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.

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.

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.

Scientific Linux 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.scientificlinux.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://download.fedora.redhat.com/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://download.fedora.redhat.com/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
- /usr/share/doc/mrepo-0.8.7/dists/ contains examples for various distributions
- Configured for Scientific Linux 6 + EPEL (x86_64 only)...

Sample MRepo configuration file

```
[sl6]
name = ScientificLinux $release ($arch)
release = 6x
arch = x86_64
metadata = repomd repoview

### ISO images
iso = SL-60-x86_64-2011-03-03-Everything-DVD?.iso

### BASE Release
#sl-base = http://ftp.scientificlinux.org/linux/scientific/6x/x86_64/os/

### Additional repositories
sl-security = http://ftp.scientificlinux.org/linux/scientific/6x/x86_64/
updates/security/
sl-fastbugs = http://ftp.scientificlinux.org/linux/scientific/6x/x86_64/
updates/fastbugs/

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

MRepo - ... continued

- Copy ISO(s) to /var/mrepo/iso to save you downloading everything (see sl-base in mrepo config example);
- 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.

Spanner in the works ... just add SELinux

- By default SELinux is enabled.
- Because of how mrepo works (caches in /var/mrepo and servers via /var/www/mrepo), all the files are not going to be served by httpd.
- /var/mrepo/<cache> should be httpd_content_t

```
[root@sl6repo ~]# semanage fcontext -a -t httpd_sys_content_t /var/mrepo/sl6-x86_64\(/.*\)?
[root@sl6repo ~]# semanage fcontext -l | grep mrepo
/var/mrepo/sl6-x86_64(/.*)?        all files      system_u:object_r:httpd_sys_content_t:s0
[root@sl6repo local_repo]# restorecon -R -v /var/mrepo/sl6-x86_64/
restorecon reset /var/mrepo/sl6-x86_64/sl-errata context unconfined_u:object_r:var_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/mrepo/sl6-x86_64/sl-contrib context unconfined_u:object_r:var_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
...snip...
```

SELinux

Before you turn off SELinux, ask yourself:

"What if my repository server is compromised?"

reposync

- create the repo file for yum; for instance /etc/yum.repos.d/epel-puppet.repo :

```
[epel-puppet]
name=epel puppet
baseurl=http://tmz.fedorapeople.org/repo/puppet/epel/6/$basearch/
enabled=1
gpgcheck=1
gpgkey=http://tmz.fedorapeople.org/repo/RPM-GPG-KEY-tmz
```

- and sync the repo:

```
[root@sl6repo ~]# rm -rf /var/www/mrepo/reposync/
[root@sl6repo ~]# mkdir /var/www/mrepo/reposync
[root@sl6repo ~]# reposync -p !$ -a x86_64 -r epel-puppet -nreposync -p /var/
www/mrepo/reposync/ -a x86_64 -r epel-puppet -n
[epel-puppet: 1     of 3     ] Downloading facter-1.5.9-0.3.rc5.el6.noarch.rpm
facter-1.5.9-0.3.rc5.el6.noarch.rpm                      |  62 kB     00:01
[epel-puppet: 2     of 3     ] Downloading puppet-2.6.7-1.el6.noarch.rpm
puppet-2.6.7-1.el6.noarch.rpm                            | 807 kB     00:03
[epel-puppet: 3     of 3     ] Downloading puppet-server-2.6.7-1.el6.noarch.rpm
puppet-server-2.6.7-1.el6.noarch.rpm                     |  20 kB     00:00
[root@sl6repo ~]#
```

createrepo

- create your new repository:

```
[root@sl6repo ~]# ls -l /var/www/mrepo/reposync/epel-puppet/
total 896
-rw-r--r--. 1 root root  63672 Apr  8 09:51 facter-1.5.9-0.3.rc5.el6.noarch.rpm
-rw-r--r--. 1 root root 826744 Mar 25 12:00 puppet-2.6.7-1.el6.noarch.rpm
-rw-r--r--. 1 root root  20792 Mar 25 12:00 puppet-server-2.6.7-1.el6.noarch.rpm
[root@sl6repo ~]# createrepo /var/www/mrepo/reposync/epel-puppet/
3/3 - facter-1.5.9-0.3.rc5.el6.noarch.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
[root@sl6repo ~]#
```

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

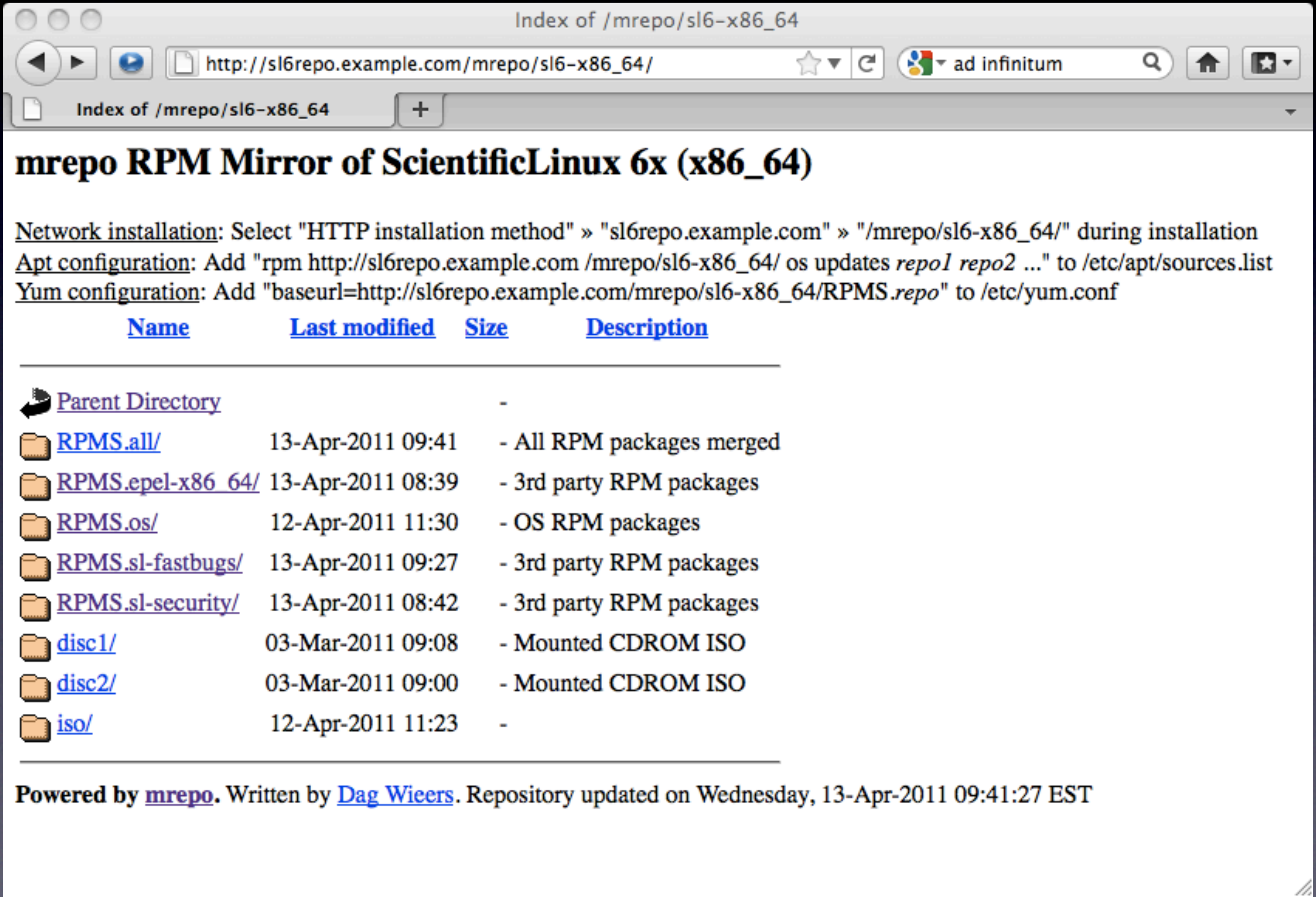
together at last

- to keep this up to date create a cronjob (e.g.: /etc/cron.d/reposync_epel-puppet):

```
#0 3 * * * root reposync -p /var/www/mrepo/reposync/ -a x86_64 -r epel-
puppet -n -q && createrepo /var/www/mrepo/reposync/epel-puppet > /dev/null
0 3 * * * root reposync -p /var/www/mrepo/reposync/ -a x86_64 -r epel-
puppet -n && createrepo /var/www/mrepo/reposync/epel-puppet
```

- quiet (hashed out) or verbose (active);
- reposync keeps all files it downloads (-d to age out files)
  - based in -p /var/www/mrepo/reposync ; and
  - creates -r epel-puppet
- createrepo acts on /var/www/mrepo/reposync/epel-puppet

Vendor and EPEL


Index of /mrepo/sl6-x86_64
http://sl6repo.example.com/mrepo/sl6-x86_64/
ad infinitum
Index of /mrepo/sl6-x86_64
+

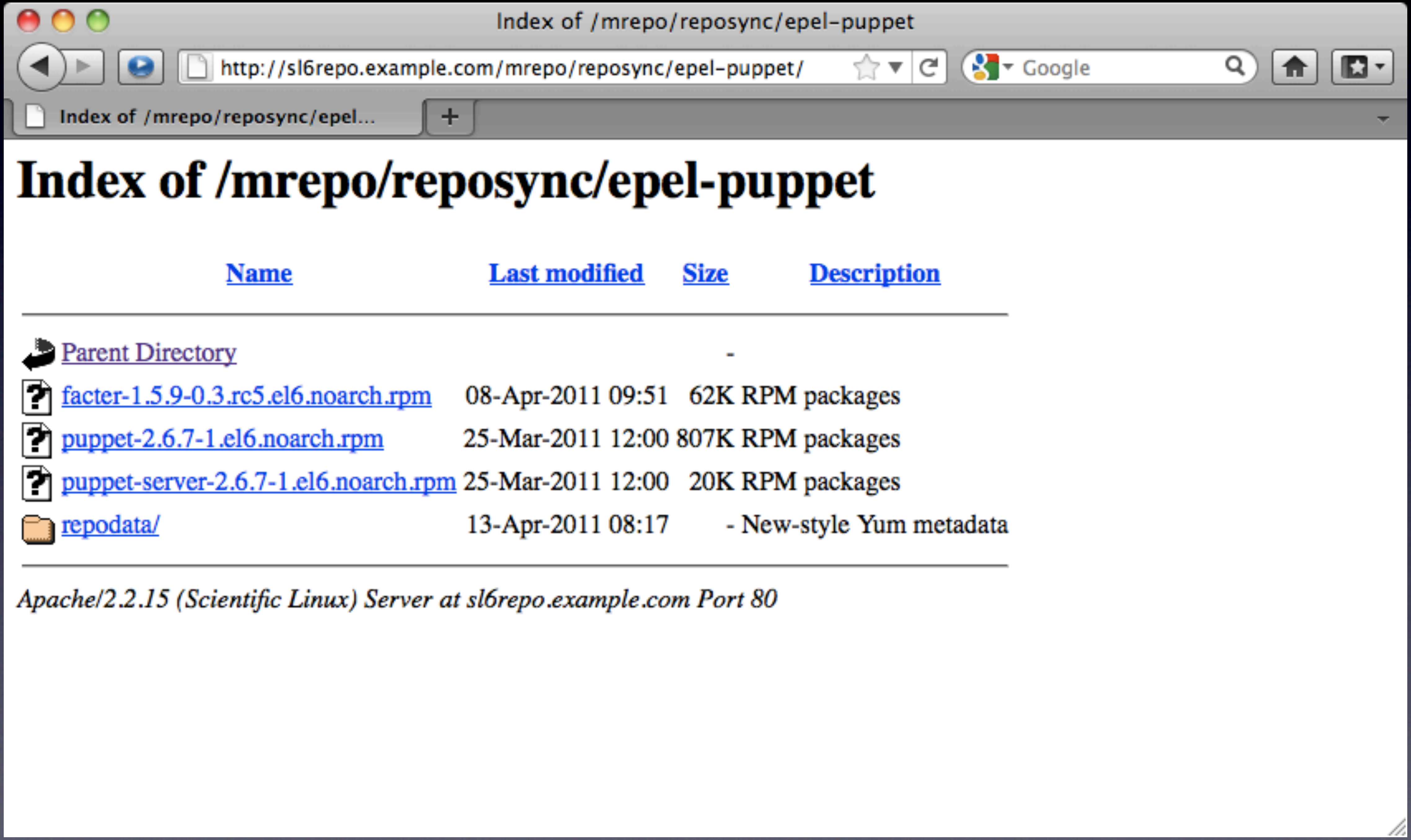
mrepo RPM Mirror of ScientificLinux 6x (x86_64)

Network installation: Select "HTTP installation method" » "sl6repo.example.com" » "/mrepo/sl6-x86_64/" during installation
Apt configuration: Add "rpm http://sl6repo.example.com /mrepo/sl6-x86_64/ os updates *repo1 repo2* ..." to /etc/apt/sources.list
Yum configuration: Add "baseurl=http://sl6repo.example.com/mrepo/sl6-x86_64/RPMS.*repo*" to /etc/yum.conf

Name	Last modified	Size	Description
Parent Directory		-	
RPMS.all/	13-Apr-2011 09:41	-	All RPM packages merged
RPMS.epel-x86_64/	13-Apr-2011 08:39	-	3rd party RPM packages
RPMS.os/	12-Apr-2011 11:30	-	OS RPM packages
RPMS.sl-fastbugs/	13-Apr-2011 09:27	-	3rd party RPM packages
RPMS.sl-security/	13-Apr-2011 08:42	-	3rd party RPM packages
disc1/	03-Mar-2011 09:08	-	Mounted CDROM ISO
disc2/	03-Mar-2011 09:00	-	Mounted CDROM ISO
iso/	12-Apr-2011 11:23	-	

Powered by mrepo. Written by Dag Wieers. Repository updated on Wednesday, 13-Apr-2011 09:41:27 EST

reposync & createrepo



Index of /mrepo/reposync/epel-puppet
http://sl6repo.example.com/mrepo/reposync/epel-puppet/
Google
Index of /mrepo/reposync/epel...
+

Index of /mrepo/reposync/epel-puppet

Name	Last modified	Size	Description
Parent Directory		-	
facter-1.5.9-0.3.rc5.el6.noarch.rpm	08-Apr-2011 09:51	62K	RPM packages
puppet-2.6.7-1.el6.noarch.rpm	25-Mar-2011 12:00	807K	RPM packages
puppet-server-2.6.7-1.el6.noarch.rpm	25-Mar-2011 12:00	20K	RPM packages
repodata/	13-Apr-2011 08:17	-	New-style Yum metadata

Apache/2.2.15 (Scientific Linux) Server at sl6repo.example.com Port 80

```
[local_sl_os_x86_64]
name=Scientific Linux 6 - x86_64
baseurl=http://sl6repo.example.com/mrepo/sl6-x86_64/RPMS.os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-sl file:///etc/pki/rpm-gpg/RPM-GPG-KEY-dawson

[local_sl-security_x86_64]
name=Scientific Linux 6 - x86_64 - security updates
baseurl=http://sl6repo.example.com/mrepo/sl6-x86_64/RPMS.sl-security/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-sl file:///etc/pki/rpm-gpg/RPM-GPG-KEY-dawson

[local_sl-fastbugs_x86_64]
name=Scientific Linux 6 - x86_64 - fastbug updates
baseurl=http://sl6repo.example.com/mrepo/sl6-x86_64/RPMS.sl-fastbugs/
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-sl file:///etc/pki/rpm-gpg/RPM-GPG-KEY-dawson

[local_epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
baseurl=http://sl6repo.example.com/mrepo/sl6-x86_64/RPMS.epel-x86_64/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

[local_epel-puppet]
name=Local EPEL puppet by TMZ
baseurl=http://sl6repo.example.com/mrepo/reposync/epel-puppet
enabled=1
gpgcheck=1
gpgkey=http://tmz.fedorapeople.org/repo/RPM-GPG-KEY-tmz
```

Final Thoughts

- gpg key - the repo file (previous slide) refers to a location on the client file system so it likely would be deployed via epel-release*.rpm
    - good to get updated keys;
    - bad if its repo files circumvent your local mirror.
        - but you could:
            - just clear the repo files; and
            - then make them immutable.

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;

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

- ~33MB 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;
- 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

kickstart file

```
### SL 6  #####
install
#url --url http://192.168.1.8/mrepo/rhel6-server-x86_64/
url --url http://192.168.1.8/mrepo/sl6-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
sl6puppetmaster.example.com --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
  - rootpw - make sure you change this once the system is booted.

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

partitioning & packages

- Do NOT make /boot a fancy filesystem;
- If you have more than one drive / RAID set, mention in clearpart, create a physical volume and volume group.
- Explicitly install packages either:
  - by group, e.g.:“@Core” ;
  - by name, e.g.:“openldap-servers”
  - exclude by prefacing a “-”, e.g.:“-arts”

```
bootloader --location=mbr --driveorder=sda
clearpart --all --drives=sda --initlabel
part /boot --fstype ext4 --size=128 --ondisk=sda
part pv.1 --size=100 --grow --ondisk=sda
volgroup VolGroup00 --pesize=32768 pv.1
logvol / --fstype ext4 --name=LogVol_root --vgname=VolGroup00 --size=1536
logvol /usr --fstype ext4 --name=LogVol_usr --vgname=VolGroup00 --size=3072
logvol /opt --fstype ext4 --name=LogVol_opt --vgname=VolGroup00 --size=2048
logvol /home --fstype ext4 --name=LogVol_home --vgname=VolGroup00 --size=512
logvol /tmp --fstype ext4 --name=LogVol_tmp --vgname=VolGroup00 --size=1024
logvol /var --fstype ext4 --name=LogVol_var --vgname=VolGroup00 --size=100 --grow

%packages

%end
```

%pre install

- Runs of the ISO - like the rescue environment;
- Most useful for workarounds:
  - Copy the custom RPMs you want to install, of 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;
  - off the ISO - like the rescue environment.

```
%post --nochroot

mkdir /mnt/sysimage/opt/sbin
mkdir /mnt/sysimage/mnt/dvd
mkdir /mnt/sysimage/mnt/nfs
mkdir /mnt/sysimage/mnt/samba
```

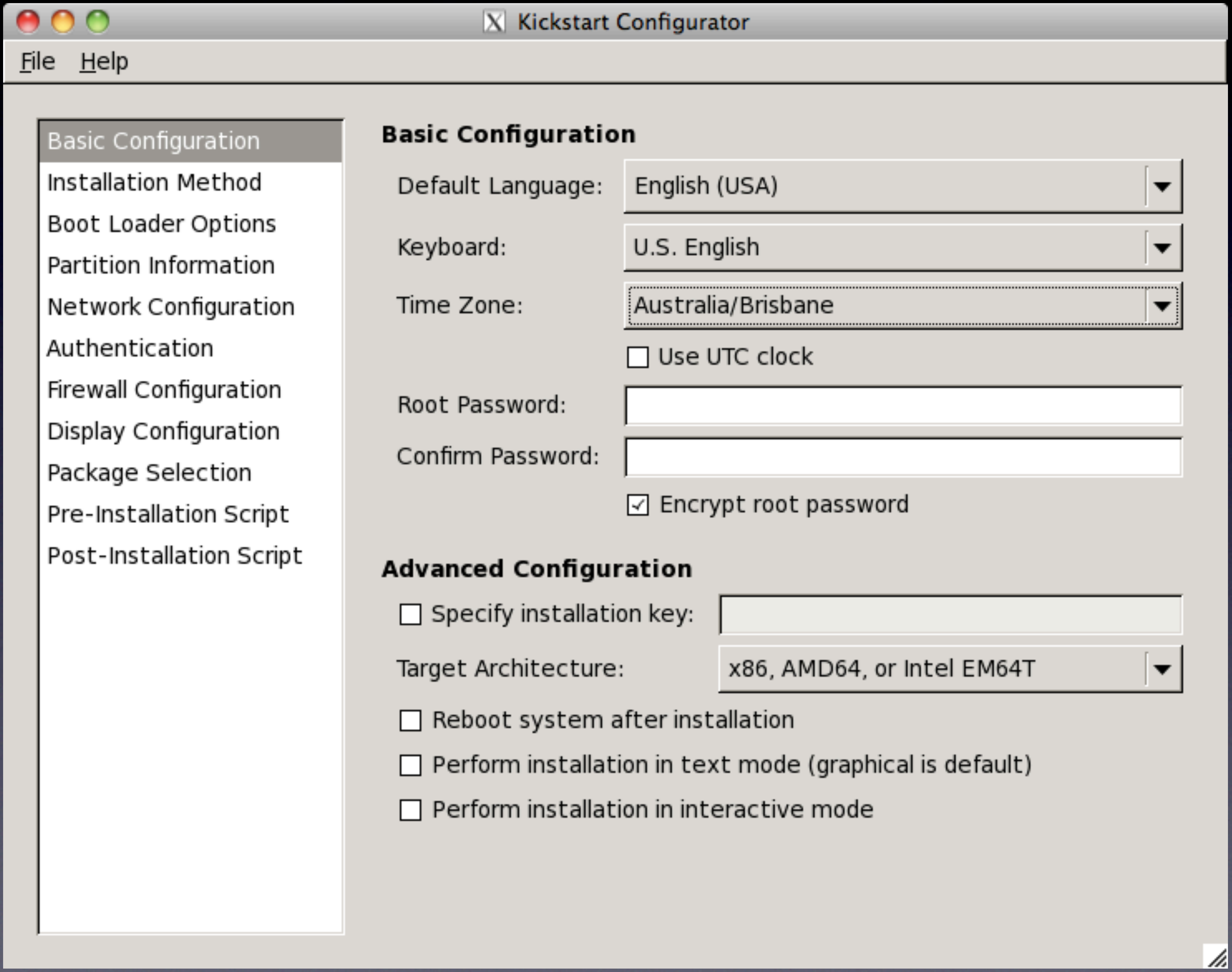
%post install chrooted

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

%post install chrooted

```
%post

rm -vf ` find / -name "TRANS.TBL" `

> /etc/yum.repos.d/epel.repo
> /etc/yum.repos.d/epel-testing.repo
> /etc/yum.repos.d/sl.repo
> /etc/yum.repos.d/sl-updates.repo
chattr +i /etc/yum.repos.d/epel*repo /etc/yum.repos.d/sl*repo

rpm -iv http://192.168.1.8/mrepo/sl6-x86_64/RPMS.epel-x86_64/epel-
release-6-5.noarch.rpm

wget http://192.168.1.8/local_repo/local.repo -O /etc/yum.repos.d/
local.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

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

There's a X11 tool for that



Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script

Basic Configuration

Default Language: English (USA)

Keyboard: U.S. English

Time Zone: Australia/Brisbane

□ Use UTC clock

Root Password:

Confirm Password:

☑ Encrypt root password

Advanced Configuration

□ Specify installation key:

Target Architecture: x86, AMD64, or Intel EM64T

□ Reboot system after installation

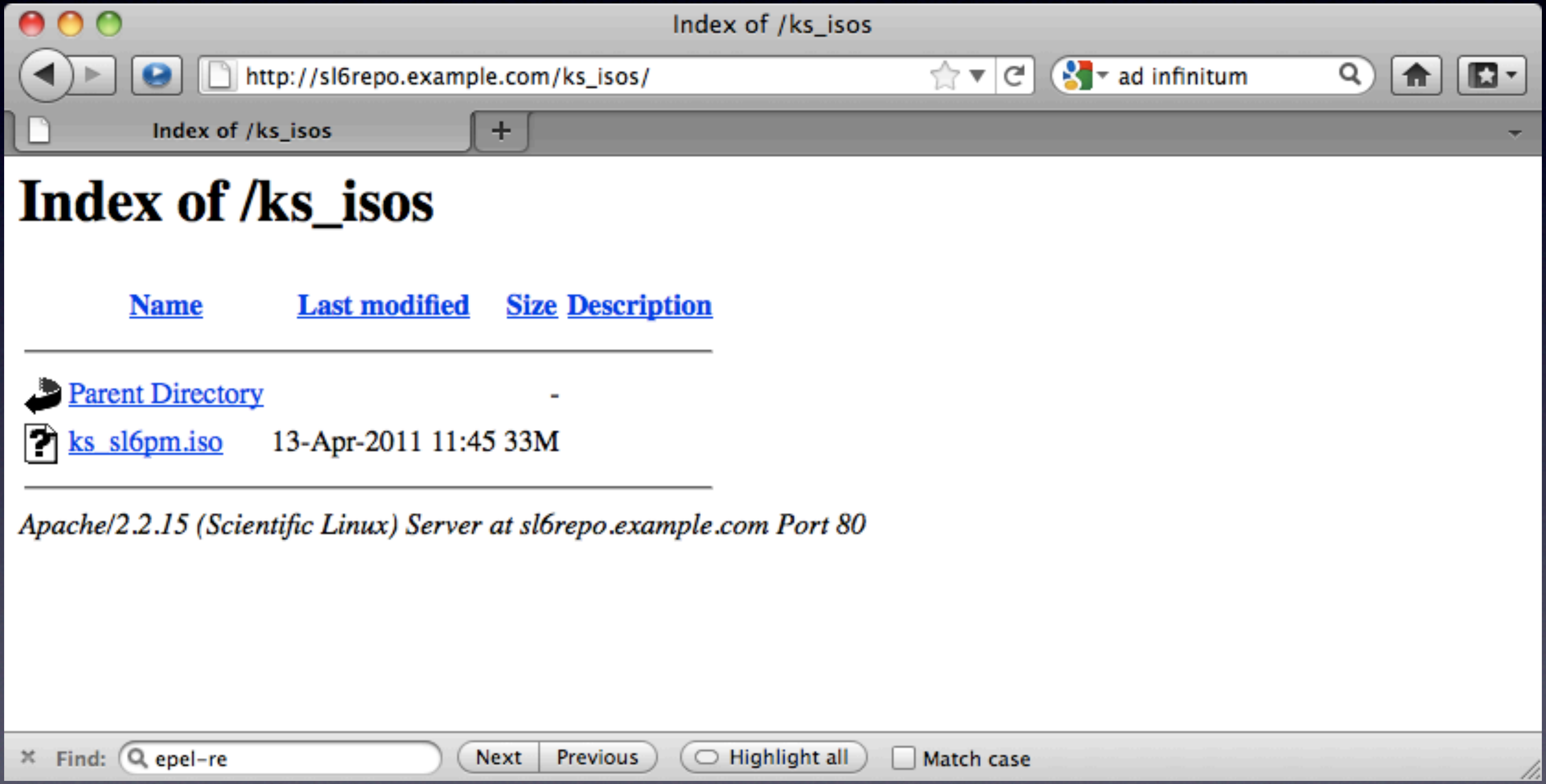
□ Perform installation in text mode (graphical is default)

□ Perform installation in interactive mode

Build the bootable ISO

```
[root@sl6repo ~]# mkdir kickstart
[root@sl6repo ~]# vi kickstart/ks.cfg
[root@sl6repo ~]# mount -o loop /var/mrepo/iso/SL-60-x86_64-2011-03-03-Everything-
DVD1.iso /mnt/
[root@sl6repo ~]# cp -r /mnt/isolinux ./kickstart/
[root@sl6repo ~]# echo -e "label custom\n  kernel vmlinuz\n  append ks=cdrom:/ks.cfg
initrd=initrd.img text" >> kickstart/isolinux/isolinux.cfg
[root@sl6repo ~]# sed -i 's:^default.*$:default custom:' kickstart/isolinux/isolinux.cfg
[root@sl6repo ~]# sed -i 's:^timeout.*$:timeout 5:' kickstart/isolinux/isolinux.cfg
[root@sl6repo ~]# mkisofs -r -N -allow-leading-dots -d -J -T -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -V "kickstart sl6puppetmaster" -boot-load-size 4 -boot-
info-table -o /var/www/html/ks_isos/ks_sl6pm.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
 29.63% done, estimate finish Wed Apr 13 11:45:58 2011
 59.27% done, estimate finish Wed Apr 13 11:45:58 2011
 88.82% done, estimate finish Wed Apr 13 11:45:58 2011
Total translation table size: 4701
Total rockridge attributes bytes: 1438
Total directory bytes: 2650
Path table size(bytes): 26
Max brk space used 0
16898 extents written (33 MB)
[root@sl6repo ~]#
```

Assuming you have a working httpd server



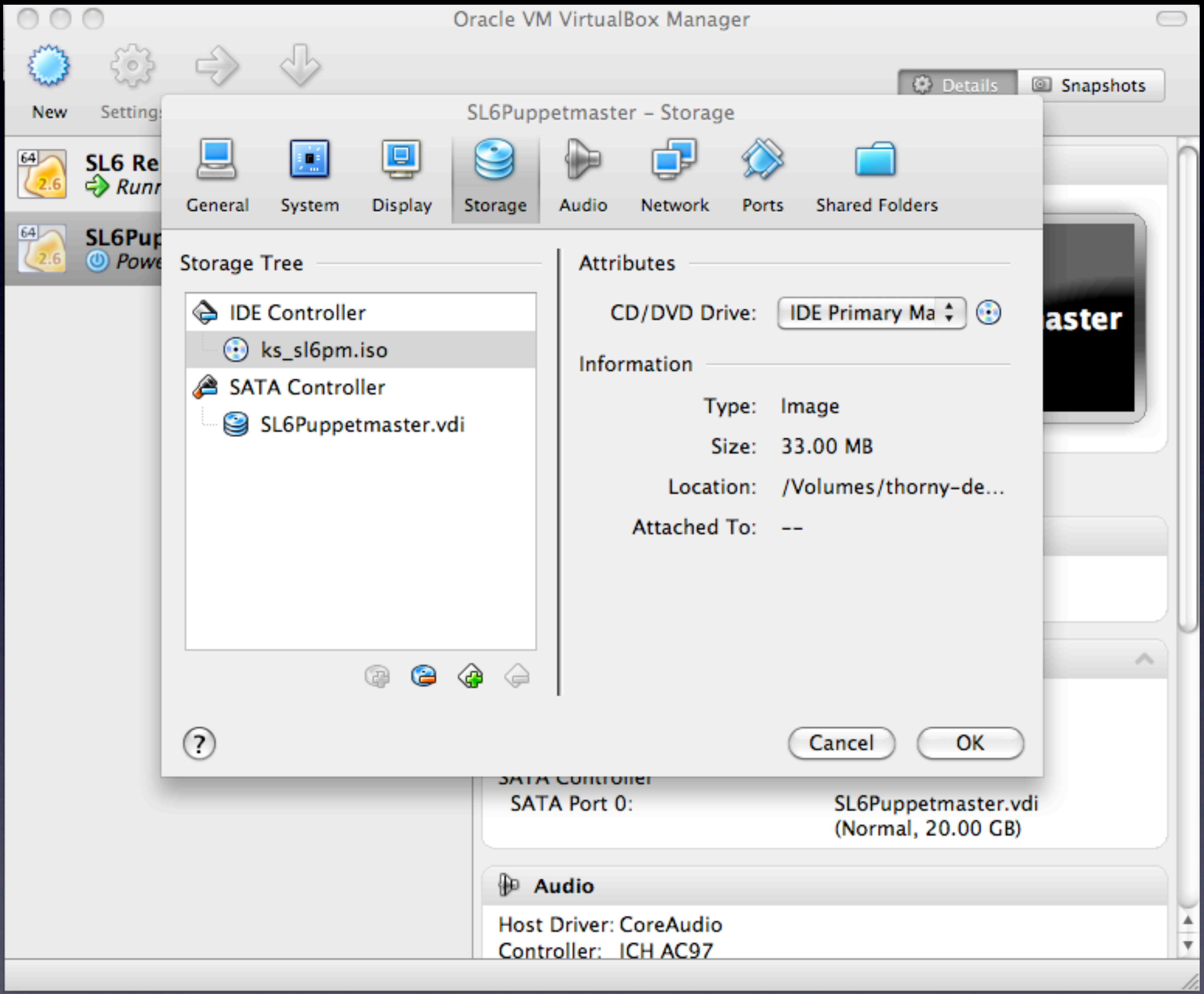
Index of /ks_isos

Name	Last modified	Size	Description
Parent Directory		-	
ks_sl6pm.iso	13-Apr-2011 11:45	33M	

Apache/2.2.15 (Scientific Linux) Server at sl6repo.example.com Port 80

Mount disk

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



Storage Tree

- IDE Controller
  - ks_sl6pm.iso
- SATA Controller
  - SL6Puppetmaster.vdi

Attributes

CD/DVD Drive: IDE Primary Ma

Information

Type: Image

Size: 33.00 MB

Location: /Volumes/thorny-de...

Attached To: --

SATA Controller

SATA Port 0: SL6Puppetmaster.vdi (Normal, 20.00 GB)

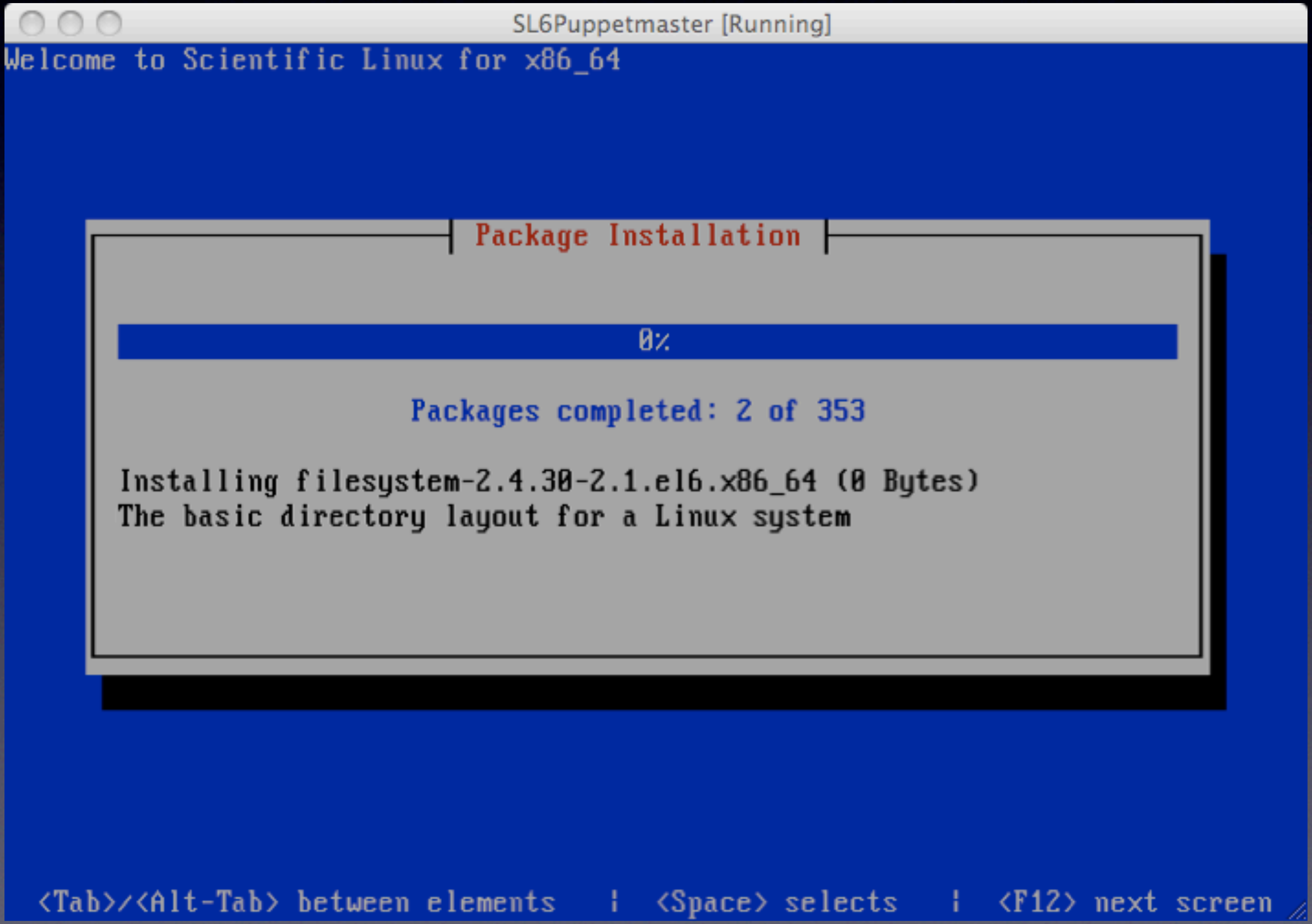
Host Driver: CoreAudio

Controller: ICH AC97

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

Install

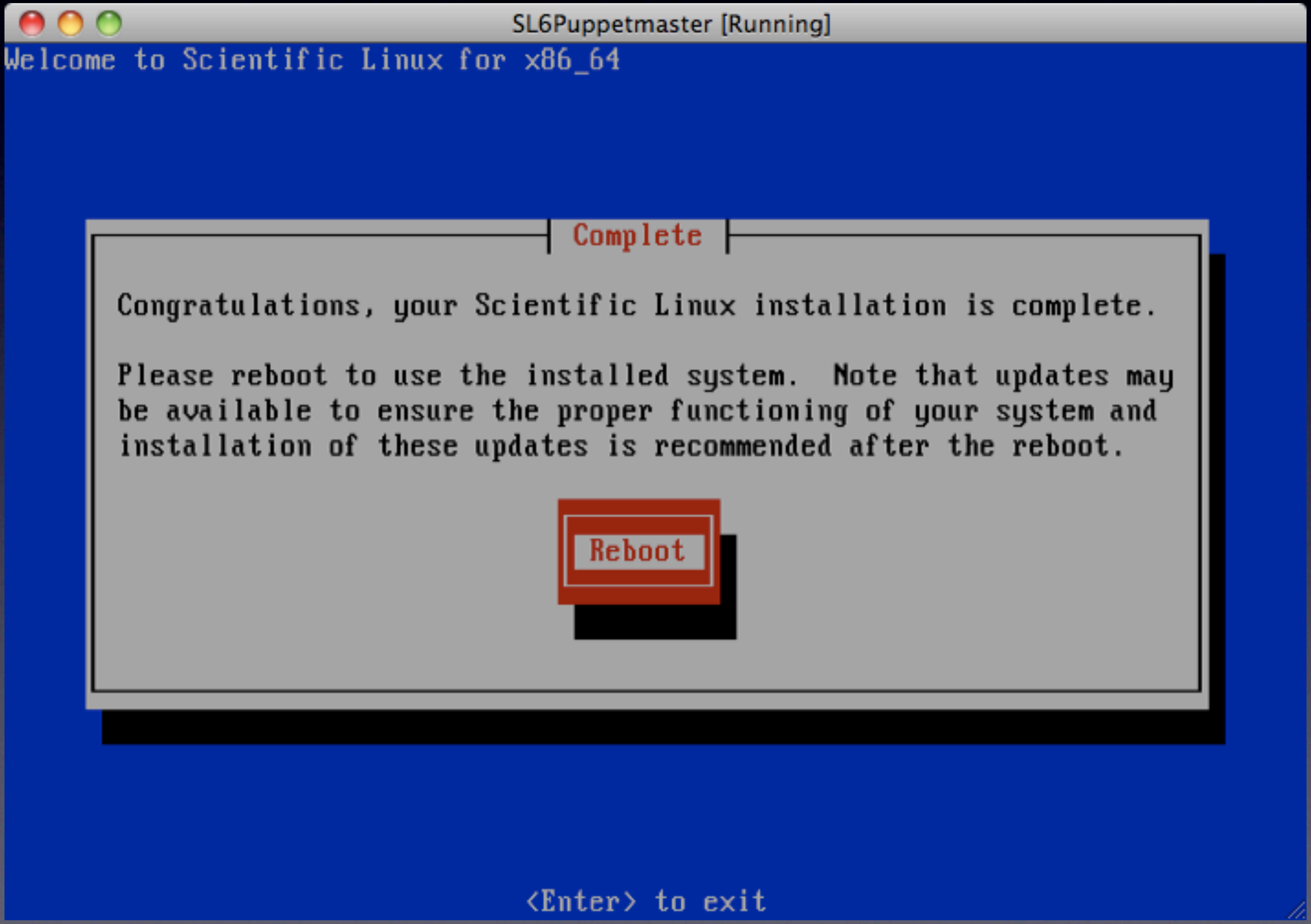
- You should not need to touch a thing.



```
Welcome to Scientific Linux for x86_64

                         Package Installation

                                  0%

                     Packages completed: 2 of 353

   Installing filesystem-2.4.30-2.1.el6.x86_64 (0 Bytes)
   The basic directory layout for a Linux system

  <Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

Reboot

- Make sure you unmount the ISO!!



Welcome to Scientific Linux for x86_64

Congratulations, your Scientific Linux installation is complete.

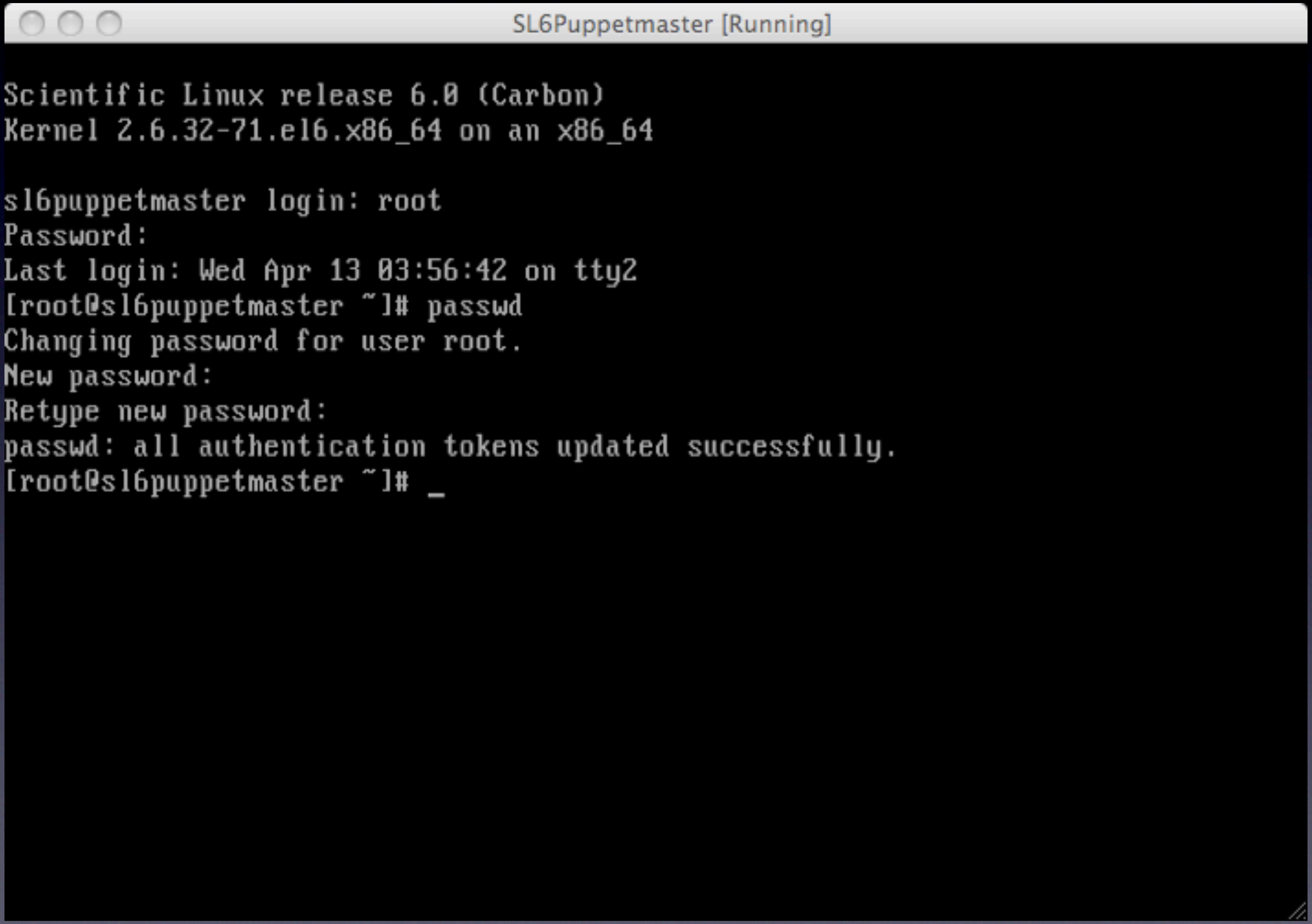
Please reboot to use the installed system. Note that updates may be available to ensure the proper functioning of your system and installation of these updates is recommended after the reboot.

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

First Boot



```
Scientific Linux release 6.0 (Carbon)
Kernel 2.6.32-71.el6.x86_64 on an x86_64

sl6puppetmaster login: root
Password:
Last login: Wed Apr 13 03:56:42 on tty2
[root@sl6puppetmaster ~]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@sl6puppetmaster ~]# _
```

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;
- Requires libselinux-ruby which is in the "RHEL Server Optional" add-on channel;

Create a module

- This module will be called “puppet_conf”
- 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

Example of a file type

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

$operatingsystem ?

$operatingsystem ?

```
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",
    }
}
```

Create a module

- Determine your modulepath:

```
[root@sl6puppetmaster ~]# puppet --configprint modulepath
/etc/puppet/modules:/usr/share/puppet/modules
```

- Create your module’s directory structure:

```
[root@s...r ~]# mkdir -p /etc/puppet/modules/puppet_conf
[root@s...r ~]# mkdir /etc/puppet/modules/puppet_conf/manifests
[root@s...r ~]# mkdir /etc/puppet/modules/puppet_conf/files
[root@s...r ~]# mkdir /etc/puppet/modules/puppet_conf/templates
```

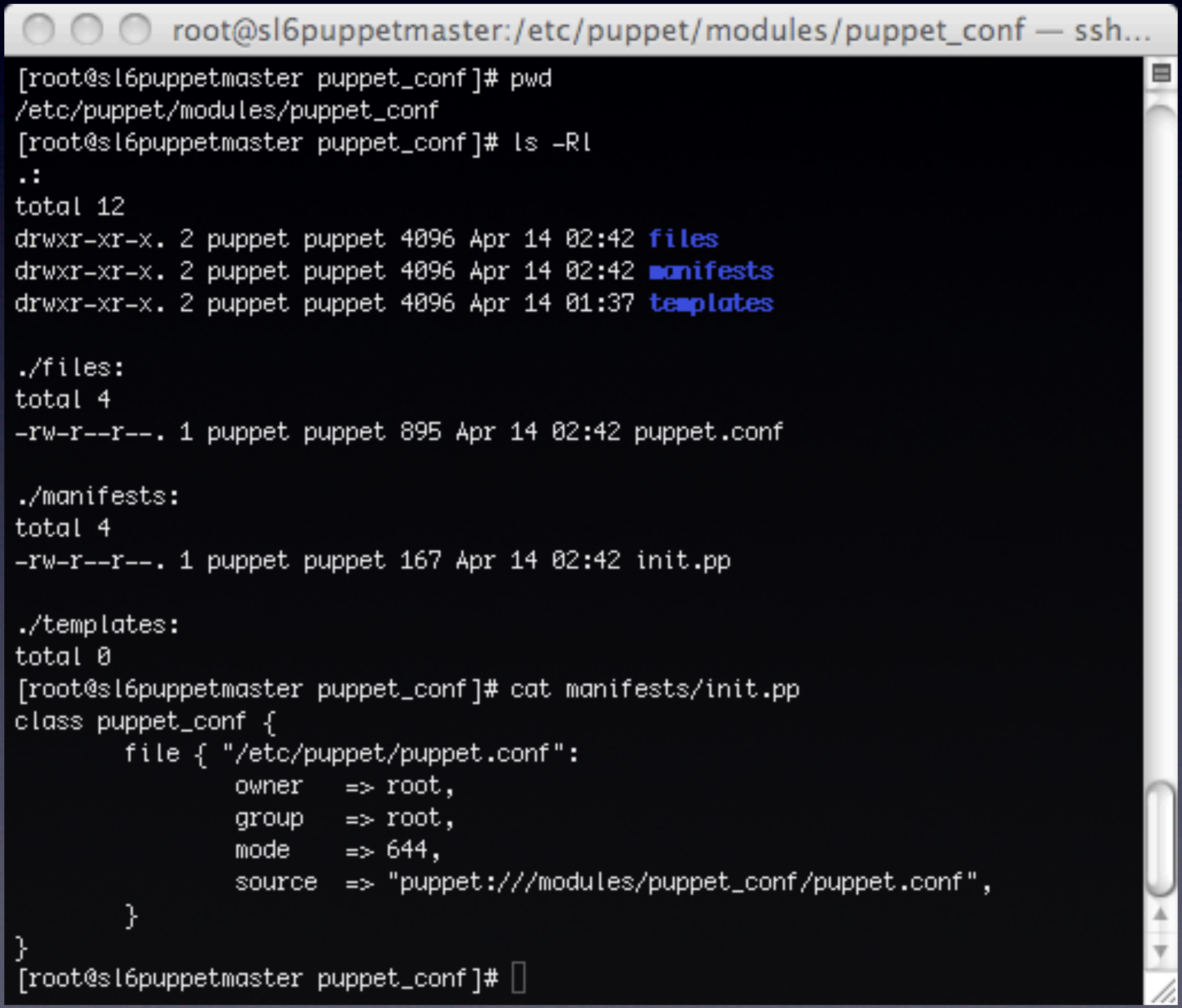
- Create your module’s init.pp:

```
[root@s...r ~]# vi /etc/puppet/modules/puppet_conf/manifests/init.pp
```

- ... and put in what’s on the previous slide.

One more thing...

- make the module and contents owned by puppet:puppet


root@sl6puppetmaster:/etc/puppet/modules/puppet_conf — ssh...

```
[root@sl6puppetmaster puppet_conf]# pwd
/etc/puppet/modules/puppet_conf
[root@sl6puppetmaster puppet_conf]# ls -Rl
.:
total 12
drwxr-xr-x. 2 puppet puppet 4096 Apr 14 02:42 files
drwxr-xr-x. 2 puppet puppet 4096 Apr 14 02:42 manifests
drwxr-xr-x. 2 puppet puppet 4096 Apr 14 01:37 templates

./files:
total 4
-rw-r--r--. 1 puppet puppet 895 Apr 14 02:42 puppet.conf

./manifests:
total 4
-rw-r--r--. 1 puppet puppet 167 Apr 14 02:42 init.pp

./templates:
total 0
[root@sl6puppetmaster puppet_conf]# cat manifests/init.pp
class puppet_conf {
        file { "/etc/puppet/puppet.conf":
                owner   => root,
                group   => root,
                mode    => 644,
                source  => "puppet:///modules/puppet_conf/puppet.conf",
        }
}
[root@sl6puppetmaster puppet_conf]# 
```

puppet.conf

- straight copy from your default rpm provided server configuration, with the addition of:
  - server = sl6puppetmaster.example.com
- at the bottom of the file in the [agent] section.

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.

Remaining configuration

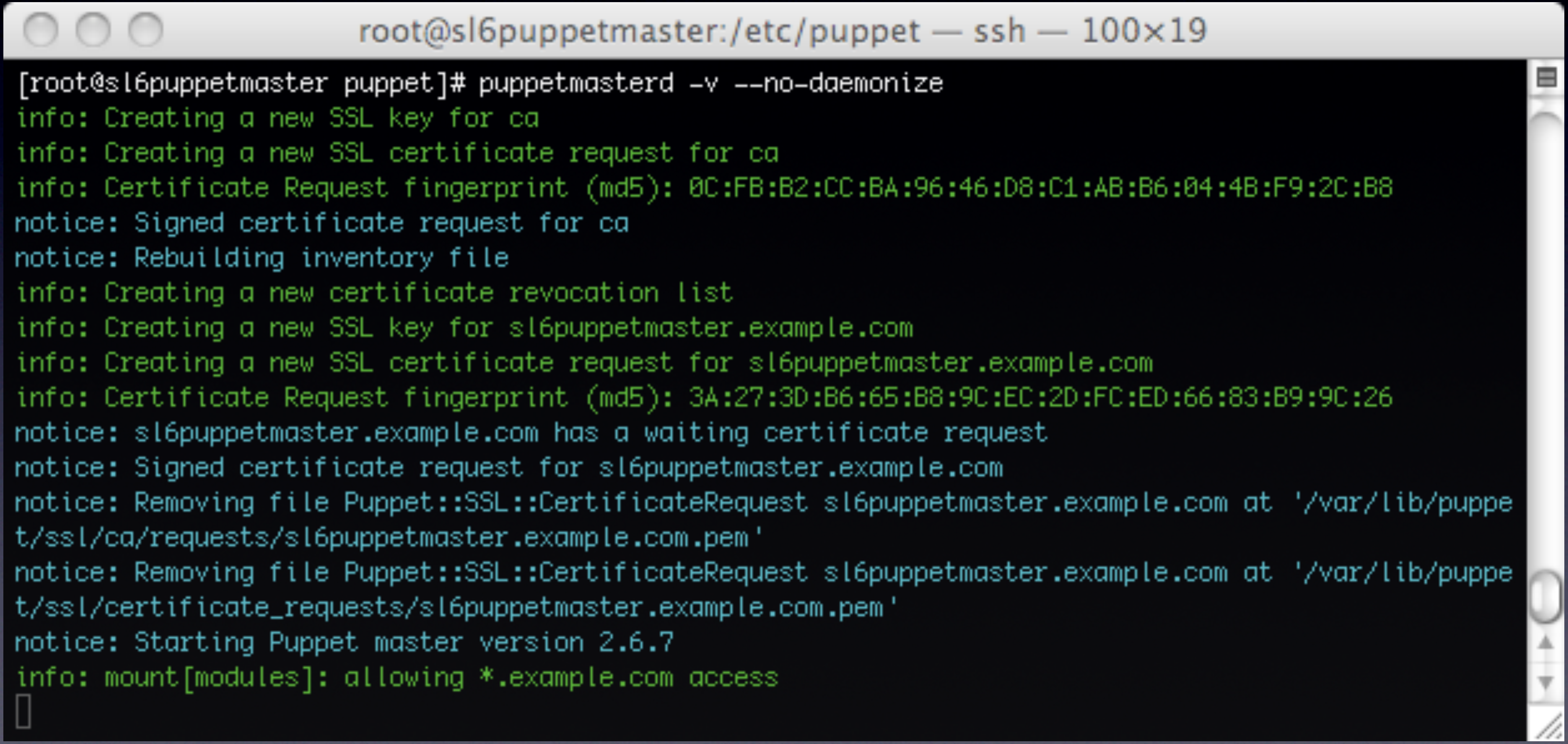
- /etc/puppet/fileserver.conf - allow everyone to modules:

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

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

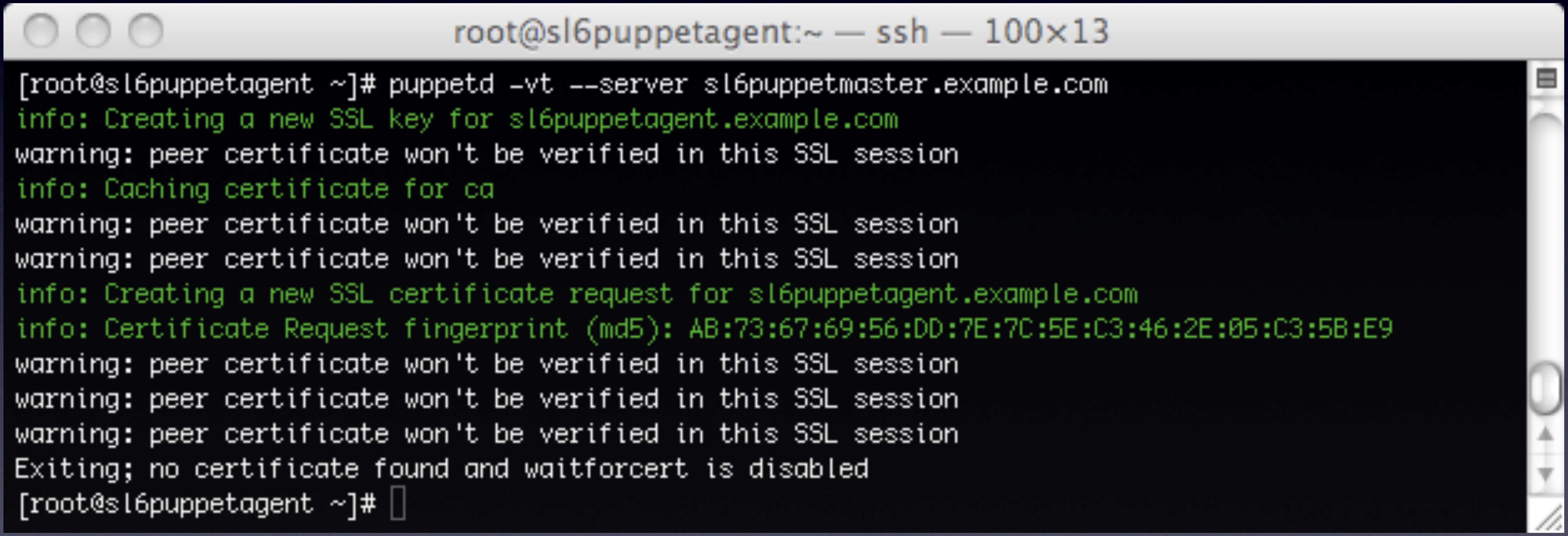
```
node default {
        include puppet_conf
}
```

puppetmasterd starts



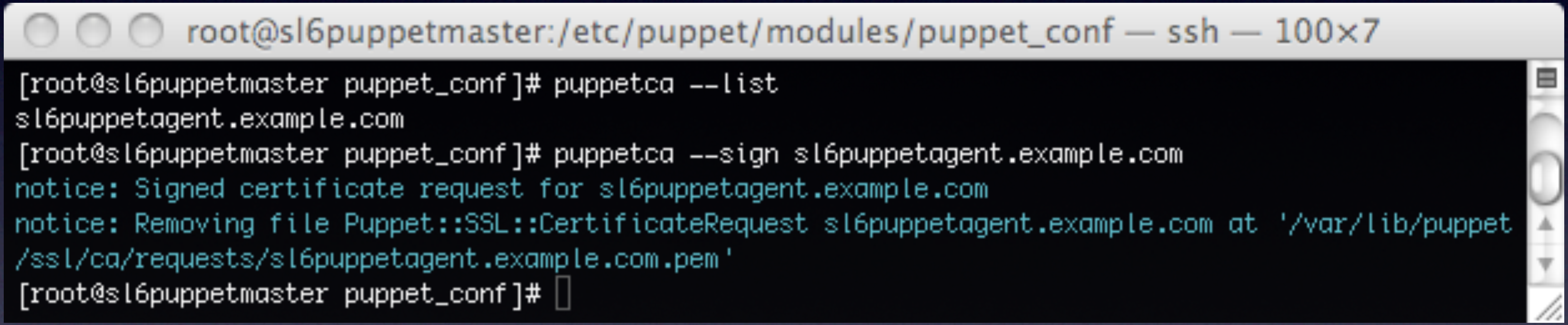
```
[root@sl6puppetmaster puppet]# puppetmasterd -v --no-daemonize
info: Creating a new SSL key for ca
info: Creating a new SSL certificate request for ca
info: Certificate Request fingerprint (md5): 0C:FB:B2:CC:BA:96:46:D8:C1:AB:B6:04:4B:F9:2C:B8
notice: Signed certificate request for ca
notice: Rebuilding inventory file
info: Creating a new certificate revocation list
info: Creating a new SSL key for sl6puppetmaster.example.com
info: Creating a new SSL certificate request for sl6puppetmaster.example.com
info: Certificate Request fingerprint (md5): 3A:27:3D:B6:65:B8:9C:EC:2D:FC:ED:66:83:B9:9C:26
notice: sl6puppetmaster.example.com has a waiting certificate request
notice: Signed certificate request for sl6puppetmaster.example.com
notice: Removing file Puppet::SSL::CertificateRequest sl6puppetmaster.example.com at '/var/lib/puppe
t/ssl/ca/requests/sl6puppetmaster.example.com.pem'
notice: Removing file Puppet::SSL::CertificateRequest sl6puppetmaster.example.com at '/var/lib/puppe
t/ssl/certificate_requests/sl6puppetmaster.example.com.pem'
notice: Starting Puppet master version 2.6.7
info: mount[modules]: allowing *.example.com access
```

Then your client connects



```
[root@sl6puppetagent ~]# puppetd -vt --server sl6puppetmaster.example.com
info: Creating a new SSL key for sl6puppetagent.example.com
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 sl6puppetagent.example.com
info: Certificate Request fingerprint (md5): AB:73:67:69:56:DD:7E:7C:5E:C3:46:2E:05:C3:5B:E9
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
[root@sl6puppetagent ~]# 
```

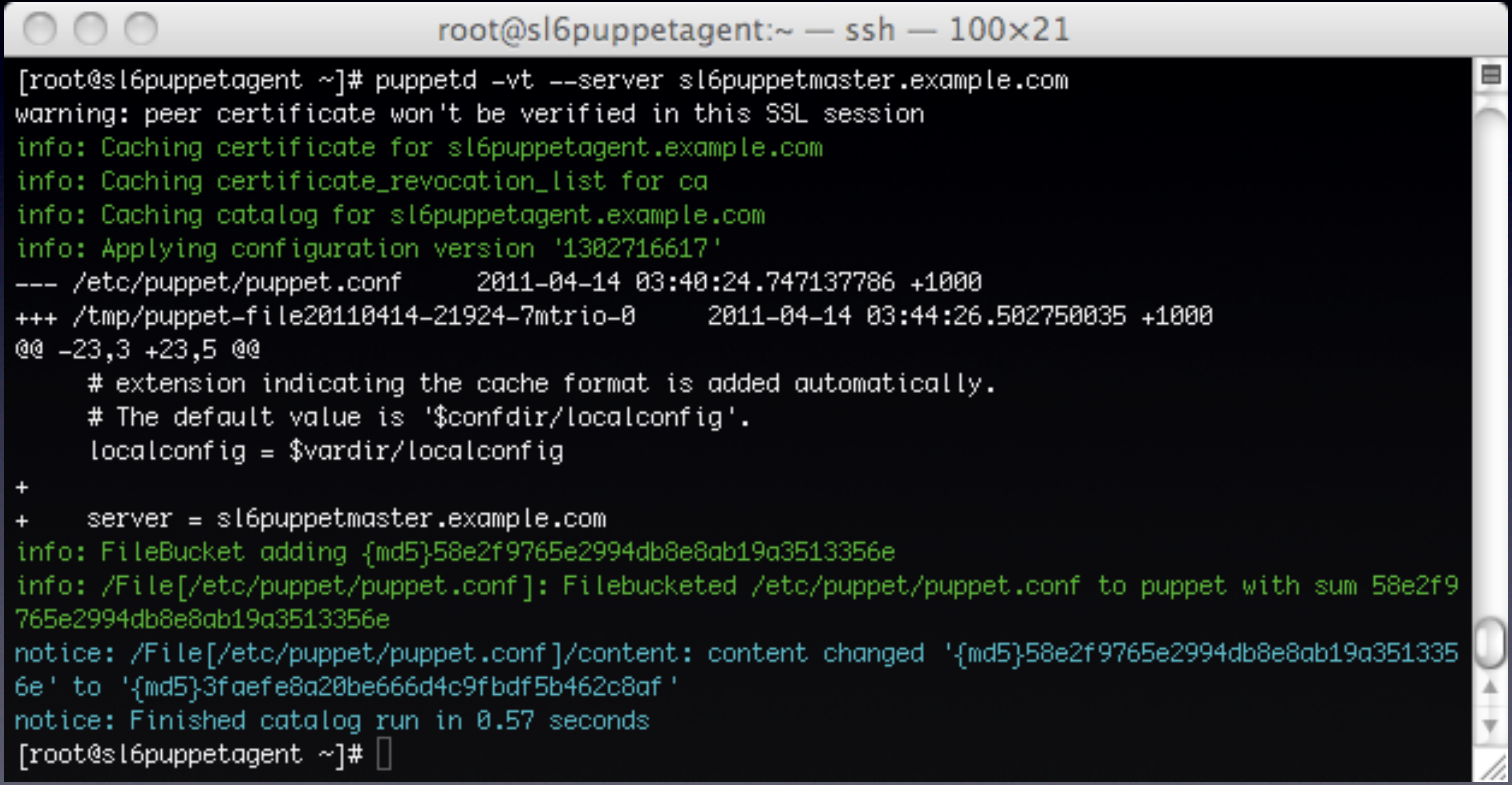
You sign the client



root@sl6puppetmaster:/etc/puppet/modules/puppet_conf — ssh — 100×7

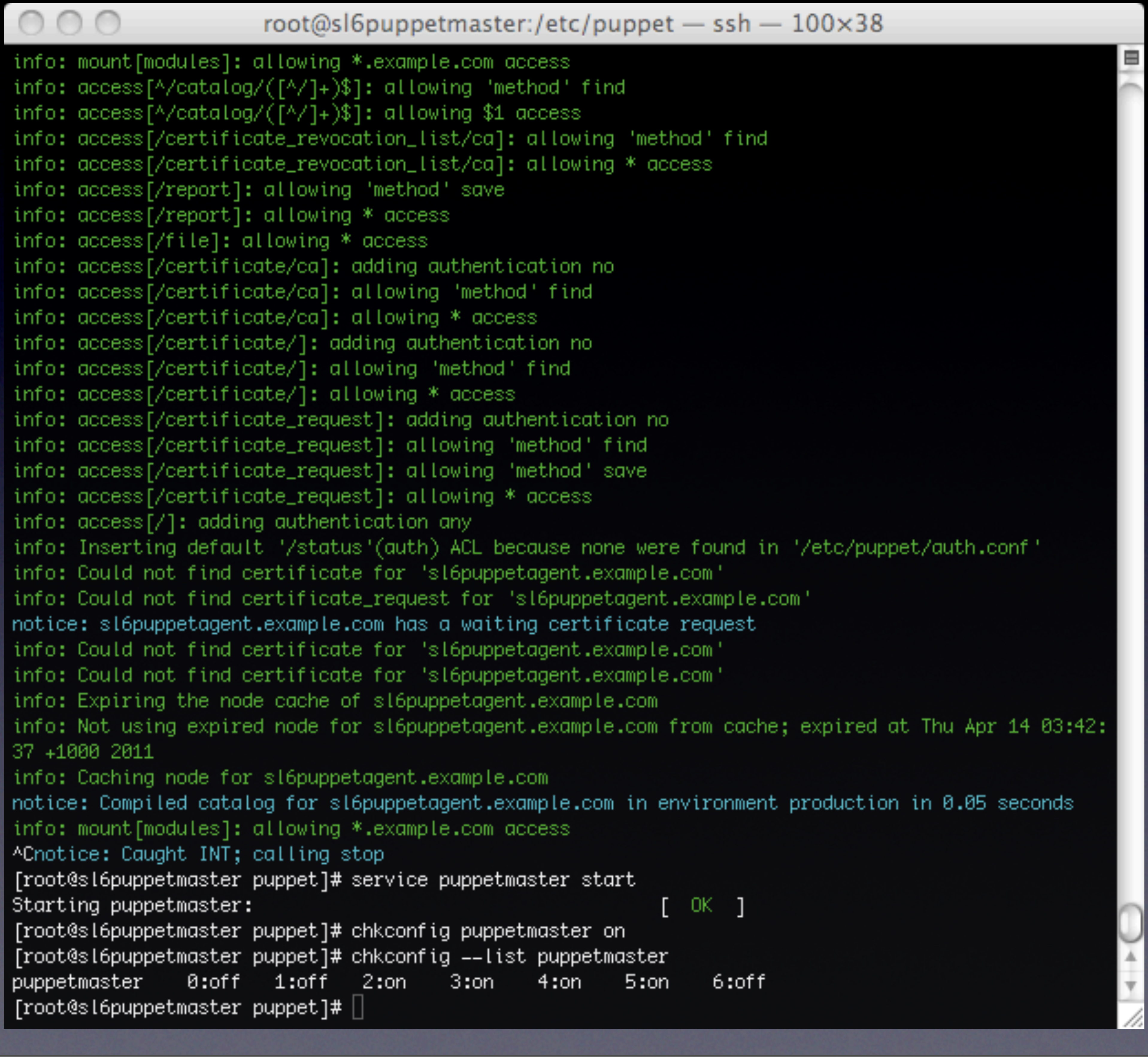
```
[root@sl6puppetmaster puppet_conf]# puppetca --list
sl6puppetagent.example.com
[root@sl6puppetmaster puppet_conf]# puppetca --sign sl6puppetagent.example.com
notice: Signed certificate request for sl6puppetagent.example.com
notice: Removing file Puppet::SSL::CertificateRequest sl6puppetagent.example.com at '/var/lib/puppet
/ssl/ca/requests/sl6puppetagent.example.com.pem'
[root@sl6puppetmaster puppet_conf]# 
```

re-run the client



```
[root@sl6puppetagent ~]# puppetd -vt --server sl6puppetmaster.example.com
warning: peer certificate won't be verified in this SSL session
info: Caching certificate for sl6puppetagent.example.com
info: Caching certificate_revocation_list for ca
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302716617'
--- /etc/puppet/puppet.conf     2011-04-14 03:40:24.747137786 +1000
+++ /tmp/puppet-file20110414-21924-7mtrio-0     2011-04-14 03:44:26.502750035 +1000
@@ -23,3 +23,5 @@
     # extension indicating the cache format is added automatically.
     # The default value is '$confdir/localconfig'.
     localconfig = $vardir/localconfig
+
+    server = sl6puppetmaster.example.com
info: FileBucket adding {md5}58e2f9765e2994db8e8ab19a3513356e
info: /File[/etc/puppet/puppet.conf]: Filebucketed /etc/puppet/puppet.conf to puppet with sum 58e2f9
765e2994db8e8ab19a3513356e
notice: /File[/etc/puppet/puppet.conf]/content: content changed '{md5}58e2f9765e2994db8e8ab19a351335
6e' to '{md5}3faefe8a20be666d4c9fbdf5b462c8af'
notice: Finished catalog run in 0.57 seconds
[root@sl6puppetagent ~]# 
```

You see that it is good



```
info: mount[modules]: allowing *.example.com access
info: access[^/catalog/([^/]+)$]: allowing 'method' find
info: access[^/catalog/([^/]+)$]: allowing $1 access
info: access[/certificate_revocation_list/ca]: allowing 'method' find
info: access[/certificate_revocation_list/ca]: allowing * access
info: access[/report]: allowing 'method' save
info: access[/report]: allowing * access
info: access[/file]: allowing * access
info: access[/certificate/ca]: adding authentication no
info: access[/certificate/ca]: allowing 'method' find
info: access[/certificate/ca]: allowing * access
info: access[/certificate/]: adding authentication no
info: access[/certificate/]: allowing 'method' find
info: access[/certificate/]: allowing * access
info: access[/certificate_request]: adding authentication no
info: access[/certificate_request]: allowing 'method' find
info: access[/certificate_request]: allowing 'method' save
info: access[/certificate_request]: allowing * access
info: access[/]: adding authentication any
info: Inserting default '/status'(auth) ACL because none were found in '/etc/puppet/auth.conf'
info: Could not find certificate for 'sl6puppetagent.example.com'
info: Could not find certificate_request for 'sl6puppetagent.example.com'
notice: sl6puppetagent.example.com has a waiting certificate request
info: Could not find certificate for 'sl6puppetagent.example.com'
info: Could not find certificate for 'sl6puppetagent.example.com'
info: Expiring the node cache of sl6puppetagent.example.com
info: Not using expired node for sl6puppetagent.example.com from cache; expired at Thu Apr 14 03:42:
37 +1000 2011
info: Caching node for sl6puppetagent.example.com
notice: Compiled catalog for sl6puppetagent.example.com in environment production in 0.05 seconds
info: mount[modules]: allowing *.example.com access
^Cnotice: Caught INT; calling stop
[root@sl6puppetmaster puppet]# service puppetmaster start
Starting puppetmaster:                                     [  OK  ]
[root@sl6puppetmaster puppet]# chkconfig puppetmaster on
[root@sl6puppetmaster puppet]# chkconfig --list puppetmaster
puppetmaster    0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@sl6puppetmaster puppet]# 
```

If it’s not good

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.

Summary so far

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
```

- don’t forget to:
  - chown the module; and
  - include sshd_config in site.pp

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@sl6puppetagent ~]# puppetd -vt
err: Could not retrieve catalog from remote server: Error 400 on
SERVER: Could not find class sysAdmins at /etc/puppet/manifests/
site.pp:4 on node sl6puppetagent.example.com
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@sl6puppetagent ~]# egrep "chakkerz|foo|sysadmins" /etc/{passwd,shadow,group}
[root@sl6puppetagent ~]# ls -l /home
total 16
drwx------. 2 root root 16384 Apr 14 04:28 lost+found
[root@sl6puppetagent ~]#
```

... and after on Linux

```
[root@sl6puppetagent ~]# 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.:15077:0:99999:7:::
/etc/shadow:foo:$1$m16B30$AYeyT/XyRpEHmEym7fDmK/:15077:0:99999:7:::
/etc/group:sysadmins:x:1000:
[root@sl6puppetagent ~]# ls -l /home
total 24
drwx------. 2 chakkerz sysadmins  4096 Apr 14 07:53 chakkerz
drwx------. 2 foo      sysadmins  4096 Apr 14 07:53 foo
drwx------. 2 root     root      16384 Apr 14 04:28 lost+found
[root@sl6puppetagent ~]#
```

... 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

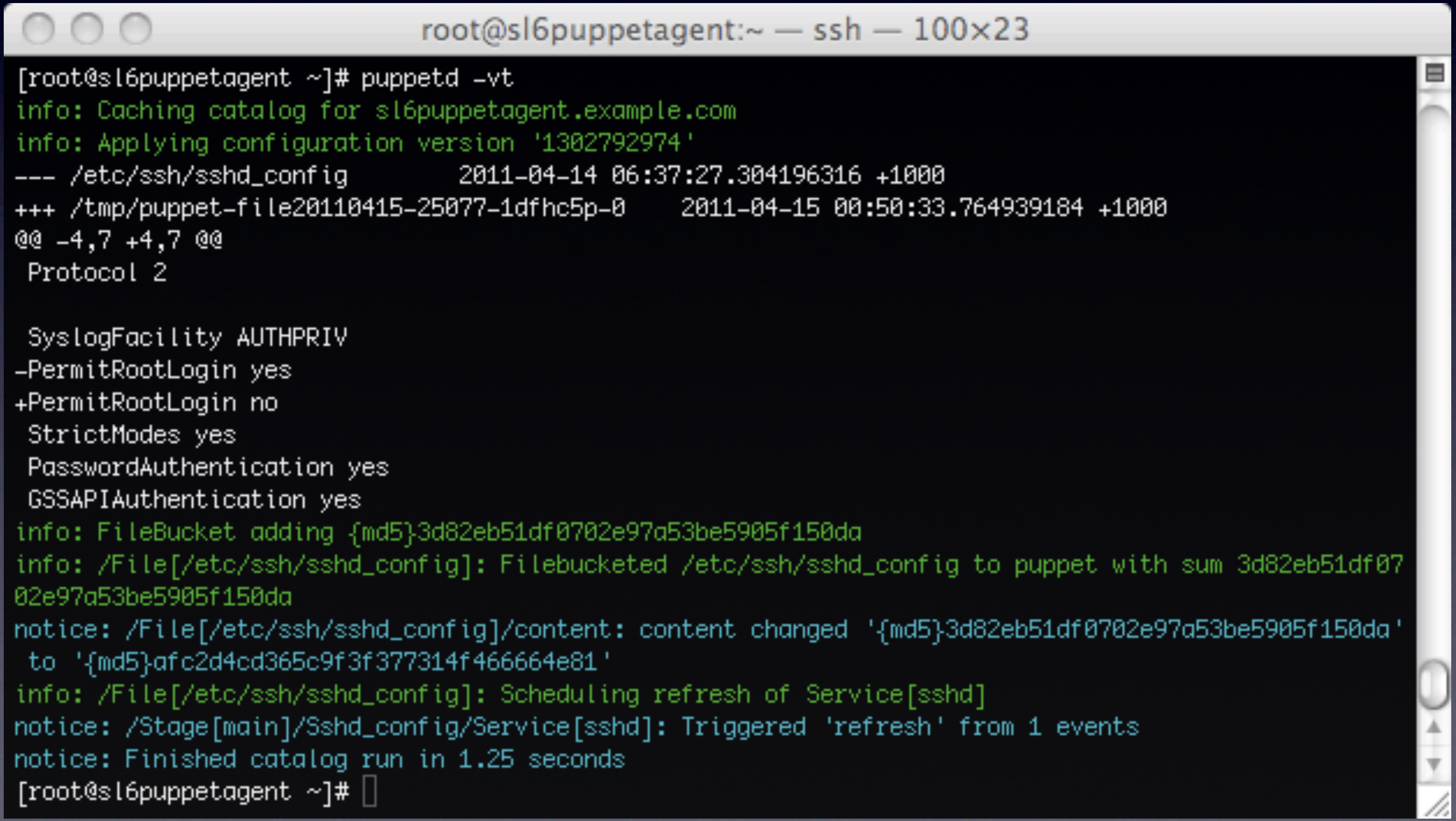
```
class sshd_config
{
    if ($operatingsystem == darwin) {
        $sshd_file_path = "/etc/sshd_config"
        $sshd_service  = "com.openssh.sshd"
    }
    else {
        $sshd_file_path   = "/etc/ssh/sshd_config"
        $sshd_service  = "sshd"
    }

    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@sl6puppetagent:~ — ssh — 100×23

```
[root@sl6puppetagent ~]# puppetd -vt
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302792974'
--- /etc/ssh/sshd_config        2011-04-14 06:37:27.304196316 +1000
+++ /tmp/puppet-file20110415-25077-1dfhc5p-0    2011-04-15 00:50:33.764939184 +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[/etc/ssh/sshd_config]: Filebucketed /etc/ssh/sshd_config to puppet with sum 3d82eb51df07
02e97a53be5905f150da
notice: /File[/etc/ssh/sshd_config]/content: content changed '{md5}3d82eb51df0702e97a53be5905f150da'
 to '{md5}afc2d4cd365c9f3f377314f466664e81'
info: /File[/etc/ssh/sshd_config]: Scheduling refresh of Service[sshd]
notice: /Stage[main]/Sshd_config/Service[sshd]: Triggered 'refresh' from 1 events
notice: Finished catalog run in 1.25 seconds
[root@sl6puppetagent ~]# 
```

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@sl6puppetagent ~]# grep baz /etc/passwd
baz:x:498:496:Baz Contrived:/home/baz:/bin/bash
[root@sl6puppetagent ~]# 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

package type

```
class packages {
   package  { "nano":           ensure   => absent,   }

   package  { "elinks":         ensure   => installed,   }
   package  { "telnet":         ensure   => installed,   }
}
```

- results in:

```
[root@sl6puppetagent state]# rpm -q nano elinks telnet
nano-2.0.9-7.el6.x86_64
package elinks is not installed
package telnet is not installed
[root@sl6puppetagent state]# puppetd -vt 2>&1 1> /dev/null
[root@sl6puppetagent state]# rpm -q nano elinks telnet
package nano is not installed
elinks-0.12-0.20.pre5.el6.x86_64
telnet-0.17-46.el6.x86_64
```

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

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,
    }
}
```

exec type result SL

```
[root@sl6puppetagent ~]# ls /tmp/puppet*
ls: cannot access /tmp/puppet*: No such file or directory
[root@sl6puppetagent ~]# puppetd -vt
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302785098'
notice: /Stage[main]/Execute/Exec[touch a file just once]/returns: executed
successfully
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns:
executed successfully
notice: Finished catalog run in 1.57 seconds
[root@sl6puppetagent ~]# ls -l /tmp/puppet*
-rw-r--r--. 1 root root 7570 Apr 14 22:39 /tmp/puppet.top
-rw-r--r--. 1 root root    0 Apr 14 22:39 /tmp/puppet.touch.once
[root@sl6puppetagent ~]# puppetd -vt
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302785098'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns:
executed successfully
notice: Finished catalog run in 1.64 seconds
[root@sl6puppetagent ~]# ls -l /tmp/puppet*
-rw-r--r--. 1 root root 15140 Apr 14 22:41 /tmp/puppet.top
-rw-r--r--. 1 root root     0 Apr 14 22:39 /tmp/puppet.touch.once
[root@sl6puppetagent ~]#
```

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.com
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

Summary so far

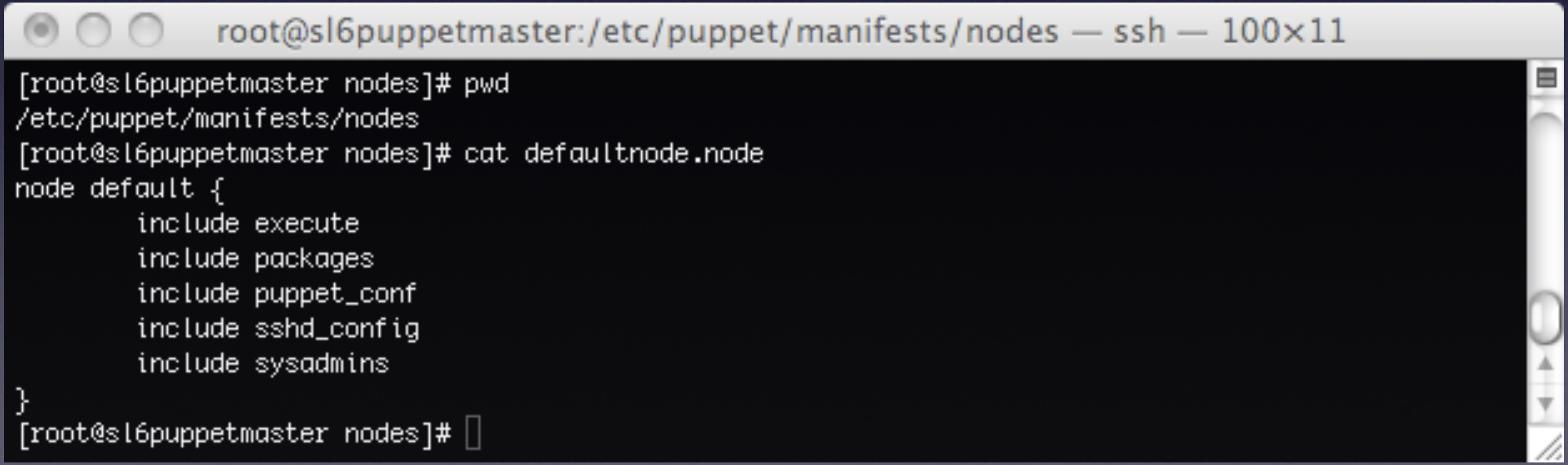
- Resource types:
  - files, directories and templates;
  - users and groups;
  - package and exec;
- Ordering;
- Coming up with strange puppet examples.

nodes

- You need this to customise specific hosts;
- Setting this up the first time feels buggy and the syntax strikes me as counter intuitive;
- This will also cover inheritance.

nodes - step 1

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

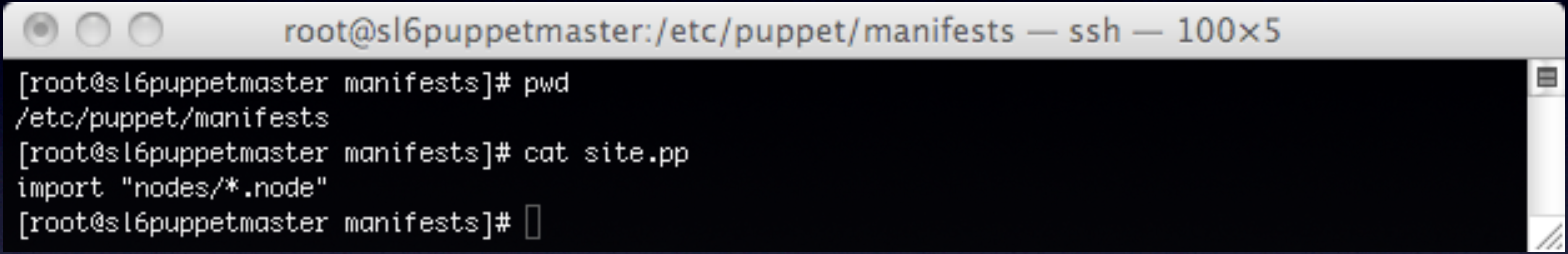


root@sl6puppetmaster:/etc/puppet/manifests/nodes — ssh — 100×11

```
[root@sl6puppetmaster nodes]# pwd
/etc/puppet/manifests/nodes
[root@sl6puppetmaster nodes]# cat defaultnode.node
node default {
        include execute
        include packages
        include puppet_conf
        include sshd_config
        include sysadmins
}
[root@sl6puppetmaster nodes]# 
```

nodes - step 2

- create a new site.pp:



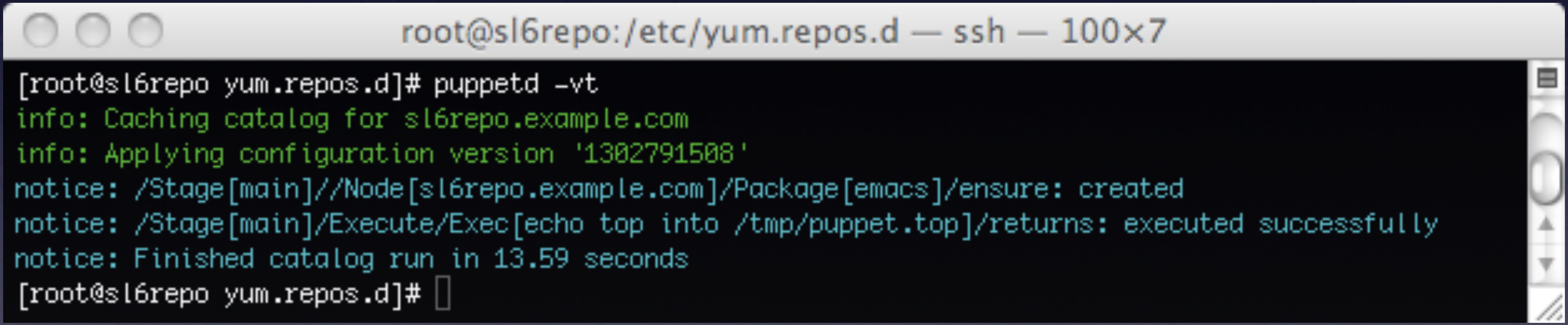
```
[root@sl6puppetmaster manifests]# pwd
/etc/puppet/manifests
[root@sl6puppetmaster manifests]# cat site.pp
import "nodes/*.node"
[root@sl6puppetmaster manifests]# 
```

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

nodes - step 3

- create nodes/sl6repo.node

```
node "sl6repo.example.com" inherits default {
    package { "emacs":        ensure => installed,    }
}
```

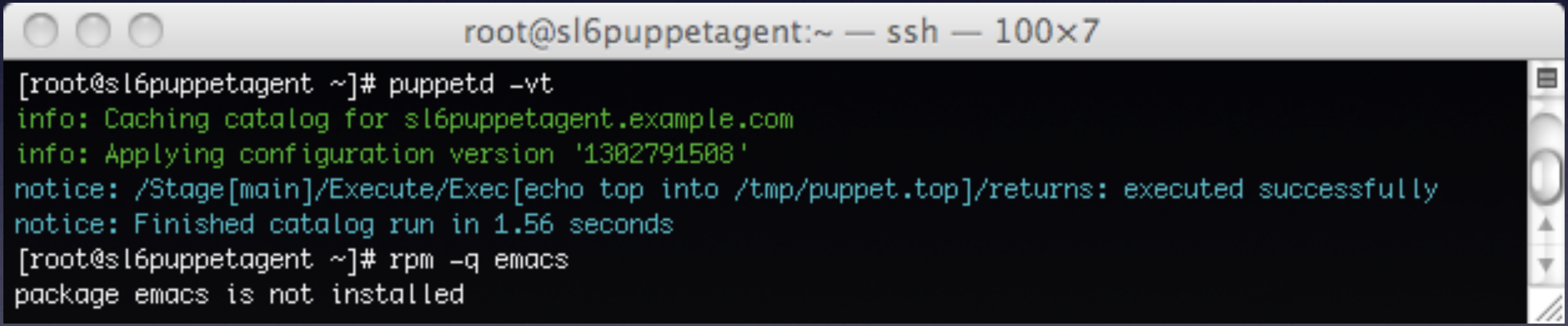


```
[root@sl6repo yum.repos.d]# puppetd -vt
info: Caching catalog for sl6repo.example.com
info: Applying configuration version '1302791508'
notice: /Stage[main]//Node[sl6repo.example.com]/Package[emacs]/ensure: created
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns: executed successfully
notice: Finished catalog run in 13.59 seconds
[root@sl6repo yum.repos.d]# 
```

nodes - step 4

- create nodes/sl6puppetagent.node

```
node "sl6puppetagent.example.com" inherits default {

}
```



```
[root@sl6puppetagent ~]# puppetd -vt
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302791508'
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns: executed successfully
notice: Finished catalog run in 1.56 seconds
[root@sl6puppetagent ~]# rpm -q emacs
package emacs is not installed
```

nodes - admissions

nodes - admissions

- It was not smooth:
  - puppetca ; and
  - puppetmaster --no-daemonize ;
- and then ...

Then SELinux struck

- setenforce 0

custom facts and conditional

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

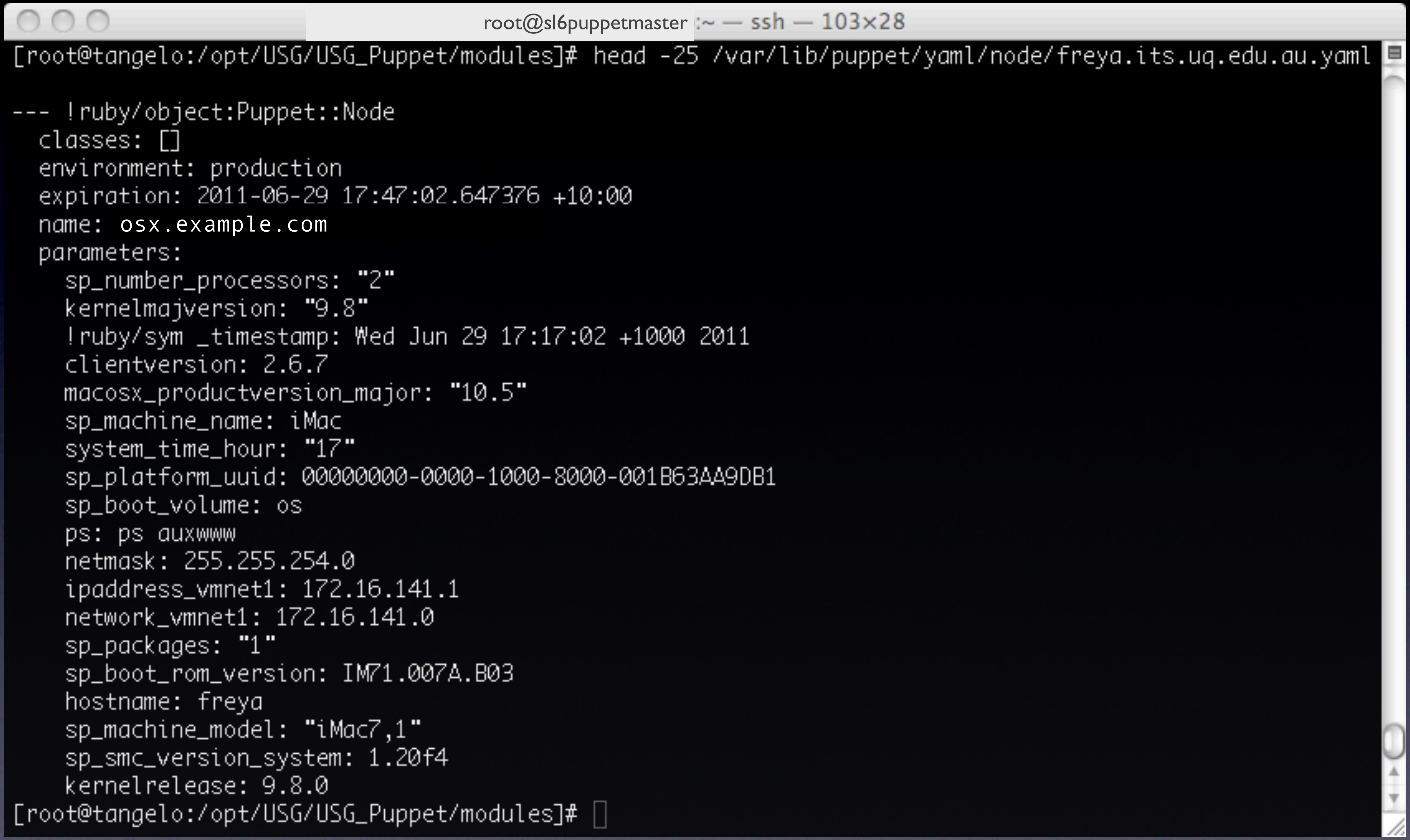
what is a fact?

- 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 %>

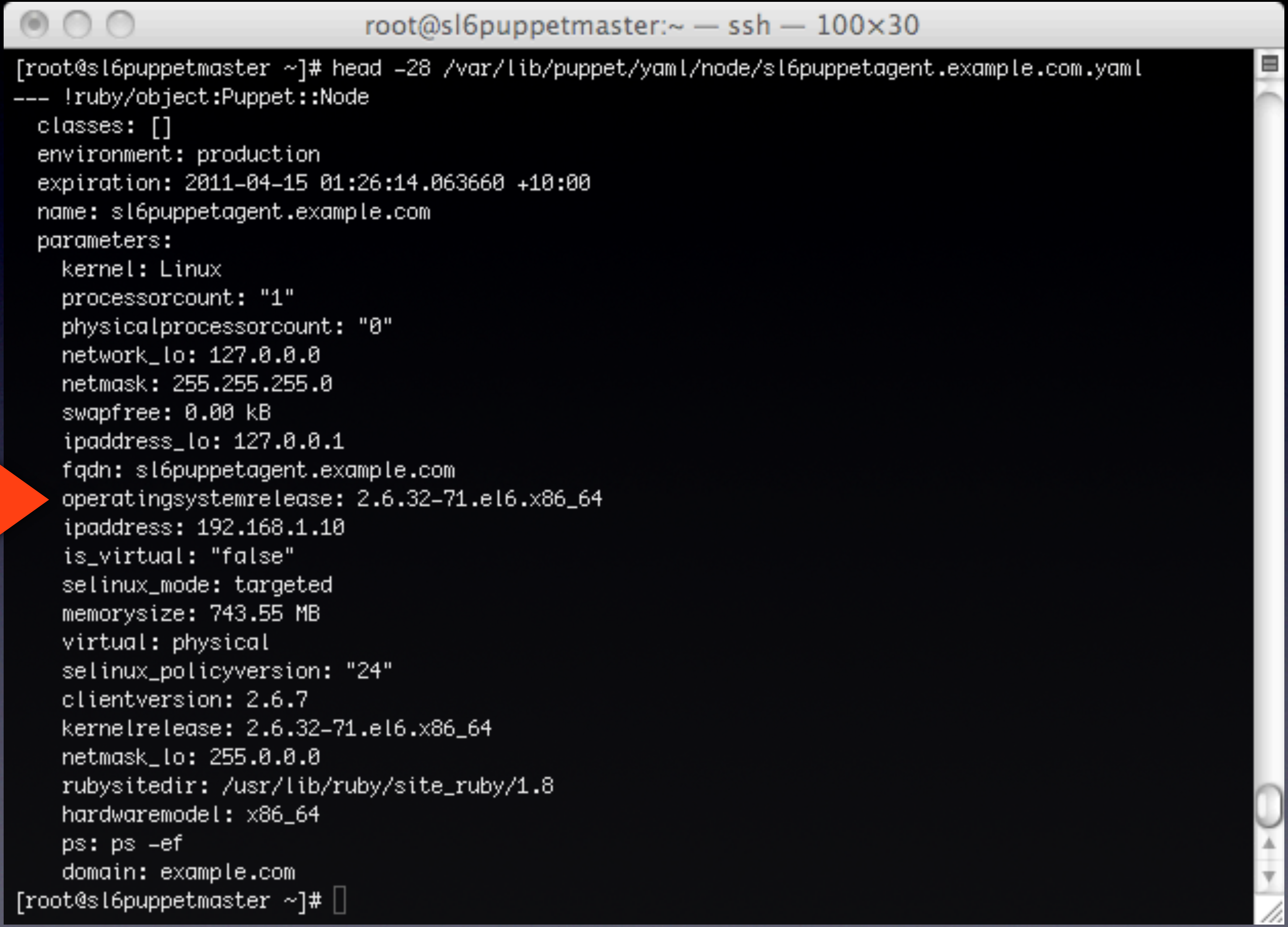
- used fact $ipaddress in sshd_config.erb template,
  - in nodes and classes they are addressed with a $ before their name;
  - in templates there is no $.

big brother is watching



```
[root@tangelo:/opt/USG/USG_Puppet/modules]# head -25 /var/lib/puppet/yaml/node/freya.its.uq.edu.au.yaml

--- !ruby/object:Puppet::Node
  classes: []
  environment: production
  expiration: 2011-06-29 17:47:02.647376 +10:00
  name: osx.example.com
  parameters:
    sp_number_processors: "2"
    kernelmajversion: "9.8"
    !ruby/sym _timestamp: Wed Jun 29 17:17:02 +1000 2011
    clientversion: 2.6.7
    macosx_productversion_major: "10.5"
    sp_machine_name: iMac
    system_time_hour: "17"
    sp_platform_uuid: 00000000-0000-1000-8000-001B63AA9DB1
    sp_boot_volume: os
    ps: ps auxwww
    netmask: 255.255.254.0
    ipaddress_vmnet1: 172.16.141.1
    network_vmnet1: 172.16.141.0
    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
[root@tangelo:/opt/USG/USG_Puppet/modules]# 
```

big brother is watching

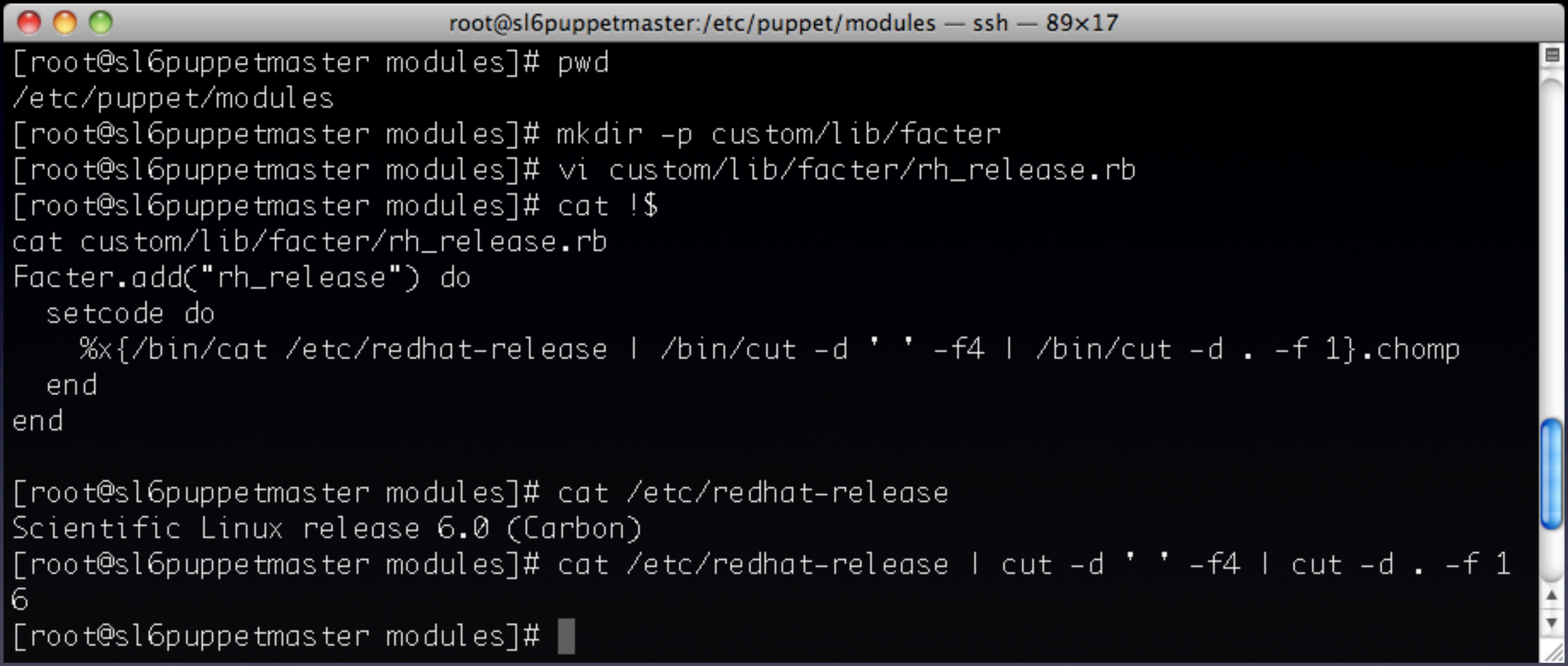


```
[root@sl6puppetmaster ~]# head -28 /var/lib/puppet/yaml/node/sl6puppetagent.example.com.yaml
--- !ruby/object:Puppet::Node
  classes: []
  environment: production
  expiration: 2011-04-15 01:26:14.063660 +10:00
  name: sl6puppetagent.example.com
  parameters:
    kernel: Linux
    processorcount: "1"
    physicalprocessorcount: "0"
    network_lo: 127.0.0.0
    netmask: 255.255.255.0
    swapfree: 0.00 kB
    ipaddress_lo: 127.0.0.1
    fqdn: sl6puppetagent.example.com
    operatingsystemrelease: 2.6.32-71.el6.x86_64
    ipaddress: 192.168.1.10
    is_virtual: "false"
    selinux_mode: targeted
    memorysize: 743.55 MB
    virtual: physical
    selinux_policyversion: "24"
    clientversion: 2.6.7
    kernelrelease: 2.6.32-71.el6.x86_64
    netmask_lo: 255.0.0.0
    rubysitedir: /usr/lib/ruby/site_ruby/1.8
    hardwaremodel: x86_64
    ps: ps -ef
    domain: example.com
[root@sl6puppetmaster ~]#
```

contrivances

- there should be $operatingsystemrelease telling you “6.0” or “6.1”, but we want “6”;
- Scientific Linux has a bug:
  - http://projects.puppetlabs.com/issues/6679
  - now fixed; reports as “Scientific”
- but see $operatingsystemrelease on the previous slide and:
  - http://projects.puppetlabs.com/issues/7682

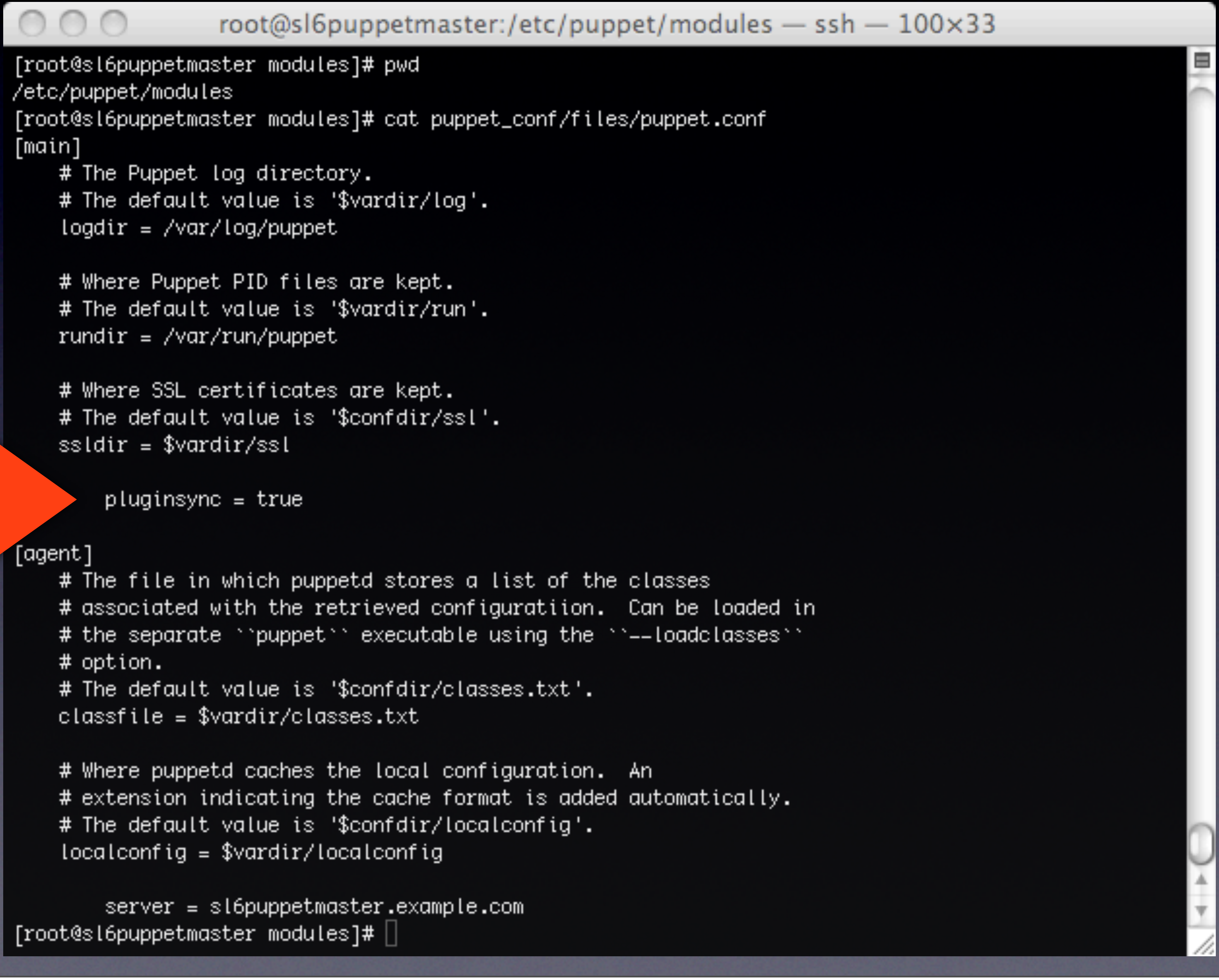
rh_release.rb



```
[root@sl6puppetmaster modules]# pwd
/etc/puppet/modules
[root@sl6puppetmaster modules]# mkdir -p custom/lib/facter
[root@sl6puppetmaster modules]# vi custom/lib/facter/rh_release.rb
[root@sl6puppetmaster modules]# cat !$
cat custom/lib/facter/rh_release.rb
Facter.add("rh_release") do
  setcode do
    %x{/bin/cat /etc/redhat-release | /bin/cut -d ' ' -f4 | /bin/cut -d . -f 1}.chomp
  end
end

[root@sl6puppetmaster modules]# cat /etc/redhat-release
Scientific Linux release 6.0 (Carbon)
[root@sl6puppetmaster modules]# cat /etc/redhat-release | cut -d ' ' -f4 | cut -d . -f 1
6
[root@sl6puppetmaster modules]# 
```

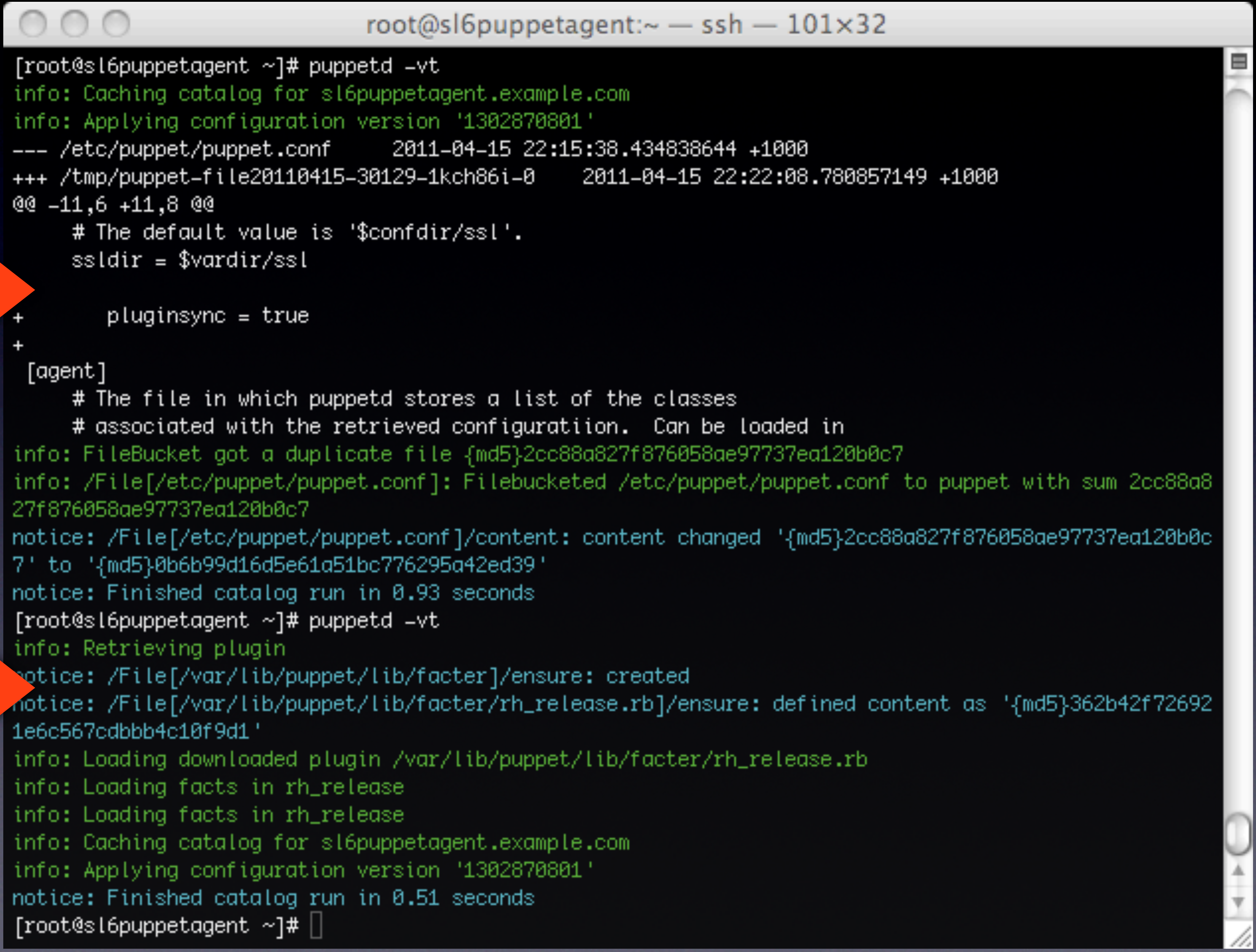
pluginsync = true

- modify puppet_conf/files/puppet.conf to:



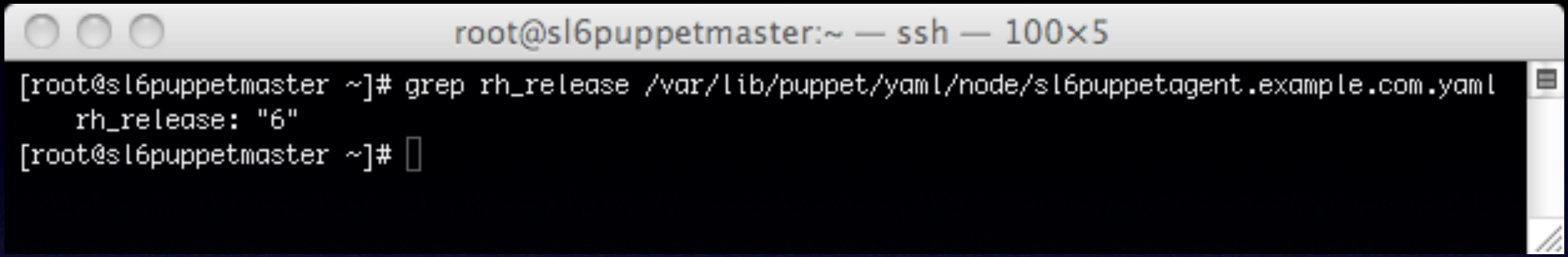
```
[root@sl6puppetmaster modules]# pwd
/etc/puppet/modules
[root@sl6puppetmaster modules]# cat puppet_conf/files/puppet.conf
[main]
    # The Puppet log directory.
    # The default value is '$vardir/log'.
    logdir = /var/log/puppet

    # 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

        pluginsync = true

[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 = sl6puppetmaster.example.com
[root@sl6puppetmaster modules]# 
```

on the client



```
[root@sl6puppetagent ~]# puppetd -vt
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302870801'
--- /etc/puppet/puppet.conf     2011-04-15 22:15:38.434838644 +1000
+++ /tmp/puppet-file20110415-30129-1kch86i-0    2011-04-15 22:22:08.780857149 +1000
@@ -11,6 +11,8 @@
     # The default value is '$confdir/ssl'.
     ssldir = $vardir/ssl

+        pluginsync = true
+
 [agent]
     # The file in which puppetd stores a list of the classes
     # associated with the retrieved configuratiion.  Can be loaded in
info: FileBucket got a duplicate file {md5}2cc88a827f876058ae97737ea120b0c7
info: /File[/etc/puppet/puppet.conf]: Filebucketed /etc/puppet/puppet.conf to puppet with sum 2cc88a8
27f876058ae97737ea120b0c7
notice: /File[/etc/puppet/puppet.conf]/content: content changed '{md5}2cc88a827f876058ae97737ea120b0c
7' to '{md5}0b6b99d16d5e61a51bc776295a42ed39'
notice: Finished catalog run in 0.93 seconds
[root@sl6puppetagent ~]# 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}362b42f72692
1e6c567cdbbb4c10f9d1'
info: Loading downloaded plugin /var/lib/puppet/lib/facter/rh_release.rb
info: Loading facts in rh_release
info: Loading facts in rh_release
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302870801'
notice: Finished catalog run in 0.51 seconds
[root@sl6puppetagent ~]# 
```

Now the server knows



```
[root@sl6puppetmaster ~]# grep rh_release /var/lib/puppet/yaml/node/sl6puppetagent.example.com.yaml
    rh_release: "6"
[root@sl6puppetmaster ~]# 
```

- ... so let’s use it ...

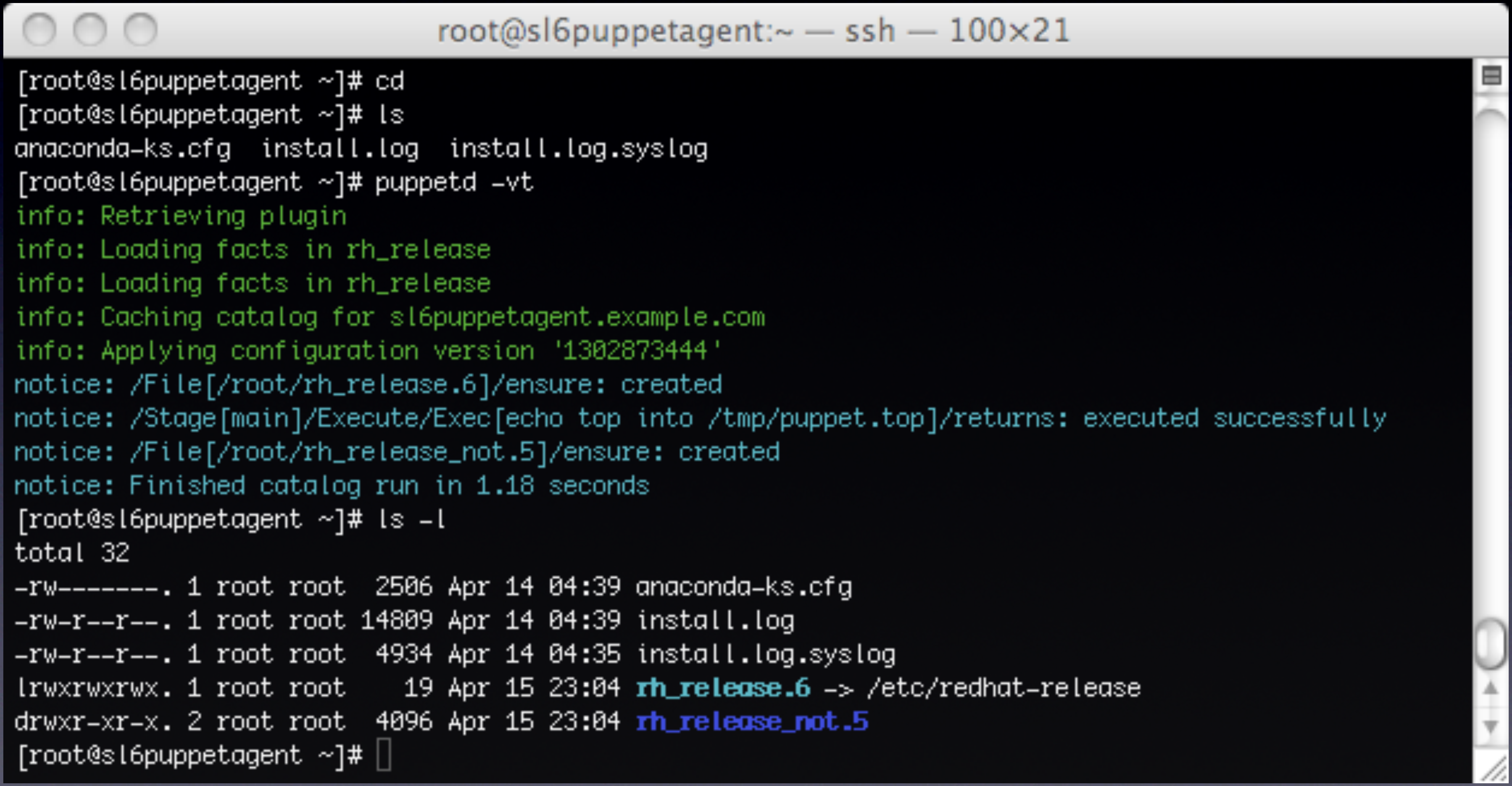
rh_release_case

```
class rh_release_case {

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

# conditionally create a directory, or install rsyslog
   if ($rh_release != "5") {
      file { "/root/rh_release_not.5":
         ensure => directory,
      }
   }
   else {
      package { "rsyslog":
         ensure => installed,
      }
   }
}
```

Remember

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

execute on the client



```
[root@sl6puppetagent ~]# cd
[root@sl6puppetagent ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@sl6puppetagent ~]# puppetd -vt
info: Retrieving plugin
info: Loading facts in rh_release
info: Loading facts in rh_release
info: Caching catalog for sl6puppetagent.example.com
info: Applying configuration version '1302873444'
notice: /File[/root/rh_release.6]/ensure: created
notice: /Stage[main]/Execute/Exec[echo top into /tmp/puppet.top]/returns: executed successfully
notice: /File[/root/rh_release_not.5]/ensure: created
notice: Finished catalog run in 1.18 seconds
[root@sl6puppetagent ~]# ls -l
total 32
-rw-------. 1 root root  2506 Apr 14 04:39 anaconda-ks.cfg
-rw-r--r--. 1 root root 14809 Apr 14 04:39 install.log
-rw-r--r--. 1 root root  4934 Apr 14 04:35 install.log.syslog
lrwxrwxrwx. 1 root root    19 Apr 15 23:04 rh_release.6 -> /etc/redhat-release
drwxr-xr-x. 2 root root  4096 Apr 15 23:04 rh_release_not.5
[root@sl6puppetagent ~]# 
```

Gigantic No-No

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 traditional programming;
    - ... used for sets of operations that are logically related;
- Defined (pun not intended) outside a class;
    - ... that is a big pitfall ...

define choices

- Choose where you use define with care:
  - odds are you will want to use it in more than one module;
  - ... but it may logically belong to a module;
  - ... can make it hard to follow.
- Don’t overdo it ...

(contrived) define example

```
class directories {
    mkdir_path { "puppet":
        path   => "/opt",
    }

    mkdir_path { "test":
        path   => "/opt/puppet",
    }

    Mkdir_path["test"] <- Mkdir_path["puppet"]
}

define mkdir_path($path) {
    file { "create a directory in $path by name $title":
        path   => "$path/$title",
        ensure => directory,
    }
}
```

- mkdir_path does not add much;
- ... might be okay if only used locally;

(contrived) define explanation

- path is explicitly passed;
- name is built-in;
- resource type file's path can use the source arguments (path and name);
    - note $ on right of => but not the left;

- Bonus: ordering using <-

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 = sl6puppetmaster.example.com
       splay = true
       runinterval = 1800
       environment = main
```

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

puppet agent as a service

```
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",
        notify => Service["com.reductivelabs.puppet"],
    }

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

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;
  - alleged to be Puppetlabs preferred method;

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

- program vs configuration;
- 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.

pitfalls in upgrading

- Migrating from 0.25.x to 2.6:
    - I upgraded my 0.25.6 to 2.6.5 no worries on RHEL5;
    - we also tried to build a brand new 2.6.5 instance on RHEL6 ... didn’t work so well;
        - do it one module at a time;

Dashboard

- Good for monitoring - discovering nodes with issues;
- Good single place to look at time trends;
- Not interactive.
- Database grows huge..

- /var/lib/puppet/yaml might be quicker.

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;
- Great for customer confidence;
- Not a replacement for documentation.