V2 Lab1 - Steps 2A & 2B

Like | Updated 6 December 2017 by Roy Mitchley | Tags: *None*

Perform this lab exercise

Atom

```
Anything in this box needs to be edited in Atom Text
```

Terminal

```
This box contains lists of commands that should executed in order on the
Terminal
```

(A) Using Puppet to Automate the provisioning of the VM

The objective of this lab is to use puppet to manage the state of the Virtual Machine. The core os and a single user are already installed, along with puppet. All other modules are brought into the VM and managed by puppet. To do this, the learner will add a new user, configure an existing one and add some packages to be installed.

- Log into the VM to begin exploring the Learning VM 

username and password are both `devops`.

Once inside the VM, there is not much installed or configured. One of the only things configured in the VM manually is puppet (and Docker and Git). Puppet is the tool that will be used to describe the state of the VM.

- Once inside the VM, open the terminal and run the following to bring in all the lab material to the VM

Terminal

```
git clone https://bitbucket.org/ibmdevopscourse/lab-material.git
```

- The repository just cloned is read only, to allow the learner make changes throughout the labs they will use a local running git server as the origin. Run the following from the lab-material directory to set this up

Terminal

```
cd lab-material
git remote remove origin
git remote add origin ssh://git@git.server/home/git/lab-material.git
git push -u origin --all
```

You may have to accept the host authenticity check by typing in `yes` in the prompt

```
devops@devops-v2: ~/lab-material
File Edit View Search Terminal Help
Resolving deltas: 100% (145/145), done.
Checking connectivity... done.
→ ~ git remote remove origin
fatal: Not a git repository (or any of the parent directories): .git
→ ~ cd lab-material
→ lab-material git:(master) git remote remove origin
→ lab-material git:(master) git remote add origin ssh://git@git.server/home/git/lab-material.git
→ lab-material git:(master) git push -u origin --all
The authenticity of host 'git.server (127.0.1.1)' can't be established.
ECDSA key fingerprint is SHA256:miEyrK51nSr6wYWFVctdAgX4NCbW+qQOedPIHFx4UHA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'git.server' (ECDSA) to the list of known hosts.
Counting objects: 283, done.
Compressing objects: 100% (117/117), done.
Writing objects: 100% (283/283), 29.75 KiB | 0 bytes/s, done.
Total 283 (delta 133), reused 265 (delta 115)
To ssh://git@git.server/home/git/lab-material.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
→ lab-material git:(master)
```

- Navigate to the puppet directory inside this folder using the terminal by typing and open the puppet manifests. From your HOME dir run the following.

```
cd lab-material/puppet
atom .
```

- Open build-vm.pp in your Atom editor using the navigation tree on the left hand side.

The puppet file is broken into a few commented sections (denoted by the #)

- The first one shows the puppet variables `$newuser = 'your.username.here'`
- Next is some configuration for the users and groups using the puppet sytax group / user
- Then the configuration to get zsh working
- Some packages used at an OS level
- Finally, the section that defines the NodeJS module we are going to use as well as some additional libraries to npm install
- Existing resources can be queried from the puppet cmd line using the following cmds back on the *Terminal*

- Lets experiment with some puppet commands. The following can be run from any directory.

```
puppet resource user
puppet resource user donal
```

```
➜  puppet puppet resource user donal
user { 'donal':
  ensure => 'absent',
}
➜  puppet
```

Create a new user for the learner.

```
in the lab-material project (~/lab-material/) open the `build-vm.pp` puppet file
```

- Change the `$newuser` variable to your name.

- Add the user to the `'devops-course'` group and to the `'sudo'` group, so the puppet config look as follows. The puppet script will give the user a new id of `1010` and put it into the `docker` and `sudo` groups as shown in the `user` section

```
user { $newuser :
  ensure           => 'present',
  gid              => '5001',
  groups           => ['docker', 'devops-course', 'sudo'],
  home             => "/home/${newuser}",
  password         => 'your.encrypted.password.here', # note thi
  password_max_age => '99999',
  password_min_age => '0',
  shell            => '/bin/zsh',
  uid              => '1010',
  managehome       => true,
} ->
```

- Generate your user a password by running by running the command below and pasting the value returned into the `'your.encrypted.password.here'` in the file. Save your file.

```
openssl passwd -1
```

Note that the "`-1`" in the command is "minus numeral one" as opposed to "minus letter ell".

```
devops@devops-v2: ~/Documents/puppet
➜  ~ cd Documents/puppet
➜  puppet openssl passwd -1
Password:
Verifying - Password:
$1$qEf/ydR9$RFsPo7ZqoVpul3pQiX/uk1
➜  puppet
```

- Validate your new puppet file; this will provide no output if successful. The command prompt may change to be appended by a yellow "X" as in the second screen-shot below which may, at first sight, appear confusing.

```
cd ~/lab-material/puppet
puppet parser validate build-vm.pp
```

HINT you can check the return code of the last executed command by executing `echo $?`. A result of `0` means successful. If a command was successful, oh-my-zsh will use the return code to display a green arrow on the left of your prompt. If unsuccessful it will be red, for example:

```
➜ lab-material git:(master) puppet parser validate build-vm.pp
Error: One or more file(s) specified did not exist:
["   build-vm.pp\n"]
Error: Try 'puppet help parser validate' for usage
➜ lab-material git:(master) cd puppet
➜ puppet git:(master) puppet parser validate build-vm.pp
➜ puppet git:(master) echo $?
0
```

```
➜ puppet git:(master) puppet parser validate build-vm.pp
➜ puppet git:(master) ✗ echo $?
0
➜ puppet git:(master) ✗
```

- Run the commands below. The `noop` flag will stage the changes and check for errors without actually applying anything.

```
cd ~/lab-material/puppet
```

- Apply the puppet manifest to the VM by running the following from the ~/lab-material/puppet directory:

```
sudo puppet apply build-vm.pp
```

Unfortunately puppet displays both warnings and errors in red, and there are often a lot of warnings You might normally look at them when something has changed in your puppet file, or if you are getting errors lower down. An example is below:

```
➜ puppet sudo puppet apply build-vm.pp
Warning: Scope(Class[Nodejs::Params]): The nodejs module might not work on Ubuntu 16.04. Sensible defaults will be attempted.
Warning: Scope(Apt::Source[nodesource]): $include_src is deprecated and will be removed in the next major release, please use $include => { 'src' => false } instead
Warning: Scope(Apt::Source[nodesource]): $required_packages is deprecated and will be removed in the next major release, please use package resources instead.
Warning: Scope(Apt::Source[nodesource]): $key_source is deprecated and will be removed in the next major release, please use $key => { 'source' => https://deb.nodesource.com/gpgkey/nodesource.gpg.key } instead.
Warning: Scope(Apt::Key[Add key: 9FD3B784BC1C6FC31A8A0A1C1655A0AB68576280 from Apt::Source nodesource]): $key_source is deprecated and will be removed in the next major release. Please use $source instead.
Notice: Compiled catalog for devops-v2.ibm.com in environment production in 0.97 seconds
Notice: /Stage[main]/Main/Group[devops-course]/ensure: created
Notice: /Stage[main]/Main/User[donal]/ensure: created
Notice: /Stage[main]/Main/Vcsrepo[/home/donal/.oh-my-zsh]/ensure: Creating repository from present
Notice: /Stage[main]/Main/Vcsrepo[/home/donal/.oh-my-zsh]/ensure: created
Notice: /Stage[main]/Apt/File[sources.list]/mode: mode changed '0664' to '0644'
Notice: /Stage[main]/Apt/Apt::Setting[conf-update-stamp]/File[/etc/apt/apt.conf.d/15update-stamp]/content: content changed '{md5}b9de0ac9e2c9854b1bb213e362dc4e41' to '{md5}0962d70c4ec78bbfa6f3544ae0c41974'
Notice: /Stage[main]/Apt/File[preferences]/ensure: created
Notice: /Stage[main]/Nodejs::Repo::Nodesource::Apt/Apt::Source[nodesource]/Apt::Setting[list-nodesource]/File[/etc/apt/sources.list.d/nodesource.list]/ensure: created
Notice: /Stage[main]/Apt::Update/Exec[apt_update]: Triggered 'refresh' from 1 events
Notice: /Stage[main]/Nodejs::Install/Package[nodejs]/ensure: ensure changed 'purged' to 'present'
Notice: /Stage[main]/Main/File[/usr/bin/node]/target: target changed '/etc/alternatives/node' to '/usr/bin/nodejs'
Notice: /Stage[main]/Main/Package[bower]/ensure: created
Notice: /Stage[main]/Main/Package[npm-cache]/ensure: created
Notice: /Stage[main]/Main/Package[grunt-cli]/ensure: created
Notice: /Stage[main]/Main/File[/home/donal/.zshrc]/ensure: defined content as '{md5}ae3126808fcd7ad0ff99fb32d535b44b'
Notice: Finished catalog run in 44.93 seconds
➜ puppet
```

NOTE: This can take a while, could be a good time for a coffee. The first time this was run in the test VM, it took around 6 minutes.

- The changes can be validated by running the following where `<NAME>` is the user you've just created

```
puppet resource user <NAME>
```

```
➜ puppet puppet resource user donal
user { 'donal':
  ensure => 'present',
  gid    => '5001',
  groups => ['sudo', 'docker', 'devops-course'],
  home   => '/home/donal',
  shell  => '/bin/zsh',
  uid    => '1010',
}
```

NOTE : For those who are curious, the puppet config can be shown by running `puppet config print` The `--debug` and `--verbose` can be set on all puppet commands to give additional information.

- The changes you have made to the puppet file should be added to the git repository you created at the beginning of the lab. Committing code frequently and running an automated build process on it shortens the feedback loop leading to improved code quality by learning from mistakes sooner. Our puppet files are not built using an automated process however it is still good practice to commit our code in chunks of work, as we have just added the code to add a new user, now is a good time to commit. To add out code to the repo, follow this work flow (you do not need to enter the comments, which includes the `#` and anything after it):

```
git status          #look through your changeset to ensure it contains
everything you                      expect
git add .           #stage your changes by adding it to the index
git commit -m 'Adding the new user to the puppet file'
                    #commit the staged changes with a meaningful message
git pull            #get the latest changes from the remote
git push            #push your changes to the remote (origin) set up in step 1
```

```
➜ lab-material git:(master) ✗ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   puppet/build-vm.pp

no changes added to commit (use "git add" and/or "git commit -a")
➜ lab-material git:(master) ✗ git add .
➜ lab-material git:(master) ✗ git commit -m "Adding the new user to the puppet file"
[master ff966fe] Adding the new user to the puppet file
 1 file changed, 4 insertions(+), 4 deletions(-)
➜ lab-material git:(master) git pull
Already up-to-date.
➜ lab-material git:(master) git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behaviour, use:

  git config --global push.default matching

To squelch this message and adopt the new behaviour now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behaviour, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 4, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 457 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
To ssh://git@git.server/home/git/lab-material.git
   c86974e..ff966fe  master -> master
➜ lab-material git:(master)
```

NOTE: This flow will be used often so glance over it when the instructions refer you to 'commit your changes'. The flow is also available in the lab cheat sheet for reference.

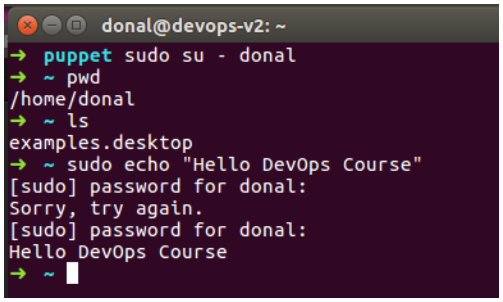
- Back on the terminal, lets validate our newly added user. Change to the new user you've created by running

```
sudo su - <USER>
```

- Validate the user has sudo access and other expected functionality as shown below by running the following, enter your users password when prompted:

```
echo $HOME # will print the path to the users home
whoami # will tell you the name of the current user
sudo echo "Hello DevOps Course" # will print Hello DevOps Course as root
```

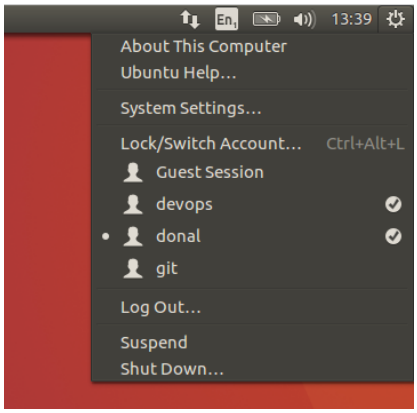
For example, if your username was "donal":



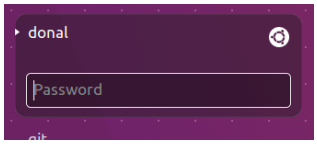
- Exit from the new users session

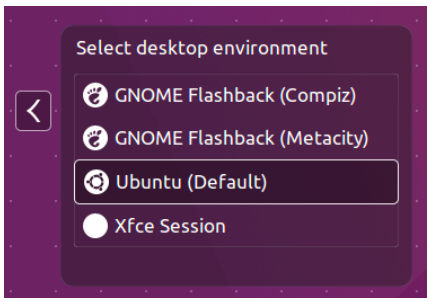
```
exit
```

- You can also login as the new user by hitting the cog in the upper right hand corner, and selecting the newly created user. This new user will have the default Ubuntu Unity Navigation bar and background.



The devops user is using a different Desktop environment based on the classic Ubuntu 10.04 desktop to improve the UI performance. To select the alternative Desktop environment, go to the login screen and hit the Ubuntu Icon by the users name. This will produce a new menu. Select the Metacity environment for a less whizzy experience which will be better for running inside the Virtualized environment.





The labs were written and tested using the devops user but it is possible to now use your own user from the after the next section (Ammend the existing user). However, the new user will not have the same shortcuts installed as the devops user so choose which one you want to use with this in mind.

Also, the devops user has that awesome Star Wars Jenkins background which makes enjoyment of the lab material 12 parsecs better.

Ammend the existing user.

- Ensure you are the devops user for this next bit by typing `whoami`. If you're still the new user type `exit`

```
whoami
```

- In this course, we will be working with files and directories owned by the devops-course group. We must add the `devops` user to the `devops-course` group so that they can also modify these files. Amend `devops` user configuration as follows:

```
in the lab-material project (~/lab-material) open the `./puppet/build-vm.pp`
puppet file
```

```
user { 'devops' :
  ensure => 'present',
  gid    => '5001',
  groups => ['devops-course', 'docker', 'sudo'],
  home   => '/home/devops'
} ->
```

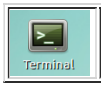
NOTE: the arrow after the above block denotes a dependency chain in puppet. This means to Puppet, that it must run that block before running the next block

- We are going to give access to the /share directory to the `'devops-course'` group. In the existing `file { "/share":` config block, add the group property as follows:

```
file { "/share":
  ensure  => 'directory',
  owner   => "devops",
  group   => 'devops-course',
  require => [ User["devops"], Group['devops-course']],
  recurse => true
}
```

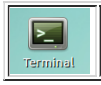
```
file { '/share':
  ensure  => 'directory',
  owner   => 'devops',
  group   => 'devops-course',
  require => [ User['devops'], Group['devops-course']],
  recurse => true
}
```

- Run the commands. You should now be able to access the things in `/share` as your new user. From this point on wards you can use either user.



```
cd ~/lab-material/puppet
sudo puppet apply build-vm.pp
```

- Close down Atom Text. Move the lab-material repo files into the share directory before continuing so that all the users created have access to the course content.



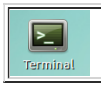
```
cd ~
mv -v /home/devops/lab-material /share
```

- We just moved the lab-material, so we must make sure we close down all of the Atom editors that are currently pointing to a directory that does not exist. Simply click the close button:

Add a new package to the VM

To add OS packaged, we can add to the existing Puppet manifest, who is already managing some of our `os packages`.

- Navigate to the `build-vm.pp` file in the lab-material project. We just closed Atom, so open it with:



```
atom /share/lab-material
```

```
in the lab-material project (/share/lab-material) open the `puppet/build-
vm.pp` puppet file
```

- Move to the `# os packages` section of the of the `build-vm.pp` and add `cowsay , fortune` and `sl` to the list of packages.

```
# os packages
package { ['vim', 'curl', 'wget', 'openssh-client', 'openssh-server', 'git-core', 'cowsay', 'sl', 'fortune']:
  ensure => present,
}
```

- Run the puppet script.

```
cd /share/lab-material/puppet
sudo puppet apply build-vm.pp
```

- You should now be able to use the previously unusable packages. Try it out!

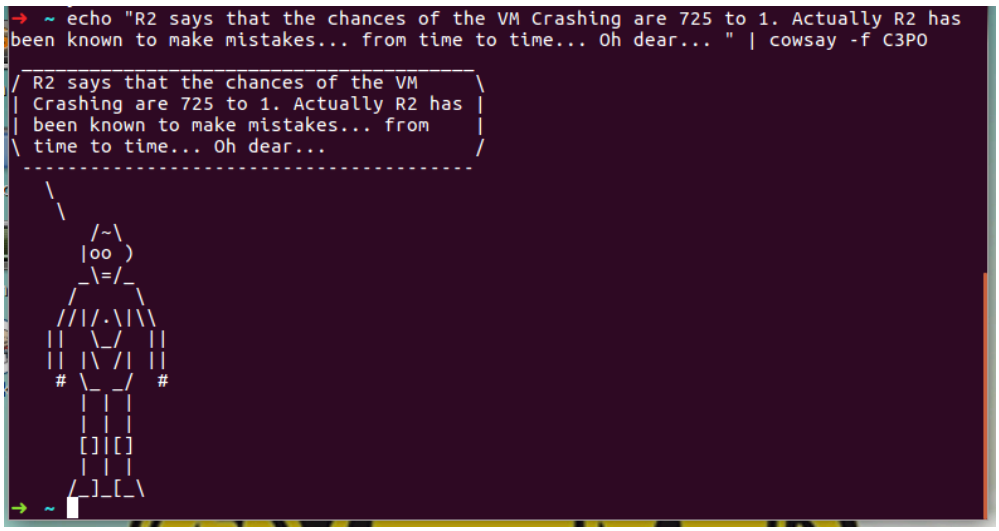
```
fortune | cowsay
sl
```

- This puppet file will be returned to when we want to add additional modules and packages to our VM. Commit your changes once more.

```
git status          #look through your changeset to ensure it contains
everything you                    expect
git add .           #stage your changes by adding it to the index
git commit -m 'Adding the new user to the puppet file'
                    #commit the staged changes with a meaningful message
git pull            #get the latest changes from the remote
git push            #push your changes to the remote (origin) set up in step 1
```

MINI EXTENSION TASK FOR FUN - Swap your cow for R2-D2 or C3P0

https://github.com/paulkaefer/cowsay-files



B) Installing Jenkins

The objective of this lab is to install Jenkins. Jenkins is used as our build server. Puppet will be used to manage Jenkins dependencies such as java. NOTE - there is a jenkins puppet module but it does not support v2 of Jenkins so we will manually add the jenkins repo and let puppet install the package.

```
in the lab-material project (/share/lab-material) open the `puppet/build-
vm.pp` puppet file
```

- Install Java and Jenkins along with some other things that are needed for Jenkins to run correctly. To do this, modify the `build-vm.pp` manifest to import the `jenkins-install.pp` manifest just after the `include git` statement. Your manifest should look as follows:

```
# modules to include

include git

# Jenkins install...
import 'jenkins-install.pp'
```

```
build-vm.pp ●   jenkins-install.pp ×
# modules to include

include git

# Jenkins install...
import 'jenkins-install.pp'
```

– At time of writing the puppet labs module for Jenkins (https://github.com/jenkinsci/puppet-jenkins) does not support v2 of Jenkins so we are going to install it as a package and set up the repo manually. This is not ideal, but until this version is fixed execute the following on the command line (as per the jenkins install instructions https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu). This has been wrapped into a configure script. Run the script using the commands below

Terminal

```
cd /share/lab-material
./scripts/jenkins-2-puppet-configure.sh
```

```
→ lab-material git:(master) ✗ ./scripts/jenkins-2-puppet-configure.sh
OK
Hit:1 http://gb.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Ign:3 http://dl.google.com/linux/chrome/deb stable InRelease
Get:4 http://gb.archive.ubuntu.com/ubuntu xenial-updates InRelease [95.7 kB]
Hit:5 http://dl.google.com/linux/chrome/deb stable Release
Hit:6 http://ppa.launchpad.net/webupd8team/sublime-text-3/ubuntu xenial InRele
ase
Get:7 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
Hit:8 http://gb.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:9 https://apt.dockerproject.org/repo ubuntu-xenial InRelease
Ign:10 http://pkg.jenkins.io/debian-stable binary/ InRelease
Ign:12 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 InRelease
Ign:13 https://dl.bintray.com/pcp/xenial xenial InRelease
Hit:14 http://pkg.jenkins.io/debian-stable binary/ Release
Get:15 https://dl.bintray.com/pcp/xenial xenial Release [1,839 B]
Get:16 http://gb.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packa
ges [319 kB]
Hit:15 https://dl.bintray.com/pcp/xenial xenial Release
Hit:17 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 Release
Get:21 http://gb.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packag
es [316 kB]
Fetched 825 kB in 1s (463 kB/s)
Reading package lists... Done
W: http://pkg.jenkins.io/debian-stable/binary/Release.gpg: Signature by key 15
0FDE3F7787E7D11EF4E12A9B7D32F2D50582E6 uses weak digest algorithm (SHA1)
```

If you get an error message with mongo gpg key expired here these can be ignored.

– Take a closer look at the jenkins-install.pp manifest file *(This is purely an explanatory step, and no action is required from you as a user)*:

Atom

```
From the lab-material project, open the puppet/jenkins-install.pp manifest
```

The file is doing the following:

1. Include the apt module, a package manager,
2. Ensure that the required version of Java is installed in order to run Jenkins.
3. Ensure that the Jenkins user is present, that they are in the correct groups and that they have the expected home directory.
4. The Keygen step is a workaround for the Jenkins module not supporting Jenkins 2 at this stage.
5. Ensure that Jenkins has the correct git configuration by using the git puppet module. This is needed to prevent git prompts to the Jenkins user, which is undesirable when running Jenkins builds.

(End Exploratory step)

– The missing step is to actually install the Jenkins package by using the package module:

Atom

```
in the lab-material project (/share/lab-material) open the `puppet/jenkins-
install.pp` puppet file
```

Atom

```
# Add jenkins package here !!
package { ['jenkins']:
    ensure     => present,
    require    => [ Package['openjdk-7-jdk'], ],
}
```

```
# Add jenkins package here !!
package {['jenkins']:
    ensure  =>  present,
    require =>  [Package['openjdk-7-jdk'],],
}
```

NOTE: that `require` is another way of managing dependency chains in puppet, for example above the `package jenkins` cannot be installed until the `package openjdk-7-jdk` is **installed by puppet**.

- Run the puppet script. This may take some time to run (potential coffee time). **Once complete, commit your changes to git.**

```
cd /share/lab-material/puppet
sudo puppet apply build-vm.pp
```

NOTE - You will now see the following warning about the use of 'import'. This is because Puppet wants you to make this a module, instead of an included manifest. This would involve introducing and configuring Puppet environments as well. This is overkill for the purpose of getting to grips with Puppet, however if you would like more information then head over to Puppet's documentation for understanding and configuring environments: https://docs.puppet.com/puppet/4.1/environments.html:

```
→ puppet git:(master) ✗ sudo puppet apply build-vm.pp
Warning: The use of 'import' is deprecated at /share/lab-material/puppet/build-vm.pp:7. See http://link
s.puppetlabs.com/puppet-import-deprecation
```

NOTE - if you see this error, just run the puppet apply again. It is caused by apt-get update not completing before running the install of Java (after we just set the PPA for it).

```
E: Package 'openjdk-7-jdk' has no installation candidate
Notice: /Package[jenkins]: Dependency Package[openjdk-7-jdk] has failures: true
Warning: /Package[jenkins]: Skipping because of failed dependencies
```

What does all of this do? (taken from the jenkins website):

- Jenkins will be launched as a daemon up on start. See `/etc/init.d/jenkins` for more details.
- The '`jenkins`' user is created to run this service.
- Log file will be placed in `/var/log/jenkins/jenkins.log`. Check this file if you are troubleshooting Jenkins.
- `/etc/default/jenkins` will capture configuration parameters for the launch like e.g JENKINS_HOME
- By default, Jenkins listen on port 8080. Access this port with your browser to start configuration.

- Navigate to http://jenkins.server:8080/ to see jenkins come alive. Or use the launcher on the Desktop

- Follow the onscreen setup instructions. Use the cmd below to quickly get the password from jenkins. Cut/paste by highlighting right-click|copy and right-click|paste on Jenkins.

```
sudo su jenkins -c "cat /var/lib/jenkins/secrets/initialAdminPassword"
```


- Install the default plugins for ease

- Create your user account

- Commit your changes once more by running the following from the /share/lab-material directory:


```
git status          #look through your changeset to ensure it contains
everything you                      expect
git add .           #stage your changes by adding it to the index
git commit -m 'Installing Jenkins'
                    #commit the staged changes with a meaningful message
git pull            #get the latest changes from the remote
git push            #push your changes to the remote (origin) set up in step 1
```

Comments

1-1 of 1 Previous Next

FIASTRU, ALEXANDRU IONUT (Alexandru Ionut) commented on 26 October 2017 Permalink
Workaround for step:
Running jenkins-2-puppet-confugre.sh

./scripts/jenkins-2-puppet-configure.sh

Due to mogodb repository key being expired the apt-get update command will fail completion.

To get rid of this go through following steps:

1) sudo apt-key list
(look for MongoDB id - should be something like /EA312927)

2) Remove the old key (expired on 08-10-2017)
sudo apt-key del EA312927

3) Add the new key (expires: 2019-10-09)
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927

4) Check the new key being added:
sudo apt-key list

Something like:
pub 4096R/EA312927 2015-10-09 [expires: 2019-10-09]
uid MongoDB 3.2 Release Signing Key <packaging@mongodb.com>
should appear for MongoDB

Show 10 25 50 items per page Previous Next