# **Exercise 1 - Puppet**

# **Getting Started by launching Centos7.2 Shell**

Now lets add repo

wget "https://apt.puppetlabs.com/puppetlabs-release-pc1-xenial.deb"

sudo dpkg -i puppetlabs-release-pc1-xenial.deb

sudo apt-get update

sudo apt-get install puppet-agent

lets change the sudoers file

run:

sudo visudo

add to secure\_path the path of puppet like in slide 17

then reboot

now lets run ad hoc commands

sudo puppet agent –version

sudo puppet config print

sudo puppet config print confdir

sudo puppet config print certname

sudo puppet config print { confdir rundir ssldir runinterval }

sudo puppet resource user <OS-User>

sudo puppet resource user <your name>

Lets write first manifest :

Vim helloworld.pp

Notify { ‘ Hello World!’ : }

Exit and save

sudo puppet parser validate helloworld.pp

sudo puppet apply helloworld.pp

now run :

sudo puppet apply –e “notify { ‘ Hello World’ : }”

did you get the same results?

Now run :

Sudo puppet resource service puppet

sudo puppet resource service puppet > puppet-service.pp

sudo puppet parser validate puppet-service.pp

now lets start the service and use puppet-service.pp to shut it down

sudo systemctl start puppet

sudo systemctl status puppet

sudo puppet apply puppet-service.pp

Now lets change the puppet-service.pp

Vi puppet-service.pp

service { 'puppet':

ensure => 'running',

enable => 'true',

}

exit and save

now run the puppet apply again

IDE syntax and highlight

lets install vim if it is not installed…

yum –y install vim

sudo puppet module install theurbanpenguin/puppet\_vim

sudo puppet module list

sudo puppet apply -e " include puppet\_vim"

now the IDE is ready with style and symtax highlight

now lets install standard library module

sudo puppet module install puppetlabs/stdlib

sudo puppet describe –list

sudo puppet describe notify

sudo puppet describe user

sudo puppet describe user –short

NameVar:

Vim file.pp

file { '/etc/motd':

ensure => 'file',

content => 'Welcome to my Server',

}

sudo puppet apply file.pp

cat /etc/motd

another way to do this :

vim file.pp

file { 'Message File':

ensure => 'file',

content => 'Welcome to my Server',

path => '/etc/motd',

}

and run puppet aply again

now let write ntp manifest

vim ntp.pp

$ntp\_conf = '#Managed by puppet

server 10.0.2.15 iburst

driftfile /var/lib/ntp/drift

'

package { 'ntp':

ensure => 'installed',

provider => 'yum',

}

file { '/etc/ntp.conf':

ensure => 'file',

content => $ntp\_conf,

owner => 'root',

group => 'wheel',

mode => '0664',

}

service {'NTP\_Service':

ensure => 'running',

enable => true,

name => 'ntpd'

}

sudo puppet parser validate ntp.pp

sudo puppet apply ntp.pp

Now lets create 2 users with puppet

First create a user with your name

Vim user.pp

user { '<yourname>':

managehome => true,

uid => '2004',

ensure => 'present',

groups => ['root','wheel','users'],

password => pw\_hash('Password1','SHA-512','random'),

}

user { 'u2':

managehome => true,

uid => '2003',

ensure => 'present',

groups => ['wheel','users'],

password => pw\_hash('Password1','SHA-512','random'),

}

sudo puppet apply user.pp

now check in your linux that users are created

now create a group.pp

group { 'admin':

ensure => 'present',

}

run the pp file

and check if group exist

now lets remove the group

vim rmgroup.pp

group { 'admin':

ensure => 'absent',

}

Create hosts file enrty

Ping tock

--should fail

Vim host.pp

host { 'centos2':

ip => '<the ip of seconfd vm>',

host\_aliases => 'tock',

}

now run the pp file

ping tock again

lets generate and distribute ssh key

ssh-keygen -t rsa

cd .ssh/

cat id\_rsa.pub

now copy the key

cd ..

vim ssh.pp

ssh\_authorized\_key { '<yourkeyname>':

user => '<yourname>',

type => 'ssh-rsa',

key => 'AAAB3NzaC1yc2EAAAADAQABAAABAQCWDo3I01jgkQJr6xfUbt3sQE4E8QjSEU0NqLpMiOs44rYbMoJMpJwt9oAZ55okxnNRrew1KxmIsgCs4JSNWENN5z5NPpE34W8MShJvSyqjYEMvnM12SJJNnGOeiwpLS7H59AjxvjMM2mkeQCwd4moZRzkEvSTlf2WhhLCd476YGTatCRS0G24vk60wqbiWY7lp9qN+tku6ti41B+WAJl6InCDqa13diiCcqLUMm0qDdQp46ndlNUcW7d02Gg0KKPs93tQYqCHqS+Q08w0EEF/IAP/V4SNH2tAXrEAPw6Sqk4ljngQ4447yRCpP8J00ivEyIIfzXn5niv7OYmx7VgEX',

}

now copy the ssh.pp file to remote server with puppet and run

sudo puppet apply ssh.pp

check you have the public key in authorized\_keys

now go back to your local vm

and try to ssh to remote server

now lets run facter

simply type

facter

facter partitions

facter kernel

facter os.family

now edit ntp.pp

with facter so it can work with Ubuntu also

use either if or case or selector