

# Puppet Enterprise Report (S3540510)

## Introduction

This document serves to show how I coded my puppet manifests files, and the reasoning for the decisions made. All of the code contained in this document and on my Git repository is my own work.

The proof of it working is included as a .txt file containing the debug output of the command 'puppet agent -t -d'. This has been accepted as sufficient and I have a record of the e-mail correspondence with Ke Deng.

## GitHub

A git.log file with all my commit history is included in the zip file.

URL: <https://github.com/wejrox/Puppet-Development>

## Note

I did not create a git on the core teaching servers as I have a git account and use github to store my repositories.

However, I did have to set up the remote on the AWS server in order to get it to work.

When I talked to Ke Deng after a lecture, I asked if I could use a GIT repository that is not on core-teaching and was told that this was acceptable.

Unfortunately, I do not have proof of this conversation other than my word.

If you attempt to run the code, it will fail when trying to mount to titan.

This is because I will not include my password in the files I submit as it is a security issue.

**All user passwords are: !!**

# **Puppet Enterprise Report (S3540510)**

## **Table of Contents**

Introduction	<b>1</b>
GitHub	<b>1</b>
Note	<b>1</b>
Create users	<b>3</b>
cgdb, fuse-sshfs	<b>4</b>
Dia2Code, MySQL-Server, Other Packages	<b>5</b>
Create groups	<b>6</b>
Ensure that httpd is run on boot and subscribes to <i>.conf</i> file	<b>7</b>
Enable SSH and subscribe to file	<b>8</b>
Agent Check-In	<b>8</b>
Default Run-Level	<b>8</b>
Timestamp display on login	<b>9</b>
Add /usr/local/bin to file path	<b>9</b>
Give becca Sudo privileges	<b>9</b>
Mount Titan over /home/becca/titan	<b>10</b>
Define Hosts	<b>10</b>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Create users</b> User management	<pre> # Creates all users class user {   require groups   require packages   user { 'becca' :     ensure      =&gt; present,     uid         =&gt; '10010510',     home        =&gt; '/home/becca',     managehome  =&gt; true,     password    =&gt; '\$1\$j.cuPw6w\$xEqjdNmz.qgs2lgZgwTqQ.',     groups      =&gt; ['sysadmin', 'cars'],     shell       =&gt; '/bin/bash',   }   user { 'fred' :     ensure      =&gt; present,     uid         =&gt; '10020510',     home        =&gt; '/home/fred',     managehome  =&gt; true,     password    =&gt; '\$1\$j.cuPw6w\$xEqjdNmz.qgs2lgZgwTqQ.',     groups      =&gt; ['trucks', 'cars', 'wheel'],     shell       =&gt; '/bin/csh',   }   user { 'wilma' :     ensure      =&gt; present,     uid         =&gt; '10030510',     home        =&gt; '/home/wilma',     managehome  =&gt; true,     password    =&gt; '\$1\$j.cuPw6w\$xEqjdNmz.qgs2lgZgwTqQ.',     groups      =&gt; ['trucks', 'cars', 'ambulances'],     shell       =&gt; '/bin/bash',     purge_ssh_keys =&gt; true,   }   # SSH Key for wilma's account   -&gt; ssh_authorized_key { 'wilmaKey' :     ensure =&gt; present,     user   =&gt; 'wilma',     type   =&gt; 'ssh-rsa',     key    =&gt; 'ThisIsWilmasKeyAbCd',   }} </pre>	<p>Every user is created in the same class, with the requirement for groups and packages to be instated prior to creating the users. This is required due to the assignment of groups to users which would fail without it, as well as the requirement for /bin/csh needing to exist before setting Fred's shell.</p> <p>Passwords are encrypted with SHA-1 on the linux machine then pasted into the file. I could not use the sha-1('!!') function as it would not allow me to log in with the password when attempting to 'su – becca'.</p> <p>Fred has been assigned the group 'wheel' as it is the redhat group to enable sudo privileges without manually changing the sudoers file.</p> <p>Wilma's ssh key is created straight after her user is created.</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>cgdb, fuse-sshfs</b> Package Management	<pre> # Ensures that all packages specified are installed and installed class packages{   # This is needed for yum or things will be missing   exec { 'getDeps' :     onlyif =&gt; '/usr/bin/yum-config-manager',     command =&gt; '/usr/bin/yum-config-manager --enable rhui-REGION-rhel-server-optional',   }    # Install cgdb   package { 'cgdb' :     ensure =&gt; 'installed',     source =&gt; 'ftp://mirror.switch.ch/pool/4/mirror/epel/7/x86_64/c/cgdb-0.6.8-1.el7.x86_64.rpm',     provider =&gt; 'rpm',   }    # Install fuse-sshfs   package { 'glib*' :     ensure =&gt; 'installed',     provider =&gt; 'yum',   }   -&gt; package { 'fuse*' :     ensure =&gt; 'installed',     provider =&gt; 'yum',   }   -&gt; package { 'fuse-sshfs' :     ensure =&gt; 'installed',     source =&gt; 'ftp://195.220.108.108/linux/dag/redhat/el7/en/x86_64/dag/RPMS/fuse-sshfs-2.5-1.el7.rf.x86_64.rpm',     provider =&gt; 'rpm',   } } </pre>	<p>As yum doesn't have as many records of packages as apt-get or other package providers, I have required a dependency download before other packages are installed.</p> <p>Gcdb does not exist on the Yum repository, so I found an RPM and installed using the rpm provider instead.</p> <p>Fuse-sshfs also does not exist on Yum, so I had to install using the same method as gcdb. As fuse-sshfs requires dependencies, I also ensured that they were downloaded and installed prior to fuse-sshfs.</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Dia2Code, MySQL-Server, Other Packages Package Management</b>	<pre># Install dia2code package { 'libxml2.so.2' :   ensure =&gt; installed, } -&gt; package { 'dia2code' :   ensure  =&gt; 'installed',   source  =&gt; 'https://downloads.sourceforge.net/project/dia2code/dia2code/0.8.3/dia2code-0.8.3-3.1.i586.rpm?r=http%3A%2F%2Fdia2code.sourceforge.net%2Fdownload.html&amp;ts=1507292790&amp;use_mirror=ncu',   provider =&gt; 'rpm', }  # Install mysql-server package { 'mysql57-community-release' :   ensure  =&gt; 'installed',   source  =&gt; 'https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm',   provider =&gt; 'rpm', }  # Install additional packages that exist on Yum \$packages = [   'openssh-server', 'httpd', 'mysql',   'tigervnc-server', 'tmux', 'lynx', 'gcc',   'gdb', 'vim', 'emacs', 'csh', ]  package { \$packages :   ensure  =&gt; 'installed',   provider =&gt; 'yum', } }</pre>	<p>Dia2Code is similar to Fuse-sshfs in that it has dependencies required, so the same method was used.</p> <p>Mysql-server was also installed using RPM as Yum did not have a reference to it.</p> <p>The remainder of packages existed on Yum after the dependency update at the top of the file, so I created an array of them,</p> <p>Then installed them all using a reference to the variable and specifying the Yum package provider.</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Create groups</b> User management	<pre># Creates all groups class groups {   group{'sysadmin':     ensure =&gt; present,     gid    =&gt; '500',   }    group{'cars':     ensure =&gt; present,     gid    =&gt; '501',   }    group{'trucks':     ensure =&gt; present,     gid    =&gt; '502',   }    group{'ambulances':     ensure =&gt; present,     gid    =&gt; '503',   } }</pre>	<p>Groups are required for each user. Groups id starts at 500 as ID's below 500 are used by services and could be taken.</p> <p>I do not think further explanation is required.</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<p><b>Ensure that httpd is run on boot and subscribes to .conf file</b></p> <p>Package Configuration</p>	<pre># Handles management of services, and initial setup class services {   file {'/var/www/s3540510':     ensure =&gt; 'directory',     owner  =&gt; 'root',     mode   =&gt; '0644',   }   # Overwrite httpd.conf with custom configuration   -&gt; file { '/etc/httpd/conf/httpd.conf' :     ensure =&gt; present,     mode   =&gt; '0644',     owner  =&gt; 'root',     group  =&gt; 'root',     source =&gt;       '/etc/puppetlabs/code/environments/production/manifests/configfiles/httpd.conf',     require =&gt; Package['httpd'],   }   # Subscribe to service (restart on file change)   -&gt; service { 'httpd' :     ensure =&gt; running,     enable =&gt; true,     subscribe =&gt; File['/etc/httpd/conf/httpd.conf'],   } }</pre>	<p>Directory to change to must exist</p> <p>Httpd is subscribed to the httpd.conf file, so it will restart whenever the file is changed. This line copies over the custom httpd.conf, which changes the root document to '/var/www/s3540510.html'</p> <p>Make sure the service is running and enabled on boot. Subscribe to httpd.conf file to trigger refresh</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Enable SSH and subscribe to file</b> Package Configuration	<pre># Ensures ssh is running and that root access is disabled class openssh {   # Overwrite sshd_config with custom one.   # Augeas cannot handle commenting.   file { ['/etc/ssh/sshd_config':     ensure =&gt; present,     mode   =&gt; '0777',     owner  =&gt; 'root',     group  =&gt; 'root',     source =&gt; '/etc/puppetlabs/code/environments/production/manifests/configfiles/sshd_config',     require =&gt; Package['openssh-server'],   }   -&gt; service { 'sshd' :     ensure  =&gt; running,     enable  =&gt; true,     require =&gt; Package['openssh-server'],     subscribe =&gt; File['/etc/ssh/sshd_config'],   } }</pre>	<p>The config file is a custom file that we were told we could use on blackboard, and is copied across to the server from a local location. This is due to Augeas being unable to uncomment lines in a file.</p> <p>Openssh will start on system boot, and restart when the sshd_config file is changed (subscribe)</p>
<b>Agent Check-In</b> Agent Configuration	<pre># Sets run interval, agent timestamp display, inclusion of /usr/local/bin, # Becca sudoing, mounting Titan onto becca's drive. class iniconfig {   augeas { 'agent_runinterval' :     context =&gt; '/etc/puppetlabs/puppet/puppet.conf/agent',     changes =&gt; 'set runinterval 20m',   } }</pre>	<p>Agent checkin 3 times per hour, achieved through augeas.</p>
<b>Default Run-Level</b> Agent Configuration	<pre>exec { 'runlevel' :   command =&gt; '/usr/bin/systemctl set-default multi-user.target', }</pre>	<p>Setting the default run-level to 3. Manual changes to the file that used to handle it no longer have any effect, so it must be achieved through a command instead.</p>



## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Timestamp display on login</b> Agent Configuration	<pre># Set the timestamp to client file { '/etc/profile.d/agent_login.sh' :   ensure =&gt; present,   mode   =&gt; '0777',   owner  =&gt; 'root',   group  =&gt; 'root',   content =&gt; 'timeStamp=`/bin/date +"%d-%m-%Y_%H.%M.%S"`; echo "Agent started running at \$timeStamp"', }</pre>	<p>A timestamp is displayed whenever any user logs into an agent.</p> <p>No requirement was specified to save this timestamp anywhere.</p>
<b>Add /usr/local/bin to file path</b> Agent Configuration	<pre># Include /usr/local/bin to user file { '/etc/profile.d/set-user-bin.sh' :   owner  =&gt; 'root',   mode   =&gt; '0644',   content =&gt; 'PATH=\$PATH:/usr/local/bin', }</pre>	<p>/usr/local/bin is added to the default system path whenever a user logs in.</p> <p>Achieved the same way as the timestamp.</p>
<b>Give becca Sudo privileges</b> Agent Configuration	<pre># Give Becca Sudoers privilege exec { 'give_sudo_becca' :   command =&gt; '/usr/sbin/usermod -aG wheel becca', }</pre>	<p>Becca is given sudo privilege by editing the sudo file.</p> <p>It is bad practice to edit it directly, so a command is run instead.</p>

## Puppet Enterprise Report (S3540510)

Task	Code	Description
<b>Mount Titan over /home/becca/titan</b> Agent Configuration	<pre># Mount titan over becca (/home/becca/titan) # Make dir to use, connect using sshfs; only if it's not mounted already exec { 'mount_titan_becca' :   command =&gt; '/usr/bin/mkdir /home/becca/titan; echo \'     "\$(cat /etc/puppetlabs/code/environments/production/manifests/configfiles/ssh_pass)"   \'   /usr/bin/sshfs -o StrictHostKeyChecking=no -o password_stdin s3540510@titan.csit.rmit.edu.au:/home/sl0/S3540510/ /home/becca/titan/',   unless =&gt; '/usr/bin/find /home/becca/titan -mindepth 1   /usr/bin/read', } }</pre>	<p>SSHFS is used to mount my personal RMIT directory from Titan into becca's home directory.</p> <p>A file containing my password is stored locally and referenced, maintaining security.</p> <p>The password is then supplied via piping to stdin, utilising the 'password_stdin' option in sshfs mounting command.</p> <p>Line breaks are required to pass puppet-lint.</p>
<b>Define Hosts</b> Host Configuration	<pre># Defines the hosts for dns use (RMIT coreteaching servers) class hosts {   host{'titan.csit.rmit.edu.au':     ip      =&gt; '131.170.5.131',     host_aliases =&gt; 'titan',   }   host{'saturn.csit.rmit.edu.au':     ip      =&gt; '131.170.5.132',     host_aliases =&gt; 'saturn',   }   host{'jupiter.csit.rmit.edu.au':     ip      =&gt; '131.170.5.135',     host_aliases =&gt; 'jupiter',   } }</pre>	<p>Host records created for internal DNS.</p> <p>Titan, Saturn and Jupiter are added.</p>