Creating a local repo (Mirror) with CentOS 6.2+

(Source: http://darktraining.com/linux/93/)

Recently I had an environment that had dozens of servers that were all on a private network. In situations like this, where there are single entry and exit points to a network doing a "yum update" on dozens of servers can really choke up the pipes.

The solution is use a local repo / yum mirror. However the documentation for CentOS 6+ is out dated. (See this wiki). So I had to figure out how to do this, and here are those steps.

For my install, I didn't want a DVD repo, I wanted a mirror of exactly what my hosts see when they call yum update. First thing, this is going to take up some room... like 8+ GB (and that was just x86_64). So do your self a favor and make sure that you are storing this on a separate disk / LVM.

In my example I make a folder called "repo"

```
mkdir /repo/centos/6.3/
```

Now I need to get the content that I want to mirror. There are alot of different ways to do this (repoman, etc) but I like the tried and true rsync method. I've selected a mirror from the following list (centos mirrors) and selected one that has the "rsync" value.

Now I need to call out for the packages, here is the command that I use:

```
rsync --progress -av --delete --delete-excluded --exclude "local*" --exclude "isos" --exclude "i386"
rsync://mirrors.kernel.org/centos/6.3/ /repo/centos/6.3/
```

That will give me a full copy of the mirror but I don't want the 32 bit versions because the servers in this environment are all 64bit, hence the exclude.

In my case, we are going to use apache to host the files, some folks do NFS though.

```
yum install httpd
```

Seriously, dont be a tool and disable selinux like everyone on the net says in these tutorials, just correctly set the context of the files:

```
chcon -Rv --type=httpd_sys_content_t /repo/
```

Because of how the repo works, you now need to make a sym link from the 6.3 folder like this:

```
ln -s /repo/centos/6.3/ /repo/centos/6
```

Allow the web server to be seen through the firewall:

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

Lets make a config file for Apache to use that points to our repo.

```
vi /etc/httpd/conf.d/00-repo.config
```

Then insert the content below, change dnsname to whatever name is resolvable on your network, if you are using bob.com, make dnsname.domain bob.com

```
<VirtualHost *:80>
ServerName dnsname.domain
```

```
ServerAlias repo
DocumentRoot /repo
ErrorLog logs/dnsname-error_log
CustomLog logs/dnsname-access_log common
<Directory "/repo/*">
Options Indexes FollowSymLinks
Order allow,deny
Allow from all
</Directory>
</VirtualHost>
```

Now start Apache

```
service httpd start
```

You should now be able to open a browser and navigate to your yum server and /repo/ and see the directory contents.

Configure the client

Now that you have the server configured, you need to congfigure the client.

Make the following edit:

```
vi /etc/yum.repos.d/CentOS-Base.repo
```

Comment out all the mirror lists:

```
mirrorlist=http://mirrorlist.centos.org/?release=6.2&arch=$basearch&repo=os
```

TO

```
#mirrorlist=http://mirrorlist.centos.org/?release=6.2&arch=$basearch&repo=os
```

Now UN comment the baseurl and make it like so

```
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
```

to

```
#baseurl=http://your.domain/centos/$releasever/os/$basearch/
```

Now make sure the client is clear of any old settings

```
yum clean all
```

Then update the host

```
yum update
```

You should see that the host calls out to the local repo and gets its files from you.

Parting thoughts

Having a local repo is important for folks that run large server farms because it allows you take control what versions EVERYONE is running. An example would be if you were running a web server farm. You want to make sure that no new host could some how get a newer version of PHP (for example) than the rest. So by using your own repo you control the versions by controlling when you choose to rsync.

Another added benefit is that you reduce outbound network traffic to a mirror which makes it faster for you and all of us by not clogging up the mirrors bandwidth.