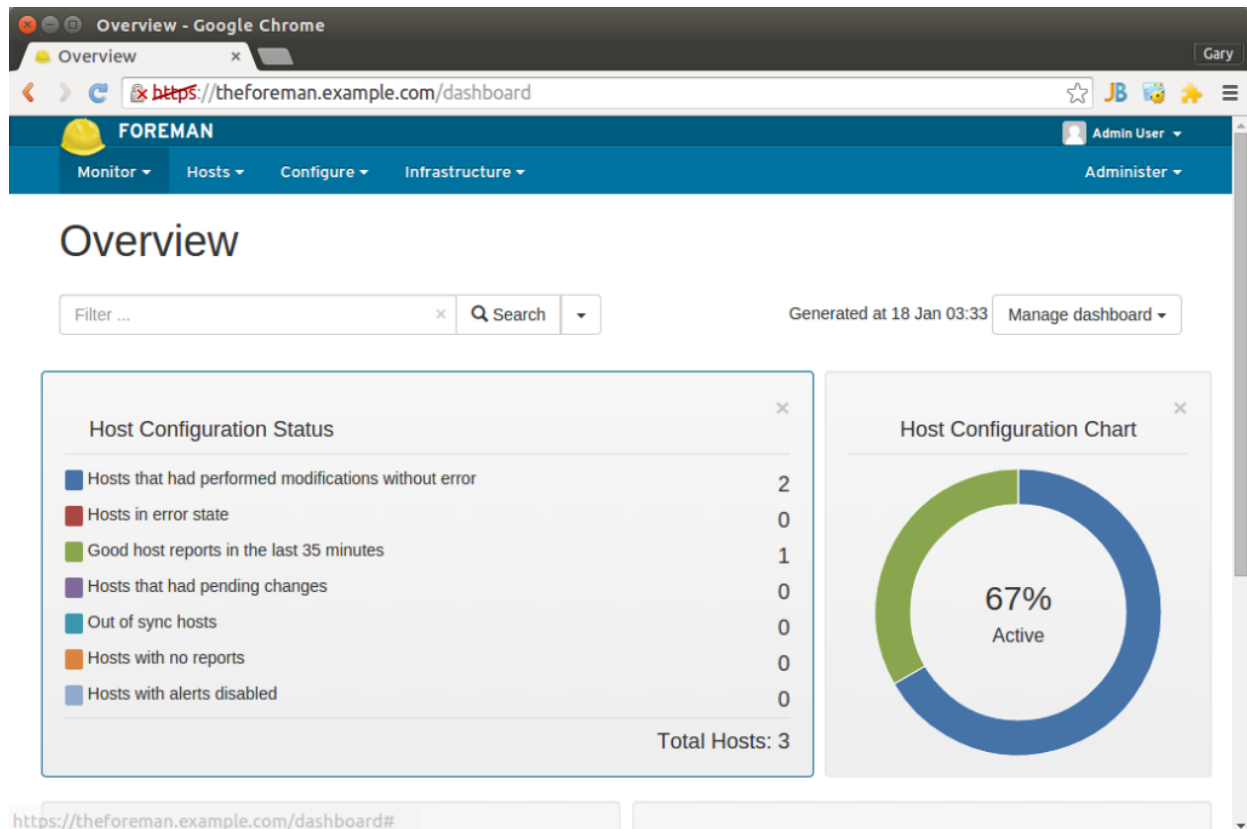


Installing Foreman and Puppet Agent on Multiple VMs Using Vagrant and VirtualBox

Automatically install and configure Foreman, the open source infrastructure lifecycle management tool, and multiple Puppet Agent VMs using Vagrant and VirtualBox.



<https://programmaticponderings.files.wordpress.com/2015/01/foreman-overview.png>

Introduction

In the last post, [Installing Puppet Master and Agents on Multiple VM Using Vagrant and VirtualBox](https://programmaticponderings.wordpress.com/2014/12/14/installing-puppet-master-and-agents-on-multiple-vm-using-vagrant-and-virtualbox/) (<https://programmaticponderings.wordpress.com/2014/12/14/installing-puppet-master-and-agents-on-multiple-vm-using-vagrant-and-virtualbox/>), we installed Puppet Master / Agent on VirtualBox VMs using Vagrant. Puppet Master is an excellent tool, but lacks the ease-of-use of [Puppet Enterprise](http://puppetlabs.com/puppet/puppet-enterprise) (<http://puppetlabs.com/puppet/puppet-enterprise>) or Foreman. In this post, we will build an almost identical environment, substituting Foreman for Puppet Master.

According to Foreman's [website](http://www.theforeman.org/learn_more.html) (http://www.theforeman.org/learn_more.html), "Foreman is an open source project that helps system administrators manage servers throughout their lifecycle, from provisioning and configuration to orchestration and monitoring. Using Puppet or Chef and Foreman's smart proxy architecture, you can easily automate repetitive tasks, quickly deploy applications, and proactively manage change, both on-premise with VMs and bare-metal or in the cloud."

Combined with Puppet Labs' [Open Source](http://puppetlabs.com/puppet/puppet-open-source) (<http://puppetlabs.com/puppet/puppet-open-source>) Puppet, Foreman is an effective solution to manage infrastructure and system configuration. Again, according to Foreman's website, the Foreman installer is a collection of Puppet modules that installs everything required for a full working Foreman setup. The installer uses native OS packaging and adds necessary configuration for the complete installation. By default, the Foreman installer will configure:

- [Apache HTTP](http://httpd.apache.org/) (<http://httpd.apache.org/>) with [SSL](http://httpd.apache.org/docs/2.2/ssl/) (<http://httpd.apache.org/docs/2.2/ssl/>) (using a Puppet-signed certificate)
- Foreman running under [mod_passenger](https://www.phusionpassenger.com/) (<https://www.phusionpassenger.com/>)
- [Smart Proxy](http://theforeman.org/manuals/1.1/index.html#Smart-Proxy) (<http://theforeman.org/manuals/1.1/index.html#Smart-Proxy>) configured for Puppet, TFTP and SSL
- Puppet master running under `mod_passenger`
- Puppet agent configured
- [TFTP](http://projects.theforeman.org/projects/smart-proxy/wiki/Tftp) (<http://projects.theforeman.org/projects/smart-proxy/wiki/Tftp>) server (under [xinetd](https://forge.puppetlabs.com/puppetlabs/xinetd) (<https://forge.puppetlabs.com/puppetlabs/xinetd>) on Red Hat platforms)

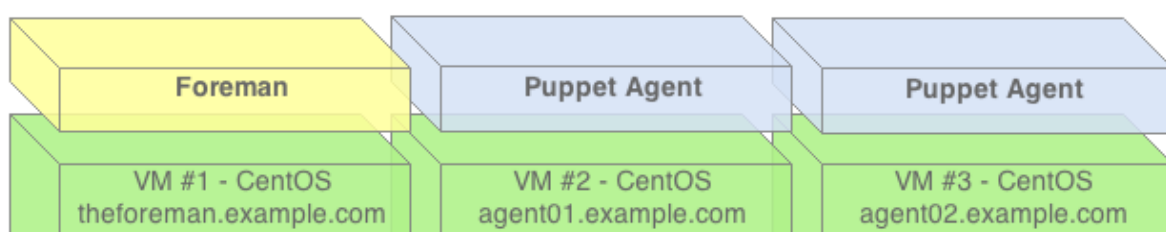
For the average Systems Engineer or Software Developer, installing and configuring Foreman, Puppet Master, Apache, Puppet Agent, and the other associated software packages listed above, is daunting. If the installation doesn't work properly, you must troubleshooting, or trying to remove and reinstall some or all the components.

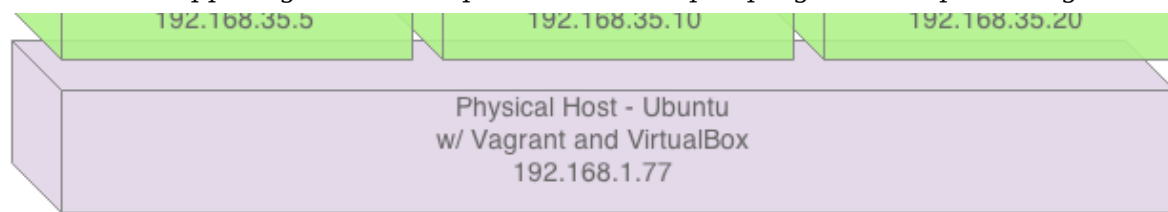
A better solution is to automate the installation of Foreman into a Docker container, or on to a VM using Vagrant. Automating the installation process guarantees accuracy and consistency. The Vagrant VirtualBox VM can be snapshotted, moved to another host, or simply destroyed and recreated, if needed.

All code for this post is available on [GitHub](https://github.com/garystafford/foreman-vagrant) (<https://github.com/garystafford/foreman-vagrant>). However, it been updated as of 8/23/2015. Changes were required to fix compatibility issues with the latest versions of Puppet 4.x and Foreman. Additionally, the version of CentOS on all VMs was updated from 6.6 to 7.1 and the version of Foreman was updated from 1.7 to 1.9.

The Post's Example

In this post, we will use [Vagrant](https://www.vagrantup.com/) (<https://www.vagrantup.com/>) and [VirtualBox](https://www.virtualbox.org/) (<https://www.virtualbox.org/>) to create three VMs. The VMs in this post will be build from a standard [CentOS 6.5 x64 base Vagrant Box](https://atlas.hashicorp.com/chef/boxes/centos-6.5) (<https://atlas.hashicorp.com/chef/boxes/centos-6.5>), located on [Atlas](https://atlas.hashicorp.com/chef/boxes/centos-6.5) (<https://atlas.hashicorp.com/chef/boxes/centos-6.5>). We will use a single JSON-format configuration file to automatically build all three VMs with Vagrant. As part of the provisioning process, using Vagrant's shell provisioner, we will execute a bootstrap shell script. The script will install Foreman and it's associated software on the first VM, and Puppet Agent on the two remaining VMs (aka Puppet 'agent nodes' or Foreman 'hosts').





(<https://programmaticponderings.files.wordpress.com/2015/01/foreman-puppet-vagrant-11.png>)

Foreman does have the ability to provision on bare-metal infrastructure and public or private clouds. However, this example would simulate an environment where you have existing nodes you want to manage with Foreman.

The Foreman bootstrap script will also download several Puppet modules. To test Foreman once the provisioning is complete, import those module's classes into Foreman and assign the classes to the hosts. The hosts will fetch and apply the configurations. You can then test for the installed instances of those module's components on the puppet agent hosts.

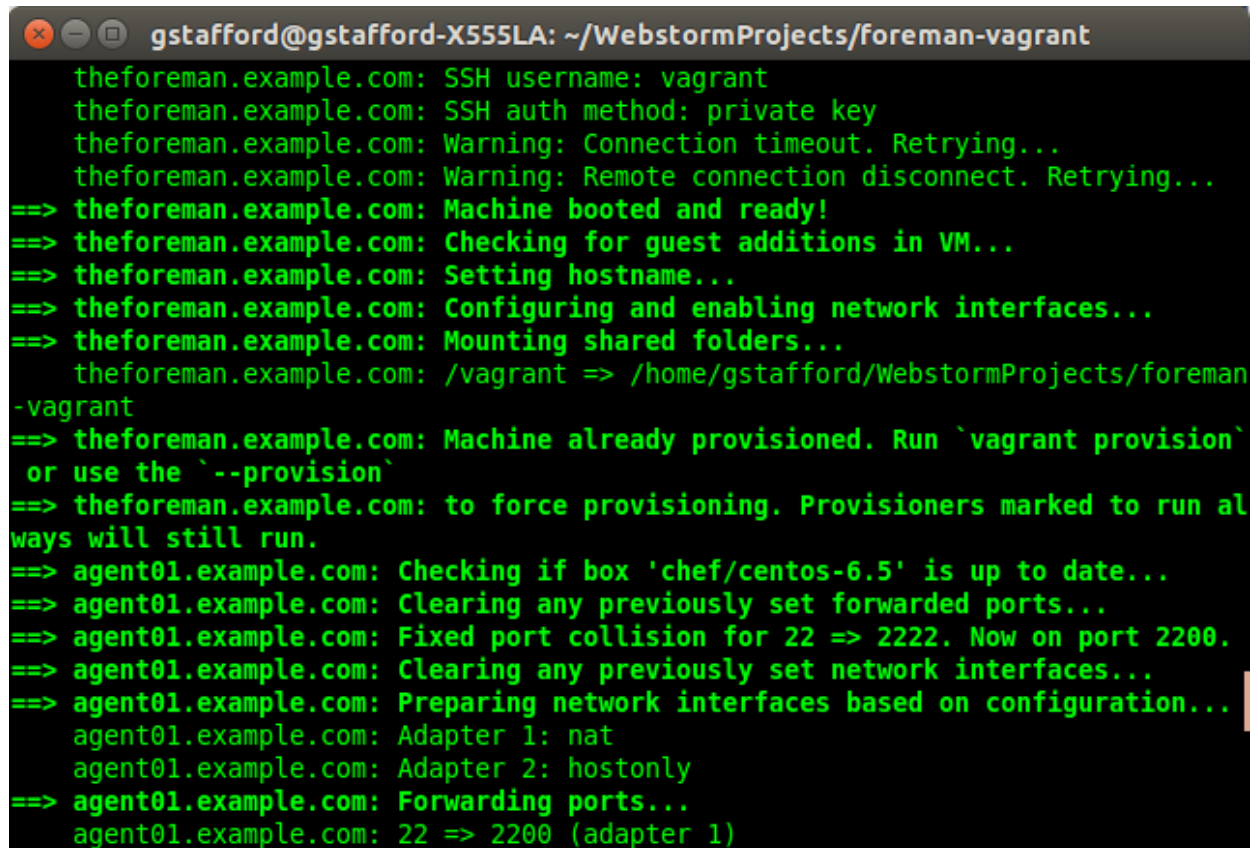
Vagrant

To begin the process, we will use the JSON-format configuration file to create the three VMs, using Vagrant and VirtualBox.

```
{
  "nodes": {
    "theforeman.example.com": {
      ":ip": "192.168.35.5",
      "ports": [],
      ":memory": 1024,
      ":bootstrap": "bootstrap-foreman.sh"
    },
    "agent01.example.com": {
      ":ip": "192.168.35.10",
      "ports": [],
      ":memory": 1024,
      ":bootstrap": "bootstrap-node.sh"
    },
    "agent02.example.com": {
      ":ip": "192.168.35.20",
      "ports": [],
      ":memory": 1024,
      ":bootstrap": "bootstrap-node.sh"
    }
  }
}
```

The Vagrant file uses the JSON-format configuration file, to provision the three VMs, using a single 'vagrant up' command. That's it, less than 30 lines of actual code in the Vagrant file to create as many VMs as you want. For this post's example, we will not need to add any VirtualBox

port mappings. However, that can also done from the JSON configuration file (see the README.md for more directions).



```
gstafford@gstafford-X555LA: ~/WebstormProjects/foreman-vagrant
theforeman.example.com: SSH username: vagrant
theforeman.example.com: SSH auth method: private key
theforeman.example.com: Warning: Connection timeout. Retrying...
theforeman.example.com: Warning: Remote connection disconnect. Retrying...
==> theforeman.example.com: Machine booted and ready!
==> theforeman.example.com: Checking for guest additions in VM...
==> theforeman.example.com: Setting hostname...
==> theforeman.example.com: Configuring and enabling network interfaces...
==> theforeman.example.com: Mounting shared folders...
theforeman.example.com: /vagrant => /home/gstafford/WebstormProjects/foreman
-vagrant
==> theforeman.example.com: Machine already provisioned. Run `vagrant provision`
or use the `--provision`
==> theforeman.example.com: to force provisioning. Provisioners marked to run al
ways will still run.
==> agent01.example.com: Checking if box 'chef/centos-6.5' is up to date...
==> agent01.example.com: Clearing any previously set forwarded ports...
==> agent01.example.com: Fixed port collision for 22 => 2222. Now on port 2200.
==> agent01.example.com: Clearing any previously set network interfaces...
==> agent01.example.com: Preparing network interfaces based on configuration...
agent01.example.com: Adapter 1: nat
agent01.example.com: Adapter 2: hostonly
==> agent01.example.com: Forwarding ports...
agent01.example.com: 22 => 2200 (adapter 1)
```

<https://programmaticponderings.files.wordpress.com/2015/01/vagrant-provisioning-the-vms.png>

Vagrant Provisioning the VMs

If you have not used the [CentOS Vagrant Box \(https://atlas.hashicorp.com/chef/boxes/centos-6.5\)](https://atlas.hashicorp.com/chef/boxes/centos-6.5), it will take a few minutes the first time for Vagrant to download the it to the local Vagrant Box repository.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# Builds single Foreman server and
# multiple Puppet Agent Nodes using JSON config file
# Gary A. Stafford - 01/15/2015

# read vm and chef configurations from JSON files
nodes_config = (JSON.parse(File.read("nodes.json"))['nodes'])

VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "chef/centos-6.5"

  nodes_config.each do |node|
    node_name = node[0] # name of node
    node_values = node[1] # content of node

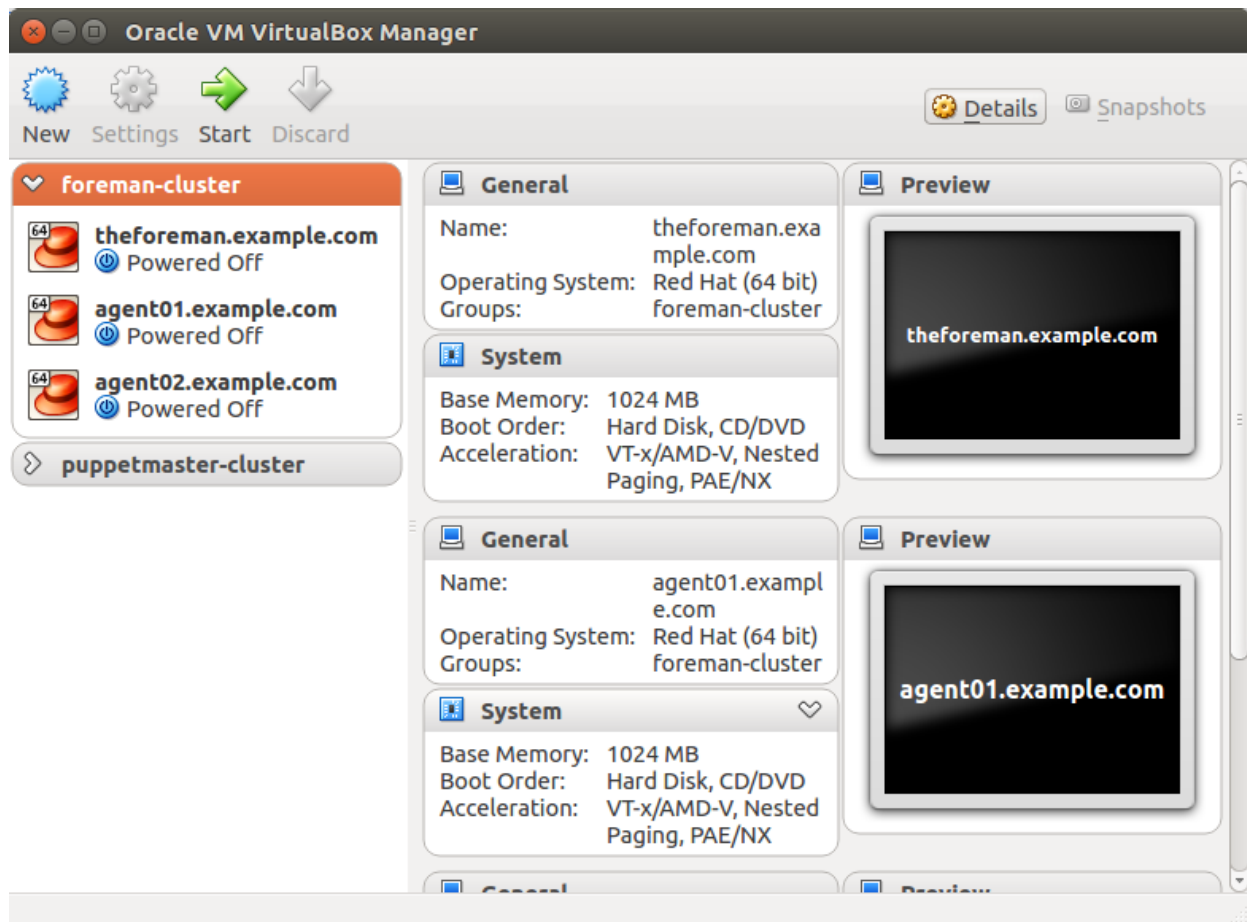
    config.vm.define node_name do |config|
      # configures all forwarding ports in JSON array
      ports = node_values['ports']
      ports.each do |port|
        config.vm.network :forwarded_port,
          host: port[:host],
          guest: port[:guest],
          id: port[:id]
      end

      config.vm.hostname = node_name
      config.vm.network :private_network, ip: node_values[:ip]

      config.vm.provider :virtualbox do |vb|
        vb.customize ["modifyvm", :id, "--memory", node_values[:memory]]
        vb.customize ["modifyvm", :id, "--name", node_name]
      end

      config.vm.provision :shell, :path => node_values[:bootstrap]
    end
  end
end
```

Once provisioned, the three VMs, also called 'Machines' by Vagrant, should appear in Oracle VM VirtualBox Manager.



(<https://programmaticponderings.files.wordpress.com/2015/01/oracle-vm-virtaulbox-manager-view.png>)

Oracle VM VirtualBox Manager View

The name of the VMs, referenced in Vagrant commands, is the parent node name in the JSON configuration file (`node_name`), such as, 'vagrant ssh theforeman.example.com'.

```
gstafford@gstafford-X555LA: ~/WebstormProjects/foreman-vagrant
gstafford@gstafford-X555LA:~/WebstormProjects/foreman-vagrant$ vagrant status
Current machine states:

theforeman.example.com    running (virtualbox)
agent01.example.com       running (virtualbox)
agent02.example.com       running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
gstafford@gstafford-X555LA:~/WebstormProjects/foreman-vagrant$ █
```

(<https://programmaticponderings.files.wordpress.com/2015/01/vagrant-status.png>)

Bootstrapping Foreman

As part of the Vagrant provisioning process ('`vagrant up`' command), a bootstrap script is executed on the VMs (shown below). This script will do almost of the installation and configuration work. Below is script for bootstrapping the Foreman VM.

```
#!/bin/sh

# Run on VM to bootstrap Foreman server
# Gary A. Stafford - 01/15/2015

if ps aux | grep "/usr/share/foreman" | grep -v grep 2> /dev/null
then
    echo "Foreman appears to all already be installed. Exiting..."
else
    # Configure /etc/hosts file
    echo "" | sudo tee --append /etc/hosts 2> /dev/null && \
    echo "192.168.35.5    theforeman.example.com    theforeman" | sudo tee

    # Update system first
    sudo yum update -y

    # Install Foreman for CentOS 6
    sudo rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch
    sudo yum -y install epel-release http://yum.theforeman.org/releases/1.
    sudo yum -y install foreman-installer && \
    sudo foreman-installer

    # First run the Puppet agent on the Foreman host which will send the f
    # automatically creating the host in Foreman's database
    sudo puppet agent --test --waitforcert=60

    # Install some optional puppet modules on Foreman server to get starte
    sudo puppet module install -i /etc/puppet/environments/production/modu
    sudo puppet module install -i /etc/puppet/environments/production/modu
    sudo puppet module install -i /etc/puppet/environments/production/modu
fi
```

Bootstrapping Puppet Agent Nodes

Below is script for bootstrapping the puppet agent nodes. The agent node bootstrap script was executed as part of the Vagrant provisioning process.

```
#!/bin/sh

# Run on VM to bootstrap Puppet Agent nodes
# Gary A. Stafford - 01/15/2015

if ps aux | grep "puppet agent" | grep -v grep 2> /dev/null
then
    echo "Puppet Agent is already installed. Moving on..."
else
    # Update system first
    sudo yum update -y

    # Install Puppet for CentOS 6
    sudo rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch
    sudo yum -y install puppet

    # Configure /etc/hosts file
    echo "" | sudo tee --append /etc/hosts 2> /dev/null && \
    echo "192.168.35.5    theforeman.example.com    theforeman" | sudo tee

    # Add agent section to /etc/puppet/puppet.conf (sets run interval to 1
    echo "" | sudo tee --append /etc/puppet/puppet.conf 2> /dev/null && \
    echo "    server = theforeman.example.com" | sudo tee --append /etc/pu
    echo "    runinterval = 120" | sudo tee --append /etc/puppet/puppet.co

    sudo service puppet stop
    sudo service puppet start

    sudo puppet resource service puppet ensure=running enable=true
    sudo puppet agent --enable
fi
```

Now that the Foreman is running, use the command, 'vagrant ssh agent01.example.com', to ssh into the first puppet agent node. Run the command below.

```
sudo puppet agent --test --waitforcert=60
```

The command above manually starts Puppet's Certificate Signing Request (CSR) process, to generate the certificates and security credentials (private and public keys) generated by Puppet's built-in certificate authority (CA). Each puppet agent node must have its certificate signed by the Foreman, first. According to Puppet's [website \(https://docs.puppetlabs.com/puppet/latest/reference/ssl_autosign.html\)](https://docs.puppetlabs.com/puppet/latest/reference/ssl_autosign.html), "Before puppet agent nodes can retrieve their configuration catalogs, they need a signed certificate from the local Puppet certificate authority (CA). When using Puppet's built-in CA (that is, not using an external CA), agents will submit a certificate signing request (CSR) to the CA Puppet Master (Foreman) and will retrieve a signed certificate once one is available."


```

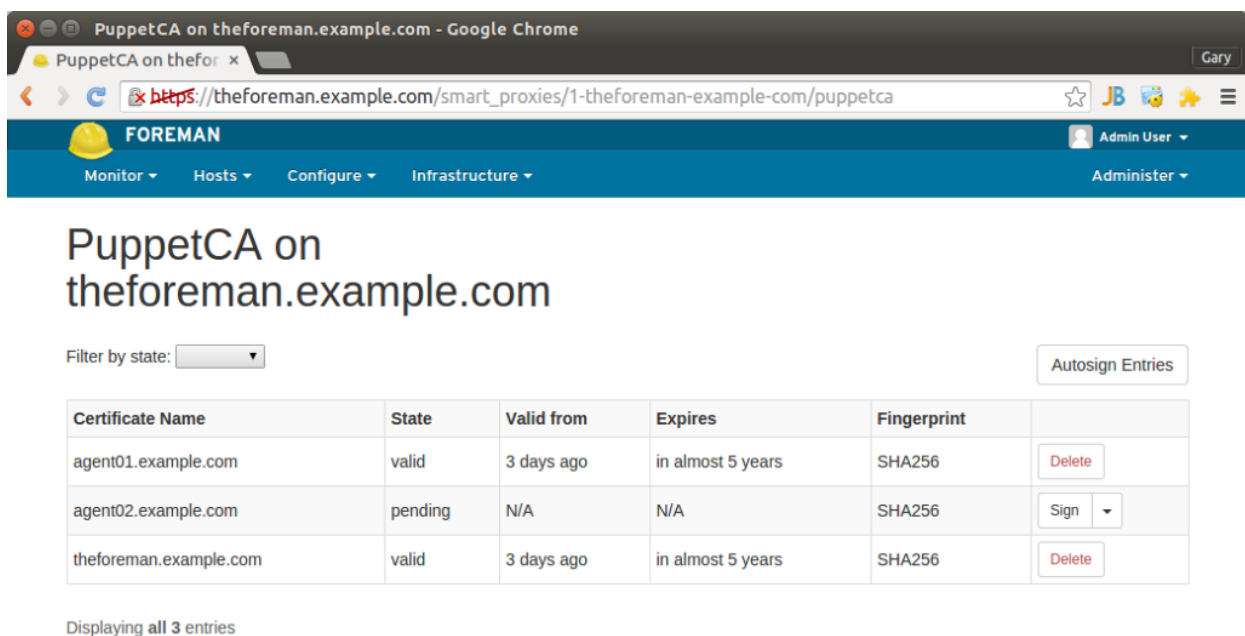
vagrant@agent02:~
gstafford@gstafford-X555LA:~$ cd WebstormProjects/foreman-vagrant/
gstafford@gstafford-X555LA:~/WebstormProjects/foreman-vagrant$ vagrant ssh agent02.example.com
Last login: Fri Mar  7 16:57:20 2014 from 10.0.2.2
[vagrant@agent02 ~]$ sudo puppet agent --test --waitforcert=60

```

(<https://programmaticponderings.files.wordpress.com/2015/01/waiting-for-cert-to-be-signed.png>)

Waiting for Certificate to be Signed by Foreman

Open the Foreman browser-based interface, running at <https://theforeman.example.com> (<https://theforeman.example.com>). Proceed to the 'Infrastructure' -> 'Smart Proxies' tab. Sign the certificate(s) from the agent nodes (shown below). The agent node will wait for the Foreman to sign the certificate, before continuing with the initial configuration.



PuppetCA on theforeman.example.com

Filter by state:

Autosign Entries

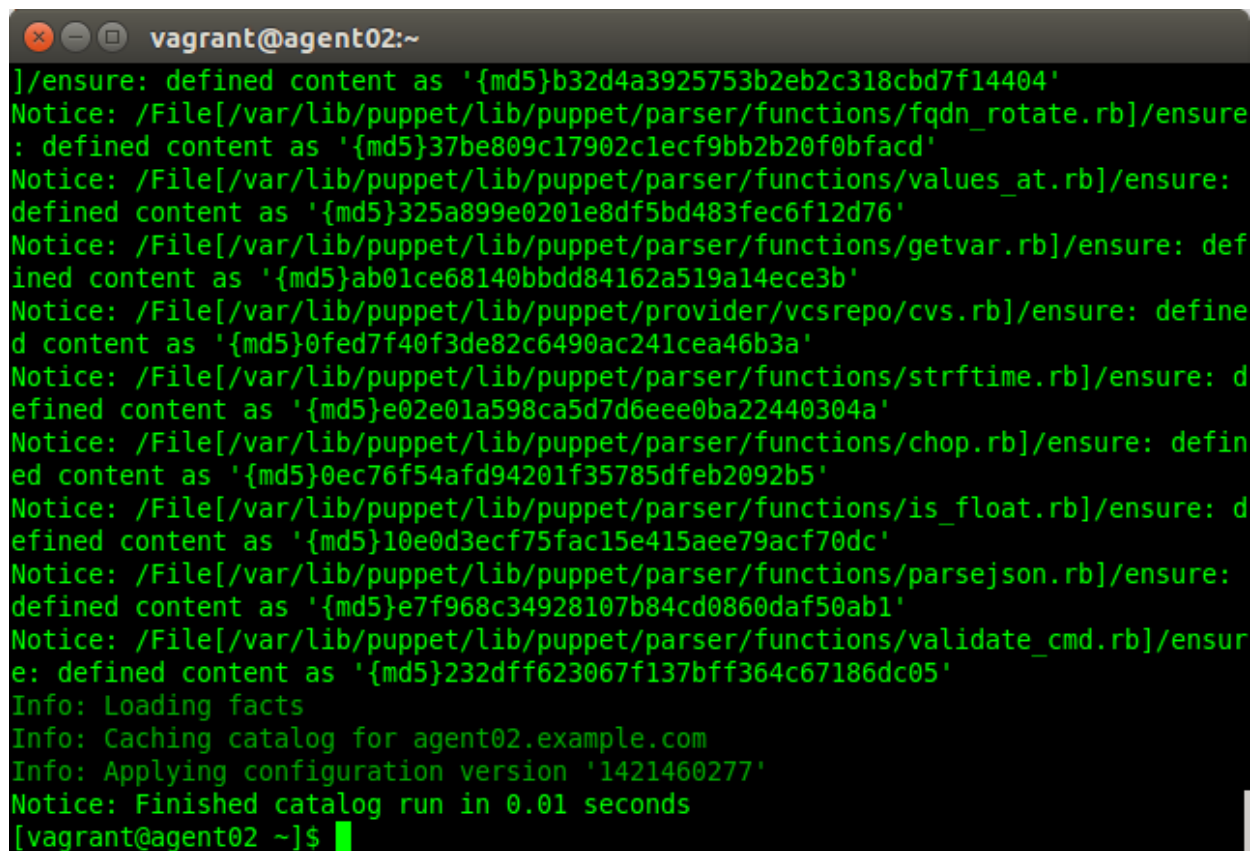
Certificate Name	State	Valid from	Expires	Fingerprint	
agent01.example.com	valid	3 days ago	in almost 5 years	SHA256	Delete
agent02.example.com	pending	N/A	N/A	SHA256	Sign
theforeman.example.com	valid	3 days ago	in almost 5 years	SHA256	Delete

Displaying all 3 entries

(<https://programmaticponderings.files.wordpress.com/2015/01/certificate-waiting-to-be-signed.png>)

Certificate Waiting to be Signed in Foreman

Once the certificate signing process is complete, the host retrieves the client configuration from the Foreman and applies it to the hosts.

A terminal window titled 'vagrant@agent02:~' showing the output of a Puppet configuration run. The output consists of several 'Notice' messages indicating that various Puppet functions have been defined with specific MD5 hashes. The functions listed include 'ensure' for 'fqdn_rotate.rb', 'values_at.rb', 'getvar.rb', 'vcsrepo/cvs.rb', 'strftime.rb', 'chop.rb', 'is_float.rb', 'parsejson.rb', and 'validate_cmd.rb'. The run concludes with 'Info: Loading facts', 'Info: Caching catalog for agent02.example.com', 'Info: Applying configuration version '1421460277'', and 'Notice: Finished catalog run in 0.01 seconds'. The prompt '[vagrant@agent02 ~]\$' is visible at the bottom.

```
vagrant@agent02:~  
]/ensure: defined content as '{md5}b32d4a3925753b2eb2c318cbd7f14404'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/fqdn_rotate.rb]/ensure:  
: defined content as '{md5}37be809c17902c1ecf9bb2b20f0bfacd'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/values_at.rb]/ensure:  
defined content as '{md5}325a899e0201e8df5bd483fec6f12d76'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/getvar.rb]/ensure: def  
ined content as '{md5}ab01ce68140bbdd84162a519a14ece3b'  
Notice: /File[/var/lib/puppet/lib/puppet/provider/vcsrepo/cvs.rb]/ensure: define  
d content as '{md5}0fed7f40f3de82c6490ac241cea46b3a'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/strftime.rb]/ensure: d  
efined content as '{md5}e02e01a598ca5d7d6eee0ba22440304a'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/chop.rb]/ensure: defin  
ed content as '{md5}0ec76f54afd94201f35785dfef2092b5'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/is_float.rb]/ensure: d  
efined content as '{md5}10e0d3ecf75fac15e415aee79acf70dc'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/parsejson.rb]/ensure:  
defined content as '{md5}e7f968c34928107b84cd0860daf50ab1'  
Notice: /File[/var/lib/puppet/lib/puppet/parser/functions/validate_cmd.rb]/ensur  
e: defined content as '{md5}232dff623067f137bff364c67186dc05'  
Info: Loading facts  
Info: Caching catalog for agent02.example.com  
Info: Applying configuration version '1421460277'  
Notice: Finished catalog run in 0.01 seconds  
[vagrant@agent02 ~]$
```

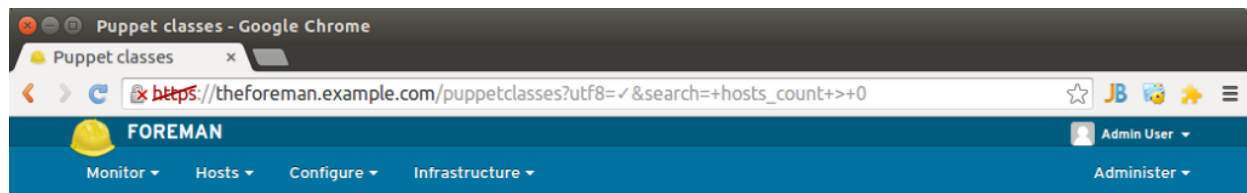
(<https://programmaticponderings.files.wordpress.com/2015/01/configuration-applied-to-agent-node.png>)

Foreman Puppet Configuration Applied to Agent Node

That's it, you should now have one host running Foreman and two puppet agent nodes.

Testing Foreman

To test Foreman, import the classes from the Puppet modules installed with the Foreman bootstrap script.



Puppet classes

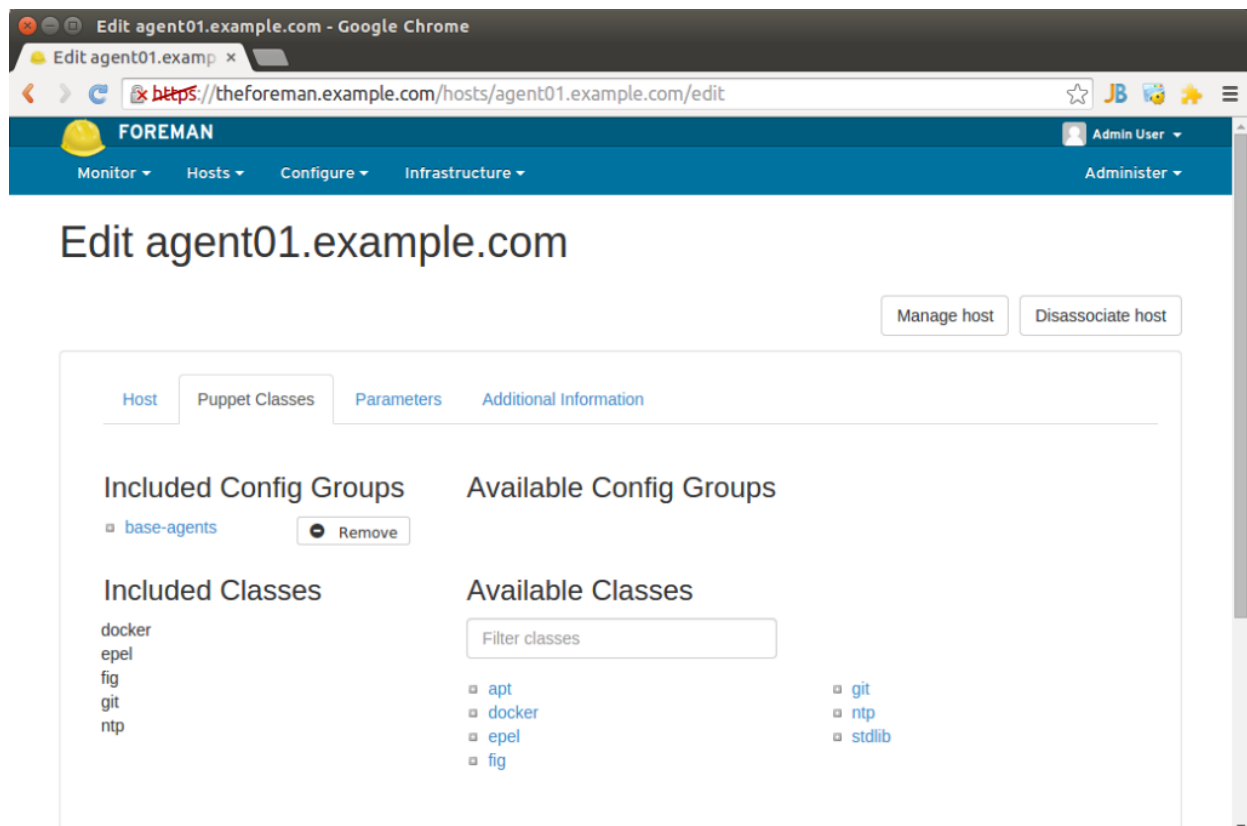
hosts_count > 0 Search

Class name	Environments and documentation	Host group	Hosts	Parameters	Variables	
docker	production		1	34	0	Delete
fig	production		1	0	0	Delete
git	production		2	1	0	Delete
ntp	production		2	24	0	Delete

Displaying all 4 entries

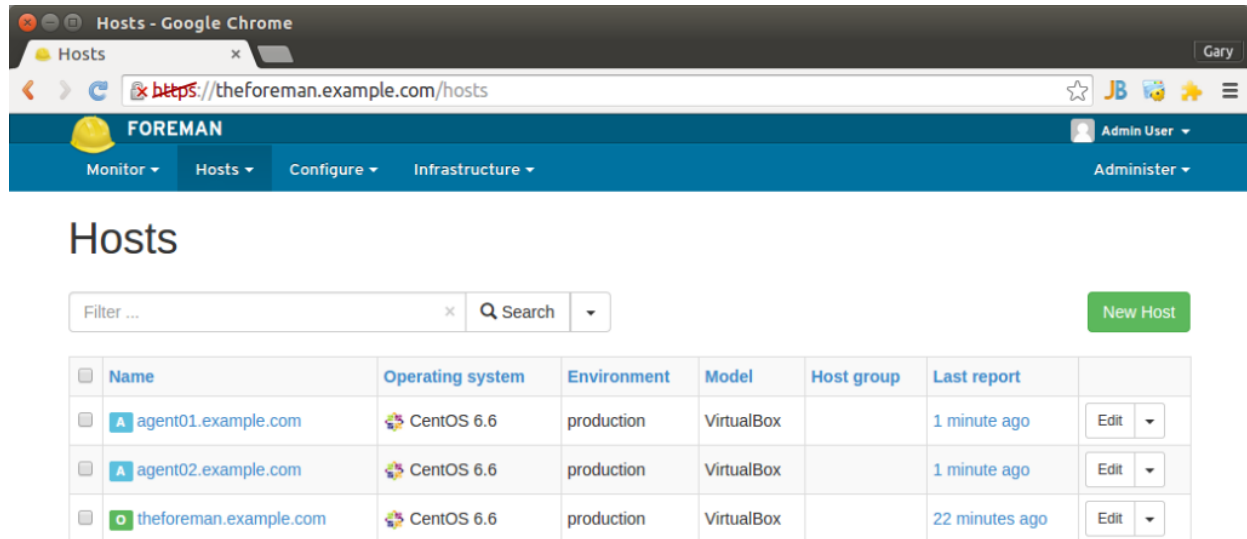
(<https://programmaticponderings.files.wordpress.com/2015/01/foreman-puppet-classes.png>)
Foreman – Puppet Classes

Next, apply [ntp](https://forge.puppetlabs.com/puppetlabs/ntp) (<https://forge.puppetlabs.com/puppetlabs/ntp>), [git](https://forge.puppetlabs.com/puppetlabs/git) (<https://forge.puppetlabs.com/puppetlabs/git>), and [Docker](https://forge.puppetlabs.com/garethr/docker) (<https://forge.puppetlabs.com/garethr/docker>) classes to both agent nodes (aka, Foreman 'hosts'), as well as the Foreman node, itself.



(<https://programmaticponderings.files.wordpress.com/2015/01/foreman-agents-puppet-classes.png>)
Foreman – Agents Puppet Classes

Every two minutes, the two agent node hosts should fetch their latest configuration from Foreman and apply it. In a few minutes, check the times reported in the 'Last report' column on the 'All Hosts' tab. If the times are two minutes or less, Foreman and Puppet Agent are working. Note we changed the `runinterval` to 120 seconds ('120s') in the bootstrap script to speed up the Puppet Agent updates for the sake of the demo. The normal default interval is 30 minutes. I recommend changing the agent node's `runinterval` back to 30 minutes ('30m') on the hosts, once everything is working to save unnecessary use of resources.



The screenshot shows the Foreman web interface in a Google Chrome browser. The page title is 'Hosts' and the URL is <https://theforeman.example.com/hosts>. The interface includes a navigation bar with 'Monitor', 'Hosts', 'Configure', and 'Infrastructure' tabs. Below the navigation bar, there is a search bar and a 'New Host' button. The main content area displays a table of hosts with the following data:

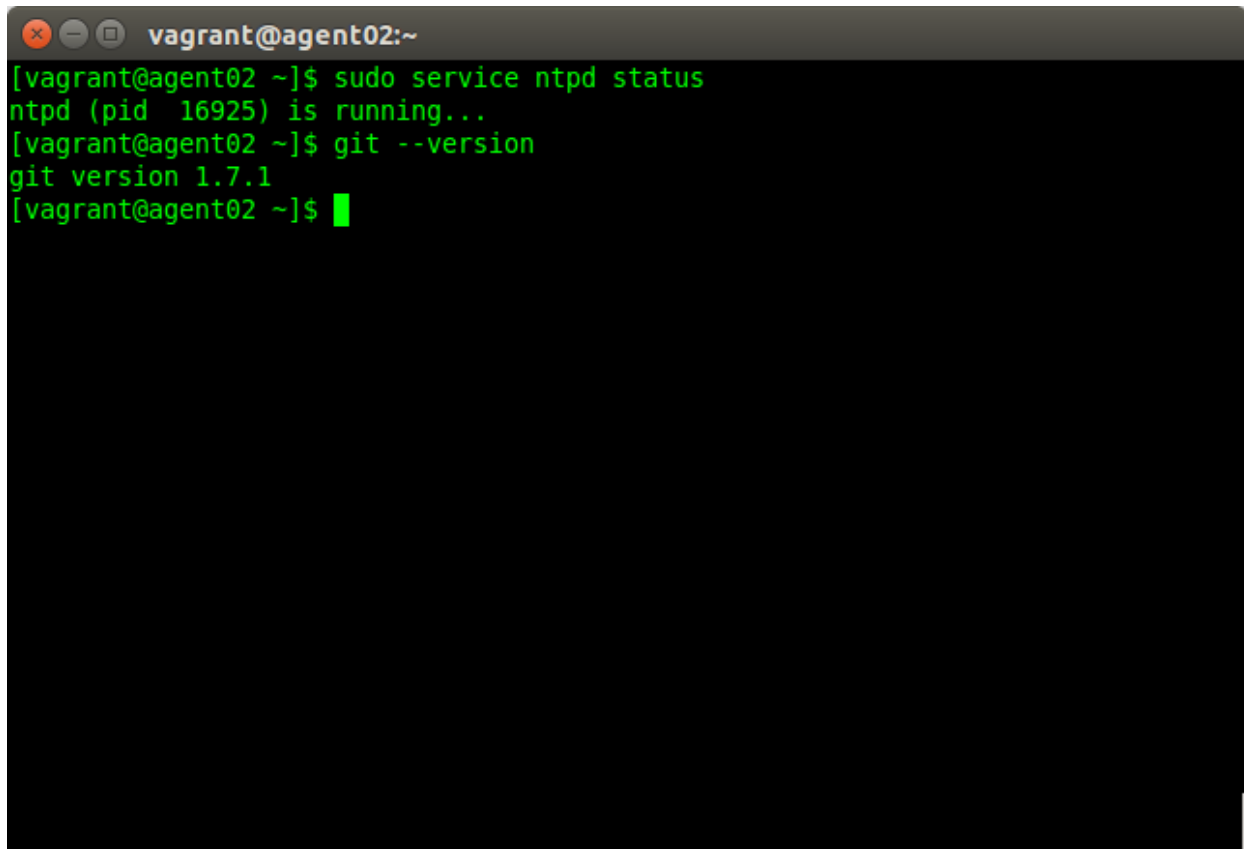
Name	Operating system	Environment	Model	Host group	Last report
agent01.example.com	CentOS 6.6	production	VirtualBox		1 minute ago
agent02.example.com	CentOS 6.6	production	VirtualBox		1 minute ago
theforeman.example.com	CentOS 6.6	production	VirtualBox		22 minutes ago

Displaying all 3 entries - 0 selected

(<https://programmaticponderings.files.wordpress.com/2015/01/foreman-hosts-reporting-back.png>)

Foreman – Hosts Reporting Back

Finally, to verify that the configuration was successfully applied to the hosts, check if `ntp` (<https://forge.puppetlabs.com/puppetlabs/ntp>), `git` (<https://forge.puppetlabs.com/puppetlabs/git>), and `Docker` (<https://forge.puppetlabs.com/garethr/docker>) are now running on the hosts.

A terminal window titled 'vagrant@agent02:~' with a dark background and green text. The terminal shows three commands and their outputs: 'sudo service ntpd status' returns 'ntpd (pid 16925) is running...', 'git --version' returns 'git version 1.7.1', and the prompt returns to '[vagrant@agent02 ~]\$' with a green cursor.

```
vagrant@agent02:~  
[vagrant@agent02 ~]$ sudo service ntpd status  
ntpd (pid 16925) is running...  
[vagrant@agent02 ~]$ git --version  
git version 1.7.1  
[vagrant@agent02 ~]$ █
```

(<https://programmaticponderings.files.wordpress.com/2015/01/agent-node-with-ntp-and-git-now-installed.png>)

Agent Node with ntp and git Now Installed

Helpful Links

All the source code this project is on [Github \(https://github.com/garystafford/foreman-vagrant\)](https://github.com/garystafford/foreman-vagrant).

Foreman:

<http://theforeman.org> (<http://theforeman.org>)

Atlas – Discover Vagrant Boxes:

<https://atlas.hashicorp.com/boxes/search> (<https://atlas.hashicorp.com/boxes/search>)

Learning Puppet – Basic Agent/Master Puppet

https://docs.puppetlabs.com/learning/agent_master_basic.html (https://docs.puppetlabs.com/learning/agent_master_basic.html#docs->-learning-puppet---basic-agent/master-puppet)

Puppet Glossary (of terms):

<https://docs.puppetlabs.com/references/glossary.html> (<https://docs.puppetlabs.com/references/glossary.html>)

agent node , Automation , CentOS , DevOps , Foreman , Puppet , Puppet Agent , Puppet Master , Vagrant , VirtualBox

This entry was posted on January 18, 2015, 3:00 pm and is filed under [DevOps](#), [Enterprise Software Development](#), [Software Development](#). You can follow any responses to this entry through [RSS 2.0](#). You can [leave a response](#), or [trackback](#) from your own site.

COMMENTS (9)

#1 by **wb** on July 9, 2015 - 2:08 pm

This is a great article! Great work.

#2 by **test** on September 9, 2015 - 2:26 pm

Excellent writeup!!! Can you do one which integrates the foreman with the openstack

#3 by **eiser** on September 30, 2015 - 6:32 pm

Great article, thanks. Quick question, Is it possible to install foreman without the puppet server? I mean, I already have a puppet server running in a different vm, I would like to

#4 by **Jon** on February 3, 2016 - 4:16 pm

During foreman vargrant up you get prompted for a password when touching /etc/hosts, what is this password?

#5 by Gary A. Stafford on February 4, 2016 - 12:11 am

vagrant

#6 by **chris** on February 6, 2016 - 9:59 pm

Hey Gary,

this seems like a really good howto, except you've missed the ID10T's guide section .
(Which I need) 😊

Basically I know Linux well and some basic puppet, but none of Vagrant/Foreman/Virtualbox /github/json (I'm really old school...)

Can you provide instructions on how to actually get all the SW onto my Centos 6.7 box and install/set it up?

Obviously I have root; I don't bother with sudo.

Incidentally I'd like to stick with the Centos 6 version of the install please.

Feel free to email me if you need more info.

thx a lot,

Chris

#7 by **Helper 1000** on September 27, 2016 - 8:03 am

Gary,

I am seeing the following issue. Please could you suggest whether some thing has been changed.

theforeman.example.com: SSH address: 127.0.0.1:2200

theforeman.example.com: SSH username: vagrant

theforeman.example.com: SSH auth method: private key

theforeman.example.com:

theforeman.example.com: Vagrant insecure key detected. Vagrant will automatically replace

theforeman.example.com: this with a newly generated keypair for better security.

theforeman.example.com:

theforeman.example.com: Inserting generated public key within guest...

theforeman.example.com: Removing insecure key from the guest if it's present...

theforeman.example.com: Key inserted! Disconnecting and reconnecting using new SSH key...

#8 by **Student** on October 15, 2016 - 10:21 am

Gary,

Do you have any solution to this issue.

#9 by Kunalsing Thakur on August 17, 2017 - 5:27 am

Getting Error:-

Downloading VirtualBox Guest Additions ISO from http://download.virtualbox.org/virtualbox/5.1.26/VBoxGuestAdditions_5.1.26.iso

Progress: 100% (Rate: 116k/s, Estimated time remaining: --:--:--)

Copy iso file C:/Users/kunalsingt/.vagrant.d/tmp/VBoxGuestAdditions_5.1.26.iso into the box /tmp/VBoxGuestAdditions.iso

mount: /dev/loop0 is write-protected, mounting read-only

Installing Virtualbox Guest Additions 5.1.26 – guest version is 4.3.22

Verifying archive integrity... All good.

Uncompressing VirtualBox 5.1.26 Guest Additions for Linux.....

VirtualBox Guest Additions installer

Removing installed version 4.3.22 of VirtualBox Guest Additions...

Copying additional installer modules ...

Installing additional modules ...

vboxadd.sh: Starting the VirtualBox Guest Additions.

Could not find the X.Org or XFree86 Window System, skipping.

An error occurred during installation of VirtualBox Guest Additions 5.1.26. Some functionality may not work as intended.

In most cases it is OK that the "Window System drivers" installation failed.

Redirecting to /bin/systemctl start vboxadd.service

Cleaning up downloaded VirtualBox Guest Additions ISO...

FAILED: Permission denied @ unlink_internal – C:/Users/kunalsingt/.vagrant.d /tmp/VBoxGuestAdditions_5.1.26.iso

You might want to delete this file manually: C:/Users/kunalsingt/.vagrant.d /tmp/VBoxGuestAdditions_5.1.26.iso

Got different reports about installed GuestAdditions version:

Virtualbox on your host claims: 4.3.22

VBoxService inside the vm claims: 5.1.26

Going on, assuming VBoxService is correct...

Got different reports about installed GuestAdditions version:

Virtualbox on your host claims: 4.3.22

VBoxService inside the vm claims: 5.1.26

Going on, assuming VBoxService is correct...

Got different reports about installed GuestAdditions version:

Virtualbox on your host claims: 4.3.22

VBoxService inside the vm claims: 5.1.26

Going on, assuming VBoxService is correct...

Restarting VM to apply changes...

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

[Blog at WordPress.com.](#)