

Cobbler 安装部署

作者: MinUnix

参考网站: <http://www.minunix.com>

一、 Cobbler 介绍

1、关于 Cobbler

Cobbler 是一个快速的构建和部署系统的服务，改变了管理员的装机民工身份。该工具使用 python 开发，小巧轻便（才 15K 行代码），使用简单的命令即可完成 PXE 网络安装环境的配置，同时还可以管理 DHCP、DNS、以及 yum 包镜像，通过配置可以集成越来越多的配置管理系统和远程脚本实用程序，同时可以部署许多不同的操作系统类型。Cobbler 支持命令行管理，Web 界面管理，同时还提供了 API 接口，可以方便二次开发使用。

Cobbler 提供以下服务集成：

PXE 服务支持

DHCP 服务管理

DNS 服务管理

电源管理

Kickstart 服务支持

yum 仓库管理

等。。。

和 Kickstart 不同的是，使用 cobbler 不会因为在局域网中启动了 dhcp 而导致有些机器因为默认从 pxe 启动在重启服务器后加载 tftp 内容导致启动终止。

2、系统版本支持

Red Hat、CentOS 、Debian、Ubuntu、SUSE、FreeBSD、ESXI 等

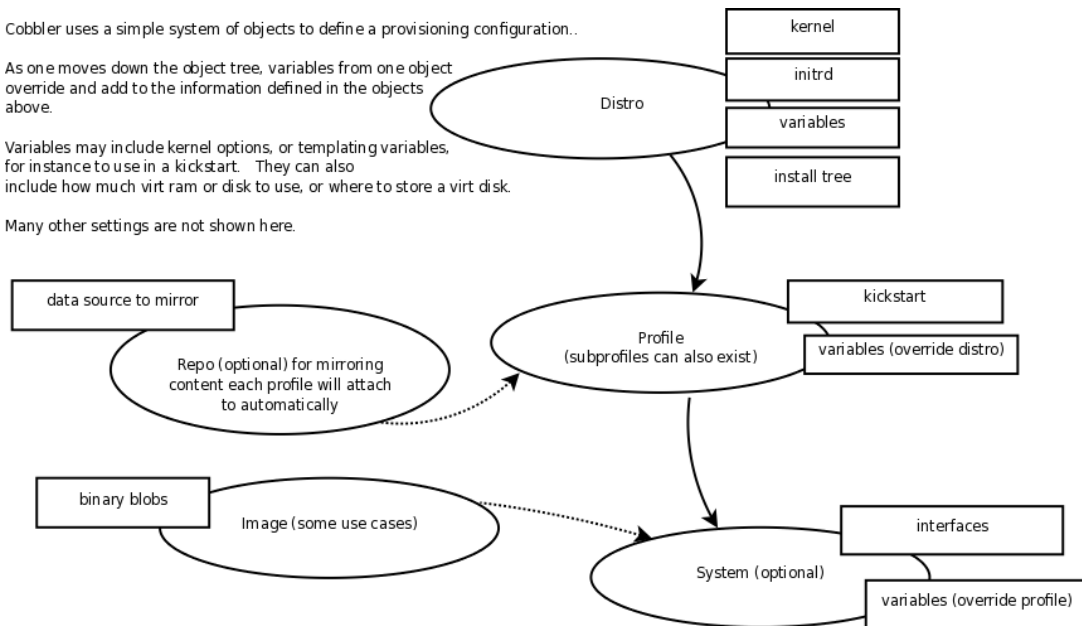
3、Cobbler 模型

Cobbler uses a simple system of objects to define a provisioning configuration..

As one moves down the object tree, variables from one object override and add to the information defined in the objects above.

Variables may include kernel options, or templating variables, for instance to use in a kickstart. They can also include how much virt ram or disk to use, or where to store a virt disk.

Many other settings are not shown here.



摘自：http://www.cobblerd.org/manuals/2.2.3/1/1_-_How_We_Model_Things.html

4、文件系统信息

Cobbler 的文件系统信息如下：

`/var/log/cobbler`

所有的 Cobbler 的日志信息的存放位置。Cobbler 的日志信息并不会转存到 `/var/log/message` 文件里面。

`/var/www/cobbler`

这是 Cobbler 通过 http 服务实现的管理所拥有的不同内容的服务，

/var/www/cobbler/web	mod_python 的 web 界面和配套服务脚本
/var/www/cobbler/webui	webapp 文件的内容
/var/www/cobbler/aux	特定的脚本
/var/www/cobbler/svc	mod_wsgi 脚本
/var/www/cobbler/images	存储所有导入发行版的 Kernel 和 initrd 镜像用于远程网络启动
/var/www/cobbler/ks_mirror	存储导入的发行版系统数据
/var/www/cobbler/repo_mirror	仓库存储目录

/var/lib/cobbler/

/var/lib/cobbler/config	存放 distros、repos、systems 和 profiles 等信息配置文件，一般都是 json 文件
/var/lib/cobbler/backups	备份 config 目录下的 rpm 的升级文件
/var/lib/cobbler/kickstarts	存放 kickstart 配置文件
/var/lib/cobbler/snippets	存放 ks 文件可以导入的脚本小片段，值得研究
/var/lib/cobbler/triggers	存放用户定义的 cobbler 命令

/usr/share/cobbler/web 这是 django 的应用程序

/etc/cobbler

cobbler.conf	cobbler 最重要的配置文件
modules.conf	auxilliary 的配置文件,控制 cobbler 的安全
users.digest	如果使用 digest 认证模块,此文件存放 web 的用户名和密码
/etc/cobbler/power	存放不同的电源模块的模板
/etc/cobbler/pxe	从 pxe 网络引导安装的不同模板
/etc/cobbler/zone_templates	存放每个 DNS 区域文件的模板
/etc/cobbler/reporting	各种报告相关的功能模板

/usr/lib/python\${VERSION}/site-packages/cobbler/
cobbler 的源代码

/usr/lib/python\${VERSION}/site-packages/cobbler/modules	
--	--

This is a directory where modules can be dropped to extend Cobbler without modifying the core. See [Modules](#) for more information.

二、Cobbler 快速指南

1. 系统版本:

```
yum -y install redhat-lsb
#lsb_release -a
LSB Version:
    :base-4.0-amd64:base-4.0-noarch:core-4.0-amd64:core-4.0-noarch:graphics-4.0-amd64
:graphics-4.0-noarch:printing-4.0-amd64:printing-4.0-noarch
Distributor ID: CentOS
Description:  CentOS release 6.2 (Final)
Release: 6.2
Codename:   Final
```

2. 软件版本:

Cobbler 版本: Cobbler 2.2.3

3. 禁用 selinux

```
#setenforce 0
```

4. 安装 Cobbler

```
# yum -y install cobbler
```

5. 修改配置文件

在启动 Cobbler 服务之前,你需要修改一些配置文件. 在修改每一个文件之前最好先备份下当前的文件.Cobblerd 的配置文件为/etc/cobbler/settings ,这个文件是 YAML 信息的格式文件.

1). 默认的加密密码: 设置一个控制 kickstart 的 root 密码

```
default_password_crypted: "$1$mF86/UHC$WvclX2t6crBz2onWxyac."
# openssl passwd -1  \\修改密码 (是数字 1 不是字母 L)
```

2). Server 和 Next_Server

server 选项设置 IP 用于为 cobbler 服务器指定地址,请不要使用 0.0.0.0, 设置一个你希望和 cobbler 服务器通过 http 和 tftp 等协议链接的 IP.

```
# default, localhost
server: 127.0.0.1
next_server 选项是 DHCP/PXE 网络引导文件被下载的 TFTP 服务器的 IP, 它将和 server 设置为同一个 IP
# default, localhost
next_server: 127.0.0.1
```

3). DHCP Management and DHCP Server Template

为了 pxe 的启动,需要一个 DHCP 服务器地址,并直接引导系统,它可以在网络中下载的引导文件到 TFTP 的服务器,cobbler 可以通过 manage_dhcp 的设置来进行管理

```
#default: don't manage
manage_dhcp: 0
```

修改 manage_dhcp 为 1 ,cobbler 将管理 dhcp, 可以通过修改 dhcp.template 来进行管理

```
# vim /etc/cobbler/dhcp.template
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers                192.168.0.1;
    option domain-name-servers 192.168.0.210,192.168.0.211;
    option subnet-mask            255.255.255.0;
    filename                      "/pxelinux.0";
    default-lease-time            21600;
```

```

        max-lease-time          43200;
        next-server              $next_server;
    }

```

4). 开启 **cobbler** 开机自动启动,并启动服务

```
chkconfig cobblerd on &&service cobblerd start
```

5). 在第一次执行同步之前,请先执行检查

```
# cobbler check
# cobbler sync
```

6). **cobbler** 导入

```
cobbler import --name=<name> --arch=<x86_64/i386> --path=<path>
```

其它命令可通过 **cobbler --help** 来进行查看

三、 安装 **Cobbler**

1. 安装 **cobbler/cobblerd** 的先决条件:

cobbler 是基于 **python** ,所以需要版本为 2.3+ 的 **python**,和以下软件包的支持:
createrepo、**httpd**、**mkisofs**、**mod_wsgi**、**python-cheetah**、**python-netaddr**、**python-simplejson**、
python-urlgrabber、**vPyYAML**(**python-yaml**)、**rsync**、**tftp-server**、**yum-utils**

2. 安装 **EPEL** 源

查看系统信息:

```
# uname -r
2.6.32-220.el6.x86_64  \\ 当前系统为 X86_64 的
# lsb_release -r
Release: 6.2  \\ 当前系统为 6.2 的系统
```

从 <http://dl.fedoraproject.org/pub/epel/> 这里找到 6.X 的系统 X86_64 位 **epel-release**

根据此方法查看您的系统所需要下载的 **EPEL** 源

```
# rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

3. **yum** 安装 **Cobbler**

```
# yum -y install cobbler httpd rsync tftp-server xinetd dhcp python-ctypes pykickstart
debmirror
```

4. 配置 **Cobbler**

4.1 配置 **cobbler** 主配置文件

```
# vim /etc/cobbler/settings
```

暂且修改以下几项::

```
manage_dhcp: 0  \\ 默认为 0 ,表示不进行管理 dhcp 服务,可以修改为 1,对其进行管理
manage_rsync: 0  \\ 默认为 0,不对 rsync 进行管理,可以修改为 1 进行管理
server: 192.168.0.7  \\ cobbler 服务器地址
next_server: 192.168.0.7  \\ dhcp 服务地址
```

4.2 配置 **tftp** **rsync**

修改 **rsync** 和 **tftp** 的配置文件:

将 **disable = yes** 修改为 **disable = no**

```
cat /etc/xinetd.d/tftp
```

```
service tftp
```

```
{
```

```

        disable                = no
    ..... (以下内容省略)
cat /etc/xinetd.d/rsync
service rsync
{
    disable = no
    ..... (以下内容省略)
4.3 启用 wsgi 模块
    取消 /etc/httpd/conf.d/wsgi.conf 文件中
LoadModule wsgi_module modules/mod_wsgi.so 行的注释
4.4 生成 Cobbler 安装系统 root 初始密码
# openssl passwd -1 -salt 'root' 'minunix'
$1$root$oZ2U5GhxaJ1b4q4tpPQnE0
将密码加入 cobbler 配置
vim /etc/cobbler/settings
    default_password_crypted: "$1$root$oZ2U5GhxaJ1b4q4tpPQnE0"
4.5 配置 DHCP 及 modules 模块
cp /etc/cobbler/dhcp.template{,.bak}
vim /etc/cobbler/dhcp.template
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers                192.168.0.1;
    option domain-name            "minunix.com";
    option domain-name-servers 192.168.0.1;
    option subnet-mask            255.255.255.0;
    range dynamic-bootp          192.168.0.100 192.168.0.254;
    filename                      "/pxelinux.0";
    default-lease-time            600;
    max-lease-time                7200;
    next-server                   $next_server;
}

```

4.6 配置开机启动

```

chkconfig httpd on && service httpd start
chkconfig dhcpd on && service dhcpd start
chkconfig xinetd on && service xinetd start
chkconfig cobblerd on && service cobblerd start

```

4.7 下载引导操作系统文件

```
cobbler get-loaders
```

4.8 配置 debmirror

```
yum -y install debmirror
```

修改/etc/debmirror.conf 文件, 注释掉:@dists 和 @arches 两行

4.9 检查 Cobbler 配置

```
cobbler check
```

同步配置: cobbler sync

四、配置管理 Cobbler

至于 Cobbler 的配置文件、日志文件等的存放位置在前面已经简述过,这里不再啰嗦了!

1. 导入系统镜像文件

上传 ISO 镜像文件到服务器/var/iso/linux 目录下

```
mkdir -p /var/iso/linux
mkdir -p /var/Linux/CentOS/X86_64/6.2
mkdir -p /var/Linux/CentOS/X86_64/6.0
mount -o loop /var/iso/linux/CentOS-6.2-x86_64-bin-DVD1.iso
/var/Linux/CentOS/X86_64/6.2
```

2. 从镜像中导入 OS 数据

注: 导入时间较长, 请耐心等待!!!

```
cobbler import --path=/var/Linux/CentOS/X86_64/6.2 --name=CentOS-6.2
--arch=x86_64
```

在正常导完之后会给出如下提示:

associating kickstarts

*** TASK COMPLETE ***

有时可能会出现卡住的现象,如果导入时间过长,可通过比对文件大小来确定是否已经正常导入

比对文件大小:

```
du -sh /var/Linux/CentOS/X86_64/6.2
du -sh /var/www/cobbler/ks_mirror/centos-6.2-x86_64
```

如果上述两个命令执行过显示的结果出入较大, 则可能文件没有正常导入

在重新导入之前最好先把之前的内容删除再导入

cobbler [distro] remove --name=[centos6.2] 方括号中的内容根据自己的情况来填写, 更多命令通过 cobbler --help 来查看

剩下其它系统导入方法类似,只是名字和路径更改下即可,导入完成之后,可通过 cobbler list 来查看导入的结果.

五、Web 界面

web 界面的管理是一个非常简便的方法, 只需要安装 cobbler_web 软件包即可

1. yum -y install cobbler_web

2. 设置用户名密码

为 cobbler 添加 minunix 用户,并设置密码

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

3. 开启 cobbler web 登录

```
vim /etc/cobbler/modules.conf
```

修改 authentication 部分中的 module,修改后如下:

```
module = authn_configfile
```

4. 重启 cobbler 与 http

5. 访问: https://192.168.0.7/cobbler_web

注: 因软件包的依赖关系笔者暂没成功完成 web 访问, 但步骤就是这样的!

六、Cobbler 安装操作系统

1. 为了增加装机的安全,可设置装机密码

```
openssl passwd -1 -salt 'root' minunix  
$1$root$oZ2U5GhxaJ1b4q4tpPQnEO
```

编辑 pxedefault.template , 加入密码部分

```
vim /etc/cobbler/pxe/pxedefault.template
```

在 TIMEOUT 上面加入一行

```
MENU MASTER PASSWD $1$root$oZ2U5GhxaJ1b4q4tpPQnEO
```

```
vim /etc/cobbler/pxe/pxepreprofile.template
```

在 LABEL 标签中加入一行 MENU PASSWD

修改完之后记得同步配置, `cobbler check && cobbler sync`

2. 自定义 PXE 菜单 TITLE

修改 pxedefault.template 文件

修改 MENU TITLE 后面的内容为自己想要的内容

eg: MENU TITLE MinUnix Install System| <http://www.minunix.com>

修改完之后记得同步配置, `cobbler check && cobbler sync`

3. 自定义安装系统

可以通过命令 `cobbler system add` 进行自定义安装某个版本的系统,
更多命令可通过 `cobbler system add --help` 获得帮助

七、 更深入研究可通过官方网站进行资料查阅

<http://www.cobblerd.org/manuals/2.2.3/> 官方文档

<https://github.com/cobbler/cobbler/wiki> git wiki

八、 更多内容请参考以下页面:

<http://www.minunix.com>

<http://docs.minunix.com>

Q 群: 279461110