# Puppet Master/Agent

Installation

## Vagrant Setup

```
$ cd learn
$ mkdir multinode
$ cd multinode
$ vagrant init puppet
```

Edit Vagrantfile

## Multi VM Setup with Vagrant

Multi VM Setup

http://docs.vagrantup.com/v2/multi-machine/index.html

## Vagrantfile

```
config.vm.define :master do |master|
   master.vm.box = "centos"
   master.vm.hostname = "master"
   master.vm.network :private_network, ip: "192.168.5.10"
 end

 config.vm.define :agent do |agent|
   agent.vm.box = "centos"
   agent.vm.hostname = "agent"
   agent.vm.network :private_network, ip: "192.168.5.11"
 end
```

## Master / Agent Boxes

We defined two VMs and gave them IP addresses.

| Master | Agent |
|---|---|
| 192.168.5.10 | 192.168.5.11 |

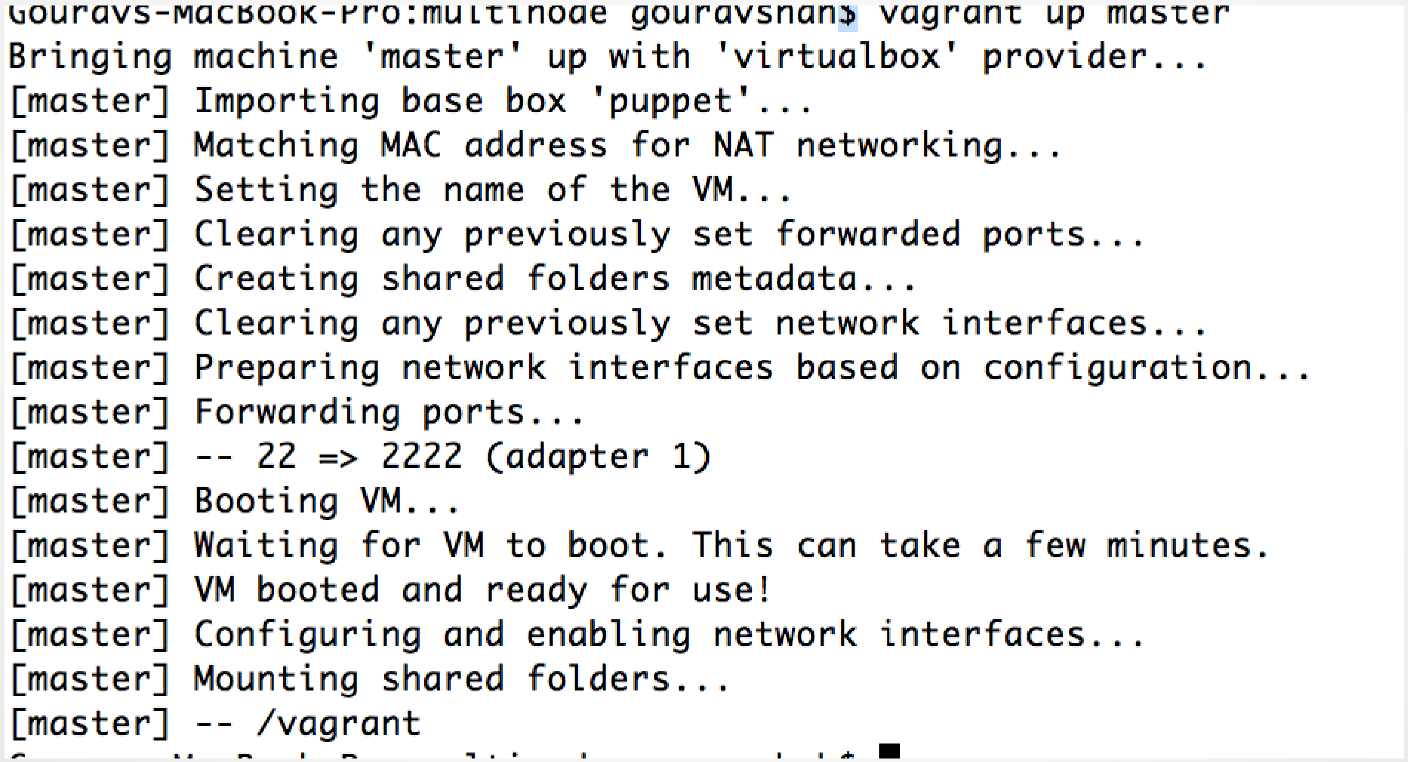
## Bringing Up Vagrant Boxes

vagrant up now becomes

- vagrant up master
- vagrant up agent

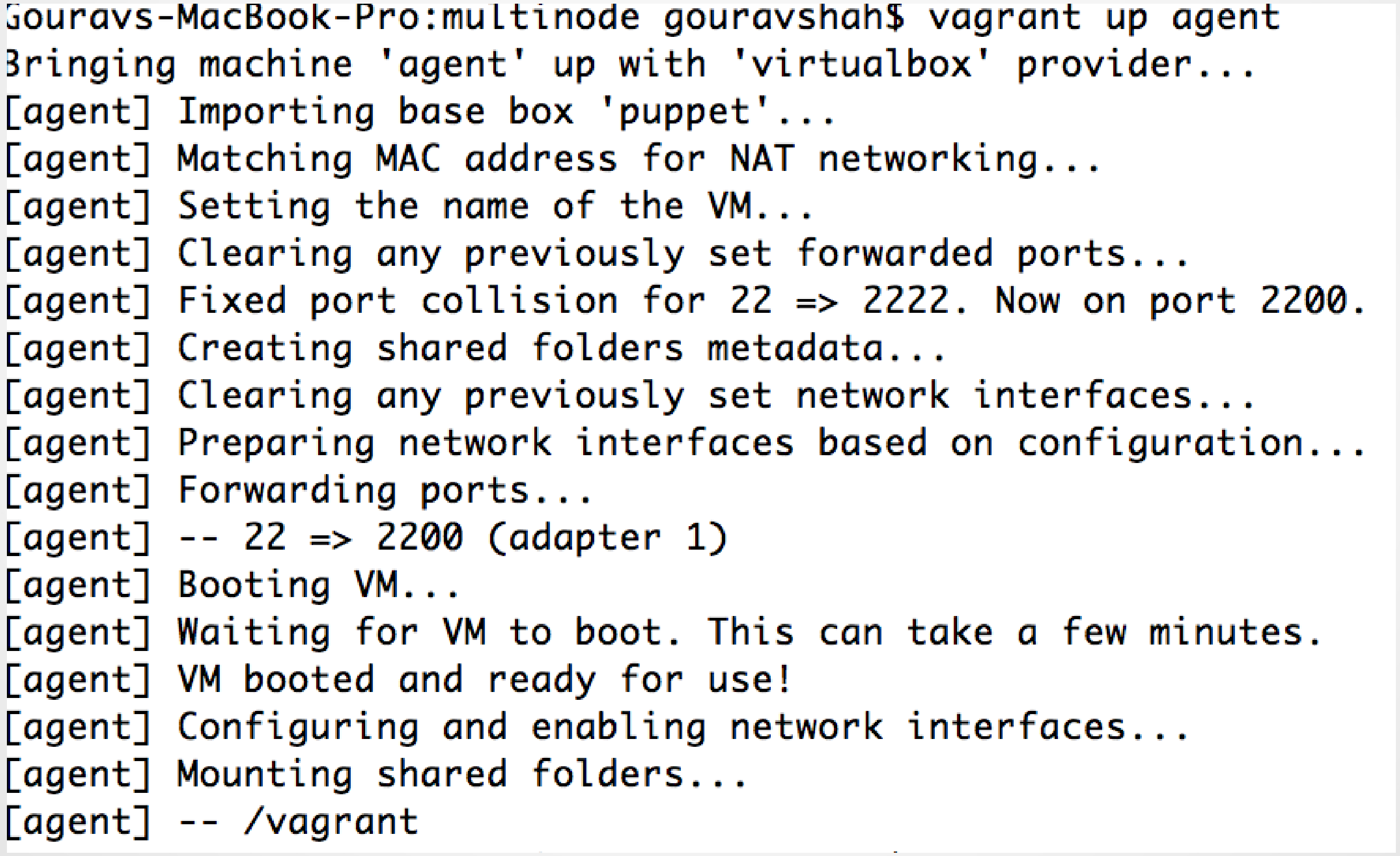
- If you use vagrant up, it will bring up both

## vagrant up master



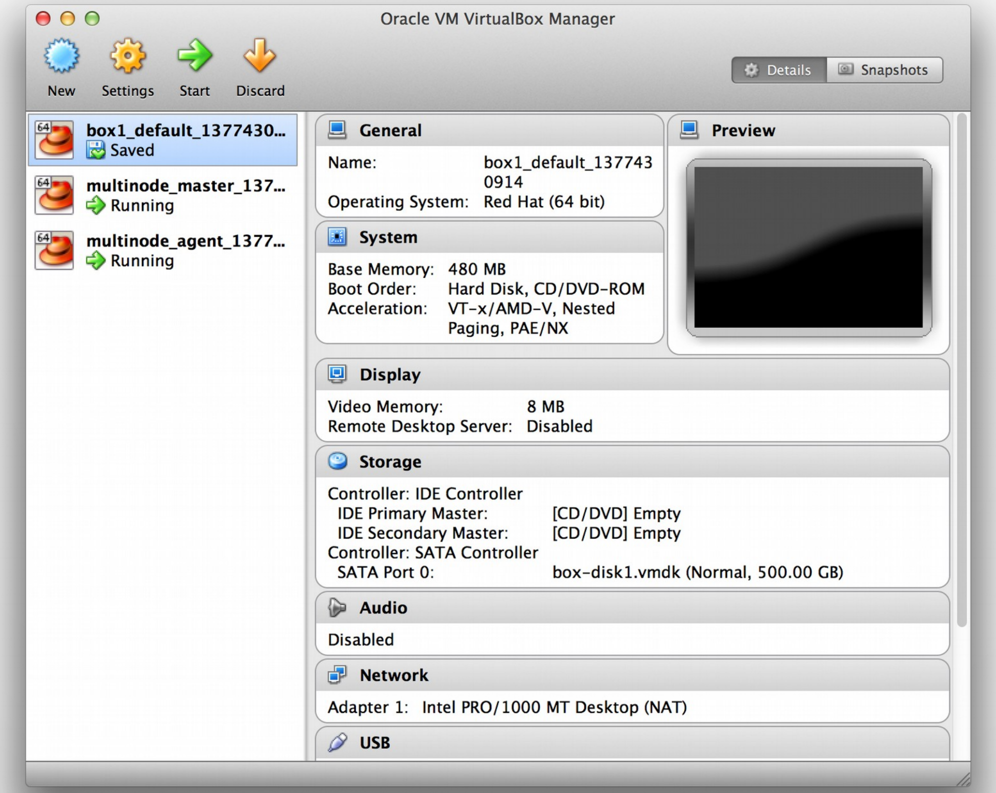
```
Gouravs-MacBook-Pro:multinode gouravshah$ vagrant up master
Bringing machine 'master' up with 'virtualbox' provider...
[master] Importing base box 'puppet'...
[master] Matching MAC address for NAT networking...
[master] Setting the name of the VM...
[master] Clearing any previously set forwarded ports...
[master] Creating shared folders metadata...
[master] Clearing any previously set network interfaces...
[master] Preparing network interfaces based on configuration...
[master] Forwarding ports...
[master] -- 22 => 2222 (adapter 1)
[master] Booting VM...
[master] Waiting for VM to boot. This can take a few minutes.
[master] VM booted and ready for use!
[master] Configuring and enabling network interfaces...
[master] Mounting shared folders...
[master] -- /vagrant
[illegible]
```

## vagrant up agent



```
Gouravs-MacBook-Pro:multinode gouravshah$ vagrant up agent
Bringing machine 'agent' up with 'virtualbox' provider...
[agent] Importing base box 'puppet'...
[agent] Matching MAC address for NAT networking...
[agent] Setting the name of the VM...
[agent] Clearing any previously set forwarded ports...
[agent] Fixed port collision for 22 => 2222. Now on port 2200.
[agent] Creating shared folders metadata...
[agent] Clearing any previously set network interfaces...
[agent] Preparing network interfaces based on configuration...
[agent] Forwarding ports...
[agent] -- 22 => 2200 (adapter 1)
[agent] Booting VM...
[agent] Waiting for VM to boot. This can take a few minutes.
[agent] VM booted and ready for use!
[agent] Configuring and enabling network interfaces...
[agent] Mounting shared folders...
[agent] -- /vagrant
```

# VirtualBox GUI



box1_default_1377430...
Saved
multinode_master_137...
Running
multinode_agent_1377...
Running

**General**

Name: box1_default_1377430914
Operating System: Red Hat (64 bit)

**System**

Base Memory: 480 MB
Boot Order: Hard Disk, CD/DVD-ROM
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX

**Display**

Video Memory: 8 MB
Remote Desktop Server: Disabled

**Storage**

Controller: IDE Controller
IDE Primary Master: [CD/DVD] Empty
IDE Secondary Master: [CD/DVD] Empty
Controller: SATA Controller
SATA Port 0: box-disk1.vmdk (Normal, 500.00 GB)

**Audio**

Disabled

**Network**

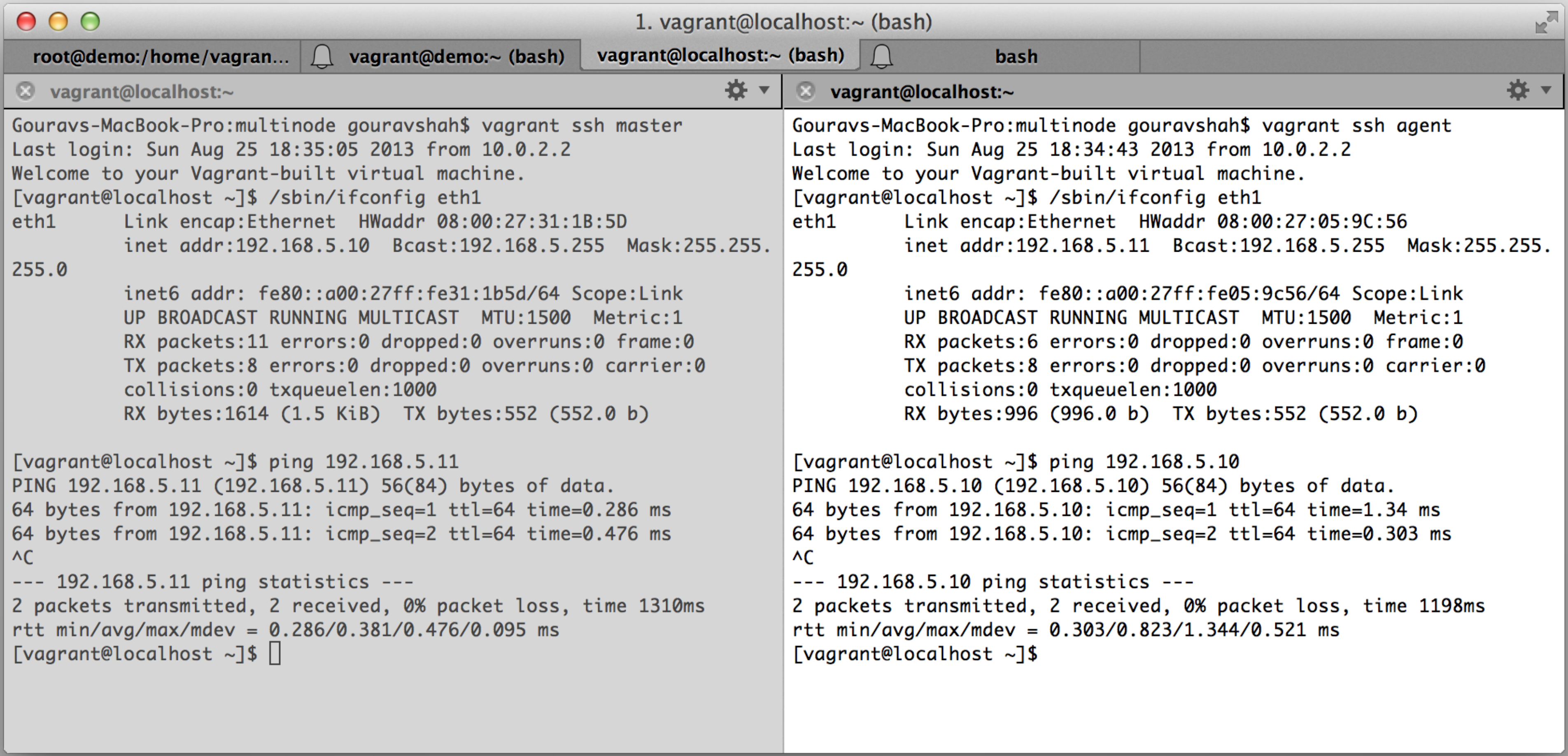
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

**USB**

# vagrant ssh

vagrant ssh master

vagrant ssh agent


1. vagrant@localhost:~ (bash)
root@demo:/home/vagran...
vagrant@demo:~ (bash)
vagrant@localhost:~ (bash)
bash
vagrant@localhost:~
vagrant@localhost:~

```
Gouravs-MacBook-Pro:multinode gouravshah$ vagrant ssh master
Last login: Sun Aug 25 18:35:05 2013 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ /sbin/ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:31:1B:5D
          inet addr:192.168.5.10  Bcast:192.168.5.255  Mask:255.255.
255.0
          inet6 addr: fe80::a00:27ff:fe31:1b5d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1614 (1.5 KiB)  TX bytes:552 (552.0 b)

[vagrant@localhost ~]$ ping 192.168.5.11
PING 192.168.5.11 (192.168.5.11) 56(84) bytes of data.
64 bytes from 192.168.5.11: icmp_seq=1 ttl=64 time=0.286 ms
64 bytes from 192.168.5.11: icmp_seq=2 ttl=64 time=0.476 ms
^C
--- 192.168.5.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1310ms
rtt min/avg/max/mdev = 0.286/0.381/0.476/0.095 ms
[vagrant@localhost ~]$
```

```
Gouravs-MacBook-Pro:multinode gouravshah$ vagrant ssh agent
Last login: Sun Aug 25 18:34:43 2013 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ /sbin/ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:05:9C:56
          inet addr:192.168.5.11  Bcast:192.168.5.255  Mask:255.255.
255.0
          inet6 addr: fe80::a00:27ff:fe05:9c56/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:996 (996.0 b)  TX bytes:552 (552.0 b)

[vagrant@localhost ~]$ ping 192.168.5.10
PING 192.168.5.10 (192.168.5.10) 56(84) bytes of data.
64 bytes from 192.168.5.10: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from 192.168.5.10: icmp_seq=2 ttl=64 time=0.303 ms
^C
--- 192.168.5.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1198ms
rtt min/avg/max/mdev = 0.303/0.823/1.344/0.521 ms
[vagrant@localhost ~]$
```

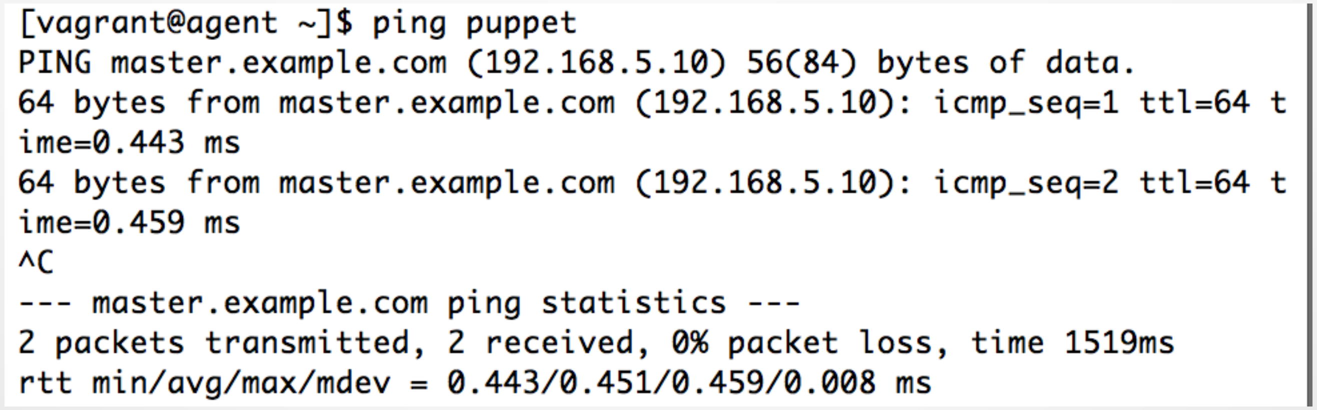
## Add hostname entries to /etc/hosts

- On both, Master and Agent
- Edit /etc/hosts file and add an entry for hostname

```
192.168.5.10   master.example.com master puppet
192.168.5.11   agent.example.com agent
```

## Validate

- Logout, login and run $hostname command on master and agent vms
- From agent vm, ping puppet
  - $ ping  puppet



```
[vagrant@agent ~]$ ping puppet
PING master.example.com (192.168.5.10) 56(84) bytes of data.
64 bytes from master.example.com (192.168.5.10): icmp_seq=1 ttl=64 t
ime=0.443 ms
64 bytes from master.example.com (192.168.5.10): icmp_seq=2 ttl=64 t
ime=0.459 ms
^C
--- master.example.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1519ms
rtt min/avg/max/mdev = 0.443/0.451/0.459/0.008 ms
```

## Enable Puppet Repository on Master

```
$ sudo su

$ cp  -r   /etc/yum.repos.d/repo,disabled/*   /etc/yum.repos.d/

$ sed  -i    's/enabled\=0/enabled\=1/g'    /etc/yum.repos.d/puppetlabs.repo
```

## Install Puppet

- On Master VM

  $ sudo yum install puppet-server

- On Agent VM

  $ sudo yum install puppet

That installed Puppet Master and Agent packages on respective hosts. Lets now look at the next steps.

## Post Install

- Agent Configurations
- Master Configurations
- Start and Enable the Puppet Services

## puppet.conf

- Puppet's main configuration file is at the following location on master and agent both
- Apart from the main config, there are additional config files which need to be created as required

**/etc/puppet/puppet.conf**

Lets look  at some configuration parameters.  These need not be changed for our installation as defaults are enough.

## On Agent Nodes

[agent]  or [main] block

- **server**: The hostname of your puppet master server. Defaults to puppet.
- **report**: Most users should set this to true.
- **pluginsync**: Most users should set this to true.
- **certname**: The sitewide unique identifier for this node. Defaults to the node’s fully qualified domain name, which is usually fine.

## Config Blocks

| | |
|---|---|
| • [main] | => all |
| • [master] | => master, cert |
| • [agent] | => agent |
| • [user] | => apply |

[main] is the least specific block

## Config Overrides

Note: puppet masters are usually also agent nodes; settings in [main] will be available to both services, and settings in the [master] and [agent] blocks will override the settings in [main].

## On Older Versions

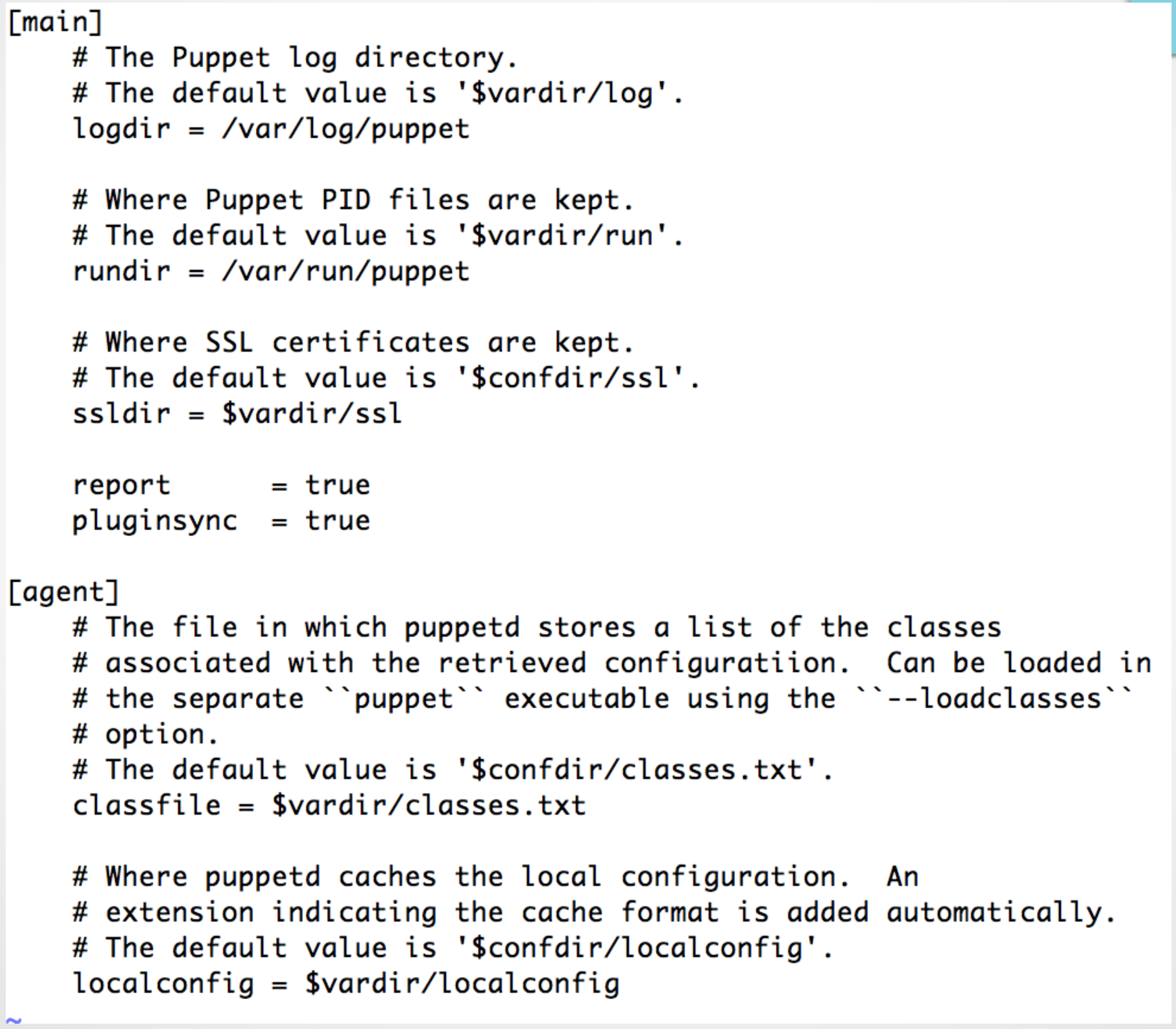
On 0.25.5 and older, Puppet Blocks were named as,

- [puppetd]
- [puppetmasterd]
- [puppet]
- [puppetca]

## Per-environment Blocks

- Are most specific
- Can override settings in the run mode block
- Settings supported:
  - modulepath
  - manifest
  - manifestdir
  - templatedir
- Corresponds to the $environment set on the agent node

# /etc/puppet/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

    report      = true
    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
~
```

## .puppet

When Puppet is not running as root (*nix) or not running with elevated privileges (Windows), it will read its config files from the .puppet directory in the current user’s home directory.

## Config Style

- Uses INI style config format
- Contains [config blocks]
- Indented groups of setting = value
- Multiple values can be separated with a comma
- $environment has special behavior
- For settings that accept only a single file or directory, you can set the owner, group, and/or mode by putting their desired states in curly braces after the value.

## Verifying Configs

--configprint <setting>

```
$ puppet master --configprint modulepath
  /etc/puppet/modules:/usr/share/puppet/modules

$puppet agent --environment testing --configprint modulepath
```

## More Configs (on Master)

- auth.conf
- fileserver.conf
- tagmail.com
- autosign.comf
- device.conf

## Auth.conf : Who gets What

- Configures Puppet's HTTP API access
- /etc/puppet/auth.conf
- ACL Stanzas
  - Specific at the top
  - Generic at the bottom

## Example auth.conf

```
# allow nodes to retrieve their own catalog
path ~ ^/catalog/([^/]+)$
method find
allow $1

# allow nodes to retrieve their own node definition
path ~ ^/node/([^/]+)$
method find
allow $1

# allow all nodes to access the certificates services
path /certificate_revocation_list/ca
method find
allow *

# allow all nodes to store their own reports
path ~ ^/report/([^/]+)$
method save
allow $1
```

## Autosign.conf

- Configured who gets certs automatically signed
- /etc/puppet/autosign.conf
- List of certnames and certname globs
- Resemble fqdns
- Use only in private networks

## Autosign.conf example

```
demo.example.com
*.example.com
*.local
```

## Device.conf

- Puppet device added in version 2.7
- Configures Network Hardware
- Requires devices be configured in /etc/puppet/device.conf

```
[device certname]
    type <type>
    url <url>
[router6.example.com]
    type cisco
    url ssh://admin:password@ad532fa.local
```

## Fileserver.conf

- Is not necessary (remember module files?)
- Creates additional mount points
- Contains mount-point stanzas
- Combination of puppet.conf and auth.conf

```
# Files in the /path/to/files directory will be served
# at puppet:///mount_point/.
[mount_point]
    path /path/to/files
    allow *.example.com
    deny *.wireless.example.com
```

## Tagmail.conf

- Notification service
- Send emails to group of users when the resources are changed
- /etc/puppet/tagmail.conf
- A comma seperated list of tags and !ntegrated tags
    - Explicit tags
    - Class names
    - all
    - log level (debug, crit, ingo, emerg etc..)

## Tagmail Prerequisites

- Set report = true on agent nodes
- Set reports = tagmail on master
- Report from address, smtp settings

## Example tagmail.conf

```
all: log-archive@example.com
webserver, !mailserver: httpadmins@example.com
emerg, crit: superman@example.com, batman@example.com, ironman@example.com
```

will mail any resource events tagged with webserver but not with mailserver to the httpadmins group; any emergency or critical events to to Superman, Batman, and Ironman, and all events to the log-archive group.

## Start and Enable Puppet Services

Using puppet resource shell

On Master:

```
$ sudo puppet resource service puppetmaster ensure=running enable=true
```

On Agent:

```
$ sudo puppet resource service puppet ensure=running enable=true
```

## If puppet master and agent were humans...

This is how they will communicate ...

Agent:   Hello Master ! What do you have for me today

Master: Alright! I verified your ID . I can see  You belong to this department, and you have been assigned that task list. Now go and complete that. And yeah.. dont forget to report me back

## Communication

The puppet agent and the puppet master server communicate via **HTTPS over host-verified SSL**.

## Workflow Diagram

http://docs.puppetlabs.com/puppet/3/reference/images/agent-master-https-sequence-large.gif

## Catalogs and Manifests

Running Puppet in agent/master mode works much the same way — the main difference is that it moves the manifests and compilation to the puppet master server. Agents don’t have to see any manifest files at all, and have no access to configuration information that isn’t in their own catalog.

source: docs.puppetlabs.com

## Puppet agent command

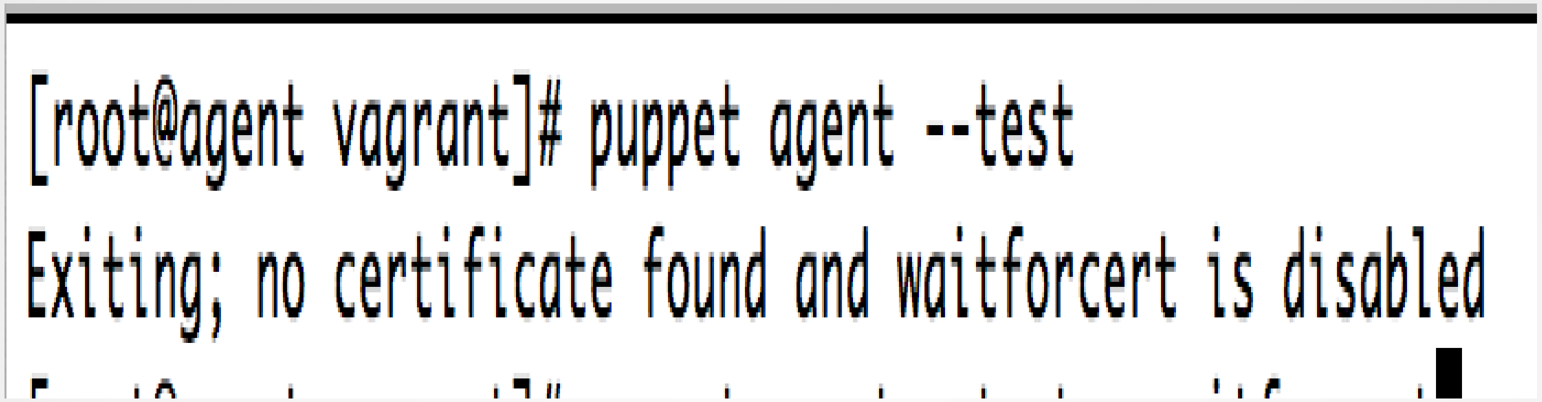
- Fetches configurations from a master server
- Has two main modes,

    - Daemonize
    - Run once and quit (--test)

## Saying Hi

```
$ puppet agent –test
```

## What Happened ?



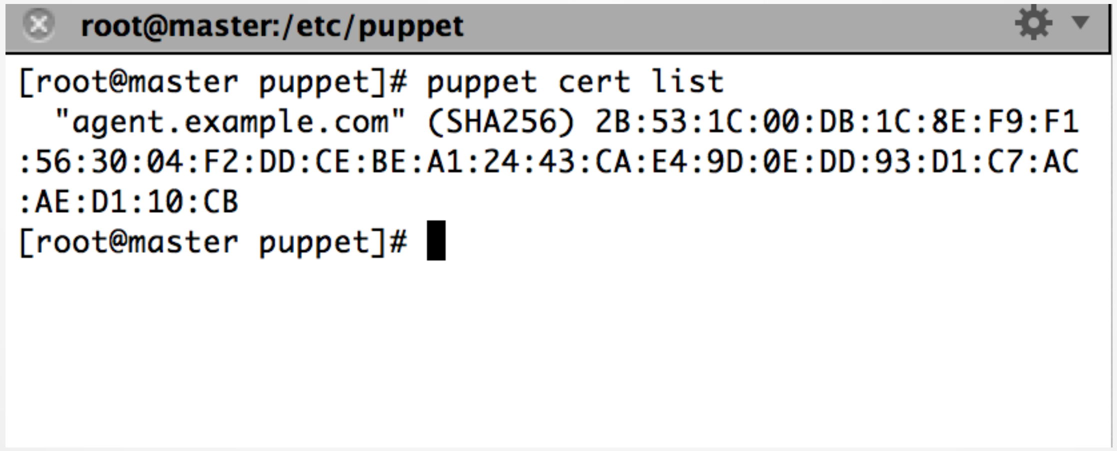
```
[root@agent vagrant]# puppet agent --test
Exiting; no certificate found and waitforcert is disabled
```

Puppet agent found the puppet master, but it got stopped at the certificate roadblock. It isn’t authorized to fetch configurations, so the master is turning it away.

## Authorizing

- Check the certificate request on master

  $ puppet cert list



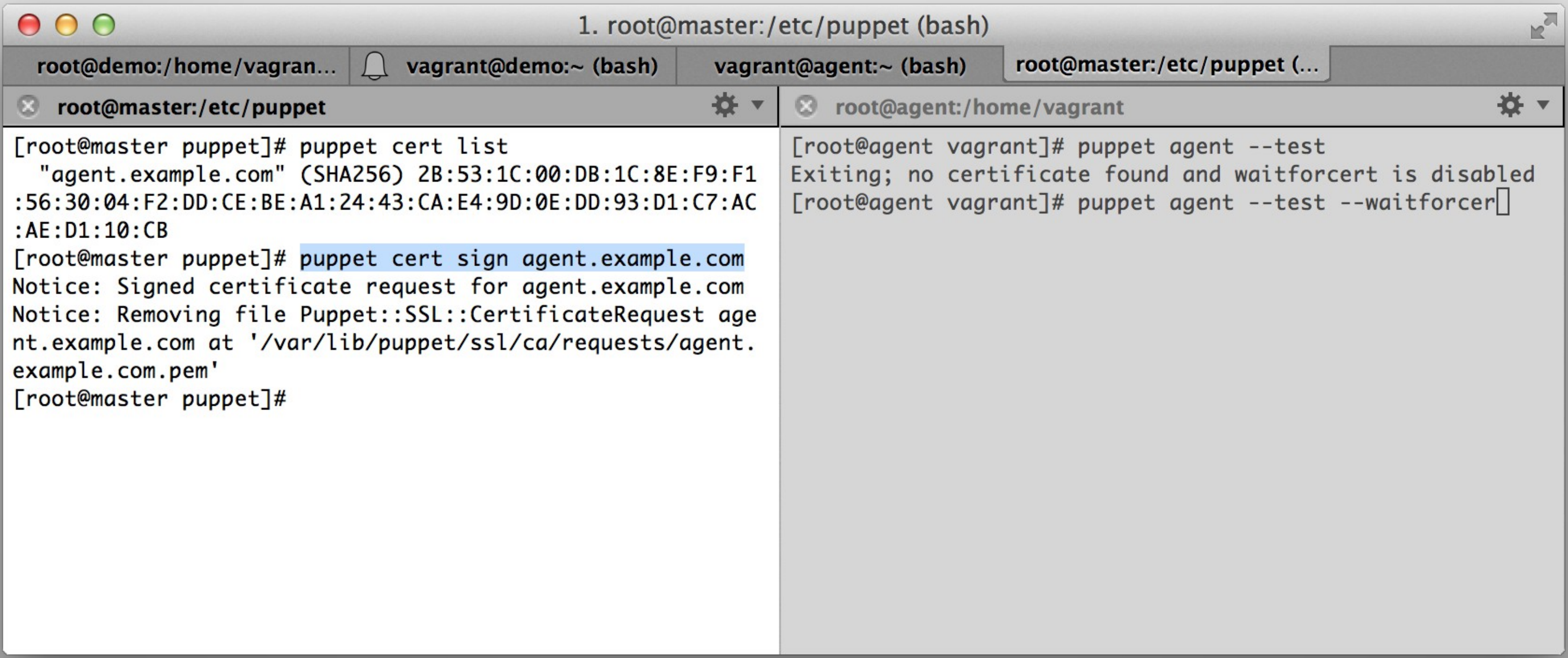
root@master:/etc/puppet

```
[root@master puppet]# puppet cert list
  "agent.example.com" (SHA256) 2B:53:1C:00:DB:1C:8E:F9:F1
:56:30:04:F2:DD:CE:BE:A1:24:43:CA:E4:9D:0E:DD:93:D1:C7:AC
:AE:D1:10:CB
[root@master puppet]# 
```

## Troubleshooting

- The VMs can ping each other
- The agent can resolve the puppet master by host name
- The agent's /etc/puppetlabs/puppet/puppet.conf file has a server setting (in the [agent] block) of puppet or learn.localdomain
- The VMs' clocks are in sync

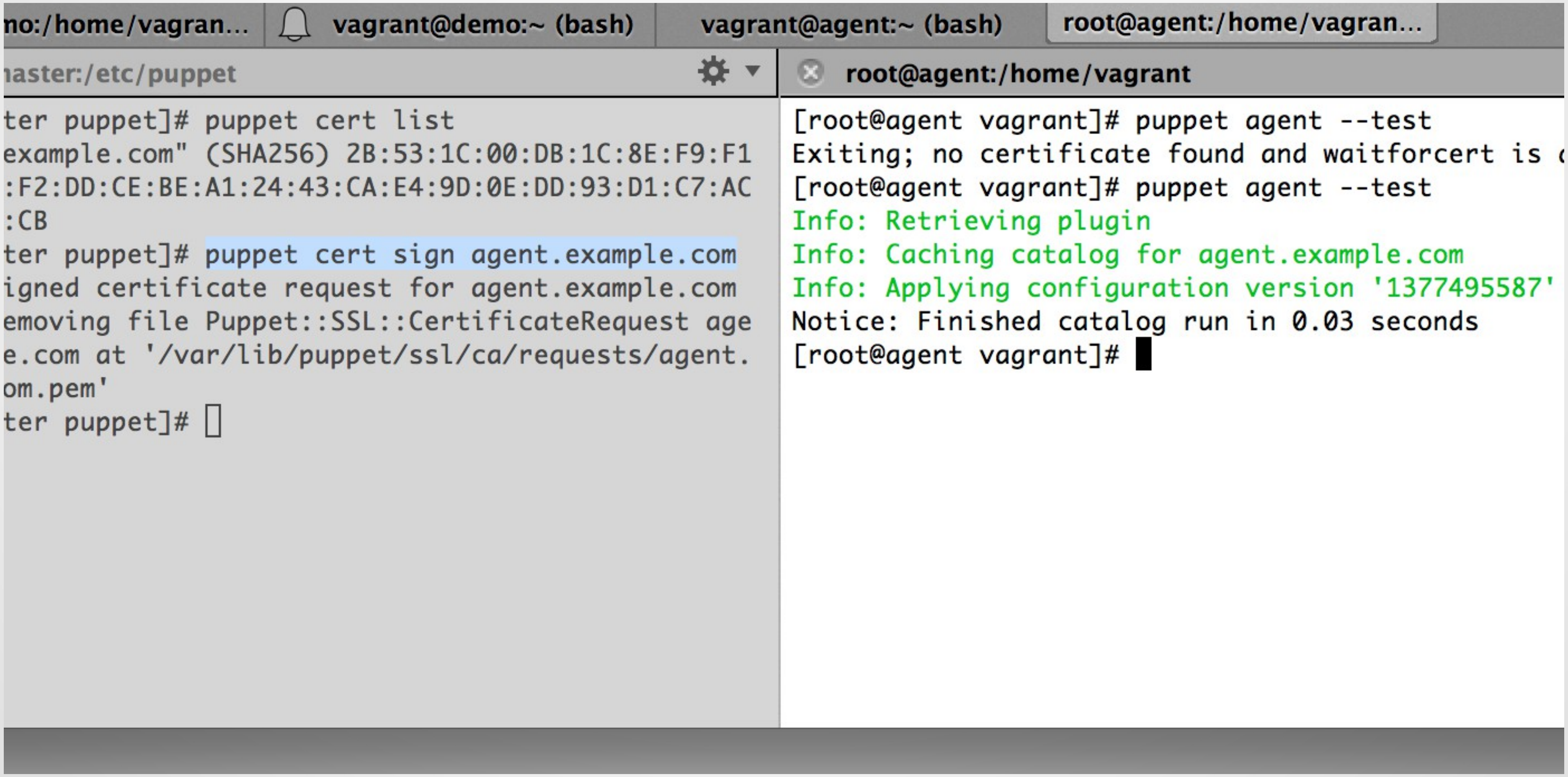
# $puppet cert sign agent.example.com



```
[root@master puppet]# puppet cert list
  "agent.example.com" (SHA256) 2B:53:1C:00:DB:1C:8E:F9:F1
:56:30:04:F2:DD:CE:BE:A1:24:43:CA:E4:9D:0E:DD:93:D1:C7:AC
:AE:D1:10:CB
[root@master puppet]# puppet cert sign agent.example.com
Notice: Signed certificate request for agent.example.com
Notice: Removing file Puppet::SSL::CertificateRequest age
nt.example.com at '/var/lib/puppet/ssl/ca/requests/agent.
example.com.pem'
[root@master puppet]#
```

```
[root@agent vagrant]# puppet agent --test
Exiting; no certificate found and waitforcert is disabled
[root@agent vagrant]# puppet agent --test --waitforcer
```

## $puppet agent --test



```
ter puppet]# puppet cert list
example.com" (SHA256) 2B:53:1C:00:DB:1C:8E:F9:F1
:F2:DD:CE:BE:A1:24:43:CA:E4:9D:0E:DD:93:D1:C7:AC
:CB
ter puppet]# puppet cert sign agent.example.com
igned certificate request for agent.example.com
emoving file Puppet::SSL::CertificateRequest age
e.com at '/var/lib/puppet/ssl/ca/requests/agent.
om.pem'
ter puppet]#
```

```
[root@agent vagrant]# puppet agent --test
Exiting; no certificate found and waitforcert is
[root@agent vagrant]# puppet agent --test
Info: Retrieving plugin
Info: Caching catalog for agent.example.com
Info: Applying configuration version '1377495587'
Notice: Finished catalog run in 0.03 seconds
[root@agent vagrant]#
```

It worked !!

## What happened ?

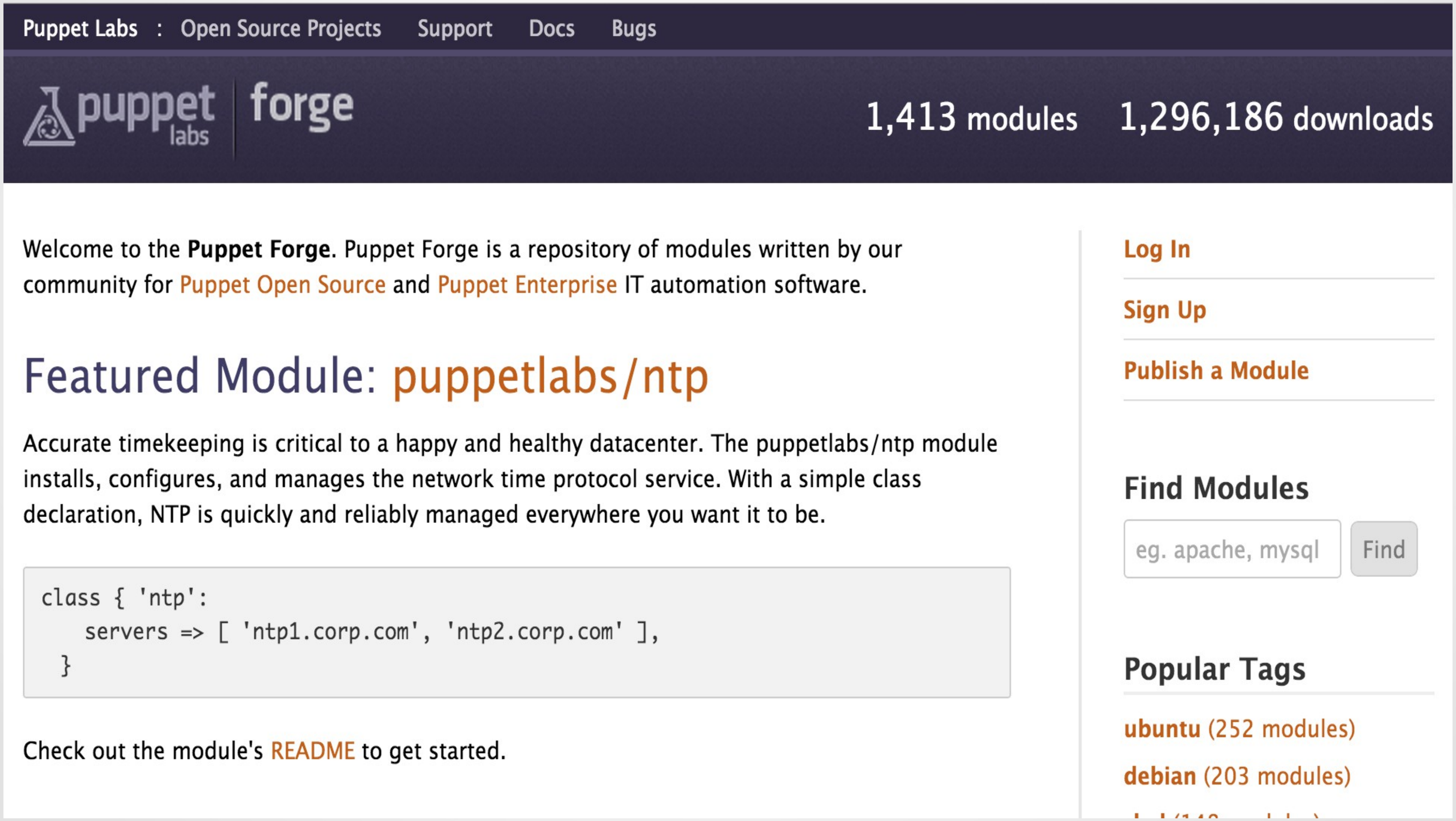
Puppet uses SSL certificates to protect communications between agents and the master. Since agents can’t do a full run without a certificate, our agent had to ask for one and then wait for the request to get approved.

Lets serve some real configurations.....

And this time we are going to see the magic of Infrastructure as a Code

## In comes puppet forge...

The Puppet Forge is a repository of pre-existing modules, written and contributed by users. These modules solve a wide variety of problems so using them can save you time and effort.



Puppet Labs : Open Source Projects Support Docs Bugs

puppet labs | forge

1,413 modules 1,296,186 downloads

Welcome to the **Puppet Forge**. Puppet Forge is a repository of modules written by our community for Puppet Open Source and Puppet Enterprise IT automation software.

## Featured Module: puppetlabs/ntp

Accurate timekeeping is critical to a happy and healthy datacenter. The puppetlabs/ntp module installs, configures, and manages the network time protocol service. With a simple class declaration, NTP is quickly and reliably managed everywhere you want it to be.

```
class { 'ntp':
   servers => [ 'ntp1.corp.com', 'ntp2.corp.com' ],
 }
```

Check out the module's README to get started.

Log In

Sign Up

Publish a Module

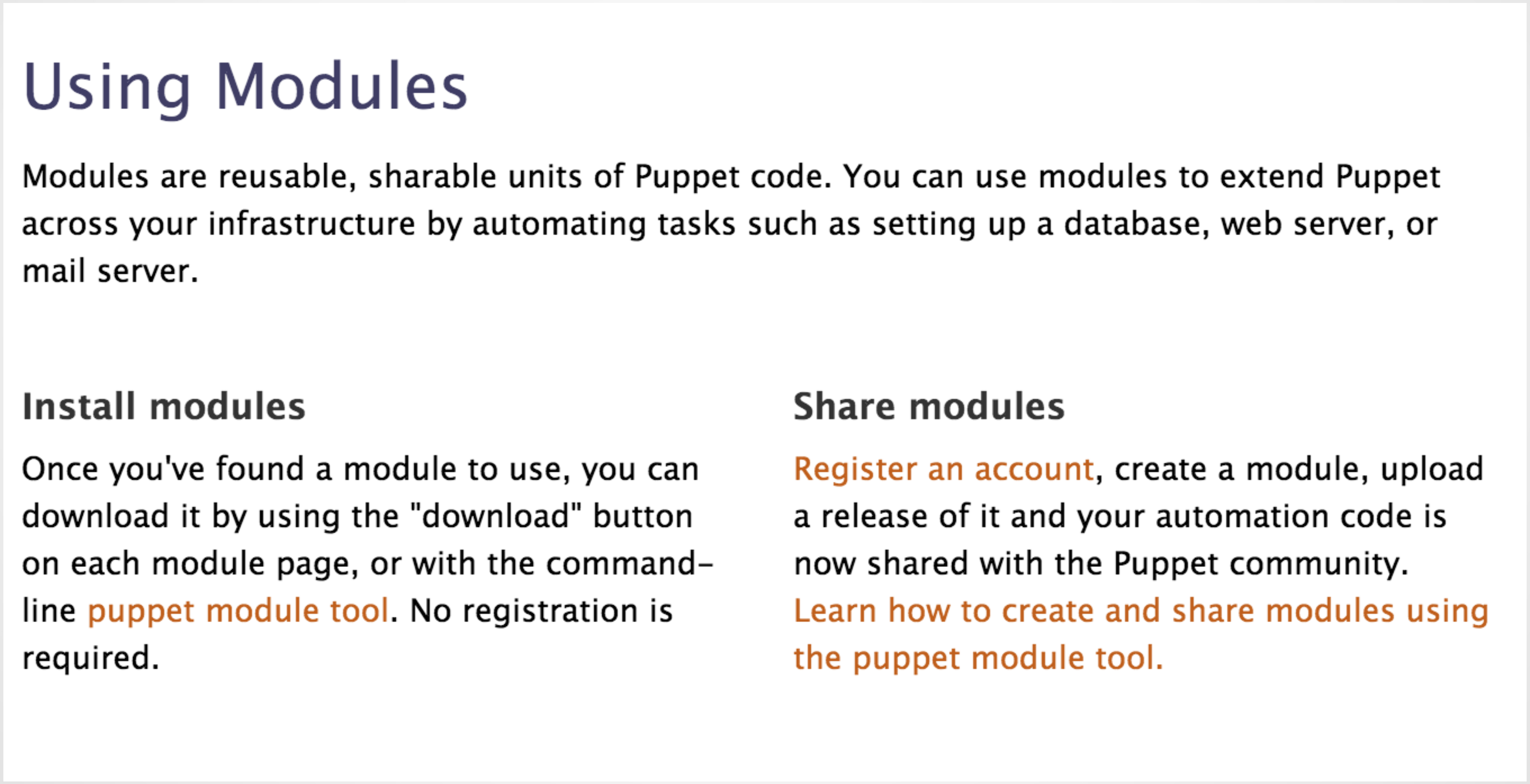
### Find Modules

### Popular Tags

**ubuntu** (252 modules)

**debian** (203 modules)

## Puppetforge



### Using Modules

Modules are reusable, sharable units of Puppet code. You can use modules to extend Puppet across your infrastructure by automating tasks such as setting up a database, web server, or mail server.

#### Install modules

Once you've found a module to use, you can download it by using the "download" button on each module page, or with the command-line puppet module tool. No registration is required.

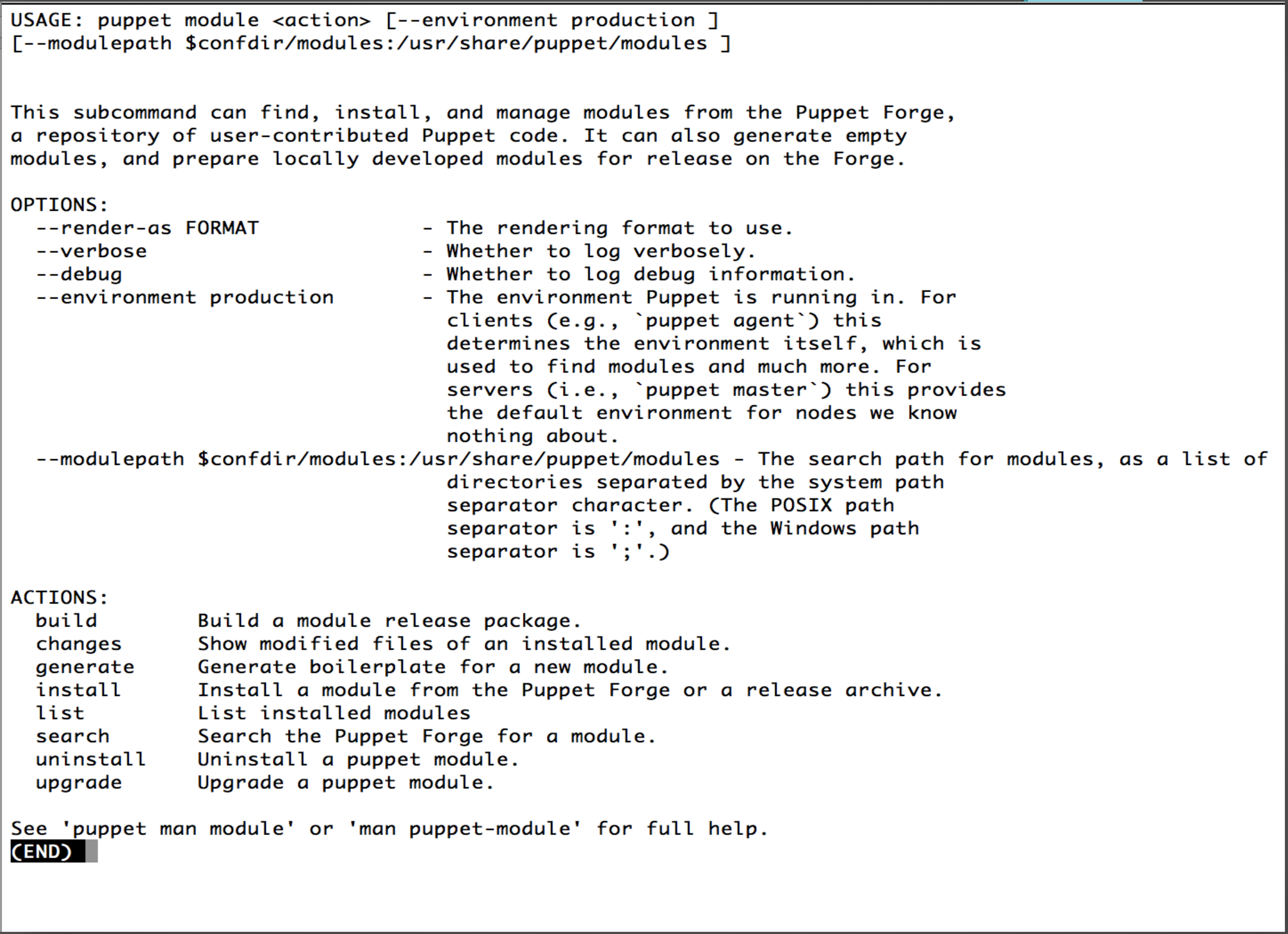
#### Share modules

Register an account, create a module, upload a release of it and your automation code is now shared with the Puppet community. Learn how to create and share modules using the puppet module tool.

## puppet module subcommand

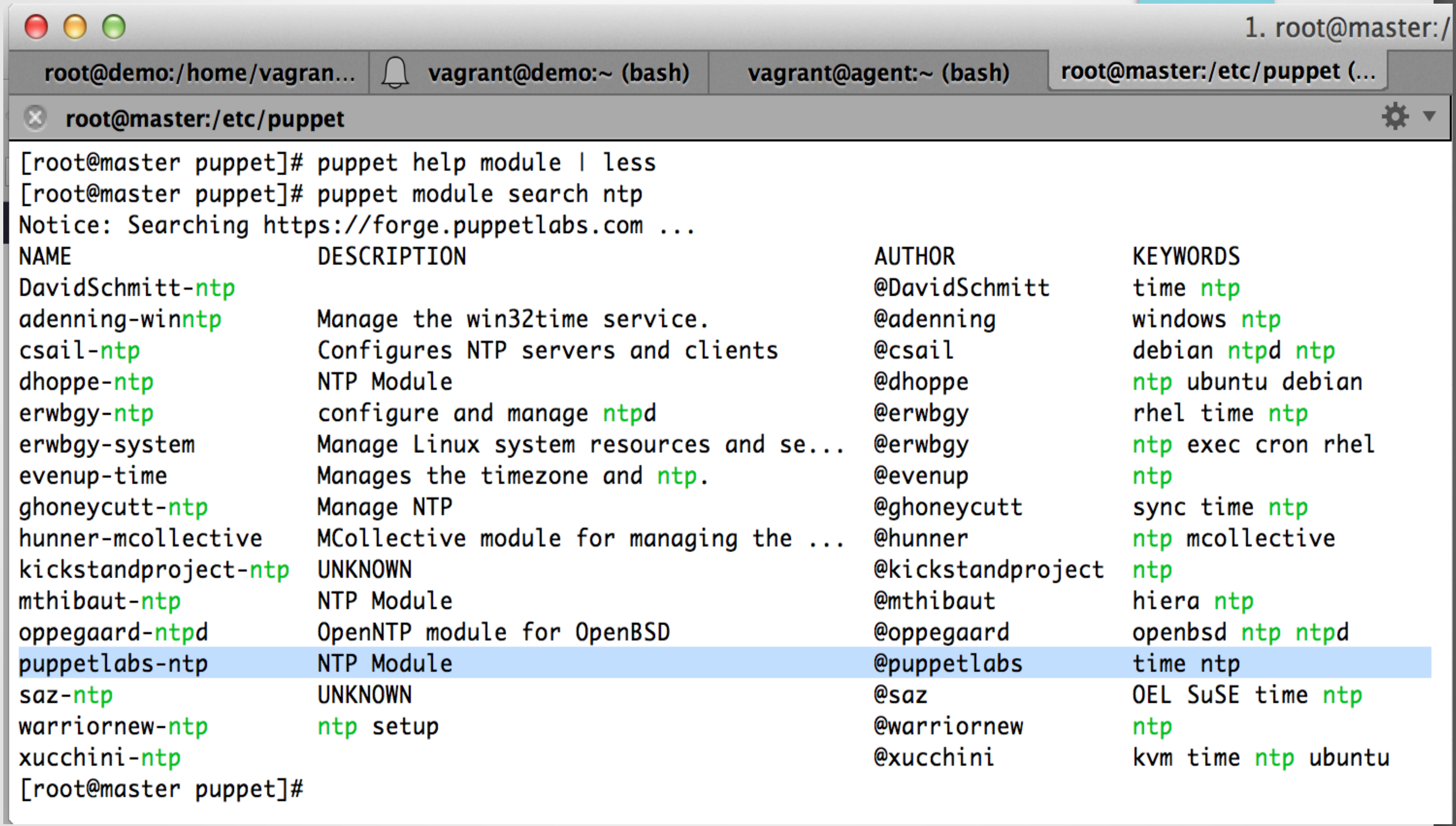
•can find, install, and manage modules from the Puppet Forge, a repository of user-contributed Puppet code.

•It can also generate empty modules, and prepare locally developed modules for release on the Forge.



```
USAGE: puppet module <action> [--environment production ]
[--modulepath $confdir/modules:/usr/share/puppet/modules ]


This subcommand can find, install, and manage modules from the Puppet Forge,
a repository of user-contributed Puppet code. It can also generate empty
modules, and prepare locally developed modules for release on the Forge.

OPTIONS:
  --render-as FORMAT             - The rendering format to use.
  --verbose                      - Whether to log verbosely.
  --debug                        - Whether to log debug information.
  --environment production       - The environment Puppet is running in. For
                                   clients (e.g., `puppet agent`) this
                                   determines the environment itself, which is
                                   used to find modules and much more. For
                                   servers (i.e., `puppet master`) this provides
                                   the default environment for nodes we know
                                   nothing about.
  --modulepath $confdir/modules:/usr/share/puppet/modules - The search path for modules, as a list of
                                   directories separated by the system path
                                   separator character. (The POSIX path
                                   separator is ':', and the Windows path
                                   separator is ';'.)

ACTIONS:
  build         Build a module release package.
  changes       Show modified files of an installed module.
  generate      Generate boilerplate for a new module.
  install       Install a module from the Puppet Forge or a release archive.
  list          List installed modules
  search        Search the Puppet Forge for a module.
  uninstall     Uninstall a puppet module.
  upgrade       Upgrade a puppet module.

See 'puppet man module' or 'man puppet-module' for full help.
(END)
```

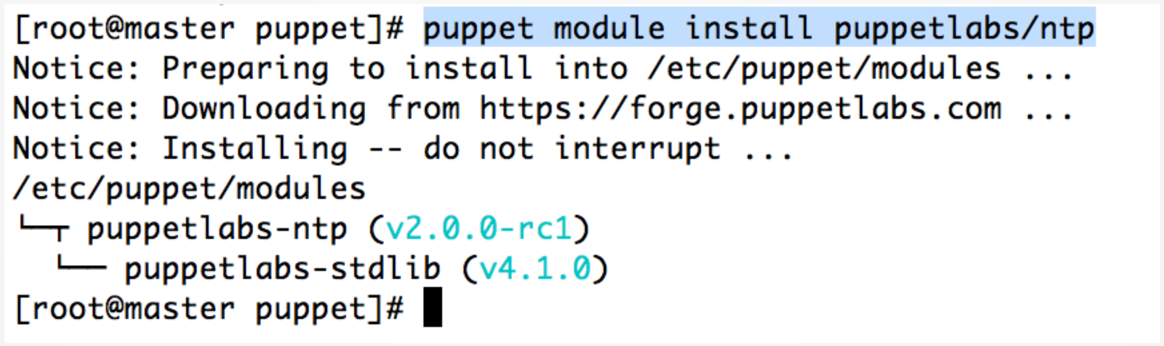
# Puppet module search ntp



```
[root@master puppet]# puppet help module | less
[root@master puppet]# puppet module search ntp
Notice: Searching https://forge.puppetlabs.com ...
NAME                   DESCRIPTION                              AUTHOR              KEYWORDS
DavidSchmitt-ntp                                                @DavidSchmitt       time ntp
adenning-winntp        Manage the win32time service.            @adenning           windows ntp
csail-ntp              Configures NTP servers and clients       @csail              debian ntpd ntp
dhoppe-ntp             NTP Module                               @dhoppe             ntp ubuntu debian
erwbgy-ntp             configure and manage ntpd                @erwbgy             rhel time ntp
erwbgy-system          Manage Linux system resources and se...  @erwbgy             ntp exec cron rhel
evenup-time            Manages the timezone and ntp.            @evenup             ntp
ghoneycutt-ntp         Manage NTP                               @ghoneycutt         sync time ntp
hunner-mcollective     MCollective module for managing the ...  @hunner             ntp mcollective
kickstandproject-ntp   UNKNOWN                                  @kickstandproject   ntp
mthibaut-ntp           NTP Module                               @mthibaut           hiera ntp
oppegaard-ntpd         OpenNTP module for OpenBSD               @oppegaard          openbsd ntp ntpd
puppetlabs-ntp         NTP Module                               @puppetlabs         time ntp
saz-ntp                UNKNOWN                                  @saz                OEL SuSE time ntp
warriornew-ntp         ntp setup                                @warriornew         ntp
xucchini-ntp                                                    @xucchini           kvm time ntp ubuntu
[root@master puppet]#
```

## Lets install ntp module

$ puppet module install puppetlabs/ntp



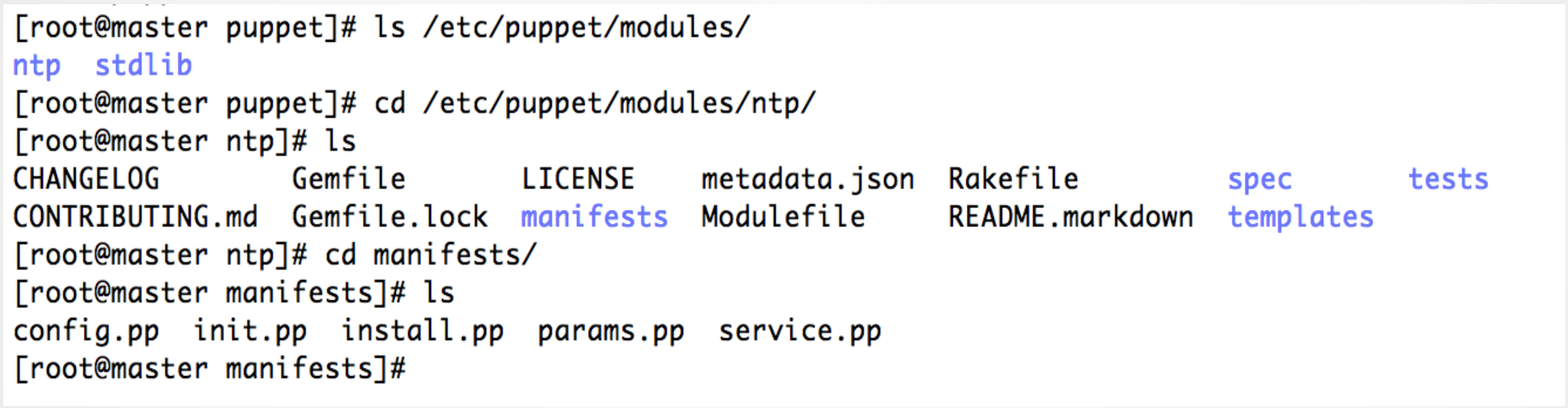
```
[root@master puppet]# puppet module install puppetlabs/ntp
Notice: Preparing to install into /etc/puppet/modules ...
Notice: Downloading from https://forge.puppetlabs.com ...
Notice: Installing -- do not interrupt ...
/etc/puppet/modules
└─┬ puppetlabs-ntp (v2.0.0-rc1)
  └── puppetlabs-stdlib (v4.1.0)
[root@master puppet]# █
```

## About NTP Module

The NTP module installs, configures, and manages the ntp service.

The NTP module handles running NTP across a range of operating systems and distributions. Where possible we use the upstream ntp templates so that the results closely match what you'd get if you modified the package default conf
files.

## Inspect

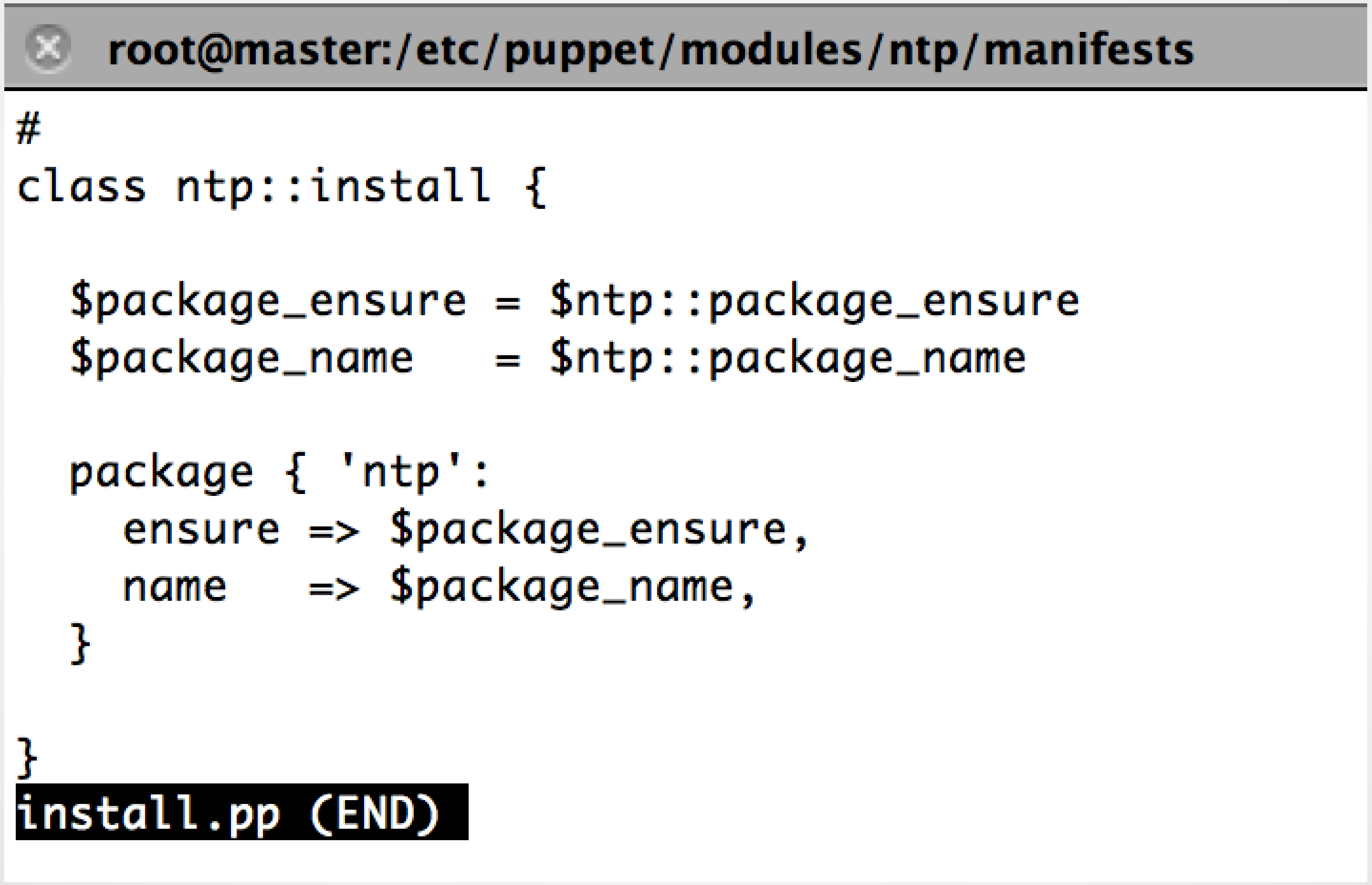


```
[root@master puppet]# ls /etc/puppet/modules/
ntp  stdlib
[root@master puppet]# cd /etc/puppet/modules/ntp/
[root@master ntp]# ls
CHANGELOG        Gemfile       LICENSE    metadata.json  Rakefile         spec       tests
CONTRIBUTING.md  Gemfile.lock  manifests  Modulefile     README.markdown  templates
[root@master ntp]# cd manifests/
[root@master manifests]# ls
config.pp  init.pp  install.pp  params.pp  service.pp
[root@master manifests]#
```

## Good Practice

Ready to use modules are great. However, you should always inspect them to know what exactly the code does.

## install.pp



root@master:/etc/puppet/modules/ntp/manifests

```
#
class ntp::install {

  $package_ensure = $ntp::package_ensure
  $package_name   = $ntp::package_name

  package { 'ntp':
    ensure => $package_ensure,
    name   => $package_name,
  }

}
install.pp (END)
```

## Exercise

- List modules installed on the system in tree format

## site.pp

```
import 'nodes.pp'
```

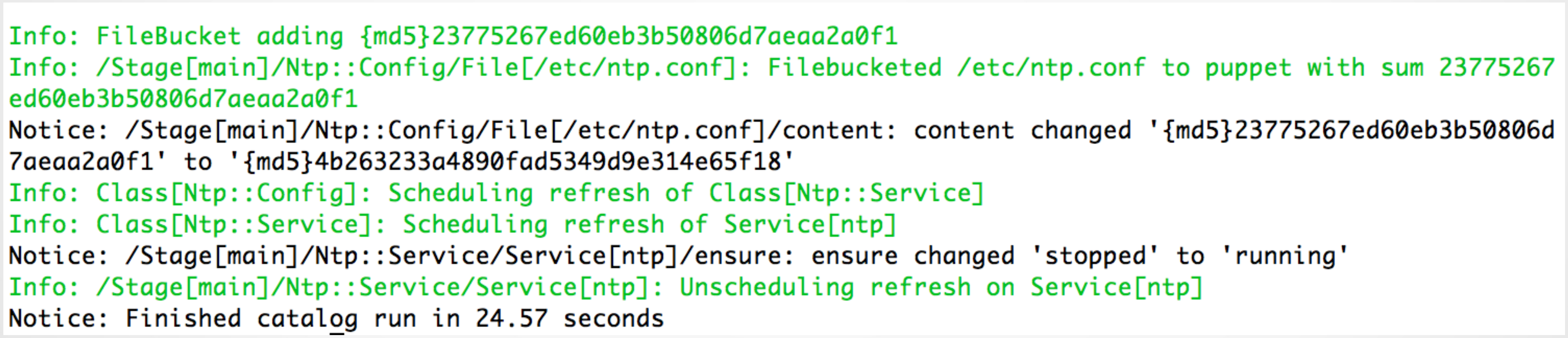
## nodes.pp

```
node 'agent.example.com' {
    include ntp
}
```

## Node Name = certname

An agent node’s name is almost always read from its certname setting, which is set at install time but can be changed later. The certname is usually (but not always) the node’s fully qualified domain name.

# puppet agent –test  (on agent)



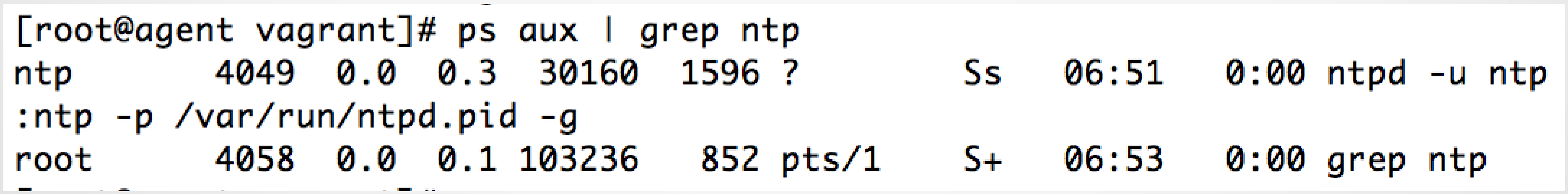
```
Info: FileBucket adding {md5}23775267ed60eb3b50806d7aeaa2a0f1
Info: /Stage[main]/Ntp::Config/File[/etc/ntp.conf]: Filebucketed /etc/ntp.conf to puppet with sum 23775267
ed60eb3b50806d7aeaa2a0f1
Notice: /Stage[main]/Ntp::Config/File[/etc/ntp.conf]/content: content changed '{md5}23775267ed60eb3b50806d
7aeaa2a0f1' to '{md5}4b263233a4890fad5349d9e314e65f18'
Info: Class[Ntp::Config]: Scheduling refresh of Class[Ntp::Service]
Info: Class[Ntp::Service]: Scheduling refresh of Service[ntp]
Notice: /Stage[main]/Ntp::Service/Service[ntp]/ensure: ensure changed 'stopped' to 'running'
Info: /Stage[main]/Ntp::Service/Service[ntp]: Unscheduling refresh on Service[ntp]
Notice: Finished catalog run in 24.57 seconds
```

## validate

```
$ ps aux | grep ntp
```

or

```
$ service ntpd status
```



```
[root@agent vagrant]# ps aux | grep ntp
ntp       4049  0.0  0.3  30160  1596 ?        Ss   06:51   0:00 ntpd -u ntp
:ntp -p /var/run/ntpd.pid -g
root      4058  0.0  0.1 103236   852 pts/1    S+   06:53   0:00 grep ntp
```

Congratulations !
You just learnt how to manage your first node with puppet master.

## Infrastructure as a Code  Beauty ..

And whats awesome about it is you just did it without writing a single line of code to install or configure NTP.

## Good Practice

Remember to contribute back to the community when you become a advanced user and write a interesting new module.