://github.com/cobbler/cobbler.git

$ cd cobbler
$ git checkout release30
```

The release30 branch corresponds to the official release version for the 3.0.x series. The master branch is the development series, and always uses an odd number for the minor version (for example, 3.1.0).

When building from source, make sure you have the correct prerequisites. Once they are, you can install Cobbler with the following command:

```
$ make install
```

This command will rewrite all configuration files on your system if you have an existing installation of Cobbler (whether it was installed via packages or from an older source tree). To preserve your existing configuration files, snippets and automatic installation files, run this command:

```
$ make devinstall
```

To install the Cobbler web GUI, use this command:

```
$ make webtest
```

This will do a full install, not just the web GUI. `make webtest` is a wrapper around `make devinstall`, so your configuration files will also be saved when running this command.

Relocating your installation

Often folks don't have a very large `/var` partition, which is what Cobbler uses by default for mirroring install trees and the like.

You'll notice you can reconfigure the webdir location just by going into `/etc/cobbler/settings`, but it's not the best way to do things – especially as the packaging process does include some files and directories in the stock path. This means that, for upgrades and the like, you'll be breaking things somewhat. Rather than attempting to reconfigure Cobbler, your Apache configuration, your file permissions, and your SELinux rules, the recommended course of action is very simple.

1. Copy everything you have already in `/var/www/cobbler` to another location – for instance, `/opt/cobbler_data`

2. Now just create a symlink or bind mount at `/var/www/cobbler` that points to `/opt/cobbler_data`.

Done. You're up and running.

If you decided to access Cobbler's data store over NFS (not recommended) you really want to mount NFS on `/var/www/cobbler` with SELinux context passed in as a parameter to mount versus the symlink. You may also have to deal with problems related to rootsquash. However if you are making a mirror of a Cobbler server for a multi-site setup, mounting read only is ok there.

Also Note: `/var/lib/cobbler` can not live on NFS, as this interferes with locking ("flock") Cobbler does around it's storage files.

Quickstart

Cobbler can be a somewhat complex system to get started with, due to the wide variety of technologies it is designed to manage, but it does support a great deal of functionality immediately after installation with little to no customization needed. Before getting started with Cobbler, you should have a good working knowledge of PXE as well as the automated installation methodology of your choosen distribution(s).

We will assume you have successfully installed Cobbler, please refer to the *Installation Guide* for instructions for your specific operating system. Finally, this guide will focus only on the CLI appllication.

Preparing your OS

SELinux

Before getting started with Cobbler, it may be convenient to either disable SELinux or set it to "permissive" mode, especially if you are unfamiliar with SELinux troubleshooting or modifying SELinux policy. Cobbler constantly evolves to assist in managing new system technologies, and the policy that ships with your OS can sometimes lag behind the feature-set we provide, resulting in AVC denials that break Cobbler's functionality.

Firewall

TBD

Changing settings

Before starting the cobblerd service, there are a few things you should modify.

Settings are stored in */etc/cobbler/settings*. This file is a YAML formatted data file, so be sure to take care when editing this file as an incorrectly formatted file will prevent cobblerd from running.

Default encrypted password

This setting controls the root password that is set for new systems during the handsoff installation.

```
default_password_crypted: "$1$bfI7WLZz$PxXetL97LkScqJFxnW7KS1"
```

You should modify this by running the following command and inserting the output into the above string (be sure to save the quote marks):

```
$ openssl passwd -1
```

Server and next_server

The *server* option sets the IP that will be used for the address of the cobbler server. **DO NOT** use 0.0.0.0, as it is not the listening address. This should be set to the IP you want hosts that are being built to contact the Cobbler server on for such protocols as HTTP and TFTP.

```
server: 127.0.0.1
```

The *next_server* option is used for DHCP/PXE as the IP of the TFTP server from which network boot files are downloaded. Usually, this will be the same IP as the server setting.

```
next_server: 127.0.0.1
```

DHCP management and DHCP server template

In order to PXE boot, you need a DHCP server to hand out addresses and direct the booting system to the TFTP server where it can download the network boot files. Cobbler can manage this for you, via the manage_dhcp setting:

```
manage_dhcp: 0
```

Change that setting to 1 so Cobbler will generate the dhcpd.conf file based on the dhcp.template that is included with Cobbler. This template will most likely need to be modified as well, based on your network settings:

```
$ vi /etc/cobbler/dhcp.template
```

For most uses, you'll only need to modify this block:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers              192.168.1.1;
    option domain-name-servers 192.168.1.210,192.168.1.211;
    option subnet-mask          255.255.255.0;
    filename                    "/pxelinux.0";
    default-lease-time          21600;
    max-lease-time              43200;
    next-server                 $next_server;
}
```

No matter what, make sure you do not modify the "next-server $next_server;" line, as that is how the next_server setting is pulled into the configuration. This file is a cheetah template, so be sure not to modify anything starting after this line:

```
#for dhcp_tag in $dhcp_tags.keys():
```

Completely going through the dhcpd.conf configuration syntax is beyond the scope of this document, but for more information see the man page for more details:

```
$ man dhcpd.conf
```

Notes on files and directories

Cobbler makes heavy use of the */var* directory. The */var/www/cobbler/ks_mirror* directory is where all of the distrubtion and repository files are copied, so you will need 5-10GB of free space per distribution you wish to import.

If you have installed Cobbler onto a system that has very little free space in the partition containing */var*, please read the "Relocating Your Installation" section of the Installation Guide to learn how you can relocate your installation properly.

Starting and enabling the Cobbler service

Once you have updated your settings, you're ready to start the service:

```
$ systemctl start cobblerd.service
$ systemctl enable cobblerd.service
$ systemctl status cobblerd.service
```

If everything has gone well, you should see output from the status command like this:

```
cobblerd.service - Cobbler Helper Daemon
    Loaded: loaded (/lib/systemd/system/cobblerd.service; enabled)
     Active: active (running) since Sun, 17 Jun 2012 13:01:28 -0500; 1min 44s ago
   Main PID: 1234 (cobblerd)
     CGroup: name=systemd:/system/cobblerd.service
             1234 /usr/bin/python /usr/bin/cobblerd -F
```

Checking for problems and your first sync

Now that the cobblerd service is up and running, it's time to check for problems. Cobbler's check command will make some suggestions, but it is important to remember that these are mainly only suggestions and probably aren't critical for basic functionality. If you are running iptables or SELinux, it is important to review any messages concering those that check may report.

```
$ cobbler check
The following are potential configuration items that you may want to fix:

1. ....
2. ....
```

Restart cobblerd and then run 'cobbler sync' to apply changes.

If you decide to follow any of the suggestions, such as installing extra packages, making configuration changes, etc., be sure to restart the cobblerd service as it suggests so the changes are applied.

Once you are done reviewing the output of "cobbler check", it is time to synchronize things for the first time. This is not critical, but a failure to properly sync at this point can reveal a configuration problem.

```
$ cobbler sync
task started: 2012-06-24_224243_sync
task started (id=Sync, time=Sun Jun 24 22:42:43 2012)
running pre-sync triggers
...
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
rendering TFTPD files
generating /etc/xinetd.d/tftp
cleaning link caches
running: find /var/lib/tftpboot/images/.link_cache -maxdepth 1 -type f -links 1 -exec␣
→rm -f '{}' ';'
received on stdout:
received on stderr:
running post-sync triggers
running python triggers from /var/lib/cobbler/triggers/sync/post/*
running python trigger cobbler.modules.sync_post_restart_services
running: dhcpd -t -q
received on stdout:
received on stderr:
running: service dhcpd restart
received on stdout:
received on stderr:
running shell triggers from /var/lib/cobbler/triggers/sync/post/*
running python triggers from /var/lib/cobbler/triggers/change/*
running python trigger cobbler.modules.scm_track
running shell triggers from /var/lib/cobbler/triggers/change/*
*** TASK COMPLETE ***
```

Assuming all went well and no errors were reported, you are ready to move on to the next step.

Importing your first distribution

Cobbler automates adding distributions and profiles via the "cobbler import" command. This command can (usually) automatically detect the type and version of the distribution your importing and create (one or more) profiles with the correct settings for you.

Download an ISO image

In order to import a distribution, you will need a DVD ISO for your distribution. **NOTE:** You must use a full DVD, and not a "Live CD" ISO. For this example, we'll be using the Fedora 17 x86_64 ISO.

Once this file is downloaded, mount it somewhere:

```
$ mount -t iso9660 -o loop,ro /path/to/isos/Fedora-17-x86_64-DVD.iso /mnt
```

Run the import

You are now ready to import the distribution. The name and path arguments are the only required options for import:

```
$ cobbler import --name=fedora17 --arch=x86_64 --path=/mnt
```

The –arch option need not be specified, as it will normally be auto-detected. We're doing so in this example in order to prevent multiple architectures from being found.

Listing objects

If no errors were reported during the import, you can view details about the distros and profiles that were created during the import.

```
$ cobbler distro list
$ cobbler profile list
```

The import command will typically create at least one distro/profile pair, which will have the same name as shown above. In some cases (for instance when a xen-based kernel is found), more than one distro/profile pair will be created.

Object details

The report command shows the details of objects in cobbler:

```
$ cobbler distro report --name=fedora17-x86_64
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) : {}
Automatic Installation Template Metadata : {'tree': 'http://@@http_server@@/cblr/
↪links/fedora17-x86_64'}
Management Classes            : []
OS Version                    : fedora17
Owners                        : ['admin']
Red Hat Management Key        : <<inherit>>
Red Hat Management Server     : <<inherit>>
Template Files                : {}
```

As you can see above, the import command filled out quite a few fields automatically, such as the breed, OS version, and initrd/kernel file locations. The "Automatic Installation Template Metadata" field (–autoinstall_meta internally) is used for miscellaneous variables, and contains the critical "tree" variable. This is used in the automated installation templates to specify the URL where the installation files can be found.

Something else to note: some fields are set to <<inherit>>. This means they will use either the default setting (found in the settings file), or (in the case of profiles, sub-profiles, and systems) will use whatever is set in the parent object.

Creating a system

Now that you have a distro and profile, you can create a system. Profiles can be used to PXE boot, but most of the features in cobbler revolve around system objects. The more information you give about a system, the more cobbler will do automatically for you.

First, we'll create a system object based on the profile that was created during the import. When creating a system, the name and profile are the only two required fields:

```
$ cobbler system add --name=test --profile=fedora17-x86_64
$ cobbler system list
test
$ cobbler system report --name=test
Name                          : test
TFTP Boot Files               : {}
Comment                       :
Enable gPXE?                  : 0
Fetchable Files               : {}
Gateway                       :
Hostname                      :
Image                         :
IPv6 Autoconfiguration        : False
IPv6 Default Device           :
Kernel Options                : {}
Kernel Options (Post Install) : {}
Automatic Installation Template: <<inherit>>
Automatic Installation Template Metadata: {}
Management Classes            : []
Management Parameters         : <<inherit>>
Name Servers                  : []
Name Servers Search Path      : []
Netboot Enabled               : True
Owners                        : ['admin']
Power Management Address       :
Power Management ID            :
Power Management Password      :
Power Management Type          : ipmitool
Power Management Username      :
Profile                       : fedora17-x86_64
Proxy                         : <<inherit>>
Red Hat Management Key         : <<inherit>>
Red Hat Management Server      : <<inherit>>
Repos Enabled                 : False
Server Override               : <<inherit>>
Status                        : production
Template Files                : {}
Virt Auto Boot                : <<inherit>>
Virt CPUs                     : <<inherit>>
Virt Disk Driver Type         : <<inherit>>
Virt File Size(GB)            : <<inherit>>
Virt Path                     : <<inherit>>
Virt RAM (MB)                 : <<inherit>>
Virt Type                     : <<inherit>>
```

The primary reason for creating a system object is network configuration. When using profiles, you're limited to DHCP interfaces, but with systems you can specify many more network configuration options.

So now we'll setup a single, simple interface in the 192.168.1/24 network:

```
$ cobbler system edit --name=test --interface=eth0 --mac=00:11:22:AA:BB:CC --ip-
→address=192.168.1.100 --netmask=255.255.255.0 --static=1 --dns-name=test.mydomain.
→com
```

The default gateway isn't specified per-NIC, so just add that separately (along with the hostname):

```
$ cobbler system edit --name=test --gateway=192.168.1.1 --hostname=test.mydomain.com
```

The –hostname field corresponds to the local system name and is returned by the "hostname" command. The –dns-name (which can be set per-NIC) should correspond to a DNS A-record tied to the IP of that interface. Neither are required, but it is a good practice to specify both. Some advanced features (like configuration management) rely on the –dns-name field for system record look-ups.

Whenever a system is edited, cobbler executes what is known as a "lite sync", which regenerates critical files like the PXE boot file in the TFTP root directory. One thing it will **NOT** do is execute service management actions, like regenerating the dhcpd.conf and restarting the DHCP service. After adding a system with a static interface it is a good idea to execute a full "cobbler sync" to ensure the dhcpd.conf file is rewritten with the correct static lease and the service is bounced.

CHAPTER 4

User Guide

Developer Guide

Patch process

You'd like to contribute features or fixes to Cobbler? Great! We'd love to have them.

It is highly recommended that you have a github.com account if you would like to contribute code. Create an account, log in, and then go to github.com/cobbler/cobbler to "fork" the project.

Create a new branch named after the feature you are working on. Do the work on your local machine, please make sure your work passes Cobbler's coding standards by using *make qa*. Only then push to your personal Github branch (e.g. github.com/yourname/cobbler).

Then use the "submit pull request" feature of Github to request that the official repo pull in your changes. Be sure to include a full description of what your change does in the comments, including what you have tested (and other things that you may have not been able to test well and need help with).

If the patch needs more work, we'll let you know in the comments.

Do not mix work on different features in different pull requests/branches if at all possible as this makes it difficult to take only some of the work at one time, and to quickly slurp in some changes why others get hammered out.

Once we merge in your pull request, you can remove the branch from your repo if you like. The AUTHORS file is created automatically when we release.

Setup

The preferred development platform is CentOS 7, you will also need the EPEL repository. Get the latest epel-release RPM from `http://download.fedoraproject.org/pub/epel/7/x86_64/repoview/epel-release.html`

Install development dependencies:

```
# yum install python-devel pyflakes python-pep8 python-sphinx rpm-build
```

Install runtime dependencies:

```
# yum install git make python-netaddr python-simplejson PyYAML python-cheetah httpd␣
↪mod_wsgi
```

Initially, to run Cobbler without using packages:

```
# git clone https://github.com/<your username>/cobbler.git
# cd cobbler
# make install
```

For each successive run, do not run make install again. To avoid blowing away your configuration, run:

```
# make webtest
```

This will install Cobbler and restart apache/cobblerd, but move your configuration files and settings aside and restore them, rather than blindly overwriting them.

You can now run Cobbler commands and access the web interface.

Branches

Cobbler has a development branch called "master" (where the action is), and branches for all releases that are in maintaince mode. All work on new features should be done against the master branch. If you want to address bugs then please target the latest release branch, the maintainers will then cherry-pick those changes into the master branch.

```
# git branch -r
# git checkout <branch>
# git checkout -b <new branch name>
```

Standards

We're not overly picky, but please follow the python PEP8 standards we want to adhere to (see Makefile).

- Always use under_scores, not camelCase.

- Always four (4) spaces, not tabs.

- Avoid one line if statements.

- Validate your code by using `make qa`.

- Keep things simple, keep in mind that this is a tool for sysadmins and not python developers.

- Use modules that are easily available (eg. EPEL) but preferrably in the base OS, otherwise they have to be packaged with the app, which usually runs afoul of distribution packaging guidelines.

- At least for now we have to support Python 2.7 for Cobbler and ython 2.6 for Koan.

You're also welcome to hang out in #cobbler and #cobbler-devel on irc.freenode.net, as there are folks around to answer questions, etc.

Contributing to the website

The github-based git repository for the http://cobbler.github.io website itself is at https://github.com/cobbler/cobbler.github.io.

If you want to contribute changes to the website, you will need jekyll (http://jekyllrb.com).

You will probably want to:

- edit the files in _dynamic
- run the generate_dynamic.sh script
- add both the .md and resulting .html files in your git commit

Mailing List

We have a development mailing list at https://fedorahosted.org/mailman/listinfo/cobbler-devel Discuss development related questions, roadmap, and other things there, rather than on the general user list.

It is a very good idea to mention your pull request (copy/paste, etc) to the development mailing list for discussion.

Debugging

If you need to debug a remote process, epdb provides some very nice capabilities beyond the standard python debugger, just insert a "import epdb; epdb.serve()" in your command line, and from the console:

```
# python -c "import epdb; epdb.connect"
```

CHAPTER 6

Indices and tables

- genindex
- modindex
- search