Linux Standard Operating Environments

What is an SOE?

- SOE - Standard Operating Environment
- Greatly reduces time to:
  - deploy new hosts - because the best way to standardise is to automate.
  - fix problems - because everything is built the same way, everything is broken the same way.
  - maintain, update and patch hosts.

What an SOE is not

- A silver bullet - an SOE does not:
  - fix a broken environment (unless you replace it);
  - replace staff (may reduce staff if overstaffed);
  - replace documentation, planning/designing or testing;
  - automate service deployment...
    - though it can be a good starting point.

What an SOE is not

- A means of improving security...
  - though it is a good way to deploy default security.
- Something you do not need until you have "x number of servers".
- A setup where you have every piece of software, required by all possible services, deployed on every server, even if they aren't going to use it.

Why would you want one

- Time saving;
- Improved documentation:
  - One shared document for the SOE; and
  - One for what makes a particular service unique.
- Disaster Recovery;
- Customer/Client confidence; and
- Ability to offload to junior staff.

And why you would not want one...

- Your Server Farm is anarchy and no two systems are alike, they are all critical and no one understands them.
- Job security.

Neither of the above reasons is valid.

You always need and want one.

Components of an SOE

- Base Operating System and approved add-ons;
  - A repository server is highly recommended;
- Defined deployment method or process;
- Centralised Configuration Management Tool;
- Clear vision of what your SOE is / is not;
- Standard Operating Procedures; and
- Documentation.

The Base Operating System

- The OS of the production environment
- This choice prefaces the OS for the development environment.
  - It makes no sense to run RHEL in production and develop on Ubuntu.
  - Use your SOE deployment for production and development.

A Repository Server

- Your first point of authority - if the package is not available here, it does not get installed (at least not on your production systems).
- Needs a sane means of choosing and adding new packages.
  - Don't end up mirroring six different versions of PHP.

Deployment method

- A means of installing the OS on your host that will bring it online to the point that it is:
  - usable;
  - secure; and
  - ready for the next step.
- Should always be the same, e.g.: Kickstart.

Deployment method

- i.e. it will probably include:
  - network configuration;
  - base firewall and other security features; and
  - base configurations (daemons, installed packages, configuration files).

Centralised Configuration Management

- You may have more than one... provided they don't conflict:
  - Kickstart with your custom scripts to do the basic deployment;
  - Puppet to customise and maintain the systems;
  - Specialised tools to manage special servers.

Clear vision

- What your SOE
  - is or is not; and
  - can or can not do.
- You achieve this through:
  - documentation;
  - SOPs; and
  - explaining it to clients and co-workers.

Monitoring

- This should not be a part of your SOE.
- You should already have it in place.
- Installation and configuration should be part of deployment.

Building a Repository Server

Purpose

Local mirror of all:

- official distro packages;
- approved for use add-on repositories; and
- approved for use packages where the overall repository is not suitable.

What it isn't

- A means of not paying for your OS licenses.
- A means for others to not pay for their OS licenses.

- Make sure you firewall it to only allow your authorised hosts in.

Purpose (revisited)

- The repository server:
  - is where the packages you use live;
  - does not need to be highly redundant; but
  - needs to be rebuildable quickly.

Backup considerations

- No need to be fully backed up, consider:
  - OS Vendor provided packages; vs
  - Expansion repositories (e.g.: EPEL) that might age out the software your service runs on.
- Method of mirroring is more important:
  - document; and
  - version control configuration files.

Source considerations

- Red Hat provides every package they release from their repository. Thus you can get packages back.
- EPEL provides (generally) the current version, and the one prior. After the packages have aged out, you will have great difficulty getting them back...
  - /var/cache/yum is not a solution.
  - keep a copy of every package (you might be using).
- Keep all your local software releases.

CentOS 6

- Major difference to RHEL:
  - No licensing fees;
  - No MRepo patching - (needed for RHEL);
  - No support.
- Potential development environment due to software / package compatibility with RHEL.

- See http://www.centos.org/

MRepo

- For RHEL6 mrepo needs to get a bunch of custom patches to work.
- Software from:
  - http://dag.wieers.com/home-made/mrepo/
  - http://packages.sw.be/mrepo/
  - http://mirror.internode.on.net/pub/epel/6/x86_64/repoview/mrepo.html
- Patches from:
  - http://lists.rpmforge.net/pipermail/tools/2010-November/001800.html

MRepo installation

- Hook your host up to EPEL and install mrepo and its dependencies.
  - `wget http://mirror.internode.on.net/pub/epel/6/x86_64/epel-release-6-5.noarch.rpm`
  - `rpm -ivh epel-release-6-5.noarch.rpm`
  - `yum install mrepo -y`
    - installs httpd and createrepo ;
    - lftp was not installed but was needed.
- Configure httpd to start at boot.

MRepo Configuration

- /etc/mrepo.conf
- /etc/mrepo.conf.d/
- /usr/share/doc/mrepo-0.8.7/dists/ contains examples for various distributions
- Configured for CentOS 6 + EPEL...:

Sample MRepo configuration file

```
[CentOS6]
name = CentOS $release ($arch)
release = 6
arch = x86_64
metadata = repomd repoview

### ISO images
iso = CentOS-6.2-x86_64-bin-DVD?.iso

### BASE Release
# not needed, using ISO

### Additional repositories
C6Updates = http://mirror.internode.on.net/pub/centos/6/updates/x86_64/
C6Extras = http://mirror.internode.on.net/pub/centos/6/extras/x86_64/
C6Plus = http://mirror.internode.on.net/pub/centos/6/centosplus/x86_64/

### Custom repository for your own RPM packages
epel-x86_64 = http://mirror.internode.on.net/pub/epel/6/x86_64
```

MRepo - ... continued

- Copy ISO(s) to /var/mrepo/iso/ to save you downloading everything;
- run `mrepo -ugvvv` ;
- edit to enable /etc/cron.d/mrepo ;
- ensure mrepo and httpd are configured to start on boot; and
- that iptables will allow the incoming connections.

iptables

- the RHCE way:

```
[root@c6repo dists]# iptables --list -n | grep 80
[root@c6repo dists]# iptables -A INPUT -m state --state NEW -m tcp -p tcp
--source 192.168.1.0/24 --dport 80 -j ACCEPT
[root@c6repo dists]# iptables --list -n | grep 80
ACCEPT     tcp  --  192.168.1.0/24      0.0.0.0/0           state NEW tcp dpt:80
[root@c6repo dists]#
```

- or just edit /etc/sysconfig/iptables

reposync ... prep

- install reposync (yum-utils);
- get and install the puppetlabs repo release:

```
[root@c6repo ~]# wget http://yum.puppetlabs.com/el/6/products/x86_64/puppetlabs-
release-6-1.noarch.rpm -q
[root@c6repo ~]# rpm -ivh puppetlabs-release-6-1.noarch.rpm warning: puppetlabs-
release-6-1.noarch.rpm: Header V4 RSA/SHA1 Signature, key ID 4bd6ec30: NOKEY
Preparing...                ########################################### [100%]
   1:puppetlabs-release     ########################################### [100%]
[root@c6repo ~]# rpm -ql puppetlabs-release-6-1
/etc/pki/rpm-gpg/RPM-GPG-KEY-puppetlabs
/etc/yum.repos.d/puppetlabs.repo
[root@c6repo ~]#
```

reposync ... configure

- copy (or move) repo file to end in reposync
- trim to suit your needs:

```
[root@c6repo yum.repos.d]# cat /etc/yum.repos.d/puppetlabs.reposync
[puppetlabs-products]
name=Puppet Labs Products 6 - $basearch
baseurl=http://yum.puppetlabs.com/el/6/products/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puppetlabs
enabled=1
gpgcheck=1

[puppetlabs-deps]
name=Puppet Labs Dependencies 6 - $basearch
baseurl=http://yum.puppetlabs.com/el/6/dependencies/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puppetlabs
enabled=1
gpgcheck=1
[root@c6repo yum.repos.d]#
```

reposync ... sync

- create a parent directory to sync to; and
- sync the repositories:

```
[root@c6repo yum.repos.d]# mkdir /var/www/mrepo/puppetlabs
[root@c6repo yum.repos.d]# reposync -c ./puppetlabs.reposync -p /var/www/mrepo/
puppetlabs -a x86_64 -r puppetlabs-products ; reposync -c ./puppetlabs.reposync -
p /var/www/mrepo/puppetlabs -a x86_64 -r puppetlabs-deps
puppetlabs-products                                      | 1.9 kB     00:00
puppetlabs-products/primary_db                           |  30 kB     00:00
[puppetlabs-products: 1     of 58    ] Downloading facter-1.6.2-1.el6.noarch.rpm
facter-1.6.2-1.el6.noarch.rpm                            |  66 kB     00:00
[puppetlabs-products: 2     of 58    ] Downloading facter-1.6.0-1.el6.noarch.rpm
facter-1.6.0-1.el6.noarch.rpm                            |  61 kB     00:00

... snip ...

[puppetlabs-deps: 12    of 12    ] Downloading
tanukiwrapper-3.5.9-1.el6.x86_64.rpm
tanukiwrapper-3.5.9-1.el6.x86_64.rpm                     | 260 kB     00:02
[root@c6repo yum.repos.d]#
```

createrepo

- create your new repositories:

```
[root@c6repo yum.repos.d]# ls /var/www/mrepo/puppetlabs/
puppetlabs-deps  puppetlabs-products
[root@c6repo yum.repos.d]# createrepo /var/www/mrepo/puppetlabs/
puppetlabs-deps/ ; createrepo /var/www/mrepo/puppetlabs/puppetlabs-
products/
12/12 - rubygem-fastthread-1.0.7-1.el6.x86_64.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
58/58 - puppet-2.7.9-2.el6.noarch.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
[root@c6repo yum.repos.d]#
```

- start (and configure to start) httpd and you are ready to go...

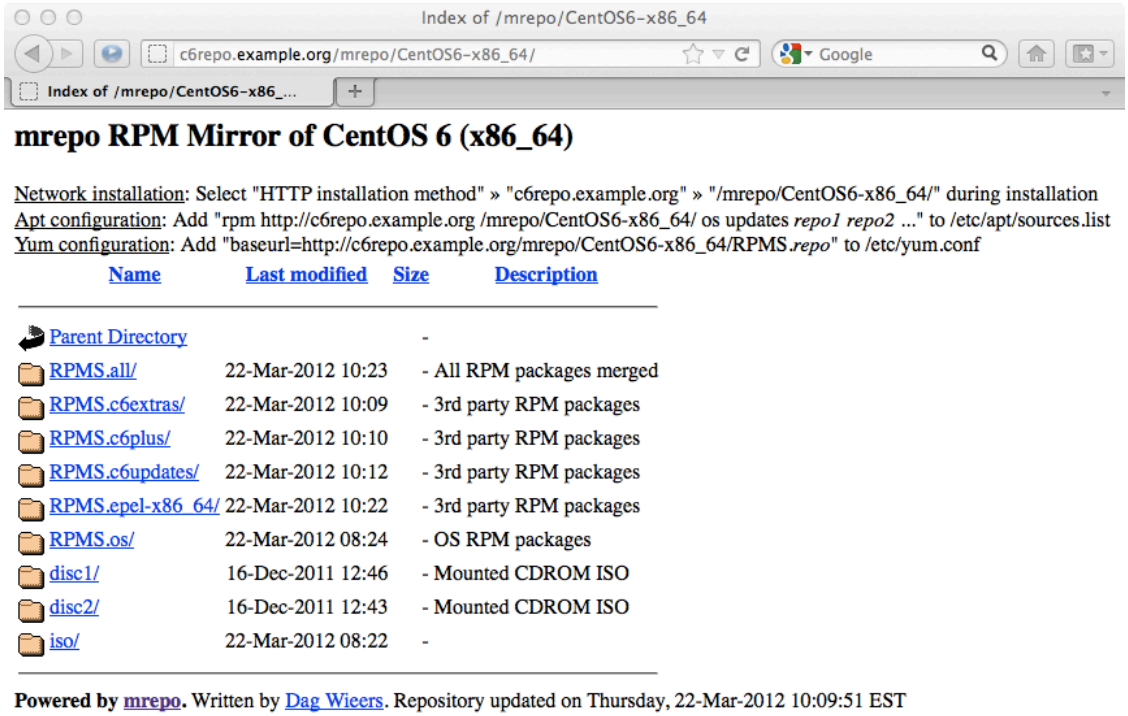
together at last

- to keep this up to date create a cronjob:

```
#0 3 * * * root reposync -n -q -c /etc/yum.repos.d/puppetlabs.reposync -
p /var/www/mrepo/puppetlabs -a x86_64 -r puppetlabs-products &&
createrepo /var/www/mrepo/puppetlabs/puppetlabs-products/ ; reposync -n -q
-c ./puppetlabs.reposync -p /var/www/mrepo/puppetlabs -a x86_64 -r
puppetlabs-deps && createrepo /var/www/mrepo/puppetlabs/puppetlabs-deps/

0 3 * * * root reposync -n -c /etc/yum.repos.d/puppetlabs.reposync -p /
var/www/mrepo/puppetlabs -a x86_64 -r puppetlabs-products && createrepo /
var/www/mrepo/puppetlabs/puppetlabs-products/ ; reposync -n -c ./
puppetlabs.reposync -p /var/www/mrepo/puppetlabs -a x86_64 -r puppetlabs-
deps && createrepo /var/www/mrepo/puppetlabs/puppetlabs-deps/
```

- -n to only download newest
- -q for quiet (hashed out) or verbose (active)
- reposync keeps all files it downloads (-d to age out files)

Vendor and EPEL



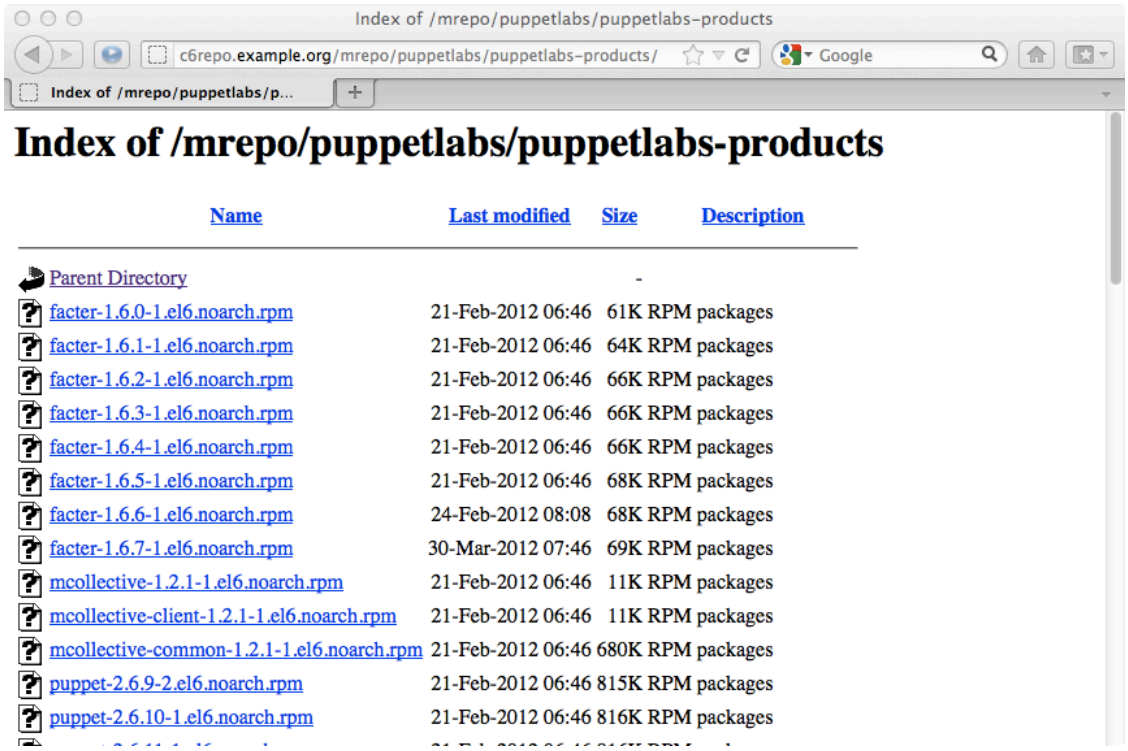
mrepo RPM Mirror of CentOS 6 (x86_64)

Network installation: Select "HTTP installation method" » "c6repo.example.org" » "/mrepo/CentOS6-x86_64/" during installation
Apt configuration: Add "rpm http://c6repo.example.org /mrepo/CentOS6-x86_64/ os updates *repo1 repo2* ..." to /etc/apt/sources.list
Yum configuration: Add "baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.*repo*" to /etc/yum.conf

Name	Last modified	Size	Description
Parent Directory		-	
RPMS.all/	22-Mar-2012 10:23	-	All RPM packages merged
RPMS.c6extras/	22-Mar-2012 10:09	-	3rd party RPM packages
RPMS.c6plus/	22-Mar-2012 10:10	-	3rd party RPM packages
RPMS.c6updates/	22-Mar-2012 10:12	-	3rd party RPM packages
RPMS.epel-x86_64/	22-Mar-2012 10:22	-	3rd party RPM packages
RPMS.os/	22-Mar-2012 08:24	-	OS RPM packages
disc1/	16-Dec-2011 12:46	-	Mounted CDROM ISO
disc2/	16-Dec-2011 12:43	-	Mounted CDROM ISO
iso/	22-Mar-2012 08:22	-	

Powered by mrepo. Written by Dag Wieers. Repository updated on Thursday, 22-Mar-2012 10:09:51 EST

reposync & createrepo



Index of /mrepo/puppetlabs/puppetlabs-products

Name	Last modified	Size	Description
Parent Directory		-	
facter-1.6.0-1.el6.noarch.rpm	21-Feb-2012 06:46	61K	RPM packages
facter-1.6.1-1.el6.noarch.rpm	21-Feb-2012 06:46	64K	RPM packages
facter-1.6.2-1.el6.noarch.rpm	21-Feb-2012 06:46	66K	RPM packages
facter-1.6.3-1.el6.noarch.rpm	21-Feb-2012 06:46	66K	RPM packages
facter-1.6.4-1.el6.noarch.rpm	21-Feb-2012 06:46	66K	RPM packages
facter-1.6.5-1.el6.noarch.rpm	21-Feb-2012 06:46	68K	RPM packages
facter-1.6.6-1.el6.noarch.rpm	24-Feb-2012 08:08	68K	RPM packages
facter-1.6.7-1.el6.noarch.rpm	30-Mar-2012 07:46	69K	RPM packages
mcollective-1.2.1-1.el6.noarch.rpm	21-Feb-2012 06:46	11K	RPM packages
mcollective-client-1.2.1-1.el6.noarch.rpm	21-Feb-2012 06:46	11K	RPM packages
mcollective-common-1.2.1-1.el6.noarch.rpm	21-Feb-2012 06:46	680K	RPM packages
puppet-2.6.9-2.el6.noarch.rpm	21-Feb-2012 06:46	815K	RPM packages
puppet-2.6.10-1.el6.noarch.rpm	21-Feb-2012 06:46	816K	RPM packages

LocalMirror.repo

```
[local_c6_base]
name=CentOS-$releasever - Base
baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.os
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

[local_c6_updates]
name=CentOS-$releasever - Updates
baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.c6updates
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

[local_c6_extras]
name=CentOS-$releasever - Extras
baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.c6extras
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

[local_c6_centosplus]
name=CentOS-$releasever - Plus
baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.c6plus
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

LocalMirror.repo

```
[local_epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
baseurl=http://c6repo.example.org/mrepo/CentOS6-x86_64/RPMS.epel-x86_64
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

[local_Puppetlabs-products]
name=Puppet Labs Products 6 - $basearch
baseurl=http://c6repo.example.org/mrepo/puppetlabs/puppetlabs-products
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puppetlabs
enabled=1
gpgcheck=1

[local_Puppetlabs-deps]
name=Puppet Labs Dependencies 6 - $basearch
baseurl=http://c6repo.example.org/mrepo/puppetlabs/puppetlabs-deps
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puppetlabs
enabled=1
gpgcheck=1
```

SELinux

- By default SELinux is enabled.
- mrepo
  - caches in /var/mrepo ; and
  - servers via /var/www/mrepo
- /var/mrepo/<cache> should be httpd_content_t

```
[root@c... ~]# ls -Zd /var/mrepo/CentOS6-x86_64/
drwxr-xr-x. root root unconfined_u:object_r:var_t:s0   /var/mrepo/CentOS6-x86_64/
[root@c... ~]# semanage fcontext -a -t httpd_sys_content_t /var/mrepo/CentOS6-x86_64\(/.*\)?
[root@c... ~]# restorecon -Rv /var/mrepo/CentOS6-x86_64/
restorecon reset /var/mrepo/CentOS6-x86_64 context unconfined_u:object_r:var_t:s0-
>unconfined_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/mrepo/CentOS6-x86_64/c6extras context unconfined_u:object_r:var_t:s0-
>unconfined_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/mrepo/CentOS6-x86_64/c6extras/drpms context
unconfined_u:object_r:var_t:s0->unconfined_u:object_r:httpd_sys_content_t:s0
...snip...
```

SELinux

Before you disable SELinux, ask yourself:

What if my repo server is compromised?

Final Thoughts

- gpg keys - the LocalMirror.repo file refers to a location on the client file system and the come from *release*.rpm
  - good to get updated keys;
  - bad if its repo files circumvent your local mirror.
    - just clear the repo files; and
    - then make them immutable:

```
[root@c6repo yum.repos.d]# > CentOS-Base.repo ; chattr +i CentOS-Base.repo
```

Final Thoughts ...continued

- redundancy - build more servers and update the baseurl in your local.repo file;
- reposync -c <config> allows specifying configuration not used by yum;

- Make sure you firewall it to only allow your authorised hosts in.

Linux Kickstart

What we are going to do

- ~34MB kickstart ISOs containing:
  - primary NIC configuration;
  - partitioning setup;
  - barebones firewall;
  - root with password “kickstart”;
  - sample post kickstart scripts;

What we are skipping

- a real default firewall;
- any real package customisation;
- default configuration files that are secure (e.g.: sshd_config).

Why kickstart ISOs?

- Issues with PXE;
- Issues with DHCP;
- Issues with kickstart;
- Evolved from a CD ISO requirement;

What you will need

- genisoimage installed;
- an ISO of the OS you are going to kickstart on the host;
- a repository server;
- a vision of:
  - your SOE; and
  - how your newly installed server(s) should be before you customise them for their role.

kickstart file

```
install
#url --url http://192.168.1.5/mrepo/rhel6-server-x86_64/
url --url http://192.168.1.5/mrepo/CentOS6-x86_64/disc1
key --skip
lang en_US.UTF-8
keyboard us

network --device eth0 --bootproto static --ip 192.168.1.9 --gateway 192.168.1.254 --netmask
255.255.255.0 --hostname c6pmaster.example.org --noipv6
# for scripting
#network --device eth0 --bootproto static --ip KS_IP --gateway KS_GATEWAY --netmask
KS_NETMASK --hostname KS_HOSTNAME --noipv6

# password is kickstart
rootpw --iscrypted $1$5YF630$HDlrn.VYFUvtPVwHDmdun0
firewall --enabled --port=22:tcp
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone Australia/Brisbane
```

base configuration

- If you are scripting this:
  - url - will likely be mostly static - use an IP
  - network - sed to replace
- rootpw - make sure you change this once the system is built.

```
[root@sl6repo ~]# grub-md5-crypt
Password:
Retype password:
$1$5YF630$HDlrn.VYFUvtPVwHDmdun0
```

partitioning

- Do NOT make /boot a fancy filesystem;
- If you have more than one drive / RAID set, mention in clearpart or configure post setup.

```
bootloader --location=mbr --driveorder=sda --append=" rhgb crashkernel=auto quiet"
clearpart --all --initlabel --drives=sda

part /boot --fstype ext4 --fsoptions "defaults,strictatime" --size=128 --ondisk=sda
part pv.1 --size=100 --grow --ondisk=sda
volgroup VolGroup00 --pesize=32768 pv.1
logvol / --fstype ext4 --fsoptions "defaults,strictatime" --name=LogVol_root --
vgname=VolGroup00 --size=2048
logvol /usr --fstype ext4 --fsoptions "defaults,strictatime" --name=LogVol_usr --
vgname=VolGroup00 --size=3072
logvol /home --fstype ext4 --fsoptions "defaults,strictatime" --name=LogVol_home --
vgname=VolGroup00 --size=1024
logvol /var --fstype ext4 --fsoptions "defaults,strictatime" --name=LogVol_var --
vgname=VolGroup00 --size=100 --grow
```

packages

- Explicitly install packages either:
  - by group, e.g.:“@Core” ;
  - by name, e.g.:“openldap-servers”
  - exclude by prefacing a “-”, e.g.:“-arts”

```
%packages
@Base
@Core
-NetworkManager
-NetworkManager-glib
-arts
%end
```

%pre

- Runs of the ISO - like the rescue environment;
  - before the installation starts.
- Most useful for workarounds:
  - Copy the custom RPMs you want to install, off the ISO to the initrd’s file system.
  - Genuine work around for a bug on physical hardware... which did not affect VMs.

%post install not chroot'ed

- Runs:
  - after installation is complete; and
  - off the ISO - like the rescue environment.

```
%post --nochroot
mkdir /mnt/sysimage/mnt/dvd
mkdir /mnt/sysimage/mnt/nfs
mkdir /mnt/sysimage/mnt/samba
```

%post install chrooted

- Does NOT run off the ISO, chroot's to newly installed system.
- Thus you can change the new system directly ...

```
%post
## Setup /opt
mkdir /var/root-opt  ; chmod 755 /var/root-opt
mkdir /opt           ; chmod 755 /opt
echo "/var/root-opt          /opt      none       bind" >> /etc/fstab
/bin/mount /opt

## Setup /tmp
mkdir /var/root-tmp  ; chmod 1777 /var/root-tmp
rm -fr /tmp          ; mkdir /tmp                ; chmod 1777 /tmp
echo "/var/root-tmp         /tmp        none        bind" >> /etc/fstab
/bin/mount /tmp
```

%post install chrooted

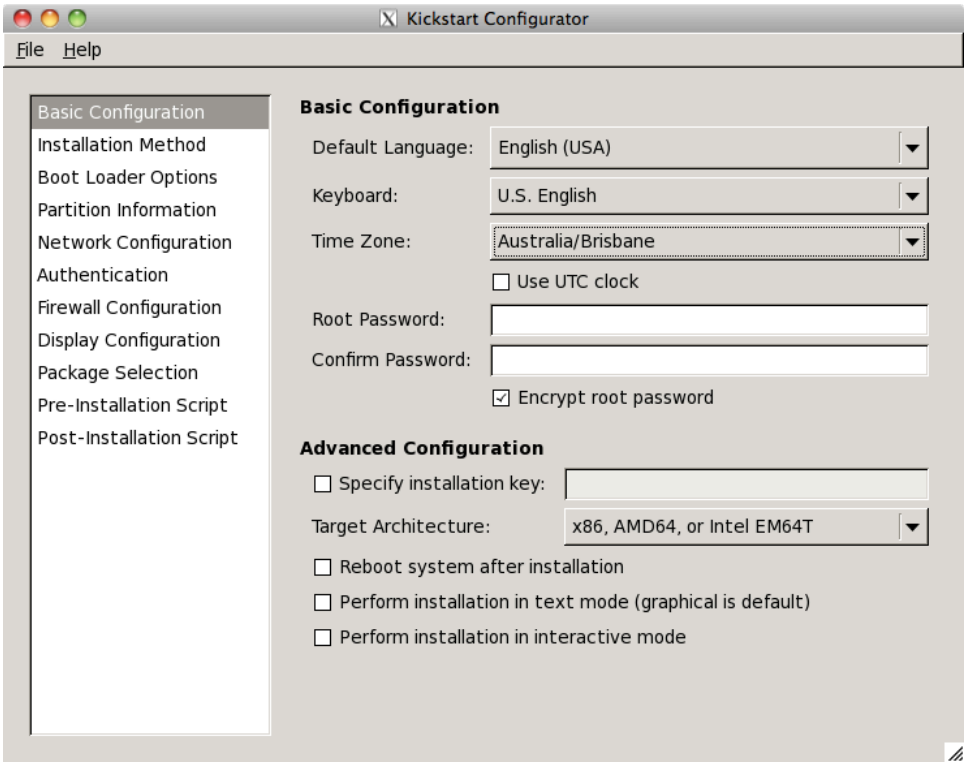
```
# ... continued

# install repo releases (keys and repo files)
rpm -i http://192.168.1.5/mrepo/CentOS6-x86_64/RPMS.epel-x86_64/epel-
release-6-5.noarch.rpm
rpm -i http://192.168.1.5/mrepo/puppetlabs/puppetlabs-products/puppetlabs-
release-6-1.noarch.rpm

# disable repofiles
for repos in `ls /etc/yum.repos.d/` ; do > /etc/yum.repos.d/$repos ; done
chattr +i /etc/yum.repos.d/*repo

# get local configuration
wget http://192.168.1.8/local_repo/LocalMirror.repo -O /etc/yum.repos.d/LocalMirror.repo
wget http://192.168.1.8/hosts/hosts -O /etc/hosts
wget http://192.168.1.8/resolv_conf/resolv.conf -O /etc/resolv.conf

# install puppet
yum clean all
yum clean metadata
yum install puppet -y
```

Things that go wrong

- a firewall (host and/or network);
- the url for the repo server;
- an error in partitioning configuration; or
- a typo in the %pre or %post sections.
- CentOS 6.2 is also finicky about installing grub:
  - make sure you have at least 768MB RAM; and
  - add this to the end of your %post (chroot'ed) section:

```
## grub-install fails consistently
grub-install /dev/sda
```

There's a X11 tool for that



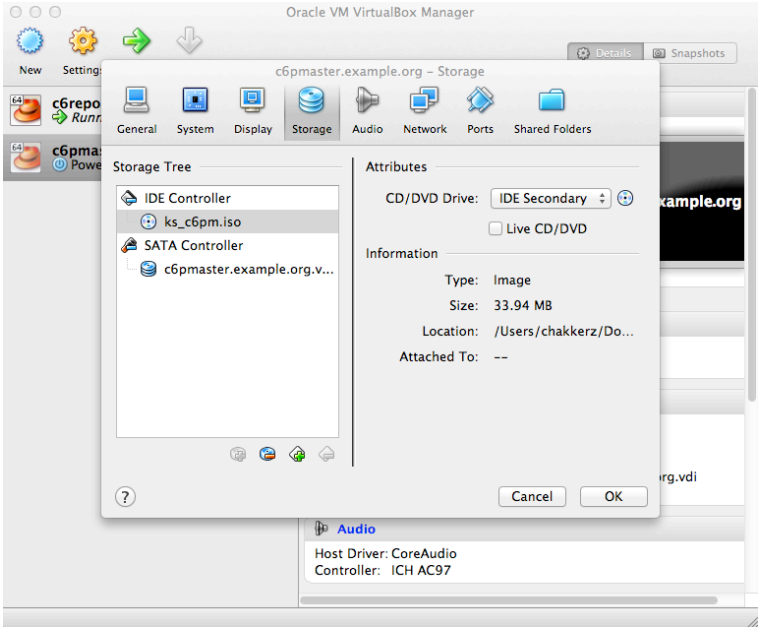
Build the bootable ISO

```
[root@c6repo ~]# mkdir kickstart ; mkdir /var/www/html/ks_isos
[root@c6repo ~]# vi kickstart/ks.cfg
[root@c6repo ~]# mount -o loop /var/mrepo/iso/CentOS-6.2-x86_64-bin-DVD1.iso /mnt/
[root@c6repo ~]# cp -r /mnt/isolinux kickstart/
[root@c6repo ~]# echo -e "label custom\n  kernel vmlinuz\n  append ks=cdrom:/ks.cfg
initrd=initrd.img text" >> kickstart/isolinux/isolinux.cfg
[root@c6repo ~]# sed -i 's:^default.*$:default custom:' kickstart/isolinux/isolinux.cfg
[root@c6repo ~]# sed -i 's:^timeout.*$:timeout 5:' kickstart/isolinux/isolinux.cfg
[root@c6repo ~]# mkisofs -r -N -allow-leading-dots -d -J -T -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -V "kickstart c6puppetmaster" -boot-load-size 4 -boot-
info-table -o /var/www/html/ks_isos/ks_c6pm.iso ./kickstart/
Warning: creating filesystem that does not conform to ISO-9660.
I: -input-charset not specified, using utf-8 (detected in locale settings)
Size of boot image is 4 sectors -> No emulation
 28.84% done, estimate finish Mon Apr  2 13:52:48 2012
 57.57% done, estimate finish Mon Apr  2 13:52:48 2012
 86.39% done, estimate finish Mon Apr  2 13:52:48 2012
Total translation table size: 4701
Total rockridge attributes bytes: 1438
Total directory bytes: 2048
Path table size(bytes): 26
Max brk space used 0
17377 extents written (33 MB)
[root@c6repo ~]#
```

If you have a working http server

Mount disk

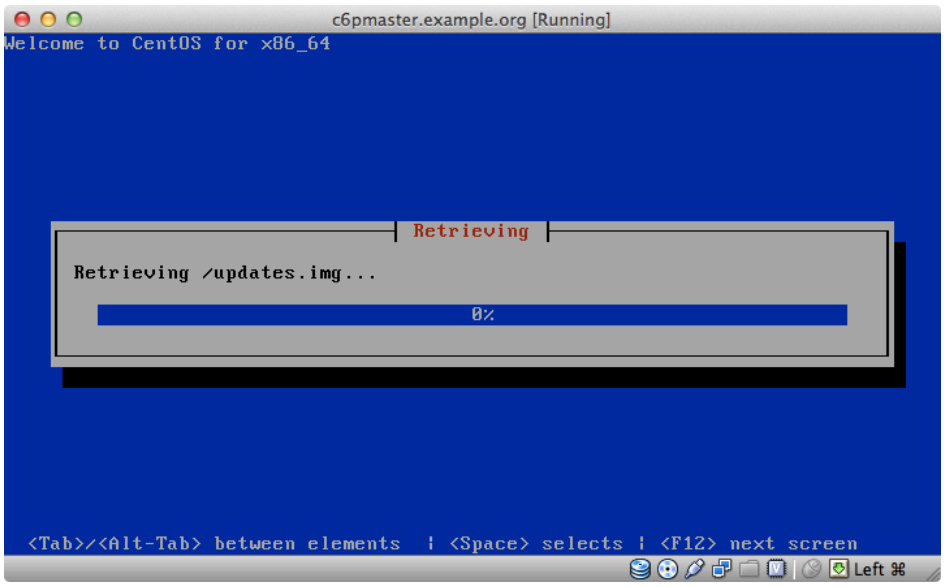
- Mount the disk via a virtual device (DRAC, *LOM, IMM, etc);



- configure the server / vm to boot of the virtual device;

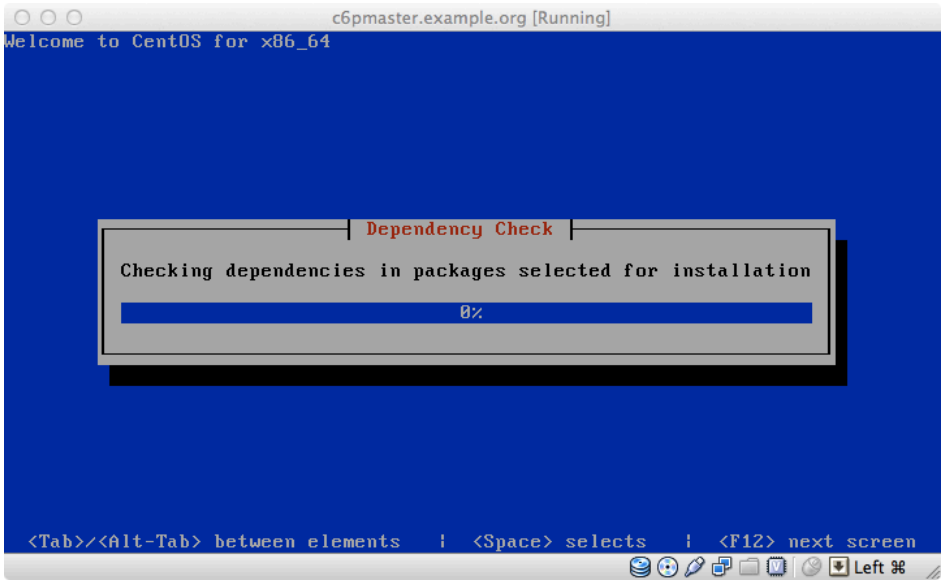
Install

- You should not need to touch a thing.



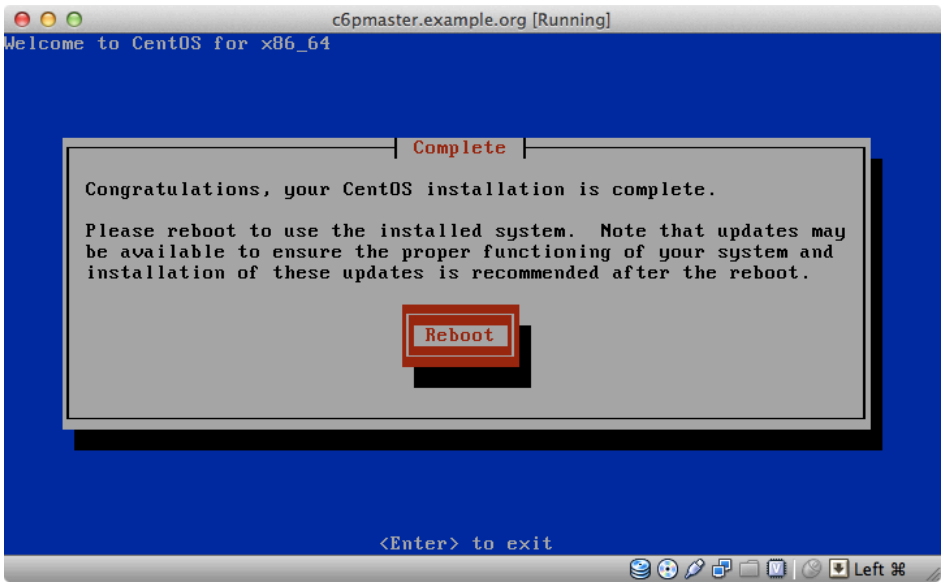
Installing

- ... unless you bungled something that is ...



Finished

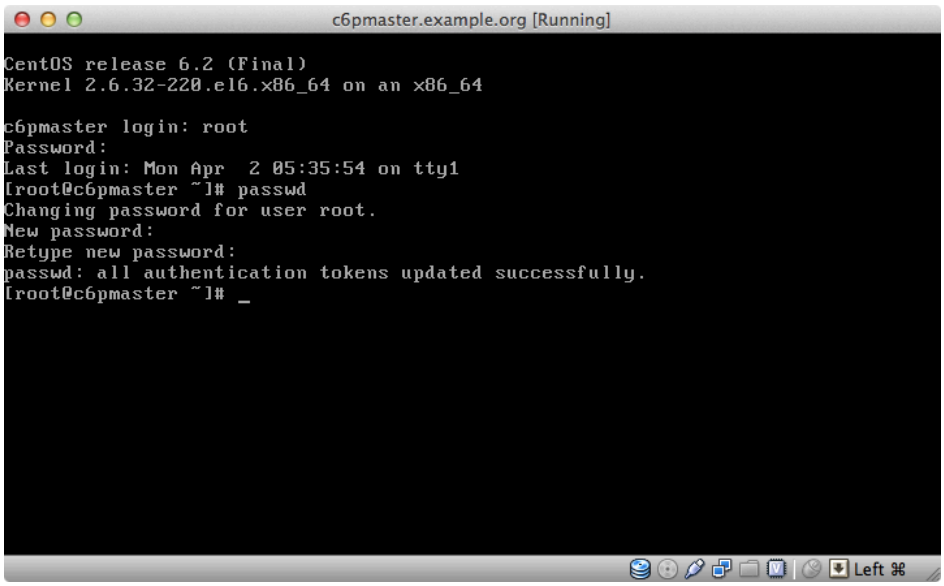
- ... so make sure you unmount the ISO!!



Finalise the build

- log on and change the root password;
- deploy your users or hook up to authentication server;
- configure any services;
- configure the host firewall and tcpwrapper;
- ... or do a lot of these things by configuring puppet.

First Boot



Introduction to Puppet

What is Puppet

Puppet Powers IT Productivity

Puppet is an enterprise systems management platform that standardizes the way IT staff deploy and manage infrastructure in the enterprise and the cloud.

By automating the provisioning, patching, and configuration of operating system and application components across infrastructure, Puppet enables IT staff to master their infrastructure even as complexity grows.

- http://www.puppetlabs.com/puppet/introduction/

Translation

- enterprise ... standardizes [sic] = lots of identical systems;
- operating systems and application components = automated service deployment;
- master infrastructure = go home on time;

Puppet Core Components

- Puppet Server;
- Puppet Agent;
- Puppetca;
- Facter.

Puppet Non-Core Components

- Augeas;
- Apache with Mongrel or Passenger;
- Custom Facts.

Puppet Configuration

- /etc/puppet/puppet.conf
- /etc/puppet/fileserver.conf
- Classes;
- Modules;
- Nodes; and
- Custom facts.

Classes vs Modules

- Both are classes but you use them differently:
  - classes = simple and atomic; vs
  - modules = larger, much more structure; self-contained with a directory structure.

Building a puppet master

- Install Software:

```
yum install puppet-server -y
```

- Installs various dependencies;
- Leaving SELinux in enforcing will require libselinux-ruby which is in the "RHEL Server Optional" add-on channel;

Hello World!

- Create a module;
  - inside it a class;
    - inside it call a file type.

- It will do just one thing:
  - deploy /etc/puppet/puppet.conf

Resource Types

- See: http://docs.puppetlabs.com/references/latest/type.html
- typically of the form:

```
type { "namevar":
 parameter => value,
 ...
 parameterN => value,
}
```

- sometimes value is wrapped in "s or 's
- value should always be followed by a , or ;

Example of a file type

```
file { "/etc/puppet/puppet.conf":
    owner  => root,
    group  => root,
    mode   => 644,
    source => "puppet:///modules/puppet_conf/puppet.conf";
}
```

Example of a class

```
class puppet_conf {
   file { "/etc/puppet/puppet.conf":
      owner  => root,
      group  => root,
      mode   => 644,
      source => "puppet:///modules/puppet_conf/puppet.conf";
   }
}
```

Create a module

- Determine your modulepath:

```
[root@c6pmaster ~]# puppet --configprint modulepath
/etc/puppet/modules:/usr/share/puppet/modules
```

- Create your module's directory structure:

```
[root@c6pmaster ~]# cd /etc/puppet/
[root@c6pmaster puppet]# mkdir -p modules/puppet_conf/manifests
[root@c6pmaster puppet]# mkdir modules/puppet_conf/{files,templates}
```

- Create your module's init.pp:

```
[root@c6pmaster puppet]# vi modules/puppet_conf/manifests/init.pp
```

- chown module to puppet:puppet

```
[root@c6pmaster puppet]# chown -R puppet:puppet modules
```

Your completed module

```
[root@c6pmaster puppet]# pwd
/etc/puppet
[root@c6pmaster puppet]# find ./modules/puppet_conf/ -exec ls -l {} \;
total 12
drwxr-xr-x. 2 puppet puppet 4096 Apr  4 22:47 files
drwxr-xr-x. 2 puppet puppet 4096 Apr  4 22:48 manifests
drwxr-xr-x. 2 puppet puppet 4096 Apr  4 22:47 templates
total 0
total 4
-rw-r--r--. 1 puppet puppet 167 Apr  4 22:48 init.pp
-rw-r--r--. 1 puppet puppet 167 Apr  4 22:48 ./modules/puppet_conf/
manifests/init.pp
total 0
[root@c6pmaster puppet]# cat modules/puppet_conf/manifests/init.pp
class puppet_conf {
    file { "/etc/puppet/puppet.conf":
        owner  => root,
        group  => root,
        mode   => 644,
        source => "puppet:///modules/puppet_conf/puppet.conf";
    }
}
```

Missing Piece: puppet.conf

- cleaned copy from default, with the addition of:
  - server = c6pmaster.example.org

```
[root@c6pmaster puppet]# cp puppet.conf modules/puppet_conf/files/
[root@c6pmaster puppet]# vi modules/puppet_conf/files/puppet.conf
[root@c6pmaster puppet]# cat modules/puppet_conf/files/puppet.conf
[main]
    logdir = /var/log/puppet
    rundir = /var/run/puppet
    ssldir = $vardir/ssl

[agent]
    classfile = $vardir/classes.txt
    localconfig = $vardir/localconfig
        server = c6pmaster.example.org
```

Before this will work

- Configure:
  - firewall to allow access on port 8140/tcp;
  - fileserver.conf;
  - site.pp;
- Accept our client system as a puppet client.

fileserver configuration

- /etc/puppet/fileserver.conf - allow everyone to modules:

```
[modules]
   allow *.example.org
```

- From the example:

```
puppet://{optional hostname}/{mount point}/{remainder of path}

source => "puppet:///modules/puppet_conf/puppet.conf",

{mount point} = modules = our modules directory as per fileserver.conf
{remainder of path} = puppet_conf/puppet.conf = puppet_conf is a module,
which has a files directory so it is handled as “puppet_conf/files” .
```

Site Manifest

- /etc/puppet/manifests/site.pp - include the puppet_conf module:

```
node default {
        include puppet_conf
}
```

puppetmasterd starts

```
[root@c6pmaster puppet]# getenforce
Enforcing
[root@c6pmaster puppet]# service puppetmaster start
Starting puppetmaster:                                     [  OK  ]
[root@c6pmaster puppet]#
```

Then your client connects

```
[root@c6pagent ~]# puppetd -vt --server=c6pmaster.example.org
info: Creating a new SSL key for c6pagent.example.org
warning: peer certificate won't be verified in this SSL session
info: Caching certificate for ca
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
info: Creating a new SSL certificate request for c6pagent.example.org
info: Certificate Request fingerprint (md5): 5E:BE:
02:B5:46:94:11:91:60:EA:7C:E8:C8:87:35:30
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
Exiting; no certificate found and waitforcert is disabled
```

You sign the client

```
[root@c6pmaster puppet]# puppetca --list
  c6pagent.example.org (5E:BE:02:B5:46:94:11:91:60:EA:7C:E8:C8:87:35:30)
[root@c6pmaster puppet]# puppetca --sign c6pagent.example.org
notice: Signed certificate request for c6pagent.example.org
notice: Removing file Puppet::SSL::CertificateRequest
c6pagent.example.org at '/var/lib/puppet/ssl/ca/requests/
c6pagent.example.org.pem'
[root@c6pmaster puppet]#
```

Re-run the client

```
[root@c6pagent ~]# puppetd -vt --server=c6pmaster.example.org
warning: peer certificate won't be verified in this SSL session
info: Caching certificate for c6pagent.example.org
info: Caching certificate_revocation_list for ca
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333548421'
notice: /File[/etc/puppet/puppet.conf]/content:
--- /etc/puppet/puppet.conf 2011-11-15 09:50:43.000000000 +1000
+++ /tmp/puppet-file20120405-1870-7nvzxy-0    2012-04-05
00:06:59.765821607 +1000
@@ -1,25 +1,13 @@
 [main]
-    # The Puppet log directory.
-    # The default value is '$vardir/log'.
     logdir = /var/log/puppet

-    # Where Puppet PID files are kept.
-    # The default value is '$vardir/run'.
     rundir = /var/run/puppet

-    # Where SSL certificates are kept.
-    # The default value is '$confdir/ssl'.
     ssldir = $vardir/ssl
```

Re-run the client ... cont

```
 [agent]
-    # The file in which puppetd stores a list of the classes
-    # associated with the retrieved configuratiion.  Can be loaded in
-    # the separate ``puppet`` executable using the ``--loadclasses``
-    # option.
-    # The default value is '$confdir/classes.txt'.
     classfile = $vardir/classes.txt

-    # Where puppetd caches the local configuration.  An
-    # extension indicating the cache format is added automatically.
-    # The default value is '$confdir/localconfig'.
     localconfig = $vardir/localconfig
+
+   server = c6pmaster.example.org

info: FileBucket adding {md5}58e2f9765e2994db8e8ab19a3513356e
info: /File[/etc/puppet/puppet.conf]: Filebucketed /etc/puppet/
puppet.conf to puppet with sum 58e2f9765e2994db8e8ab19a3513356e
notice: /File[/etc/puppet/puppet.conf]/content: content changed '{md5}
58e2f9765e2994db8e8ab19a3513356e' to '{md5}
f486e08b5a50d5515ae10299ab73e2c2'
info: Creating state file /var/lib/puppet/state/state.yaml
notice: Finished catalog run in 1.08 seconds
[root@c6pagent ~]#
```

You see that it is good

```
[root@c6pmaster puppet]# chkconfig --list puppetmaster
puppetmaster     0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@c6pmaster puppet]# chkconfig puppetmaster on
[root@c6pmaster puppet]# chkconfig --list puppetmaster
puppetmaster     0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

If it's not good

- make sure:
  - your time is in sync;
  - you are not using the short hostname of the server.
- read the error messages;
  - learn when the error message is wrong.

$operatingsystem ?

- but what if your servers aren't all Linux?
- $operatingsystem is a "fact"

```
class puppet_conf {
    file { "/etc/puppet/puppet.conf":
        owner  => root,
        group  => $operatingsystem ?{
            darwin    => wheel,
            default   => root,
        },
        mode   => 644,
        source => "puppet:///modules/puppet_conf/puppet.conf";
    }
}
```

- similar to foo = bar ? "fred" :"fish"

Summary so far

- File resource type;
- /etc/puppet/manifests/site.pp ;
- /etc/puppet/fileserver.conf ; or
- using facts to make decisions
- anything else?

More types

- File (using a templates);
- Service;
- Users, Group and Multiple Files;
- Package;
- Exec;

sshd_config

- This time we will:
  - deploy the sshd_config file from a template;
  - use a numeric GID for the group;
  - use variables; and
  - if the file is changed, restart the sshd service.

sshd_config init.pp

```
class sshd_config {
    if ($operatingsystem == darwin) {
        $sshd_file_path   = "/etc/sshd_config"
        $sshd_service     = "com.openssh.sshd"
    }
    else {
        $sshd_file_path   = "/etc/ssh/sshd_config"
        $sshd_service     = "sshd"
    }

    file { "sshd_config":
        path   => $sshd_file_path,
        owner  => root,
        group  => 0,
        mode   => 600,
        content => template("sshd_config/sshd_config.erb"),
        notify => Service[$sshd_service];
    }

    service { "$sshd_service":
        ensure => running,
        enable => true;
    }
}
```

sshd_config.erb

```
Port 22
AddressFamily any
ListenAddress <%= ipaddress %>
Protocol 2

SyslogFacility AUTHPRIV
PermitRootLogin yes
StrictModes yes
PasswordAuthentication yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
UsePAM yes
X11Forwarding yes
Subsystem sftp   /usr/libexec/openssh/sftp-server
```

and try it

```
Port 22
AddressFamily any
ListenAddress 192.168.1.10
Protocol 2

SyslogFacility AUTHPRIV
PermitRootLogin yes
StrictModes yes
PasswordAuthentication yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
UsePAM yes
X11Forwarding yes
Subsystem sftp   /usr/libexec/openssh/sftp-server
```

- chown the module;
- include sshd_config in site.pp;
- only need to run `puppetd -vt` on client.

PermitRootLogin yes

- Not a good idea, so we'll setup two users in a new module "SysAdmins";
- "sysAdmins" is a valid name for classes, but not for modules;

```
[root@c6pagent ~]# puppetd -vt
err: Could not retrieve catalog from remote server: Error 400 on
SERVER: Could not find class SysAdmins for c6pagent.example.org at /
etc/puppet/manifests/site.pp:4 on node c6pagent.example.org
warning: Not using cache on failed catalog
err: Could not retrieve catalog; skipping run
```

so “sysadmins” it is:

```
class sysadmins {

        if ($operatingsystem == darwin) {
                $home_base = "/Users"
        }
        else {
                $home_base = "/home"
        }

# continued next slide ...
```

class sysadmins part 2

```
# continued next slide ...

    user {
        "chakkerz":
            uid    => 750,
            gid    => 1000,
            comment    => "Christian Unger",
            shell  => "/bin/bash",
            home   => "$home_base/chakkerz",
            # password is chakkerz
            password   => '$1$PX5B30$XybnLRmfShFxScsAXqmid.';

        "foo":
            uid    => 751,
            gid    => 1000,
            comment    => "Foo Bar",
            shell  => "/bin/bash",
            home   => "$home_base/foo",
            # password is barry
            password   => '$1$m16B30$AYeyT/XyRpEHmEym7fDmK/';
    }

# continued next slide ...
```

class sysadmins part 3

```
# continued next slide ...

    group { "sysadmins":
        gid    => 1000,
        before => [User["chakkerz"],User["foo"],];
    }

# and then some more ...
```

class sysadmins part 4

```
# and then some more ...

    file {
        "$home_base/chakkerz":
            ensure => directory,
            owner  => chakkerz,
            group  => sysadmins,
            mode   => 700,
            require    => User["chakkerz"];

        "$home_base/foo":
            ensure => directory,
            owner  => foo,
            group  => sysadmins,
            mode   => 700,
            require    => User["foo"];
    }
}
```

Before...

```
[root@c6pagent ~]# egrep "chakkerz|foo|sysadmins" /etc/{passwd,shadow,group}
[root@c6pagent ~]# ls -l /home
total 16
drwx------. 2 root root 16384 Apr  4 23:24 lost+found
[root@c6pagent ~]#
```

... and after on Linux

```
[root@c6pagent ~]# egrep "chakkerz|foo|sysadmins" /etc/{passwd,shadow,group}
/etc/passwd:chakkerz:x:750:1000:Christian Unger:/home/chakkerz:/bin/bash
/etc/passwd:foo:x:751:1000:Foo Bar:/home/foo:/bin/bash
/etc/shadow:chakkerz:$1$PX5B30$XybnLRmfShFxScsAXqmid.:15434:0:99999:7:::
/etc/shadow:foo:$1$m16B30$AYeyT/XyRpEHmEym7fDmK/:15434:0:99999:7:::
/etc/group:sysadmins:x:1000:
[root@c6pagent ~]# ls -l /home
total 24
drwx------. 2 chakkerz sysadmins  4096 Apr  5 03:22 chakkerz
drwx------. 2 foo      sysadmins  4096 Apr  5 03:22 foo
drwx------. 2 root     root      16384 Apr  4 23:24 lost+found
[root@c6pagent ~]#
```

... and after on Darwin

```
bash-3.2# dscacheutil -q user | grep "name: chakkerz" -A7 ; dscacheutil -q user |
grep "name: foo" -A7 ; dscacheutil -q group | grep "name: sysadmins" -A3 ; ls -l /
Users/ | egrep "foo|chakkerz"
name: chakkerz
password: ********
uid: 750
gid: 1000
dir: /Users/chakkerz
shell: /bin/bash
gecos: Christian Unger

name: foo
password: ********
uid: 751
gid: 1000
dir: /Users/foo
shell: /bin/bash
gecos: Foo Bar

name: sysadmins
password:
gid: 1000

drwx------   2 chakkerz  sysadmins   68 Jun 29 16:16 chakkerz
drwx------   2 foo       sysadmins   68 Jun 29 16:16 foo
bash-3.2#
```

Ordering

- Before and Require (see sysadmins);
- Notify and Subscribe;
- Chaining.

sshd_config as it was

```
class sshd_config {
    if ($operatingsystem == darwin) {
        $sshd_file_path = "/etc/sshd_config"
        $sshd_service = "com.openssh.sshd"
    }
    else {
        $sshd_file_path   = "/etc/ssh/sshd_config"
        $sshd_service = "sshd"
    }

    file { "sshd_config":
        path   => $sshd_file_path,
        owner  => root,
        group  => 0,
        mode   => 600,
        content => template("sshd_config/sshd_config.erb"),
        notify => Service[$sshd_service];
    }

    service { "$sshd_service":
        ensure => running,
        enable => true;

    }


}
```

sshd_config subscribe

```
class sshd_config {
    if ($operatingsystem == darwin) {
        $sshd_file_path = "/etc/sshd_config"
        $sshd_service = "com.openssh.sshd"
    }
    else {
        $sshd_file_path   = "/etc/ssh/sshd_config"
        $sshd_service = "sshd"
    }

    file { "sshd_config":
        path   => $sshd_file_path,
        owner  => root,
        group  => 0,
        mode   => 600,
        content => template("sshd_config/sshd_config.erb");

    }

    service { "$sshd_service":
        ensure => running,
        enable => true,
        subscribe => File[“sshd_config”];
    }


}
```

sshd_config chained

```
class sshd_config{
    if ($operatingsystem == darwin) {
        $sshd_file_path = "/etc/sshd_config"
        $sshd_service = "com.openssh.sshd"
    }
    else {
        $sshd_file_path   = "/etc/ssh/sshd_config"
        $sshd_service = "sshd"
    }

    file { "sshd_config":
        path   => $sshd_file_path,
        owner  => root,
        group  => 0,
        mode   => 600,
        content => template("sshd_config/sshd_config.erb");

    }

    service { "$sshd_service":
        ensure => running,
        enable => true;

    }

    File["sshd_config"] ~> Service["$sshd_service"]
}
```

so update sshd_config

- So now that we can log into the host as not root, we can disable PermitRootLogin

```
[root@c6pagent ~]# puppetd -vt
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333560535'
notice: /File[sshd_config]/content:
--- /etc/ssh/sshd_config  2012-04-05 00:43:18.662536129 +1000
+++ /tmp/puppet-file20120405-21124-8rrwyy-0    2012-04-05 03:28:52.809414062 +1000
@@ -4,7 +4,7 @@
 Protocol 2

 SyslogFacility AUTHPRIV
-PermitRootLogin yes
+PermitRootLogin no
 StrictModes yes
 PasswordAuthentication yes
 GSSAPIAuthentication yes

info: FileBucket adding {md5}3d82eb51df0702e97a53be5905f150da
info: /File[sshd_config]: Filebucketed /etc/ssh/sshd_config to puppet with sum
3d82eb51df0702e97a53be5905f150da
notice: /File[sshd_config]/content: content changed '{md5}
3d82eb51df0702e97a53be5905f150da' to '{md5}afc2d4cd365c9f3f377314f466664e81'
info: /File[sshd_config]: Scheduling refresh of Service[sshd]
notice: /Stage[main]/Sshd_config/Service[sshd]: Triggered 'refresh' from 1 events
notice: Finished catalog run in 1.43 seconds
[root@c6pagent ~]#
```

Some notes about users

- unlike most examples that was very complete, if your using Linux you can skip a lot of that, e.g.:

```
"baz":
   comment    => "Baz Contrived",
   system     => true,
   managehome => true;
```

- results in:

```
[root@c6pagent ~]# grep baz /etc/passwd
baz:x:498:496:Baz Contrived:/home/baz:/bin/bash
[root@c6pagent ~]# ls -ld /home/baz/
drwx------. 2 baz baz 4096 Apr 14 22:04 /home/baz/
```

- see http://docs.puppetlabs.com/references/latest/type.html#user-3

but there is a downside

- In an enterprise setting odds are:
  - there is a centralised and authoritative UID / GID register for people and service users;
  - might not (want to) use the files deployed;
  - centralised authentication.

```
[root@c6pagent baz]# ls -lna
total 20
drwx------. 2 498 497 4096 Apr  5 03:37 .
drwxr-xr-x. 6   0   0 4096 Apr  5 03:37 ..
-rw-r--r--. 1 498 497   18 Dec  3 00:27 .bash_logout
-rw-r--r--. 1 498 497  176 Dec  3 00:27 .bash_profile
-rw-r--r--. 1 498 497  124 Dec  3 00:27 .bashrc
```

package type

```
class packages {
        package {
                "nano":                 ensure  => absent;
                "elinks":               ensure  => installed;
                "telnet":               ensure  => installed;
        }
}
```

- results in:

```
[root@c6pagent ~]# rpm -q nano elinks telnet
nano-2.0.9-7.el6.x86_64
package elinks is not installed
package telnet is not installed
[root@c6pagent ~]# puppetd -vt 2>&1 1> /dev/null
[root@c6pagent ~]# rpm -q nano elinks telnet
package nano is not installed
elinks-0.12-0.20.pre5.el6.x86_64
telnet-0.17-47.el6.x86_64
[root@c6pagent ~]#
```

providers

- This does not work in OS X unless the package provider is set to “macports”;
- in site.pp add:

```
package { provider => “macports”; }
```

- also applies to other resource types;
- http://docs.puppetlabs.com/references/stable/type.html#package
- http://www.puppetcookbook.com/posts/changing-default-package-provider.html

buy one type, get N free!

- the difference between , and ;
- completely optional;
- group:
  - alike;
  - by purpose;
  - for flow; or
  - to confuse.

exec type and variable

```
class execute {
    exec { "echo top into /tmp/puppet.top":
        command => $operatingsystem ? {
            darwin  => "/usr/bin/top -l 1 >> puppet.top",
            default => "/usr/bin/top -bn1 >> puppet.top",
        },
        cwd     => "/tmp";
    }

    $touch_once = "/tmp/puppet.touch.once"

    exec { "touch a file just once":
        command => $operatingsystem ? {
            darwin  => "/usr/bin/touch $touch_once",
            default => "/bin/touch $touch_once",
        },
        cwd     => "/",
        creates => $touch_once;
    }
}
```

two for one

```
class execute {

    $touch_once = "/tmp/puppet.touch.once"

    exec {
        "echo top into /tmp/puppet.top":
            command => $operatingsystem ? {
                darwin  => "/usr/bin/top -l 1 >> puppet.top",
                default => "/usr/bin/top -bn1 >> puppet.top",
            },
            cwd     => "/tmp";

        "touch a file just once":
            command => $operatingsystem ? {
                darwin  => "/usr/bin/touch $touch_once",
                default => "/bin/touch $touch_once",
            },
            cwd     => "/",
            creates => $touch_once;
    }
}
```

exec type result CentOS

```
[root@c6pagent ~]# puppetd -vt
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333566663'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: /Stage[main]/Execute/Exec[touch a file just once]/returns:
executed successfully
notice: Finished catalog run in 1.64 seconds
[root@c6pagent ~]# ls -l /tmp/puppet*
-rw-r--r--. 1 root root 7489 Apr  5 05:11 /tmp/puppet.top
-rw-r--r--. 1 root root    0 Apr  5 05:11 /tmp/puppet.touch.once
[root@c6pagent ~]# puppetd -vt
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333566663'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: Finished catalog run in 1.59 seconds
[root@c6pagent ~]# ls -l /tmp/puppet*
-rw-r--r--. 1 root root 14978 Apr  5 05:13 /tmp/puppet.top
-rw-r--r--. 1 root root     0 Apr  5 05:11 /tmp/puppet.touch.once
[root@c6pagent ~]#
```

exec type result OS X

```
bash-3.2# ls -l /tmp/puppet*
ls: /tmp/puppet*: No such file or directory
bash-3.2# puppetd -vt
info: Caching catalog for osx.example.org
info: Applying configuration version '1309331288'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns:
executed successfully
notice: /Stage[main]/Execute/Exec[touch a file just once]/returns: executed
successfully
notice: Finished catalog run in 14.58 seconds
bash-3.2# ls -l /tmp/puppet*
-rw-r--r--  1 root  wheel  7848 Jun 29 17:15 /tmp/puppet.top
-rw-r--r--  1 root  wheel     0 Jun 29 17:15 /tmp/puppet.touch.once
bash-3.2# puppetd -vt
info: Caching catalog for osx.example.com
info: Applying configuration version '1309331288'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns:
executed successfully
notice: Finished catalog run in 14.26 seconds
bash-3.2# ls -l /tmp/puppet*
-rw-r--r--  1 root  wheel  15696 Jun 29 17:17 /tmp/puppet.top
-rw-r--r--  1 root  wheel      0 Jun 29 17:15 /tmp/puppet.touch.once
bash-3.2#
```

Summary so far

- Resource types:
  - files, directories and templates;
  - users and groups;
  - package and exec;
- Ordering;
- Grouping types;
- Coming up with strange puppet examples.

nodes

- Needed to customise specific (groups of) hosts;
- Setting this up the first time feels buggy and the syntax strikes me as counter intuitive;
- This will also cover inheritance.
- 2.6.x bug: if you've never had a node file, puppet doesn't use your new one.

nodes - step 1

- create "nodes" inside "manifests";
  - mkdir /etc/puppet/manifest/nodes
- move site.pp to nodes/defaultnode.node .

```
[root@c6pmaster puppet]# pwd
/etc/puppet
[root@c6pmaster puppet]# mkdir manifests/nodes
[root@c6pmaster puppet]# mv manifests/site.pp manifests/nodes/
defaultnode.node
[root@c6pmaster puppet]# cat manifests/nodes/defaultnode.node
class defaultnode {
    include execute
    include packages
    include puppet_conf
    include sshd_config
    include sysadmins
}
[root@c6pmaster puppet]#
```

nodes - alternate step 1

- Uses inheritance. This has massive drawbacks later.

```
[root@c6pmaster puppet]# pwd
/etc/puppet
[root@c6pmaster puppet]# mkdir manifests/nodes
[root@c6pmaster puppet]# mv manifests/site.pp manifests/nodes/
defaultnode.node
[root@c6pmaster puppet]# cat manifests/nodes/defaultnode.node
node default {
    include execute
    include packages
    include puppet_conf
    include sshd_config
    include sysadmins
}
[root@c6pmaster puppet]#
```

nodes - step 2

- create a new site.pp:

```
[root@c6pmaster puppet]# pwd
/etc/puppet
[root@c6pmaster puppet]# echo import \"nodes/*.node\" > manifests/
site.pp
[root@c6pmaster puppet]# cat manifests/site.pp
import "nodes/*.node"
[root@c6pmaster puppet]#
```

- make sure:
  - you have quotes;
  - you have the file extension of your nodes;
  - just * does not work.

nodes - step 3

- create nodes/c6repo.node

```
node "c6repo.example.org"{
        package {
                "emacs":                ensure  => installed;
        }

        include defaultnode
}
```

- create nodes/c6pagent.node

```
node "c6pagent.example.org" {

        include defaultnode
}
```

- Does not use inheritance.

nodes - altenative step 3

- create nodes/c6repo.node

```
node "c6repo.example.org" inherits default {
        package {
                "emacs":                ensure  => installed;
        }
}
```

- create nodes/c6agent.node

```
node "c6pagent.example.org" inherits default {

}
```

- Uses inheritance.

nodes - step 4

- and apply on both client nodes:

```
...
notice: Finished catalog run in 89.68 seconds
[root@c6repo ~]# rpm -q emacs
emacs-23.1-21.el6_2.3.x86_64
[root@c6repo ~]#
```

vs

```
...
notice: Finished catalog run in 1.57 seconds
[root@c6pagent ~]# rpm -q emacs
package emacs is not installed
[root@c6pagent ~]#
```

custom facts and conditional

- verify you are on a particular version of Linux;
- use this knowledge in an if statement;

what is a fact?

- facts are ... facts about your system collected by facter;
- they are determined before the main puppet run;
- you can see them in /var/lib/puppet/yaml/nodes/<fqdn>.yaml
  - $fqdn is a fact.

<%= ipaddress %>

- used fact $ipaddress in sshd_config.erb template,
  - in nodes and classes they are addressed with a $ before their name;
  - in templates there is no $.

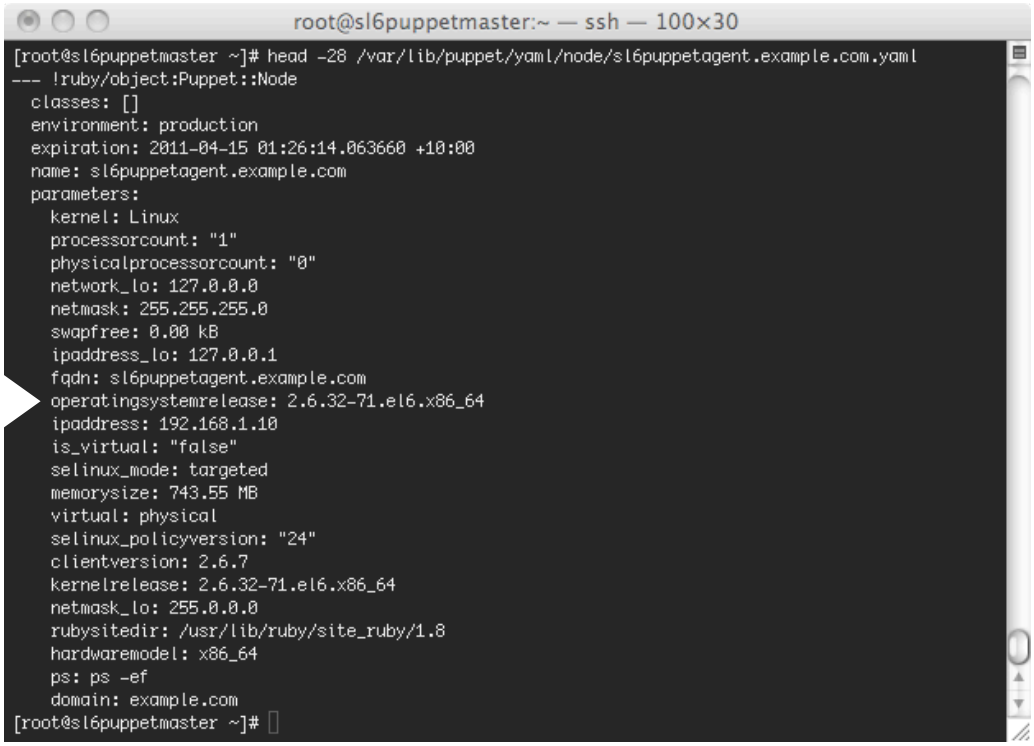
big brother is watching

```
[root@c6pmaster node]# head -23 freya.example.org.yaml
--- !ruby/object:Puppet::Node
  classes: []
  environment: production
  expiration: 2012-04-04 07:30:37.822141 +10:00
  name: freya.example.org
  parameters:
    sp_number_processors: "2"
    kernelmajversion: "9.8"
    clientversion: 2.6.7
    macosx_productversion_major: "10.5"
    sp_machine_name: iMac
    sp_boot_volume: os
    sp_platform_uuid: 00000000-0000-1000-8000-001B63AA9DB1
    ps: ps auxwww
    netmask: 255.255.254.0
    sp_packages: "1"
    sp_boot_rom_version: IM71.007A.B03
    hostname: freya
    sp_machine_model: "iMac7,1"
    sp_smc_version_system: 1.20f4
    kernelrelease: 9.8.0
    sp_current_processor_speed: 2.4 GHz
    kernel: Darwin
```

big brother is watching

```
[root@c6pmaster node]# head -24 c6pagent.example.org.yaml
--- !ruby/object:Puppet::Node
  classes: []
  environment: &id001 production
  expiration: 2012-04-05 06:34:52.886165 +10:00
  name: c6pagent.example.org
  parameters:
    serialnumber: "0"
    ipaddress_eth0: 192.168.1.10
    memoryfree: 333.30 MB
    selinux_config_policy: &id003 targeted
    id: root
    kernel: Linux
    selinux_policyversion: "24"
    netmask: 255.255.255.0
    environment: *id001
    uptime_hours: "6"
    clientcert: c6pagent.example.org
    hardwaremodel: &id002 x86_64
    productname: VirtualBox
    kernelversion: 2.6.32
    rubysitedir: /usr/lib/ruby/site_ruby/1.8
    network_lo: 127.0.0.0
    uptime_seconds: "22130"
    interfaces: "eth0,lo"
```

bug squashed



bug squashed

```
[root@c6pmaster node]# grep operating c6pagent.example.org.yaml
    operatingsystem: CentOS
    operatingsystemrelease: "6.2"
```

contrivances

- $operatingsystemrelease = “6.2”, but want “6”;
- Scientific Linux ~~has~~ had a bug:
  - http://projects.puppetlabs.com/issues/6679
  - $operatingsystem now reports as “Scientific”
- but see $operatingsystemrelease on the previous slides and:
  - http://projects.puppetlabs.com/issues/7682
- In short, facts can ruin your perfect day.

rh_release.rb

```
[root@c6pmaster ~]# cd /etc/puppet/modules/
[root@c6pmaster modules]# mkdir -p custom/lib/facter
[root@c6pmaster modules]# vi custom/lib/facter/rh_release.rb
[root@c6pmaster modules]# cat custom/lib/facter/rh_release.rb
Facter.add("rh_release") do
    setcode do
        %x{/bin/cat /etc/redhat-release | /bin/sed 's/[^0-9.]*//g' | /
bin/cut -d . -f 1}.chomp
    end
end
[root@c6pmaster modules]# /bin/cat /etc/redhat-release | /bin/sed 's/
[^0-9.]*//g' | /bin/cut -d . -f 1
6
[root@c6pmaster modules]#
```

pluginsync = true

- modify puppet_conf/files/puppet.conf to:

```
[root@c6pmaster modules]# vi puppet_conf/files/puppet.conf
[root@c6pmaster modules]# cat puppet_conf/files/puppet.conf
[main]
    logdir = /var/log/puppet

    rundir = /var/run/puppet

    ssldir = $vardir/ssl

   pluginsync = true

[agent]
    classfile = $vardir/classes.txt

    localconfig = $vardir/localconfig

   server = c6pmaster.example.org
[root@c6pmaster modules]#
```

on the client

```
[root@c6pagent ~]# puppetd -vt
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333569700'
notice: /File[/etc/puppet/puppet.conf]/content:
--- /etc/puppet/puppet.conf 2012-04-05 00:07:00.528401772 +1000
+++ /tmp/puppet-file20120405-23004-tspm02-0  2012-04-05
07:01:47.428646743 +1000
@@ -5,6 +5,8 @@

    ssldir = $vardir/ssl

+      pluginsync = true
+
 [agent]
     classfile = $vardir/classes.txt


info: FileBucket adding {md5}f486e08b5a50d5515ae10299ab73e2c2
info: /File[/etc/puppet/puppet.conf]: Filebucketed /etc/puppet/
puppet.conf to puppet with sum f486e08b5a50d5515ae10299ab73e2c2
notice: /File[/etc/puppet/puppet.conf]/content: content changed
'{md5}f486e08b5a50d5515ae10299ab73e2c2' to '{md5}
e54db7e450a10b41b3654694d7abc77b'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: Finished catalog run in 2.73 seconds
```

on the client

```
[root@c6pagent ~]# puppetd -vt
info: Retrieving plugin
notice: /File[/var/lib/puppet/lib/facter]/ensure: created
notice: /File[/var/lib/puppet/lib/facter/rh_release.rb]/ensure:
defined content as '{md5}c872f6c6d50139da8034661183d7e1b1'
info: Loading downloaded plugin /var/lib/puppet/lib/facter/
rh_release.rb
info: Loading facts in /var/lib/puppet/lib/facter/rh_release.rb
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333569700'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: Finished catalog run in 3.08 seconds
```

Now the server knows

```
[root@c6pmaster modules]# cd /var/lib/puppet/yaml/node/
[root@c6pmaster node]# grep rh_release c6pagent.example.org.yaml
    rh_release: "6"
[root@c6pmaster node]#
```

- ... so let's use it ...

rh_release_if

```
class rh_release_if {

# always symlink
    file { "/root/rh_release.$rh_release":
        ensure => "/etc/redhat-release";
    }

# conditionally install/remove some packages
    if ($rh_release == "5") {
        package { "rsyslog":
            ensure => installed;
        }
    } elsif ($rh_release == "4") {
        package { ["rsyslog", "sssd"]:
            ensure => installed;
        }
    } else {
        package { "syslogd":
            ensure => absent;
        }
    }
}
```

Remember

- to include this module we are now modifying:
  - /etc/puppet/manifests/nodes/defaultnode.node

execute on the client

```
[root@c6pagent ~]# pwd
/root
[root@c6pagent ~]# ls -l rh*
ls: cannot access rh*: No such file or directory
[root@c6pagent ~]# puppetd -vt
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/rh_release.rb
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333603327'
notice: /File[/root/rh_release.6]/ensure: created
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: Finished catalog run in 15.58 seconds
[root@c6pagent ~]# ls -l rh*
lrwxrwxrwx. 1 root root 19 Apr  5 09:40 rh_release.6 -> /etc/redhat-
release
[root@c6pagent ~]#
```

Gigantic No-No

Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
Never Never Never Never Never Never Never
use a custom facts to change the system; Never

define

- Like a function or procedure in programming;
  - ... used for sets of logically related operations;
- ~~Defined (pun not intended) outside a class;~~
- Can be inside or outside a class;
  - if defined inside a class, can be addressed directly using scope operators;
  - if defined outside a class, it becomes harder to follow the code.

define choices

- Choose where you use define with care:
  - Odds are you will want to use it in more than one module;
  - Capable of making your code very flexible, but may make it a nightmare to follow (never mind maintain).
- See "Puppet for Developers" talk.

define example

```
class local_users {
    deploy_user { "Tyler Durden":
        user    => "t.durden",
        uid     => 1001,
        gid     => 100,
        comment    => "Tyler Durden";
    }
}

define deploy_user($user, $uid, $gid, $comment) {
    file { "/home/$user":
        owner  => $uid,
        group  => $gid,
        mode   => 700,
        ensure => directory;
    }

    user { "$user":
        uid => $uid,
        gid => $gid,
        comment    => $comment,
        home   => "/home/$user",
        shell  => "/bin/bash",
        require    => File["/home/$user"];
    }
}
```

In Tyler we trust

```
[root@c6pagent home]# puppetd -vt
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/rh_release.rb
info: Caching catalog for c6pagent.example.org
info: Applying configuration version '1333605064'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/
returns: executed successfully
notice: /File[/home/t.durden]/ensure: created
notice: /Stage[main]/Local_users/Deploy_user[Tyler Durden]/
User[t.durden]/ensure: created
notice: Finished catalog run in 23.20 seconds
[root@c6pagent home]# grep t.durden /etc/passwd
t.durden:x:1001:100:Tyler Durden:/home/t.durden:/bin/bash
[root@c6pagent home]# ls -ld t*
drwx------. 2 t.durden users 4096 Apr  5 10:09 t.durden
[root@c6pagent home]# ls -ldn t*
drwx------. 2 1001 100 4096 Apr  5 10:09 t.durden
```

puppet agent as a service

- splay - true or false;
- runinterval - in seconds
  - default is 1800;
- syslogfacility - e.g.: local0
  - default is daemon;
- environment - e.g.: ... up to you ...
  - default is allegedly production ...
  - Not covered in this slide show.

puppet agent as a service

- graph - true or false;
  - default is false;
  - gives dependencies (ordering)
- report - true or false;
  - default is false;
  - needed for puppet-dashboard;
- see man puppet.conf

puppet agent as a service

```
[main]
    logdir = /var/log/puppet
    rundir = /var/run/puppet
    ssldir = $vardir/ssl
       pluginsync = true

[agent]
    classfile = $vardir/classes.txt
    localconfig = $vardir/localconfig
       server = c6pmaster.example.org
       splay = true
       runinterval = 1800
       environment = main
```

- first indent is default, second is custom;
- distribute via puppet_conf module ...

puppet agent as a service

```
class puppet_conf {
   file { "/etc/puppet/puppet.conf":
      owner  => root,
      group  => 0,
      mode   => 644,
      source => "puppet:///modules/puppet_conf/puppet.conf",
      notify => Service["puppet"];
   }

   service { "puppet":
      name   => $operatingsystem ? {
         darwin => "com.reductivelabs.puppet",
         default   => "puppet",
      },
      ensure => running,
      enable => true;
   }
}
```

- contains service resource type; and
- file resource type notifies the service.

puppet and launchd

- http://projects.puppetlabs.com/projects/1/wiki/Puppet_With_Launchd
- plist and service name will be:

```
/Library/LaunchDaemons/com.reductivelabs.puppet.plist
```

- instructions also cover puppetmaster;

PuppetNow

- for when you want to run puppet now:

```
#!/bin/bash
/sbin/service puppetd stop
/bin/rm -f /var/lib/puppet/state/puppetdlock
/usr/sbin/puppetd -vt
/sbin/service puppetd start
```

coping with real load

- Built-in file server Webrick (?) is dreadful;
- Mongrel - generally available with Linux;
  - apparently has a bad memory leak;
- Passenger - available from Puppetlabs;
  - does not have the memory leak;
  - not as good as Mongrel.

Tune

- Tune the splay and run interval times to suit:
  - remember - puppet should not be changing a lot on each run;
- Write your modules so they do not do "excessive" work; avoid
  - changing a lot on each run;
  - recursive file transfers;

Good Ideas

- Keep node specific things out of your modules;
- Build in file overrides;
- Write your modules with on or off switch (and sensible default behaviour);
- If you're really clever, build in an undo;

on / off switch

```
class sshd_config
{

    if ($skip_sshd_config != "true") {

        if ($operatingsystem == darwin) {
            $sshd_file_path = "/etc/sshd_config"
            $sshd_service    = "com.openssh.sshd"
        }
        else {
            $sshd_file_path  = "/etc/ssh/sshd_config"
            $sshd_service    = "sshd"
        }

        file { "sshd_config":
            path=> $sshd_file_path,
            owner    => root,
            group    => 0,
            mode=> 600,
            content => template("sshd_config/sshd_config.erb"),
            notify   => Service[$sshd_service],
        }

        service { "$sshd_service":
            ensure   => running,
            enable   => true,
        }
    }
}
```

- Have a proper set of naming conventions;

Change Management

- When you modify your puppet config:
  - let people know;
  - document that you changed things;
  - check your systems after they have been getting the updates;

pitfalls

- style, choose one and document it;
  - http://projects.puppetlabs.com/projects/1/wiki/Puppet_Best_Practice
- ~~SELinux~~;
- automate whenever possible - if you can write a reusable class (or module) do it sooner rather than later.

How this relates to SOE

- Puppet can maintain your SOE by:
  - completing the install process;
  - evolving your SOE by installing / removing packages;
  - deploying files and services (almost automatically) the same way every time;
- Can be really handy in DR situations;

Puppet DR

- If you built your hosts via puppet it will have a record of how to remake the node;
  - configure systems via puppet;
  - potentially difficult with SLA setups where a consultant is involved;
- Great for customer confidence;
- Not a replacement for documentation.