Integrating with Cobbler

Jesus M. Rodriguez
Principal Software Engineer
Red Hat
What is Cobbler?

- A Linux installation server for rapid setup of network install environments
- Can manage
  - DHCP
  - DNS
  - yum repos
Objects

- distro
- profile
- system
- repo – package repository for mirroring (optional)
- Image – virt guest image
API

- login
- `get_{object*(s)}` e.g. `get_distro`, `get_profiles`
- `find_{object*}`
- `get_{object*}_handle`
- `remove_{object*}`
- `copy_{object*}`
- `rename_{object*}`
- `new_{object*}`
- `modify_{object*}`
- `save_{object*}`

```python
def copy_image(self, object_id, newname, token=None):
    return self.copy_item("image", object_id, newname, token)

def rename_item(self, what, object_id, newname, token=None):
    """
    Renames an object specified by object_id to a new name.
    """
    self._log("rename_item%s" % what, object_id, object_id, token=token)
    obj = self.__get_object(object_id)
    return self.api.rename_item(what, obj, newname)

def rename_distro(self, object_id, newname, token=None):
    return self.rename_item("distro", object_id, newname, token)

def rename_profile(self, object_id, newname, token=None):
    return self.rename_item("profile", object_id, newname, token)

def rename_system(self, object_id, newname, token=None):
    return self.rename_item("system", object_id, newname, token)
```
Language bindings

- Java via cobbler4j
  - cobbler4j directory of cobbler checkout
- Ruby via rubygem-cobbler
  - contrib/ruby of cobbler checkout
- XML-RPC
cobbler4j

- Each cobbler object has a mirror in cobbler4j
  - Distro, Profile, Repo, Image, SystemRecord
  - Auto-generated from python api
- Operate on the object not the connection
- Requires cobbler 2.0 as it uses xapi_object_edit
- Seeded from Spacewalk code
- Developed because XML-RPC from Java is a pain (as are many things)
```java
import java.util.*;
import org.fedorahosted.cobbler.*;
import org.fedorahosted.cobbler.autogen.*;

// REQUIRES: cobbler 2.0
public class Cobbler4jExample {
    public static void main(String[] args) {
        // gets connection and login at the same time
        CobblerConnection conn = new CobblerConnection(
            "http://localhost", "testing", "testing");

        Finder finder = Finder.getInstance();

        // see if there are any distros there already
        List<Distro> distros = (List<Distro>) finder.listItems(conn, ObjectType.DISTR)
                            .toSet();
        System.out.println("there are " + distros.size() + " distros");

        // create a new distro, name, kernel and initrd are required
        Distro distro = new Distro(conn);
        distro.setName("cobbler-integration-java");
        distro.setKernel("/tmp/cobbler.fake");
        distro.setInitrd("/tmp/cobbler.fake");

        // in order to persist an object, we must call commit method
        distro.commit();

        // see our new distro
        distros = (List<Distro>) finder.listItems(conn, ObjectType.DISTRO);
        System.out.println("there are now " + distros.size() + " distros");
        for (Distro d: distros) {
            System.out.println(d.toString());
        }

        // cleanup
        distro.remove();
    }
}
```
XML-RPC

- Cobbler can also be controlled by basic XML-RPC calls
- Most languages have an XML-RPC library
  - Python – xmlrpclib
  - Perl – FrontierRPC
  - Java – Redstone XML-RPC & Apache XML-RPC
  - Ruby - xmlrpc
python via XML-RPC example

```python
#!/usr/bin/python
from xmlrpcclient import *

# standard XML-RPC proxy
conn = ServerProxy("http://localhost/cobbler_api")

# authenticate with cobbler (requires authn-testing to work)
token = conn.login("testing", "testing")

# see if there are any distros there already
distros = conn.get_distros()
print("there are %d distros" % len(distros))

# create a new distro, name, kernel and initrd are required
distro = conn.new_distro(token)
conn.modify_distro(distro, "name", "cobbler-integration", token)
conn.modify_distro(distro, "kernel", "/tmp/cobbler.fake", token)
conn.modify_distro(distro, "initrd", "/tmp/cobbler.fake", token)

# in order to persist an object, we must call the save method
conn.save_distro(distro, token)

# see our new distro
distros = conn.get_distros()
print("there are now %d distros" % len(distros))
for distro in distros:
    print(distro)

# cleanup
if conn.remove_distro("cobbler-integration", token):
    print("we removed distro 'cobbler-integration"")

# done
conn.logout(token)
```
Integration Strategies

- Standalone
- Master
- Slave
  - Synchronization required
Standalone

- Simply use Cobbler as a provisioning service
- Cobbler handles everything
  - PXE
  - DHCP
  - DNS
- Some light integration via scripts
Master

- Store all system & provisioning data in Cobbler
- Optionally control
  - DHCP
  - DNS
  - PXE
- Application contains most logic
Slave

- Application is canonical source
- Application has most of the logic
- Sync data to Cobbler
- Interacting with Cobbler requires special attention
Projects integrating with Cobbler

Beaker

oVirt

cobbler

reliam

Symbolic

RAQ
• Developed a Java binding
  • Seeded cobbler4j
• Sync distros & profiles
  • Creation in Spacewalk affects Cobbler immediately
  • System profiles created at provision time
• Spacewalk is canonical source for data
Spacewalk + Cobbler Why?

- Had own provisioning system
  - Cobbler has momentum and community
- Cobbler used as a slave
  - Faster than ripping out existing system
  - Wizard UI easier
Spacewalk Lessons

- Syncing could've been avoided using Master approach
- Maintaining previous functionality was difficult
- Uses Cobbler XML-RPC api
- Cobbler is the canonical source for system records
  - Looking at a system in the UI is actually showing the record from Cobbler.
- Used Cobbler python api
- For reliability switched to XML-RPC api
- Uses own wrapper for provision and power api
- Uses trigger to determine distro family, pushes to Beaker.
● Uses trigger to determine distro family, pushes to Beaker.

● Systems are kept in Beaker
  ● Data pushed to Cobbler at provision/power cycle time.
Roadmap

- cobbler4j (java binding)
- new team coming up to speed on cobbler
- get a 2.0.x update release out ASAP
- update roadmap
Contact

• IRC
  • #cobbler on freenode
  • #cobbler-devel on freenode

• Mailing lists
  • User list -
    https://fedorahosted.org/mailman/listinfo/cobbler-list
  • Developer list -
    https://fedorahosted.org/mailman/listinfo/cobbler-devel

• Website
  • https://fedorahosted.org/cobbler/
Q & A
• “Master Cobbler” by MGChan - http://www.flickr.com/people/mgchan/
• “The Cobbler North Peak And Ben Lomond (Arrochar, Argyll, Scotland) by Maurits Euro Courier - http://www.flickr.com/people/mauritsp
• “Blackberry Cobbler – 11” by haleysuzanne - http://www.flickr.com/people/haleysuzanne/
• “Street cobbles” by freefotouk - http://www.flickr.com/people/freefotouk
• “cobbles 2” by mrcharly - http://www.flickr.com/people/mrcharly
• “Thinking to walk over there” by mark79_xp - http://www.flickr.com/people/mark79_xp
• “beaker” by quidquid - http://www.flickr.com/people/quietquid/
• “The Road to Ribblesdale” by fatboyke - http://www.flickr.com/people/fatboyke/
• “Questions” by Oberazzi - http://www.flickr.com/people/oberazzi/