Foreman - Feature #1822

Puppetize host feature -- auto install puppet on new or legacy host

08/13/2012 05:09 PM - Corey Osman

Status:	New		
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

In the past I used ssh and a for loop to install puppet on hosts that don't have puppet.

I had written a bash script to perform the initial puppet install for my particular environment but since everyone's environment will be different this feature will need to be highly customizable via the foreman "script library".

Essentially what I am after is a button on the new host page that says "puppetize" which uses ssh to login to the box and installs puppet via a script defined by the user (or provided for environments that fit a particular model). All that would be required to enter a new host is:

1. ip address
2. ssh credentials or keys (one time use, don't store)
3. custom puppetize script or default puppetize script

Once this puppetize script runs and installs on the newly created puppetnode, it should create the host automatically in forman via puppet with all the necessary details provided.

Additionally, most folks will also elect to use this feature in batch operation to install puppet on a number of hosts. This batch operation would need to be capable of parsing a csv file that contains all the hosts and other useful info (credentials?, name of puppetize script, ...)

I can also envision this feature being placed inside the foremancli as well. Where one could write their own parser and batch mode operations or create a cli batch mode puppetize feature.

History

#1 - 08/13/2012 08:09 PM - Brian Gupta

Ok, first let me say my initial reaction to this is that I feel that this it is out of scope. (Perhaps I am misunderstanding the use case.)

A couple of thoughts, elaborating:

1) For new hosts, I think the "correct" Foreman pattern is to use a provisioning template that installs and bootstraps puppet
2) For existing hosts, unless we institute a script runner system like Rundeck, (which already can integrate with Foreman) then this is out of scope for foreman itself, especially if the script pulls it's metadata out of a CSV file?

If I needed to install puppet on a bunch of foreman hosts today, I would use capistrano, fabric, m-collective or ssh-in-a-for-loop, to iterate over a list of hosts either queried directly from the API, or in the case of a shell script wrapping ssh I would escape out to foremancli to generate a hostlist.

Also why a CSV file? Wouldn't you get your metadata directly from foreman?

-Brian

P.S. - Long term, I do believe running commands via ssh is in scope for the foreman cli tooling, but I think it will assume that the hosts are already under some sort of configuration management.

#2 - 08/13/2012 08:45 PM - Corey Osman

Brian,

IMHO, most companies will already have an existing infrastructure that does not have puppet installed anywhere. So if you can imagine 1000 nodes that don't have puppet and were not original provisioned by foreman the only way to install puppet to to use a script and ssh.

What I am suggesting is a "tool" to automate this mandatory procedure. This procedure is something everyone will eventually do for "legacy nodes". Why make users reinvent the wheel everytime? Why can't foreman provide a mechanism to puppetize a node with procedures that are known to work and have been tested for other users?

Say for instance that a new hire comes along and asks how do I install puppet? Which answer would you prefer to provide? (Click puppetize or run ssh script on host you don't have access to yet using fabric, capistrano, or other tool which you will have to learn). The goal here is to provide a tool that assists in the initial rollout of puppet and foreman.

Imagine the cli tool doing this (find and install): where the cli scans the network and compares hosts not in foreman

```
foremancli --installpuppet  `foremancli --nonpuppethosts`
```

or

or provide a list and don't perform scan

```
foremancli --installpuppet  `cat nonpuppethosts.csv`
```

Example script I wrote that suited my environment and auto-installed puppet via ssh and for loop:

```
#!/bin/bash
SSLDELETE=removessl
ARCH=`uname -i`
rpm -qa puppet > /dev/null

VERS="poison"
linuxversion=`lsb_release -r | cut -f2 |cut -f1 -d.`

function setuppuppet () {
    cat > /etc/puppet/namespaceauth.conf << EOF
[puppetrunner]
        allow puppetprod1.mycompany.corp

EOF

    cat >> /etc/puppet/puppet.conf << EOF
        listen=true
        report=true

EOF

    cat >> /etc/sysconfig/puppet << EOF
PUPPET_SERVER=puppetprod1.mycompany.corp
PUPPET_EXTRA_OPTS=--waitforcert=500
EOF
    /sbin/service puppet restart

}

# Function routine for install yum if it doesn't exist
function installyum (){

rpm -ivh http://download.fedora.redhat.com/pub/epel/4/x86_64/python-elementtree-1.2.6-0.6.el4.x86_64.rpm
rpm -ivh http://download.fedora.redhat.com/pub/epel/4/x86_64/sqlite-3.3.6-0.3.el4.x86_64.rpm
rpm -ivh http://download.fedora.redhat.com/pub/epel/4/x86_64/python-sqlite-1.1.7-0.1.2.2.el4.x86_64.rpm
rpm -ivh http://download.fedora.redhat.com/pub/epel/4/x86_64/python-urlgrabber-2.9.8-0.3.el4.noarch.rpm
rpm -ivh http://download.fedora.redhat.com/pub/epel/4/x86_64/yum-2.4.3-0.5.el4.noarch.rpm

echo "[main]" > /etc/yum.conf
echo "cachedir=/var/cache/yum" >> /etc/yum.conf
echo "keepcache=0" >> /etc/yum.conf
echo "debuglevel=2" >> /etc/yum.conf
echo "logfile=/var/log/yum.log" >> /etc/yum.conf
echo "distroverpkg=redhat-release" >> /etc/yum.conf
echo "tolerant=1" >> /etc/yum.conf
echo "exactarch=1" >> /etc/yum.conf
echo "obsoletes=1" >> /etc/yum.conf
echo "gpgcheck=1" >> /etc/yum.conf
echo "plugins=1" >> /etc/yum.conf

echo "# Note: yum-RHN-plugin doesn't honor this." >> /etc/yum.conf
echo "#metadata_expire=1h" >> /etc/yum.conf
echo "# Default." >> /etc/yum.conf
echo "# installonly_limit = 3" >> /etc/yum.conf

echo "# PUT YOUR REPOS HERE OR IN separate files named file.repo" >> /etc/yum.conf
echo "# in /etc/yum.repos.d" >> /etc/yum.conf
}
function installrepo() {

    echo "Installing new repo  ${filesum}"
    if [ -f /etc/yum.repos.d/internalcompany.repo ]; then
        rm -f /etc/yum.repos.d/internalcompany.repo
    fi

}
function installpuppet {
# Lets make sure puppet is installed and configured properly
if [  "${VERS}" == "0" ]; then
        echo "puppet already installed"
        # Lets make sure puppet
        setuppuppet
        /sbin/service puppet restart
        exit 0
fi
# Lets find out which version of Linux were running (4.6, 5.5, 4.8, ...)
if [ $linuxversion -ge 6 ]; then
# Configure for RHEL6
    if [ "${ARCH}" == "i386" ]; then
       echo "i386 puppet installing on $linuxversion"
       wget "http://10.0.20.130/cblr/svc/op/yum/profile/centos6.1-i386" --output-document=/etc/yum.repos.d/cobbler-config.repo
       rpm -Uvh http://download.fedora.redhat.com/pub/epel/6/i386/epel-release-5-4.noarch.rpm
    else
        echo "x86_64 puppet installing $linuxversion"
        wget "http://10.0.20.130/cblr/svc/op/yum/profile/RHEL6-x86_64" --output-document=/etc/yum.repos.d/cobbler-config.repo
        rpm -Uvh http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-release-6-5.noarch.rpm
    fi
    wget "http://10.0.20.130/repos/internalcompany-rhel5.repo" --output-document=/etc/yum.repos.d/internalcompany.repo

elif [ $linuxversion -ge 5 ]; then
# Configure for RHEL5
    if [ "${ARCH}" == "i386" ]; then
       echo "i386 puppet installing on $linuxversion"
       wget "http://10.0.20.130/cblr/svc/op/yum/profile/centos5.3-i386" --output-document=/etc/yum.repos.d/cobbler-config.repo
       rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
    else
        echo "x86_64 puppet installing $linuxversion"
        wget "http://10.0.20.130/cblr/svc/op/yum/profile/RHEL5.7-x86_64" --output-document=/etc/yum.repos.d/cobbler-config.repo
        rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
    fi
    wget "http://10.0.20.130/repos/internalcompany-rhel5.repo" --output-document=/etc/yum.repos.d/internalcompany.repo

else
# Configure for RHEL4
    if [ "${ARCH}" == "i386" ]; then
       echo "i386 puppet installing"
       wget "http://10.0.20.130/cblr/svc/op/yum/profile/RHEL4.8-i386" --output-document=/etc/yum.repos.d/cobbler-config.repo
       rpm -Uvh http://download.fedora.redhat.com/pub/epel/4/i386/epel-release-4-10.noarch.rpm
    else
        echo "x86_64 puppet installing"
        wget "http://10.0.20.130/cblr/svc/op/yum/profile/RHEL4.8-x86_64" --output-document=/etc/yum.repos.d/cobbler-config.repo
       rpm -Uvh http://download.fedora.redhat.com/pub/epel/4/x86_64/epel-release-4-10.noarch.rpm
    fi
    wget "http://10.0.20.130/repos/internalcompany-rhel4.repo" --output-document=/etc/yum.repos.d/internalcompany.repo

fi
```

```
echo "Installing puppet" 
    yum -y install ruby
    if [ "${SSLDELETE}" == "removessl" ]; then
        rm -rf /var/lib/puppet/ssl
    fi
    yum clean metadata
    rpm -e puppet
    rpm -e facter
    yum -y install puppet facter
    /sbin/chkconfig puppet on
    setuppuppet

}
# Install Yum if not present
if [ ! -f /etc/yum.conf ]; then
        echo "Installing Yum" 
        installyum
fi
# Lets get the updated internalcompanyrepo
installrepo
installpuppet
```

#3 - 08/14/2012 02:27 AM - Ohad Levy

I could see a way to import a non managed node to foreman.
e.g you have a host that is not running puppet, but you want it to show up in a host list (maybe not the default host list) and you want to collect its inventory (assuming you could install something like facter).

when you install a new host, you could have a way to select an existing non managed host and reinstall it, or enter all details manually.

it would then be easy to ask foreman api to return the list of hosts which are non managed by puppet using the regular search conditions etc.

does that make sense?

#4 - 08/14/2012 02:28 AM - Ohad Levy

oh, and the right way imho to solve the script issue, is simply to use the script template kind ;)

then you could do something like

curl http://foreman/unattended/script | bash