**Puppet Master Installation:**

wget https://apt.puppetlabs.com/puppet6-release-xenial.deb

sudo dpkg -i puppet6-release-xenial.deb

apt update

apt-get install puppetserver

puppetserver ca setup

service puppetserver start

**Puppet Agent Installation:**

wget https://apt.puppetlabs.com/puppet6-release-xenial.deb

sudo dpkg -i puppet6-release-xenial.deb

apt update

apt-get install puppet-agent

Installing Puppet Using Ansbile:

**Puppet Agent Config:**

[main]

certname = puppetclient2.local

server = puppet

environment = development

runinterval = 1h

**User Manifests:**

user { 'produser':

ensure => 'present',

home => '/home/produser',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

uid => '501',

}

user { 'devuser':

ensure => 'present',

home => '/home/devuser',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

uid => '501',

}

**Puppet Docker Lab:**

docker network create --subnet 172.20.0.0/16 --gateway 172.20.0.254 puppet\_nw

docker run --rm -dit --name puppet --hostname puppet -p 8140:8140 --network puppet\_nw sreeharshav/puppetmaster:nocasetupV2

Login to the puppet server and check the server status

Create the autosign file and allow auto sign

root@puppet:/# cat /etc/puppetlabs/puppet/autosign.conf

\*.local

\*.ec2.internal

root@puppet:/# nano /etc/puppetlabs/puppet/puppet.conf

[master]

vardir = /opt/puppetlabs/server/data/puppetserver

logdir = /var/log/puppetlabs/puppetserver

rundir = /var/run/puppetlabs/puppetserver

pidfile = /var/run/puppetlabs/puppetserver/puppetserver.pid

codedir = /etc/puppetlabs/code

autosign = true

root@puppet:/#puppetserver ca setup

root@puppet:/# service puppetserver restart

docker run --rm -dit --name puppetclient1 --hostname puppetclient1 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient1 puppet agent -t &&

docker run --rm -dit --name puppetclient2 --hostname puppetclient2 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient2 puppet agent -t

docker run --rm -dit --name puppetclient3 --hostname puppetclient3 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient3 puppet agent -t

docker run --rm -dit --name puppetclient4 --hostname puppetclient4 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient4 puppet agent -t

docker run --rm -dit --name puppetclient5 --hostname puppetclient5 -p 8000:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient5 puppet agent -t

docker run --rm -dit --name puppetclient6 --hostname puppetclient6 -p 8100:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient6 puppet agent -t

docker run --rm -dit --name webserver1 --hostname webserver1 -p 8000:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it webserver1 puppet agent -t

docker run --rm -dit --name webserver2 --hostname webserver2 -p 8001:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it webserver2 puppet agent -t

docker run --rm -dit --name webserver3 --hostname webserver3 -p 8002:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it webserver3 puppet agent -t

docker exec -it puppetclient2 sed -i "2i certname = puppetagent2" /etc/puppetlabs/puppet/puppet.conf

On Ansible controller do below:

Cd /root/.ansible/roles

Git clone <https://github.com/mavrick202/ansiblepuppetinstall.git>

Update the ansible host file

Deploy the Puppet Master Playbook

Check the communications from ansible to all clients

Deploy puppet master playbook.

root@controller:~/.ansible/roles# ansible-playbook -i hosts master.yml

Restart the puppet service

root@puppet:/# service puppetserver restart

cp -rf production/ development

Deploy Puppetclient Playbook

root@controller:~/.ansible/roles# ansible-playbook -i hosts prod.yml

ansible -i hosts -a "cat /etc/puppetlabs/puppet/puppet.conf" puppetclient1

root@controller:~/.ansible/roles# ansible-playbook -i hosts dev.yml

ansible -i hosts -a "cat /etc/puppetlabs/puppet/puppet.conf" puppetclient11

docker run --rm -dit --name puppetclient7 --hostname puppetclient7 --network puppet\_nw sreeharshav/ubuntu16:withssh2.0\_V2

docker run --rm -dit --name puppetclient17 --hostname puppetclient17 --network puppet\_nw sreeharshav/ubuntu16:withssh2.0\_V2

docker stop $(docker ps -q --filter name=^puppet)

Running Puppet Agent:

i=1

while [ $i -lt 5 ]; do docker exec -it puppetclient$i puppet agent -t; i=$((i+1)); done

#!/bin/bash

docker exec -it puppetclient1 puppet agent -t

docker exec -it puppetclient2 puppet agent -t

docker exec -it puppetclient3 puppet agent -t

docker exec -it puppetclient4 puppet agent -t

docker exec -it puppetclient5 puppet agent -t

docker exec -it puppet puppet agent -t

**Puppet Relationship & Ordering:**

[**https://puppet.com/docs/puppet/6.2/lang\_relationships.html**](https://puppet.com/docs/puppet/6.2/lang_relationships.html)

node default {

include files

}

class files {

file { '/tmp/file1':

ensure => file,

mode => '0600',

#require => File['/tmp/file2'],

#before => File['/tmp/file2'],

}

file { '/tmp/file2':

ensure => file,

mode => '0600',

#before => File['/tmp/file1'],

}

File['/tmp/file1'] -> File['/tmp/file2']

}

**Installing Nginx:**

class mynginx {

exec { 'apt update':

command => "/usr/bin/apt update",

}

exec { 'Dele Repo if Exists':

command => "/bin/rm -rf /tmp/dockertest1"

}

exec { 'Clone Repo':

command => "/usr/bin/git clone -b DevOpsB10 https://github.com/mavrick202/dockertest1.git /tmp/dockertest1",

}

package { 'nginx':

ensure => installed,

}

file { "/var/www/html" :

ensure => directory,

source => "/tmp/dockertest1",

recurse => true,

backup => false,

}

service { "nginx":

ensure => running,

enable => true,

hasrestart => true,

hasstatus => true,

# pattern => 'nginx',

}

Exec['apt update'] -> Exec['Dele Repo if Exists'] -> Exec['Clone Repo'] -> Package['nginx'] -> File['/var/www/html'] ~> Service['nginx']

}

----------------------------------------------------------------------------------------

Puppet Pamaterized Clases:

class users ($ensure, $username) {

user { $username:

ensure => $ensure,

home => '/home/produser',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

node puppetclient1 {

class {'users':

username => 'testuser1',

ensure => 'present',

}

}

node puppetclient2 {

class {'users':

username => 'sreeharsha',

ensure => 'present',

}

}

node puppetclient3 {

class {'users':

username => 'testuser3',

ensure => 'present',

}

}

node puppetclient4 {

class {'users':

username => 'testuser4',

ensure => 'present',

}

}

Puppet Paramatrized Class from Module:

node default {

class {'newusers::users':

username => 'testuser1',

ensure => 'present',

}

}

========================================================

**Puppet Defined Types:**

node default {

notify {'THIS IS DEFAULT CONFIG':}

}

node puppetclient1 {

users::testusers {'testuser1': }

users::testusers {'testuser2': }

users::testusers {'testuser3': }

users::testusers {'testuser4': }

users::testusers {'testuser5': }

}

define users::testusers {

user {$name:

ensure => 'present',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

class testing {}

init.pp

-------

node 'puppetclient2.ec2.internal' {

$newusers = ["one","two","three"]

users::testusers {[$newusers]:}

}

node 'puppetclient1.ec2.internal' {

$newusers = ["four","five","six"]

users::testusers {[$newusers]:}

}

/etc/puppetlabs/code/environments/production/modules/users/manifests

define users::testusers {

user {$name:

ensure => 'present',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

node default {

notify {'DEFAULT\_CONFIG':}

}

node puppetclient1 {

$myusers=['Sree','Harsha','Phani','Prasad','Rajesh']

users::definecass {[$myusers]:}

}

node puppetclient2 {

$myusers=['Sree111','Harsha111','Phani111','Prasad111','Rajesh111']

users::definecass {[$myusers]:}

}

**Multiple user using defined type in init.pp:**

node default {

include users

users::defineusers{'sree10':

mystate => 'absent',

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree10'

}

users::defineusers{'sree11':

mystate => 'absent',

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree11'

}

users::defineusers{'sree12':

mystate => 'absent',

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree12'

}

}

=================DEFINE-WITH-USERS==========================

node default {

new\_user2 {'testuser1': username => 'testuser1'}

new\_user2 {'testuser2': username => 'testuser2'}

new\_user2 {'testuser3': username => 'testuser3'}

new\_user2 {'testuser4': username => 'testuser4'}

new\_user2 {'testuser5': username => 'testuser5'}

}

define new\_user2($username){

user{$username:

ensure => present,

}

}

=====================DEFINE-IN-MODULE================================

root@puppet:/etc/puppetlabs/code/environments/production/modules/myusers/manifests# cat defineusers.pp

# @summary A short summary of the purpose of this class

#

# A description of what this class does

#

# @example

# include myusers::defineusers

define myusers::defineusers ($username) {

user {$username:

ensure => present,

}

}

#Node declaration with the defined type in the modue myusers::defineusers

node default {

class {new\_user:

username => 'testuser1',

}

#Accessing a module myusers::defineusers which mentioned above.

myusers::defineusers {'adminuser20': username => 'adminuser20'}

myusers::defineusers {'adminuser30': username => 'adminuser30'}

myusers::defineusers {'adminuser40': username => 'adminuser40'}

myusers::defineusers {'adminuser50': username => 'adminuser50'}

}

===============================================================

**The define type code for multi users:**

root@puppet:/etc/puppetlabs/code/environments/production/modules/users/manifests# cat defineusers.pp

# A description of what this class does

#

# @summary A short summary of the purpose of this class

#

# @example

# include users::defineusers

define users::defineusers ($mystate, $myname, $mypassword, $myshell){

user {$myname:

ensure => $mystate,

password => $mypassword,

password\_max\_age => '99999',

password\_min\_age => '0',

shell => $myshell,

}

}

**Heira Example Usage:**

cat /etc/puppetlabs/code/environments/production/hiera.yaml

# <ENVIRONMENT>/hiera.yaml

---

version: 5

hierarchy:

- name: Common

path: common.yaml

defaults:

data\_hash: yaml\_data

datadir: data

cat /etc/puppetlabs/code/environments/production/data/common.yaml

---

ldap\_servers:

- 10.132.17.196

- 10.132.17.195

users:

joe:

home: '/home/joe'

jenkins:

password: 'mysecret'

dns\_servers:

- 8.8.8.8

- 8.8.4.4

users::myname: 'sreeharsha'

users::mypassword: 'India@123456'

users::myshell: '/bin/bash'

users::mystate: 'absent'

users::defineusers::mystate: 'absent'

users::mydefine::uname: 'testuser1'

users::mydefine::state: 'absent'

root@puppet:/etc/puppetlabs/code/environments/production/modules/users/manifests# puppet lookup users::myname

--- sreeharsha

root@puppet:/etc/puppetlabs/code/environments/production/modules/users/manifests# puppet lookup users::defineusers::mystate

--- absent

**Main init.pp file for hiera testing:**

node default {

include users

users::mydefine {'uname1':

uname => lookup('users::mydefine::uname'),

state => lookup('users::mydefine::state'),

}

users::mydefine {'uname2':

uname => 'Testuser2',

state => lookup('users::mydefine::state'),

}

users::mydefine {'uname3':

uname => 'Testuser3',

state => lookup('users::mydefine::state'),

}

users::defineusers{'sree10':

mystate => lookup('users::defineusers::mystate'),

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree10'

}

users::defineusers{'sree11':

mystate => lookup('users::defineusers::mystate'),

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree11'

}

users::defineusers{'sree12':

mystate => lookup('users::defineusers::mystate'),

mypassword => 'India@123456',

myshell => '/bin/bash',

myname => 'sree12'

}

}

END=6

for ((i=1;i<=END;i++)); do

docker exec -it puppetclient$i puppet agent -t

done

for ((i=1;i<7;i++)); do

docker exec -it puppetclient$i puppet agent -t

done

MAIN MANIFEST CLASSES:

--------------------------------------

class produsers {

user { 'produser1':

ensure => 'absent',

home => '/home/produser1',

password => '$1$JX/MczRP$C9G1TbcDl3SYpQ0UH0w1A.',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

user { 'produser2':

ensure => 'absent',

home => '/home/produser2',

password => '$1$JX/MczRP$C9G1TbcDl3SYpQ0UH0w1A.',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

class devusers {

user { 'devuser1':

ensure => 'absent',

home => '/home/devuser1',

password => '$1$JX/MczRP$C9G1TbcDl3SYpQ0UH0w1A.',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

user { 'devuser2':

ensure => 'absent',

home => '/home/devuser2',

password => '$1$JX/MczRP$C9G1TbcDl3SYpQ0UH0w1A.',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

PUPPET CONDITIONAL AND LOOPS:

<https://puppet.com/docs/puppet/6.2/lang_conditional.html>

PUPPET APPLY WITH FACTS:

cat /tmp/ostype.pp

if ($osfamily == "RedHat"){

$message = "This machine OS is of the type $osfamily \n"

}

elsif ($osfamily == "Debian"){

$message = "This machine OS is of the type $osfamily \n"

}

else {

$message = "This machine is unknown \n"

}

file { "/tmp/machinetype.txt":

ensure => file,

content => "$message"

}

==============================INHERITANCE=========================================

node default {

#notify {'TEST\_MESSAGE':}

include four

}

class one{

notify {'CMG FROM CLASS-1':}

}

class two inherits one{

notify {'CMG FROM CLASS-2':}

}

class three inherits two{

notify {'CMG FROM CLASS-3':}

}

class four inherits three {

notify {'CMG FROM CLASS-4':}

}

=======================ORDER-EXECUTION===========================================

node default {

#notify {'TEST\_MESSAGE':}

include myfiles

}

class myfiles{

file { '/tmp/file1':

ensure => file,

content => 'This is Testfile1',

mode => '0600',

}

file { '/tmp/file2':

ensure => file,

content => 'This is Testfile2',

mode => '0600',

}

file { '/tmp/file3':

ensure => file,

content => 'This is Testfile3',

mode => '0600',

}

File['/tmp/file3'] -> File['/tmp/file2'] -> File['/tmp/file1']

}

https://codingbee.net/puppet/puppet-class-parameters

puppet inheritance

https://codingbee.net/puppet/puppet-class-inheritance

puppet Relationships and ordering

https://puppet.com/docs/puppet/5.3/lang\_relationships.html

Writing Better Puppet Modules

https://elatov.github.io/2014/09/writing-better-puppet-modules/

Adding debug messages to your manifests

<https://www.puppetcookbook.com/posts/simple-debug-messages.html>

==========================PROFILE & ROLES==========================

root@puppet:/etc/puppetlabs/code/environments/production/modules/profile# cat manifests/test1.pp

# @summary A short summary of the purpose of this class

#

# A description of what this class does

#

# @example

# include profile::test1

class profile::test1 {

notify{'PROFILE-TEST-1':}

}

root@puppet:/etc/puppetlabs/code/environments/production/modules/profile# cat manifests/main.pp

# @summary A short summary of the purpose of this class

#

# A description of what this class does

#

# @example

# include profile::main

class profile::main {

include profile::test1

include profile::test2

include profile::test3

}

node default {

#include profile::test1

#include profile::test2

#include profile::test3

include role::main

}

ROLE and PROFILE Testing:

Modules Created as below:

root@puppet:/etc/puppetlabs/code/environments/production/modules# ll

total 44

drwxr-xr-x 1 root root 4096 Jan 28 02:03 ./

drwxr-xr-x 1 root root 4096 Jan 16 2019 ../

drwxr-xr-x 10 root root 4096 Jan 28 01:54 prodmodule1/

drwxr-xr-x 10 root root 4096 Jan 28 01:54 prodmodule2/

drwxr-xr-x 10 root root 4096 Jan 28 01:54 prodmodule3/

drwxr-xr-x 10 root root 4096 Jan 28 02:03 prodprofile/

drwxr-xr-x 11 root root 4096 Jan 28 02:03 profile/

drwxr-xr-x 10 root root 4096 Jan 28 01:47 role/

drwxr-xr-x 10 root root 4096 Jan 28 01:48 testmodule1/

drwxr-xr-x 10 root root 4096 Jan 28 01:48 testmodule2/

drwxr-xr-x 10 root root 4096 Jan 28 01:48 testmodule3/

node default {

include testmodule1::main

include testmodule2::main

include testmodule3::main

include prodmodule1::main

include prodmodule2::main

include prodmodule3::main

}

node default {

include profile::main

include prodprofile::main

}

node default {

include role::main

}