Deploying FreeBSD systems with Foreman and mfsBSD

Martin Matuška
mm@FreeBSD.org

AsiaBSDcon 2014
16.03.2014

FreeBSD + Foreman

FreeBSD®

About Foreman

What is Foreman?

- life cycle management and orchestration tool
- designed to manage distributed architectures
- performs unattended system installations
- deployment - configuration - (basic) monitoring

Where does Foreman come from?

- founded in 2009 by Ohad Levy and Paul Kelly
- now a Red Hat community project
- key component of commercial RH Satellite 6

Foreman - technical information

- Programming language: Ruby (on rails / passenger)
- Database: PostgreSQL
- Integrated configuration management: Puppet (preliminary Chef support available)
- Two primary components: **Foreman** and **Smart Proxy**

Host OS support:

- Automated installation available in RHEL/CentOS Linux
- Smart Proxy available in FreeBSD ports (net/foreman-proxy)

Foreman Architecture 1/3

Foreman is the central component. It does:

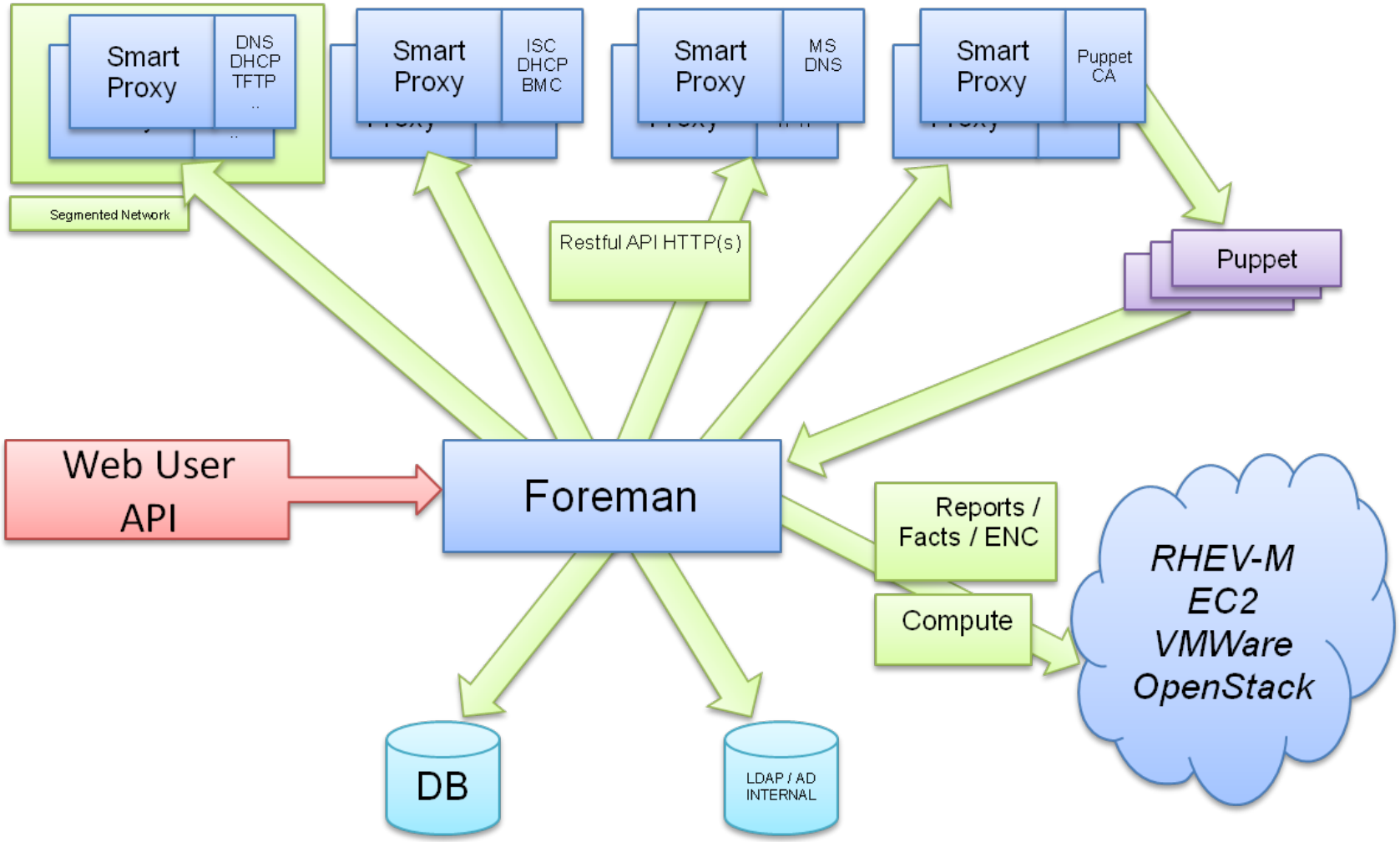
- provide a web interface and a RESTful API
- perform orchestration tasks
- deliver configuration information to deployed hosts
- manage the Smart Proxies
- talk to various API's (e.g. virtualization)
- extendable with various plugins (e.g Foreman hooks)

Foreman Architecture 2/3

Smart proxies are site components. They provide access to:

- DHCP (via ISC or MS DHCP Server)
- DNS (via dynamic updates or MS dnscmd)
- TFTP (pxelinux)
- BMC (via freeipmi or ipmitool)
- Puppet CA (proxy)

Foreman architecture 3/3



Depolyable systems

What systems can be deployed with Foreman?

- Hardware systems
  - Any system with a MAC address bootable via DHCP
  - Foreman does not manage full lifecycle
- Virtual systems
  - Compute resources via Ruby library "fog":
    EC2, GCE, Libvirt, OpenStack, oVirt, Rackspace, vSphere
  - Foreman manages full lifecycle (create/destroy)

Foreman Configuration

Important keywords:

- Infrastructure:
  Domains, Subnets, Compute resources, Compute profiles
- Provisioning:
  Operating Systems, Provisioning Templates, Installation Media, Architectures
- Configuration:
  Host Groups, Environments, Puppet Classes, Smart Variables

Foreman host deployment process

How does the system deployment with Foreman work?

- A host is created by Foreman
- Host boots a boot image via DHCP and TFTP
- Configuration data is downloaded from Foreman via HTTP(s)
- Host performs installation and configuration tasks
- Foreman is told the host is built
- Reports and/or facts are submitted to Foreman

About mfsBSD

What is mfsBSD?

- toolset to create mfsroot-based distributions of FreeBSD
- runs completely from memory
- very useful for boot and rescue images
- FreeBSD support for dedicated servers

mfsBSD and Foreman

mfsBSD needs to be customized for Foreman:

- special startup script (rc.local)
- get foreman_url from a source
- process provisioning data (or script) from Foreman

Deploying FreeBSD with Foreman

- requires Foreman configuration
  (Operating System, Provision Template, Host Group)
- boot via syslinux's memdisk
- special mfsBSD pxeboot image
- name of TFTP image: **FreeBSD-[arch]-[release]-mfs.img**
- `http://mfsbsd.vx.sk/files/foreman/FreeBSD-x86_64-10.0-mfs.img`

Sample PXE template

```
# foreman_url=<%= foreman_url %>
default freebsd
label freebsd
kernel memdisk
append initrd=<%= @initrd %> harddisk raw
```

Sample provisioning script 1/2

```
/root/bin/destroygeom -d ada0 || exit 1
/root/bin/zfsinstall -d ada0 -s 2G -u <%= @mediapath %> || exit 1

/bin/echo '<%= root_pass %>' | pw -V /mnt/etc usermod root -H 0

cat >> /mnt/etc/rc.conf <<EOF
hostname="<%= @host %>"
ifconfig_vmx0="inet <%= @host.ip %> netmask <%= @host.subnet.mask %>"
defaultrouter="<%= @host.subnet.gateway %>"
sshd_enable="YES"
ntpd_enable="YES"
EOF

echo 'PermitRootLogin yes' >> /mnt/etc/ssh/sshd_config

...
```

Sample provisioning script 2/2

```
...

cat >> /mnt/etc/resolv.conf <<EOF
domain <%= @host.domain %>
nameserver <%= @host.subnet.dns_primary %>
<% unless @host.subnet.dns_secondary.empty? -%>
nameserver <%= @host.subnet.dns_secondary %>
<% end -%>
EOF

cp /mnt/usr/share/zoneinfo/Europe/Berlin /etc/localtime
touch /mnt/etc/wall_cmos_clock

fetch -q --no-verify-hostname --no-verify-peer -o /dev/null \
  <%= foreman_url %>
sleep 5
reboot
```

Questions?

Thank you for your attention!