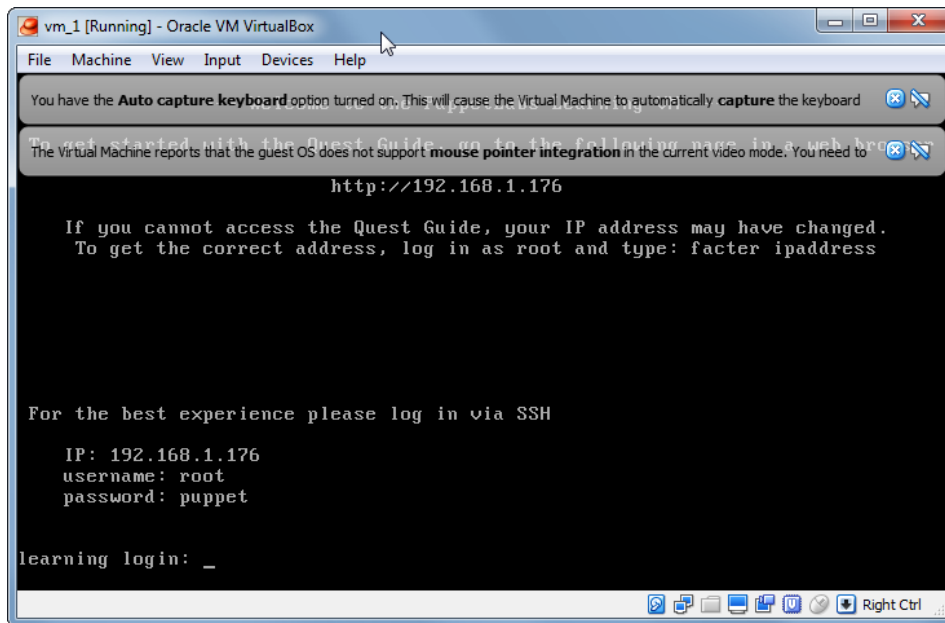


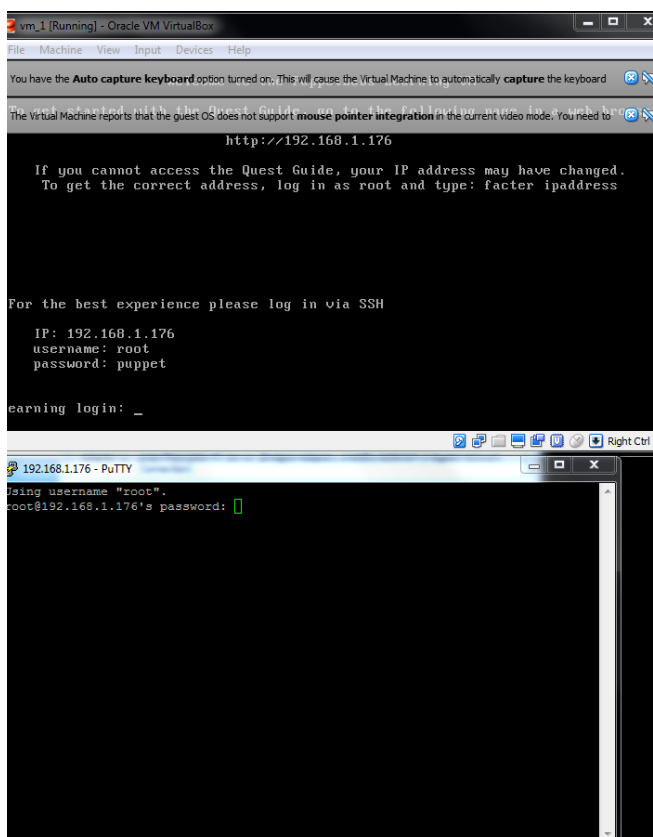
Quest To Learn Puppet

Background Info: Puppet is an open-source software configuration management tool. It runs on many Unix-like systems as well as on Microsoft Windows, and includes its own declarative language to describe system configuration.

- 1) Make note of Address e.g. 192.168.1.176



- 2) Enter connection details in Putty
root@<IPADDRESS>



3) Type Password after connecting e.g. puppet

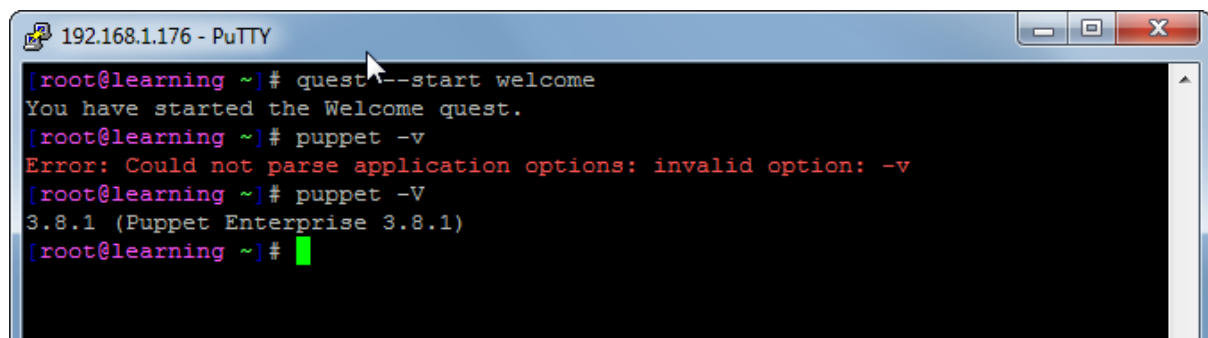
5) To access the quest guide enter the learning VM's IP address in your browser's address bar. (Be sure to use `http://<ADDRESS>` for the Quest Guide, as `https://<ADDRESS>`. This will take you to the PE console.

What is Puppet?

Puppet is an open-source IT automation tool. The Puppet Domain Specific Language (DSL) is a Ruby-based coding language that provides a precise and adaptable way to describe a desired state for each machine in your infrastructure. Once you've described a desired state, Puppet does the work to bring your systems in line and keep them there.

Task 1:

1) Check Version of Puppet Installed



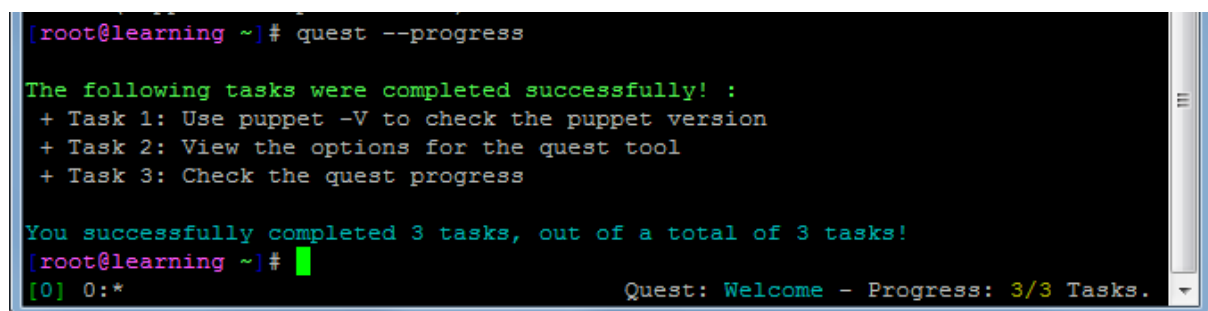
```
192.168.1.176 - PuTTY
[root@learning ~]# quest --start welcome
You have started the Welcome quest.
[root@learning ~]# puppet -v
Error: Could not parse application options: invalid option: -v
[root@learning ~]# puppet -V
3.8.1 (Puppet Enterprise 3.8.1)
[root@learning ~]#
```

Task 2 &3:

quest --help # Lists of all the options for the quest command

quest --progress # Display details of tasks completed

bash_history # Use if don't see your progress register i.e. file hasn't been initialized



```
[root@learning ~]# quest --progress

The following tasks were completed successfully! :
+ Task 1: Use puppet -V to check the puppet version
+ Task 2: View the options for the quest tool
+ Task 3: Check the quest progress

You successfully completed 3 tasks, out of a total of 3 tasks!
[root@learning ~]#
[0] 0:* Quest: Welcome - Progress: 3/3 Tasks.
```

Key Information:

You're going to get tripped up if you don't have a solid understanding of things like 'resources', 'classes', 'manifests', and 'modules'.

The Power of Puppet

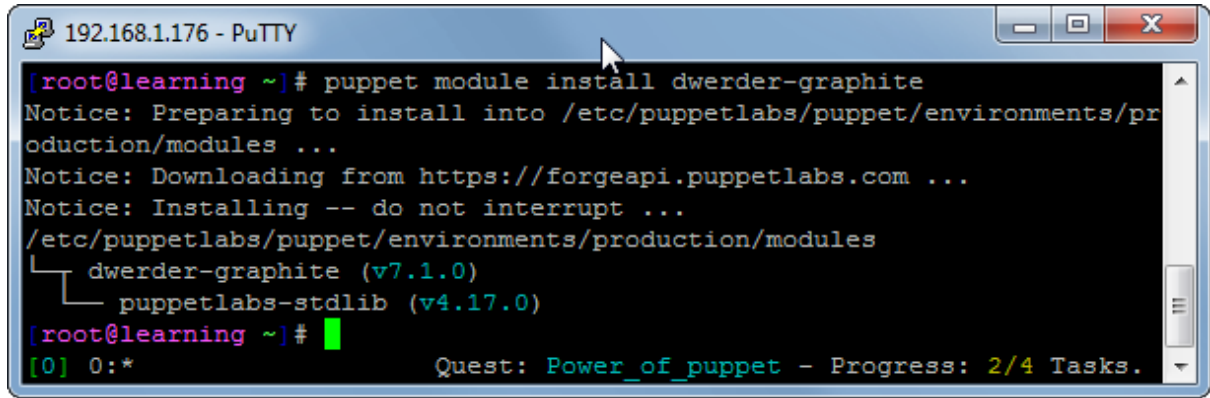
Task 1

Lheidar

```
# quest --start power_of_puppet
```

Task 2

```
# puppet module install dwerder-graphite
```



```
192.168.1.176 - PuTTY
[root@learning ~]# puppet module install dwerder-graphite
Notice: Preparing to install into /etc/puppetlabs/puppet/environments/pr
oduction/modules ...
Notice: Downloading from https://forgeapi.puppetlabs.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/puppet/environments/production/modules
└─ dwerder-graphite (v7.1.0)
   └─ puppetlabs-stdlib (v4.17.0)
[root@learning ~]#
```

[0] 0:* Quest: Power_of_puppet - Progress: 2/4 Tasks.

Task 3

Access the PE console

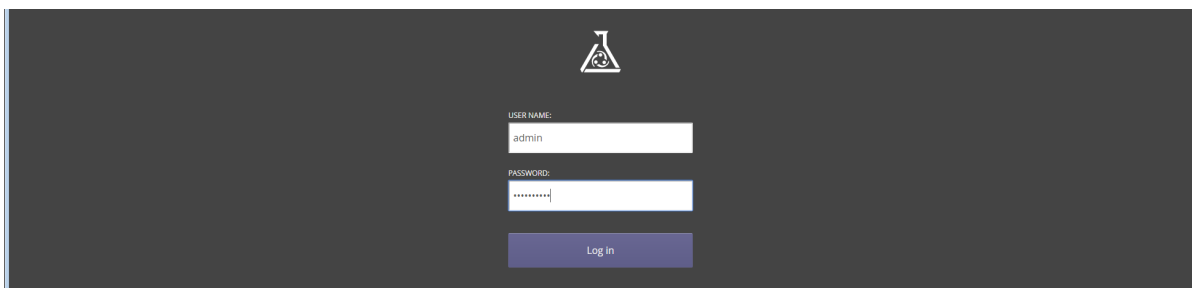
```
# facter ipaddress
```

<https://192.168.1.176/auth/login?redirect=%2F>

username: admin

password: puppetlabs

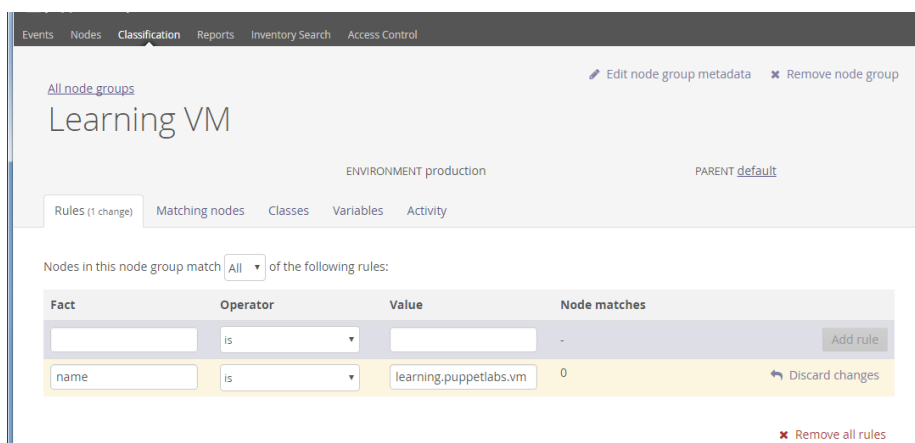
Note: bypass/continue notice in browser to continue to the console



The screenshot shows a login page with a dark background and a white triangle icon at the top. Below the icon are two input fields: 'USER NAME:' with 'admin' entered, and 'PASSWORD:' with masked characters. A 'Log in' button is positioned below the password field.

Creating Node Group

1) Classification Tab



The screenshot shows the 'Classification' tab in the Puppet Enterprise interface. The page title is 'Learning VM'. Below the title, there are tabs for 'Rules (1 change)', 'Matching nodes', 'Classes', 'Variables', and 'Activity'. The 'Rules' tab is active, showing a table of rules. The table has columns for 'Fact', 'Operator', 'Value', and 'Node matches'. There is one rule defined: 'name' is 'learning.puppetlabs.vm' with 0 matches. At the bottom right, there are buttons for 'Add rule', 'Discard changes', and 'Remove all rules'.

Fact	Operator	Value	Node matches
name	is	learning.puppetlabs.vm	0

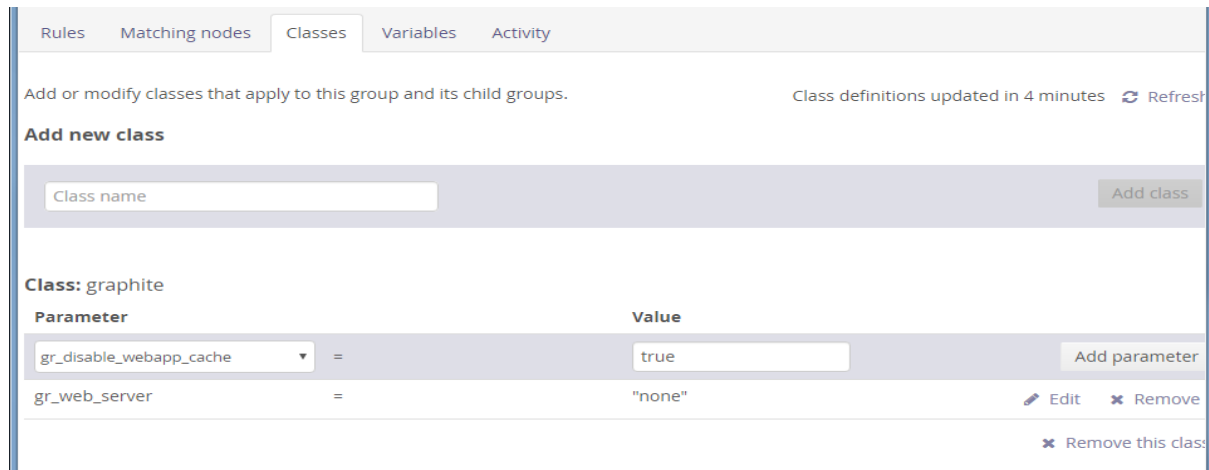
Lheidar

- 2) Creating rule that will match Learning VM's domain name: learning.puppetlabs.vm
- 3) Committing Changes

Adding a class

- 1) Enter graphite in the Class name text box, and then click the Add class button.

Note: To avoid conflicts disable Graphite's default web server settings.



Rules Matching nodes **Classes** Variables Activity

Add or modify classes that apply to this group and its child groups. Class definitions updated in 4 minutes Refresh

Add new class

Class name Add class

Class: graphite

Parameter		Value	
gr_disable_webapp_cache	=	true	Add parameter
gr_web_server	=	"none"	Edit Remove

Remove this class

Under Class: graphite section, set gr_web_server parameter to "none", and the gr_disable_webapp_cache parameter to true

Task 4

Note: puppet master node controlling a collection of agent nodes. Installing modules is a puppet master thing and puppet runs are a puppet agent thing.

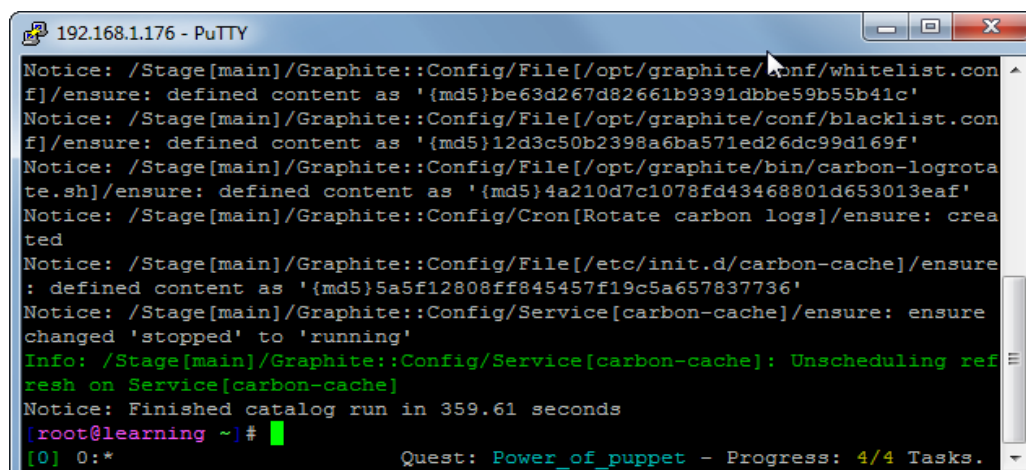
Problem Encountered:

Error: Issue with installed Graphite local dependencies

Solution:

Use required cached versions of all the modules. The command below installs the modules for all of the quests.

for m in /usr/src/forge/*; do puppet module install \$m ; done



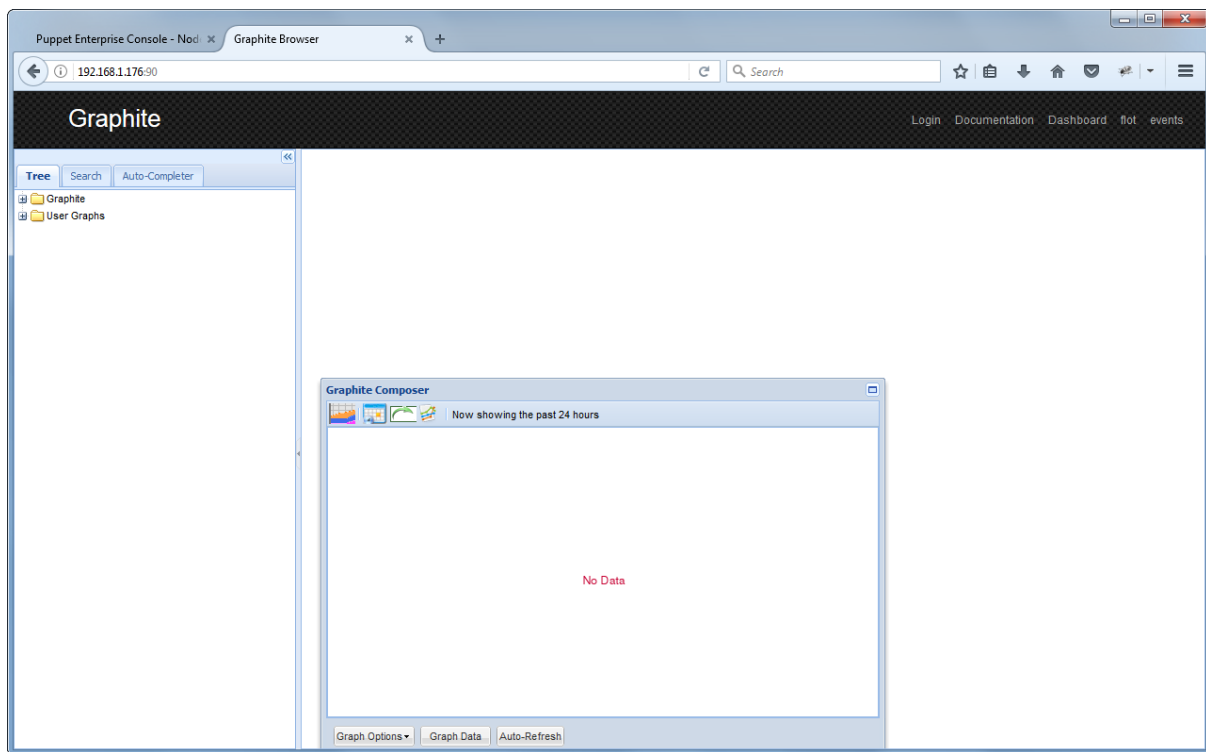
```
192.168.1.176 - PuTTY
Notice: /Stage[main]/Graphite::Config/File[/opt/graphite/conf/whitelist.conf]/ensure: defined content as '{md5}be63d267d82661b9391dbbe59b55b41c'
Notice: /Stage[main]/Graphite::Config/File[/opt/graphite/conf/blacklist.conf]/ensure: defined content as '{md5}12d3c50b2398a6ba571ed26dc99d169f'
Notice: /Stage[main]/Graphite::Config/File[/opt/graphite/bin/carbon-logrotate.sh]/ensure: defined content as '{md5}4a210d7c1078fd43468801d653013eaf'
Notice: /Stage[main]/Graphite::Config/Cron[Rotate carbon logs]/ensure: created
Notice: /Stage[main]/Graphite::Config/File[/etc/init.d/carbon-cache]/ensure: defined content as '{md5}5a5f12808ff845457f19c5a657837736'
Notice: /Stage[main]/Graphite::Config/Service[carbon-cache]/ensure: ensure changed 'stopped' to 'running'
Info: /Stage[main]/Graphite::Config/Service[carbon-cache]: Unscheduling refresh on Service[carbon-cache]
Notice: Finished catalog run in 359.61 seconds
[root@learning ~]#
[0] 0:* Quest: Power_of_puppet - Progress: 4/4 Tasks.
```

Lheidar

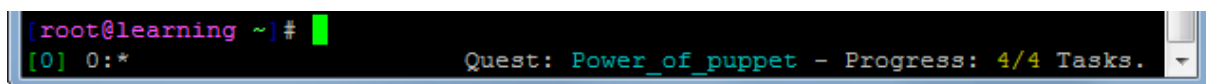
#command to manually trigger a puppet run
puppet agent --test

Graphite is up and running, its API is available for generating graphs suitable for including in a dashboard.

Below is Graphite console running on port 90.



Completion of Task:

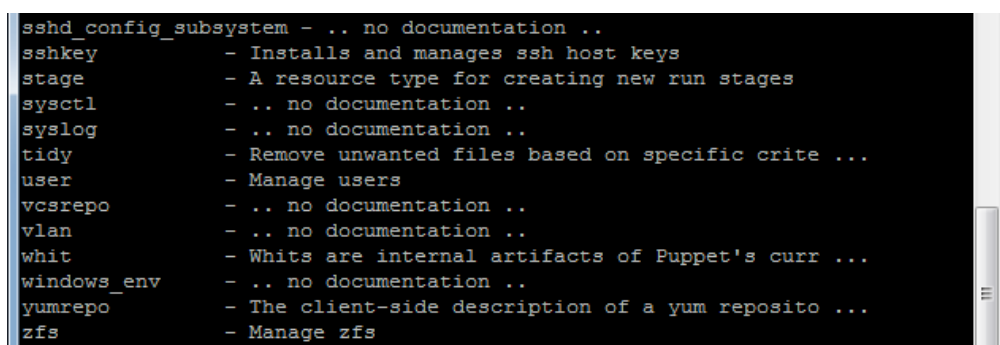


Resources:

Task 1:

Listing all the built-in resources types available

puppet describe --list



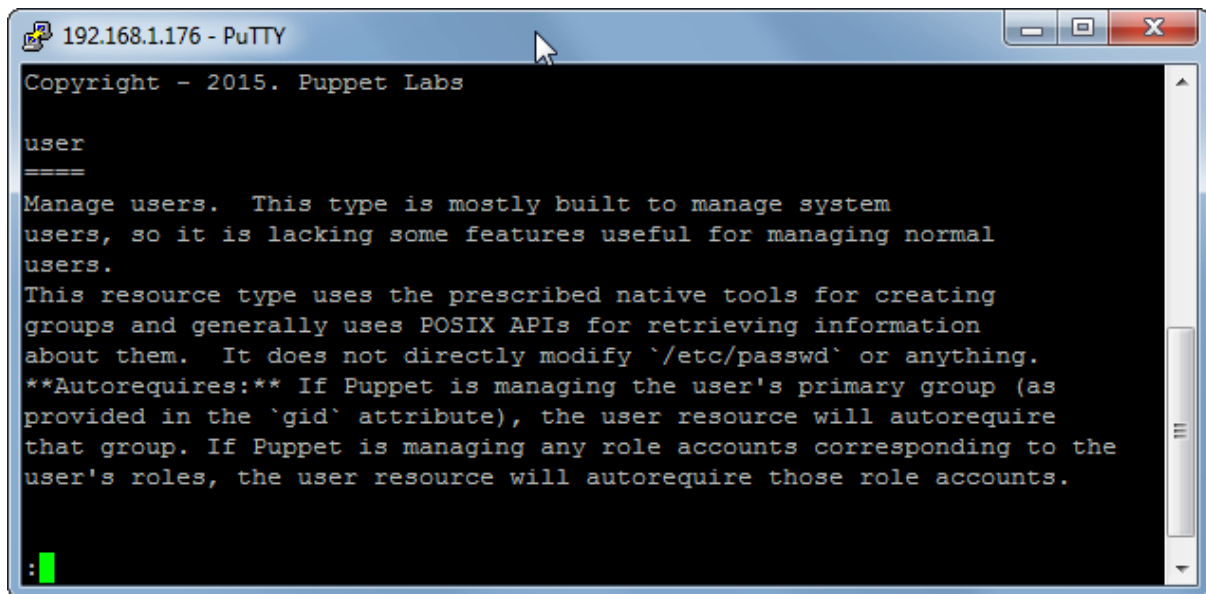
Lheidar

Puppet describe tool to get a description of the user type, including a list of its parameters.

Commands:

puppet describe user | less

jk - key mapping



```
192.168.1.176 - PuTTY
Copyright - 2015. Puppet Labs

user
====
Manage users.  This type is mostly built to manage system
users, so it is lacking some features useful for managing normal
users.
This resource type uses the prescribed native tools for creating
groups and generally uses POSIX APIs for retrieving information
about them.  It does not directly modify `/etc/passwd` or anything.
**Autorequires:** If Puppet is managing the user's primary group (as
provided in the `gid` attribute), the user resource will autorequire
that group. If Puppet is managing any role accounts corresponding to the
user's roles, the user resource will autorequire those role accounts.

:
```

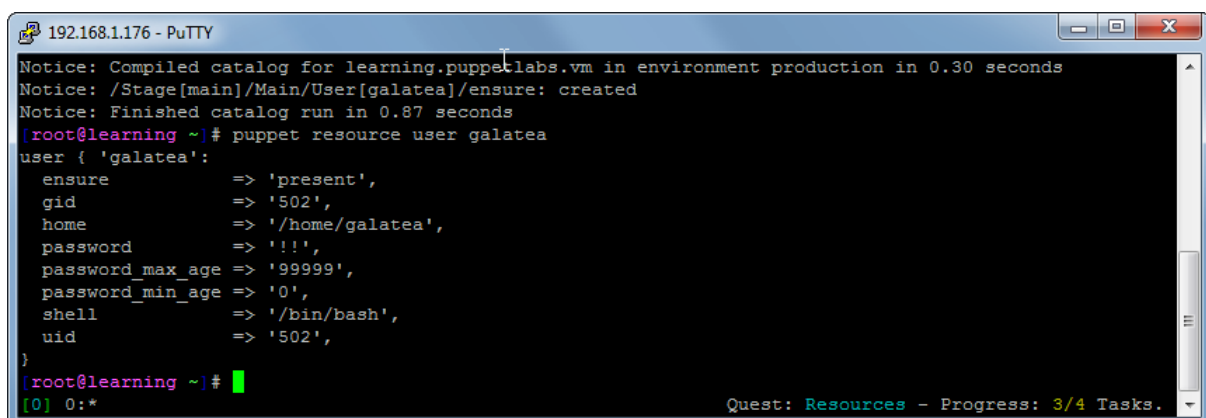
Task 3:

#Creating a new user called galatea

puppet apply -e "user { 'galatea': ensure => 'present', }"

#Puppet resource tool to take a look at user galatea

puppet resource user galatea

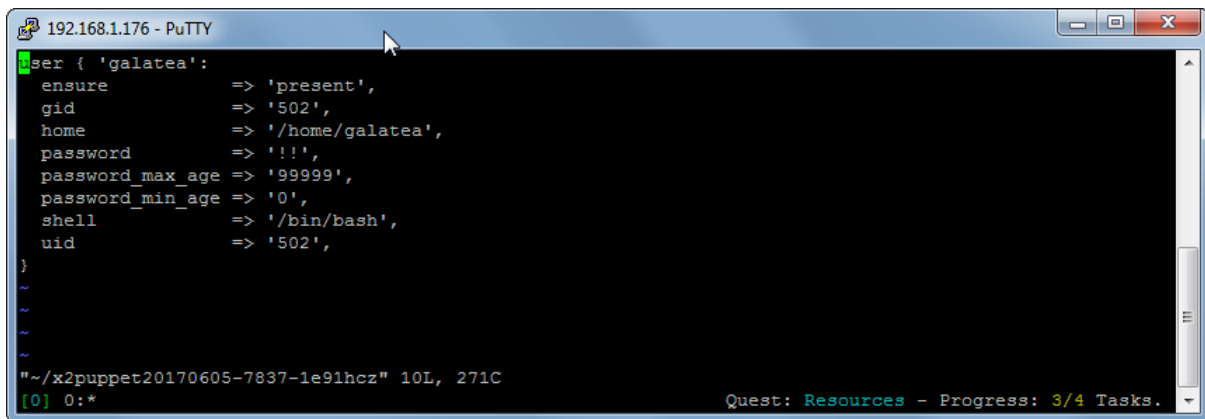


```
192.168.1.176 - PuTTY
Notice: Compiled catalog for learning.puppetlabs.vm in environment production in 0.30 seconds
Notice: /Stage[main]/Main/User[galatea]/ensure: created
Notice: Finished catalog run in 0.87 seconds
[root@learning ~]# puppet resource user galatea
user { 'galatea':
  ensure      => 'present',
  gid         => '502',
  home        => '/home/galatea',
  password    => '!!!',
  password_max_age => '99999',
  password_min_age => '0',
  shell       => '/bin/bash',
  uid         => '502',
}
[root@learning ~]#
```

Drops current state of a resource into a text editor where you can make any changes you like

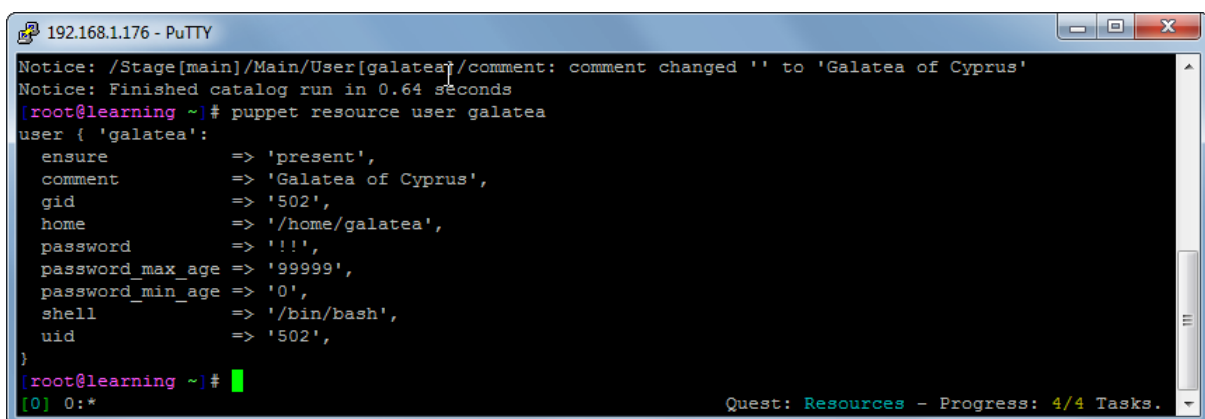
puppet resource -e user galatea

Lheidar



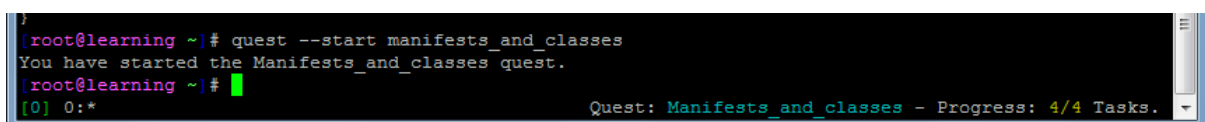
```
192.168.1.176 - PuTTY
user { 'galatea':
  ensure      => 'present',
  gid         => '502',
  home        => '/home/galatea',
  password    => '!!!',
  password_max_age => '99999',
  password_min_age => '0',
  shell       => '/bin/bash',
  uid         => '502',
}
~
~
~
~
"/x2puppet20170605-7837-1e91hcz" 10L, 271C
[0] 0:* Quest: Resources - Progress: 3/4 Tasks.
```

Puppet resource tool used again to inspect the result to see changes to added comment



```
192.168.1.176 - PuTTY
Notice: /Stage[main]/Main/User[galatea]/comment: comment changed '' to 'Galatea of Cyprus'
Notice: Finished catalog run in 0.64 seconds
[root@learning ~]# puppet resource user galatea
user { 'galatea':
  ensure      => 'present',
  comment     => 'Galatea of Cyprus',
  gid         => '502',
  home        => '/home/galatea',
  password    => '!!!',
  password_max_age => '99999',
  password_min_age => '0',
  shell       => '/bin/bash',
  uid         => '502',
}
[root@learning ~]#
[0] 0:* Quest: Resources - Progress: 4/4 Tasks.
```

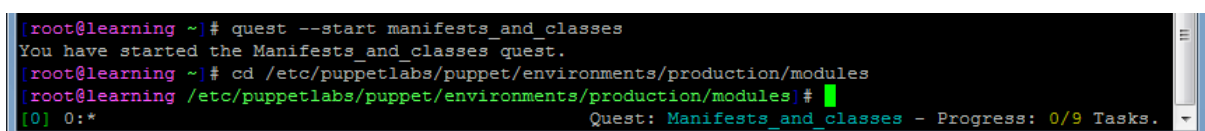
Manifests and Classes:



```
[root@learning ~]# quest --start manifests_and_classes
You have started the Manifests_and_classes quest.
[root@learning ~]#
[0] 0:* Quest: Manifests_and_classes - Progress: 4/4 Tasks.
```

Note: In Puppet, classes are singleton, which means that a class can only be declared once on a given node. Puppet's classes are different than the kind of classes you may have encountered in Object Oriented programming

Navigating to the correct directory directories



```
[root@learning ~]# quest --start manifests_and_classes
You have started the Manifests_and_classes quest.
[root@learning ~]# cd /etc/puppetlabs/puppet/environments/production/modules
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Manifests_and_classes - Progress: 0/9 Tasks.
```

Class definition for cowsay.pp manifest below:



```
192.168.1.176 - PuTTY
class cowsayings::cowsay {
  package { ['cowsay']:
    ensure => 'present',
  }
}
~
~
~
~
[0] 0:* Quest: Manifests_and_classes - Progress: 0/9 Tasks.
```

#puppet parser tool to check the syntax of your new manifest:
puppet parser validate cowsayings/manifests/cowsay.pp

Note: parser will return nothing if there are no errors

1) Declaring the cowsay class with the include keyword within the Vim Editor

Task 3:

```
puppet apply --noop cowsayings/tests/cowsay.pp
```

Using Puppet Apply Command:

```
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# cowsay puppet is awesome!
```

```
< Puppet is awesome! >
-----
      \      ^__^
       (oo)\_____)
          (__)\\       )\/\
               ||----w |
               ||     ||
```

```
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
```

```
[0] 0:*                               Quest: Manifests_and_classes - Progress: 3/9 Tasks
```


Lheidar

Fortune package:

you can provide your cow with a whole database of wisdom.

Task 4:

Creating new manifest and defining fortune class

vim cowsayings/manifests/fortune.pp



```
192.168.1.176 - PuTTY
class cowsayings::fortune {
  package { ['fortune-mod'];
    ensure => 'present',
  }
}
```

Task 5

manifests syntax with the puppet parser validate

puppet parser validate cowsayings/manifests/fortune.pp

vim cowsayings/tests/fortune.pp



```
192.168.1.176 - PuTTY
include cowsayings::fortune
```

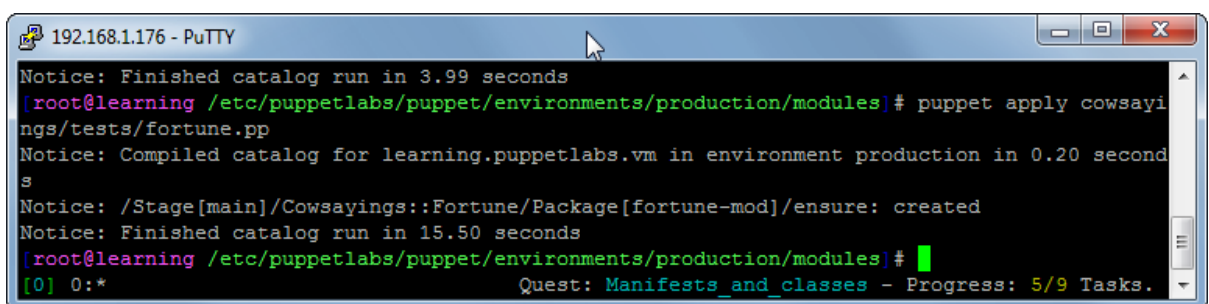
Task 6

Apply the cowsayings/tests/fortune.pp manifest with the --noop flag

puppet apply --noop cowsayings/tests/fortune.pp

without the --noop flag

puppet apply cowsayings/tests/fortune.pp



```
192.168.1.176 - PuTTY
Notice: Finished catalog run in 3.99 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# puppet apply cowsayings/tests/fortune.pp
Notice: Compiled catalog for learning.puppetlabs.vm in environment production in 0.20 seconds
Notice: /Stage[main]/Cowsayings::Fortune/Package[fortune-mod]/ensure: created
Notice: Finished catalog run in 15.50 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Manifests_and_classes - Progress: 5/9 Tasks.
```



```
192.168.1.176 - PuTTY
\ -- John Berryman /
-----
  \      ^__^
   (oo)\_____)
    (__)\       )\/\
       ||----w |
       ||     ||

[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Manifests_and_classes - Progress: 6/9 Tasks.
```

Lheidar

Output Result:

Installed two packages that can work together to do something more interesting than either would do on its own.

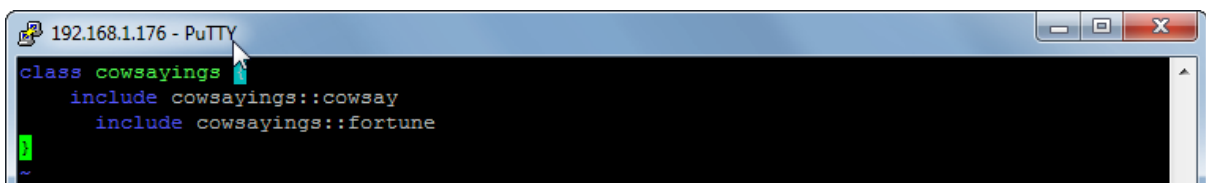
Main class: init.pp

Module will gather several classes that work together into a single class to let you declare everything at once.

The main class shares the name of the module itself, but instead of following the pattern of naming the manifest for the class it contains, Puppet recognizes the special file name init.pp for the manifest that will contain a module's main class.

Task 7

Create an init.pp manifest to declare the cowsayings::cowsay and cowsayings::fortune classes



```
192.168.1.176 - PuTTY
class cowsayings
  include cowsayings::cowsay
  include cowsayings::fortune
end
```

check syntax with the puppet parser tool

puppet parser validate cowsayings/manifests/init.pp

Task 8

Using puppet resource tool to delete both packages to test functionality of your new cowsayings class

puppet resource package fortune-mod ensure=absent

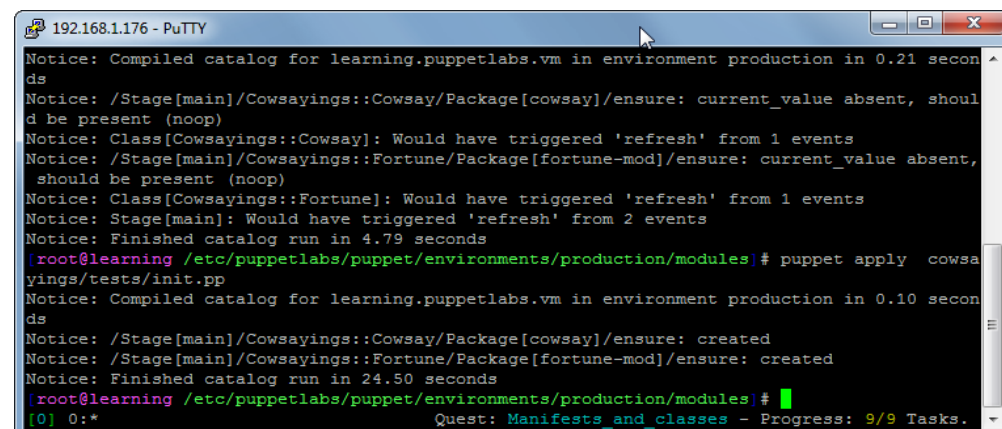
puppet resource package cowsay ensure=absent

Task 9

puppet apply --noop cowsayings/tests/init.pp

puppet apply cowsayings/tests/init.pp

Task Completion:



```
192.168.1.176 - PuTTY
Notice: Compiled catalog for learning.puppetlabs.vm in environment production in 0.21 seconds
Notice: /Stage[main]/Cowsayings::Cowsay/Package[cowsay]/ensure: current_value absent, should be present (noop)
Notice: Class[Cowsayings::Cowsay]: Would have triggered 'refresh' from 1 events
Notice: /Stage[main]/Cowsayings::Fortune/Package[fortune-mod]/ensure: current_value absent, should be present (noop)
Notice: Class[Cowsayings::Fortune]: Would have triggered 'refresh' from 1 events
Notice: Stage[main]: Would have triggered 'refresh' from 2 events
Notice: Finished catalog run in 4.79 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# puppet apply cowsayings/tests/init.pp
Notice: Compiled catalog for learning.puppetlabs.vm in environment production in 0.10 seconds
Notice: /Stage[main]/Cowsayings::Cowsay/Package[cowsay]/ensure: created
Notice: /Stage[main]/Cowsayings::Fortune/Package[fortune-mod]/ensure: created
Notice: Finished catalog run in 24.50 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:*
Quest: Manifests and classes - Progress: 9/9 Tasks.
```

Learning outcome:

A class is a collection of related resources and other classes which can be declared as a single unit. This means that unlike classes in object oriented programming, a Puppet class can only be declared a single time on a given node.

A manifest is a file containing Puppet code, and appended with the .pp extension. In this quest, we used manifests in the ./manifests directory each to define a single class, and used a corresponding test manifest in the ./tests directory to declare each of those classes.

Modules

Task 1:

#running the puppet master command with the --configprint flag and the modulepath argument:

```
puppet master --configprint modulepath
```

Task 2:

#Changing working directory to the modulepath

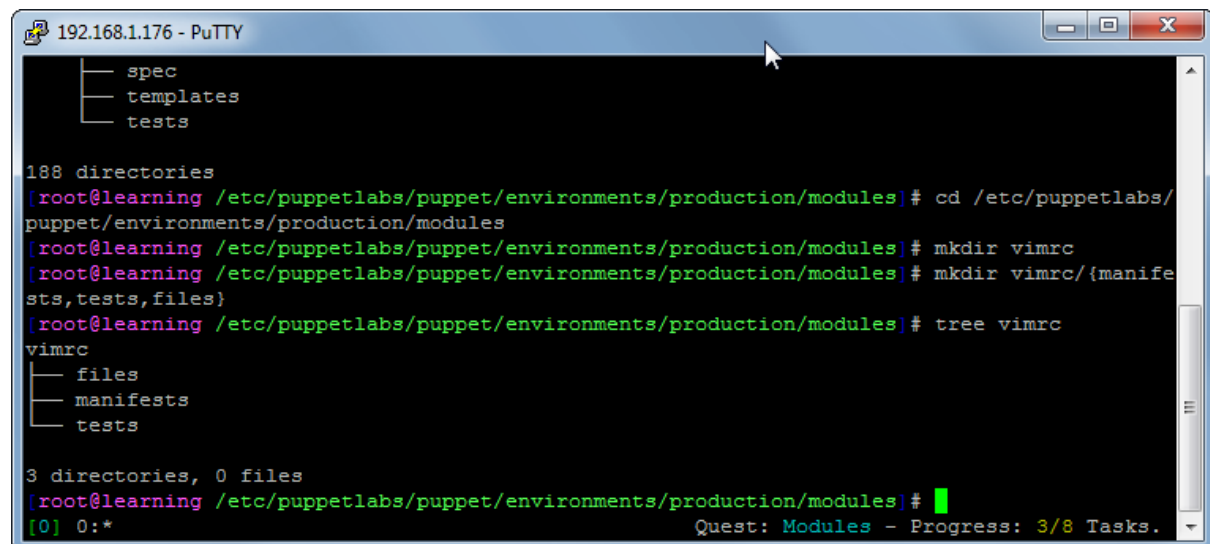
```
cd /etc/puppetlabs/puppet/environments/production/modules
```

Task 3:

mkdir command to create your module directory

```
mkdir vimrc
```

#tree vimrc command to take a look at your new module



```
192.168.1.176 - PuTTY
├── spec
├── templates
└── tests

188 directories
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# cd /etc/puppetlabs/
puppet/environments/production/modules
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# mkdir vimrc
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# mkdir vimrc/{manife
sts,tests,files}
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# tree vimrc
vimrc
├── files
├── manifests
└── tests

3 directories, 0 files
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# █
[0] 0:* Quest: Modules - Progress: 3/8 Tasks.
```

Task 4:

copy the existing .vimrc file into the files directory of your new module.

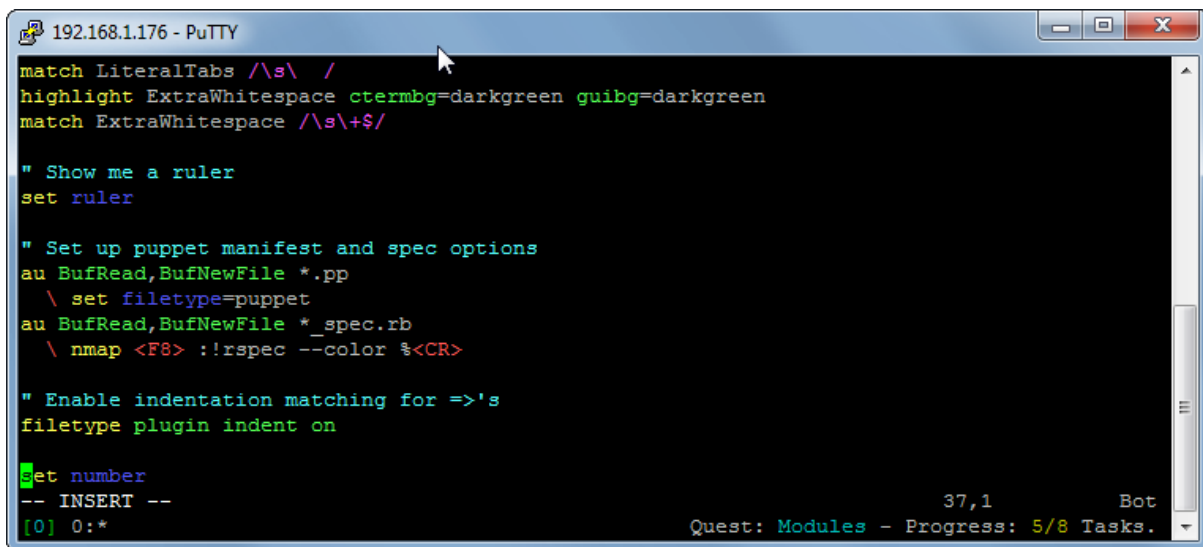
```
cp ~/.vimrc vimrc/files/vimrc
```

Task 5:

Edit file in vim editor to make an addition.

Lheidar

#line added to the end of the file for Vim to turn on line numbering

A screenshot of a PuTTY terminal window titled '192.168.1.176 - PuTTY'. The terminal displays vimrc configuration commands in a syntax-highlighted format. The commands include: 'match LiteralTabs /\s\ /', 'highlight ExtraWhitespace ctermbg=darkgreen guibg=darkgreen', 'match ExtraWhitespace /\s\+\$/', a comment '" Show me a ruler', 'set ruler', a comment '" Set up puppet manifest and spec options', 'au BufRead,BufNewFile *.pp \ set filetype=puppet', 'au BufRead,BufNewFile *_spec.rb \ nmap <F8> :!rspec --color %<CR>', a comment '" Enable indentation matching for =>'s', 'filetype plugin indent on', 'set number', and '-- INSERT --'. At the bottom, it shows '[0] 0:*' and a progress bar 'Quest: Modules - Progress: 5/8 Tasks.' with '37,1' and 'Bot' indicators.

Task 6:

Combine together init.pp manifest contains the following:

```
class vimrc {  
  
  file { ['/root/.vimrc':  
    ensure => 'present',  
    source => 'puppet:///modules/vimrc/vimrc',  
  }  
}
```

#validate your syntax

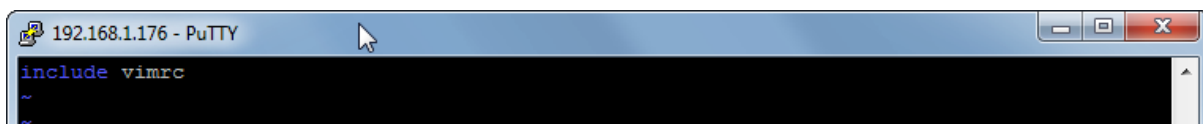
puppet parser validate vimrc/manifests/init.pp

Task 7:

create a manifest called init.pp in the vimrc/tests directory.

vim vimrc/tests/init.pp

Declaring manifest it for it to have an effect.

A screenshot of a PuTTY terminal window titled '192.168.1.176 - PuTTY'. The terminal shows the command 'include vimrc' being entered. Below it, there are two tilde characters '~' on separate lines, indicating the end of the command or a prompt.

Task 8:

#Apply with the --noop flag

puppet apply --noop vimrc/tests/init.pp

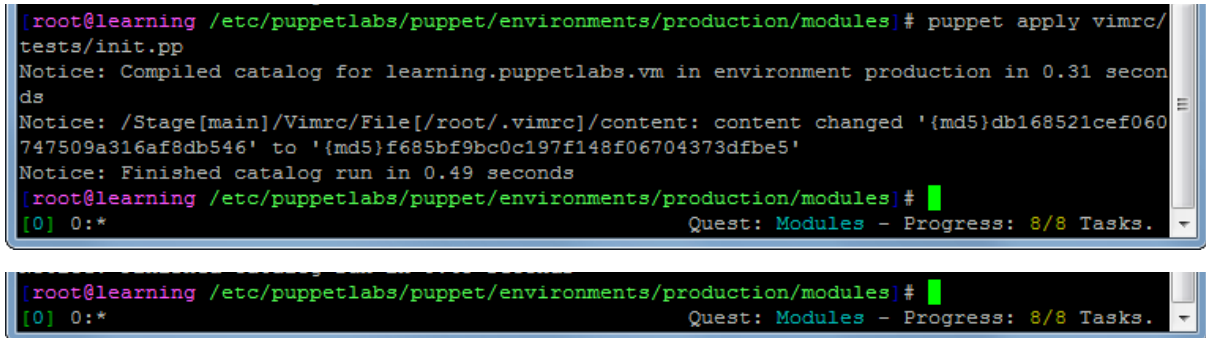
#Apply the new manifest for real

puppet apply vimrc/tests/init.pp

Lheidar

Compares md5 hash of the target file against that of the specified source file to check if any changes need to be made.

Puppet knew target file did not match desired state, and changed it to match the source file you had specified.



```
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# puppet apply vimrc/
tests/init.pp
Notice: Compiled catalog for learning.puppetlabs.vm in environment production in 0.31 seconds
Notice: /Stage[main]/Vimrc/File[/root/.vimrc]/content: content changed '{md5}db168521cef060747509a316af8db546' to '{md5}f685bf9bc0c197f148f06704373dfbe5'
Notice: Finished catalog run in 0.49 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Modules - Progress: 8/8 Tasks.
```

```
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Modules - Progress: 8/8 Tasks.
```

Learning Outcome:

Understood structure and purpose of Puppet modules. Created a module directory structure, and wrote the class you need to manage a configuration file for Vim. Observed Puppet uses md5 hashes to determine whether a target file matches the specified source file.

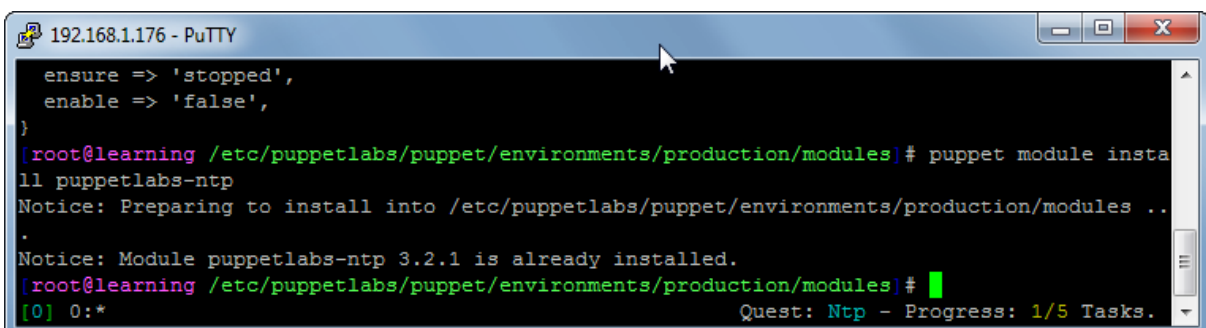
NTP

Network Time Protocol (NTP) lets you keep time millisecond-accurate within your network while staying synchronized to Coordinated Universal Time (UTC) by way of publicly accessible timeservers

Task 1:

Install the Puppet Labs ntp module

puppet module install puppetlabs-ntp

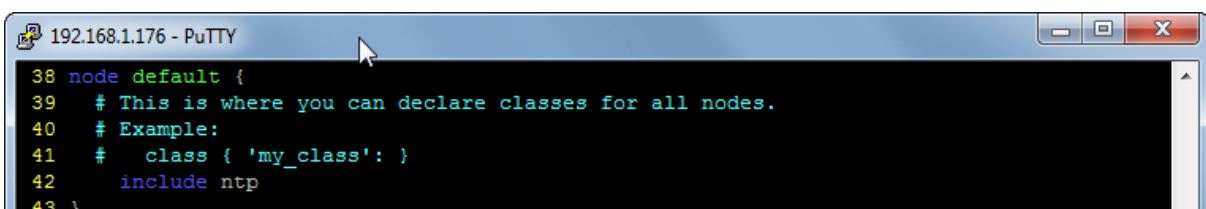


```
192.168.1.176 - PuTTY
ensure => 'stopped',
enable => 'false',
}
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# puppet module install puppetlabs-ntp
Notice: Preparing to install into /etc/puppetlabs/puppet/environments/production/modules ..
.
Notice: Module puppetlabs-ntp 3.2.1 is already installed.
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
[0] 0:* Quest: Ntp - Progress: 1/5 Tasks.
```

Note: site.pp is the first manifest the Puppet agent checks when it connects to the master. It defines global settings and resource defaults that will apply to all nodes in your infrastructure.

Task 2:

Include syntax to add the ntp class to your default node definition



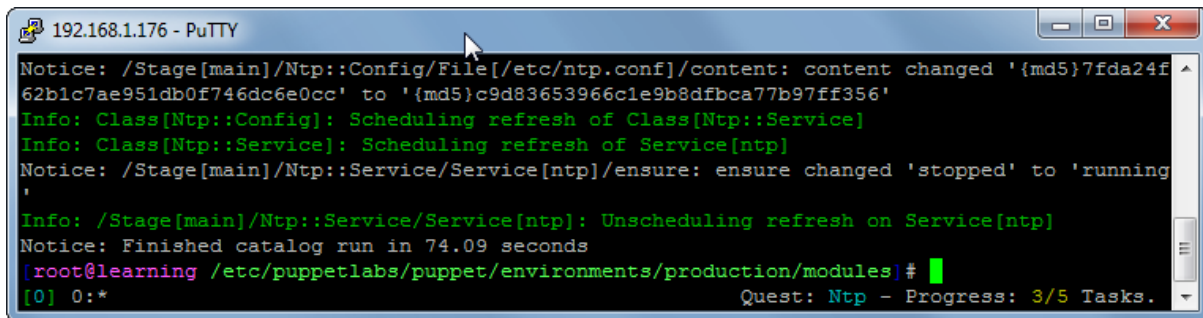
```
192.168.1.176 - PuTTY
38 node default {
39   # This is where you can declare classes for all nodes.
40   # Example:
41   #   class { 'my_class': }
42   include ntp
43 }
```

Lheidar

Note: puppet agent tool is useful for learning and testing, but that in a production environment you would want to let the puppet agent run as scheduled, every 30 minutes, by default.

```
#trigger a puppet run.  
puppet agent -t
```

Class has been successfully applied, and service is running.



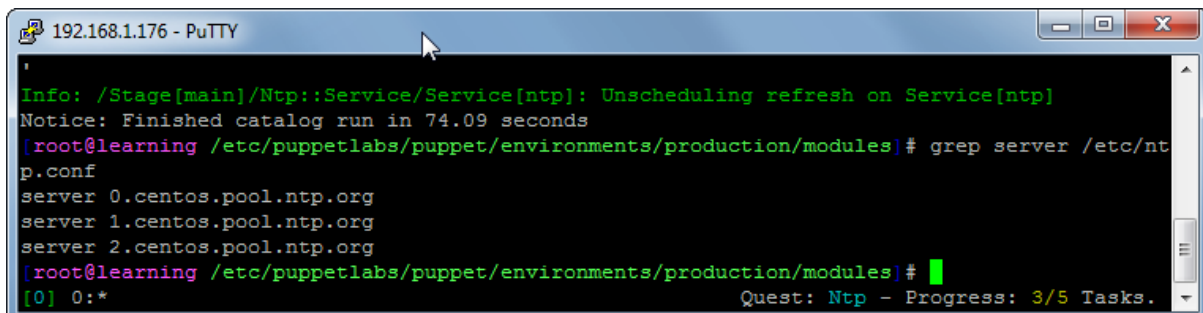
```
192.168.1.176 - PuTTY  
Notice: /Stage[main]/Ntp::Config/File[/etc/ntp.conf]/content: content changed '{md5}7fda24f62b1c7ae951db0f746dc6e0cc' to '{md5}c9d83653966c1e9b8dfbca77b97ff356'  
Info: Class[Ntp::Config]: Scheduling refresh of Class[Ntp::Service]  
Info: Class[Ntp::Service]: Scheduling refresh 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 74.09 seconds  
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#  
[0] 0:* Quest: Ntp - Progress: 3/5 Tasks.
```

#ntp class includes default settings for most of its parameters

Command below to see what servers were specified by default

```
grep server /etc/ntp.conf
```

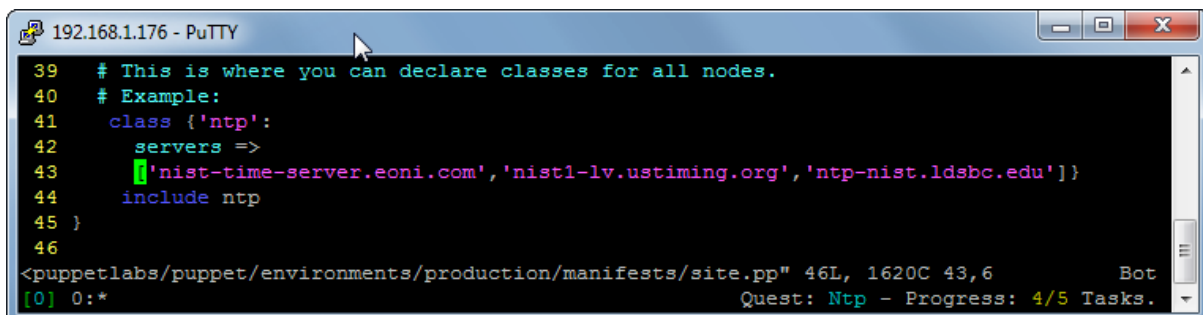
Output for list of the default servers:



```
192.168.1.176 - PuTTY  
Info: /Stage[main]/Ntp::Service/Service[ntp]: Unscheduling refresh on Service[ntp]  
Notice: Finished catalog run in 74.09 seconds  
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# grep server /etc/ntp.conf  
server 0.centos.pool.ntp.org  
server 1.centos.pool.ntp.org  
server 2.centos.pool.ntp.org  
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#  
[0] 0:* Quest: Ntp - Progress: 3/5 Tasks.
```

Task 4:

#replacing the include ntp line with a parameterized class declaration



```
192.168.1.176 - PuTTY  
39 # This is where you can declare classes for all nodes.  
40 # Example:  
41 class {'ntp':  
42     servers =>  
43     ['nist-time-server.eoni.com', 'nist1-lv.ustiming.org', 'ntp-nist.ldsbc.edu']  
44     include ntp  
45 }  
46  
<puppetlabs/puppet/environments/production/manifests/site.pp" 46L, 1620C 43,6 Bot  
[0] 0:* Quest: Ntp - Progress: 4/5 Tasks.
```

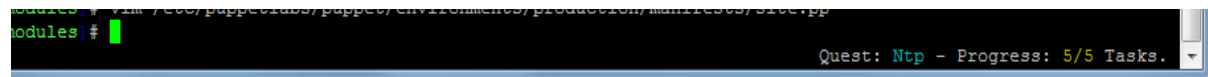
Note: servers parameter in our class declaration takes a list of servers as a value, not just one. This list of values, separated by commas (,) and wrapped in brackets ([]), is called an array. Arrays allow you assign a list of values to a single variable or attribute

Lheidar

Task 5:

#using puppet parser tool to validate syntax, use puppet agent tool to trigger puppet run.

```
# puppet parser validate /etc/puppetlabs/puppet/environments/production/manifests/site.pp
# puppet agent -t
```



Learning Outcome:

Downloading modules from the Puppet Forge with puppet module tool. Common Package/File/Service pattern, and how it's used by the NTP module to install, configure, and run the ntpd service.

Rather than running tests, learned how to use the site.pp manifest to include classes within a node declaration.

Getting the ntpd service running, covering class parameters and how they can be used to set class parameters as class is declared.

MySQL

The Puppet Labs MySQL module is a great example of how a well-written module can build on Puppet's foundation to simplify a complex configuration task without sacrificing robustness and control.

Task 1:

fetching the puppetlabs-mysql module from the Puppet Forge
puppet module install puppetlabs-mysql

Task 2:

vim /etc/puppetlabs/puppet/environments/production/manifests/site.pp

Using class parameters, specify a root password and set the server's max connections to '1024.'

```
37
38 node default {
39   # This is where you can declare classes for all nodes.
40   # Example:
41   class {'ntp':
42     servers =>
43       ['nist-time-server.eoni.com', 'nist1-lv.ustiming.org', 'ntp-nist.ldsbc.edu']}
44
45     class { ['::mysql::server':
46       root_password => 'testpassword',
47       override_options => { 'mysqld' => { 'max_connections' => '1024' } }},
48     "/etc/puppetlabs/puppet/environments/production/manifests/site.pp" 49L, 1771C
49 }
50 }
```

Task 3:

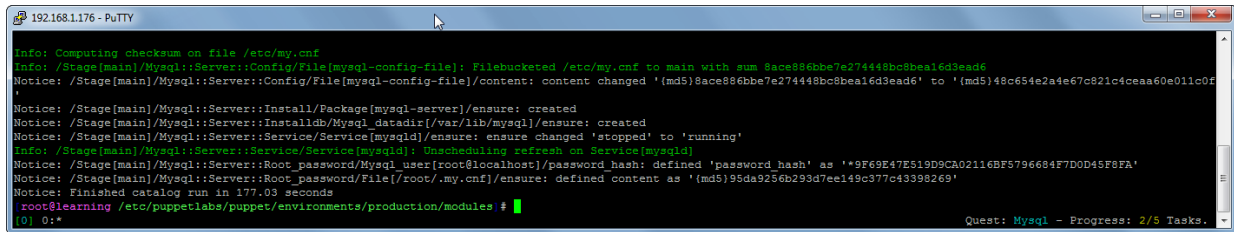
#parser validate tool to check your syntax

puppet parser validate /etc/puppetlabs/puppet/environments/production/manifests/site.pp

#trigger a puppet run

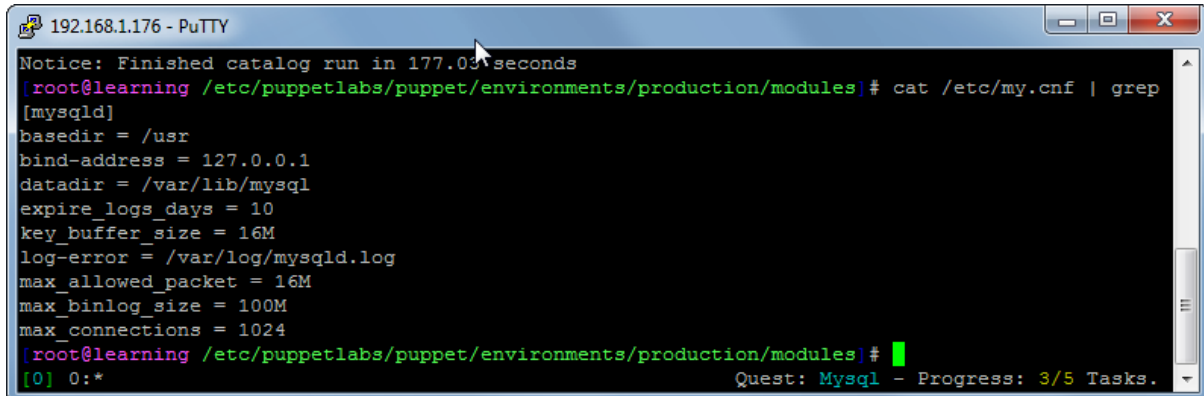
puppet agent -t

Lheidar



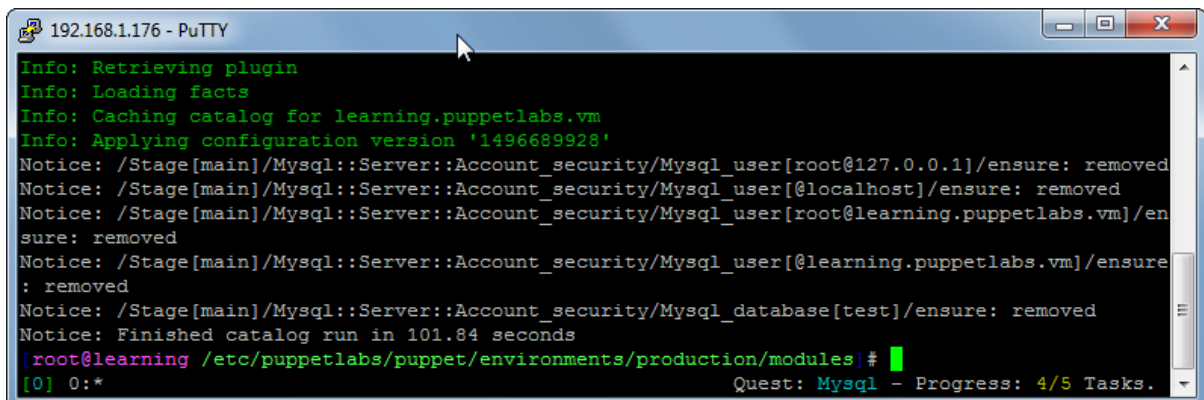
```
192.168.1.176 - PuTTY
Info: Computing checksum on file /etc/my.cnf
Info: /Stage[main]/Mysql::Server::Config/File[mysql-config-file]: Filebucketed /etc/my.cnf to main with sum 8ace88bbe7e274448bc8be16d3ead6
Notice: /Stage[main]/Mysql::Server::Config/File[mysql-config-file]/content: content changed '(md5)8ace88bbe7e274448bc8be16d3ead6' to '(md5)48c654e2a4e67c821c4ceaa60e011c0f'
Notice: /Stage[main]/Mysql::Server::Install/Package[mysql-server]/ensure: created
Notice: /Stage[main]/Mysql::Server::Installdb/Mysql_datadir[/var/lib/mysql]/ensure: created
Notice: /Stage[main]/Mysql::Server::Service/Service[mysqld]/ensure: ensure changed 'stopped' to 'running'
Info: /Stage[main]/Mysql::Server::Service[mysqld]: Unscheduling refresh on Service[mysqld]
Notice: /Stage[main]/Mysql::Server::Root_password/Mysql_user[root@localhost]/password_hash: defined 'password_hash' as '**9F69E47E519D9CA02116BF5796684F7D0D45F8FA'
Notice: /Stage[main]/Mysql::Server::Root_password/File[/root/.my.cnf]/ensure: defined content as '(md5)95da9256b293d7ee149c377c43398269'
Notice: Finished catalog run in 177.03 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
```

Result of the 'max_connections' override option set



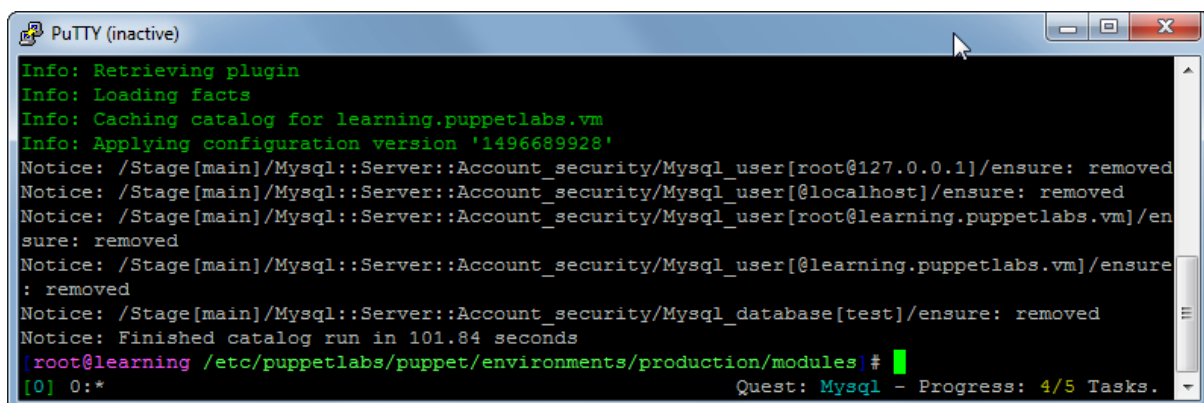
```
192.168.1.176 - PuTTY
Notice: Finished catalog run in 177.03 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]# cat /etc/my.cnf | grep
[mysqld]
basedir = /usr
bind-address = 127.0.0.1
datadir = /var/lib/mysql
expire_logs_days = 10
key_buffer_size = 16M
log-error = /var/log/mysql.log
max_allowed_packet = 16M
max_binlog_size = 100M
max_connections = 1024
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
```

include the mysql::server::account_security to manifest class



```
192.168.1.176 - PuTTY
Info: Retrieving plugin
Info: Loading facts
Info: Caching catalog for learning.puppetlabs.vm
Info: Applying configuration version '1496689928'
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[root@127.0.0.1]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[@localhost]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[root@learning.puppetlabs.vm]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[@learning.puppetlabs.vm]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_database[test]/ensure: removed
Notice: Finished catalog run in 101.84 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
```

Completion Level



```
PuTTY (inactive)
Info: Retrieving plugin
Info: Loading facts
Info: Caching catalog for learning.puppetlabs.vm
Info: Applying configuration version '1496689928'
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[root@127.0.0.1]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[@localhost]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[root@learning.puppetlabs.vm]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_user[@learning.puppetlabs.vm]/ensure: removed
Notice: /Stage[main]/Mysql::Server::Account_security/Mysql_database[test]/ensure: removed
Notice: Finished catalog run in 101.84 seconds
[root@learning /etc/puppetlabs/puppet/environments/production/modules]#
```