First edit the hostnames:

1. Rename the puppet server to puppet in /etc/hostname & /etc/hosts
2. Rename the puppet agent clients and add to puppet server /etc/hosts.

FROM ubuntu:18.04

RUN apt update

RUN apt install -y wget git

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

RUN dpkg -i puppet6-release-bionic.deb

RUN apt update

RUN apt-get install puppet-agent -y

RUN cp /opt/puppetlabs/puppet/bin/puppet /usr/local/bin/puppet

sreeharshav/puppetagent:V2 => 16.04

sreeharshav/puppetagent:V3 => 18.04

**Puppet Master Installation:**

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

sudo dpkg -i puppet6-release-bionic.deb

apt update

apt-get install puppetserver -y

puppetserver ca setup

service puppetserver start

**Puppet Agent Installation:**

apt update

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

sudo dpkg -i puppet6-release-bionic.deb

apt update

apt-get install puppet-agent -y

**Puppet Master Installation on Ubuntu 18.04 Bionic:**

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

sudo dpkg -i puppet6-release-bionic.deb

apt update

apt-get install puppetserver -y

puppetserver ca setup

service puppetserver start

**Puppet Agent Installation Ubuntu 18.04:**

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

sudo dpkg -i puppet6-release-bionic.deb

apt update

apt-get install puppet-agent -y

Installing Puppet Using Ansbile:

**Puppet Agent Config:** nano /etc/puppetlabs/puppet/puppet.conf

[main]

certname = puppetclient2.local

server = puppet

environment = development

runinterval = 1h

**User Manifests:**

Default Node Config:

node default {

notify {'This\_Is\_Default\_Config':}

include files

include apps

}

class files {

file { '/tmp/testfile1':

ensure => file,

content => "Testing The Puppet Agent Schedule. Welcome To Pupept Class."

}

}

class apps {

package { 'stress':

ensure => installed,

provider => apt

}

package { 'jq':

ensure => installed,

provider => apt

}

}

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 APPLY:**

Its not mandatory to have the puppet server to apply the configuration. Sometimes companies uses puppet called as masterless puppet.

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-masterless-puppet-environment-on-ubuntu-14-04>

**Node Definitions:**

node default {

notify {'Default\_Node\_Config':}

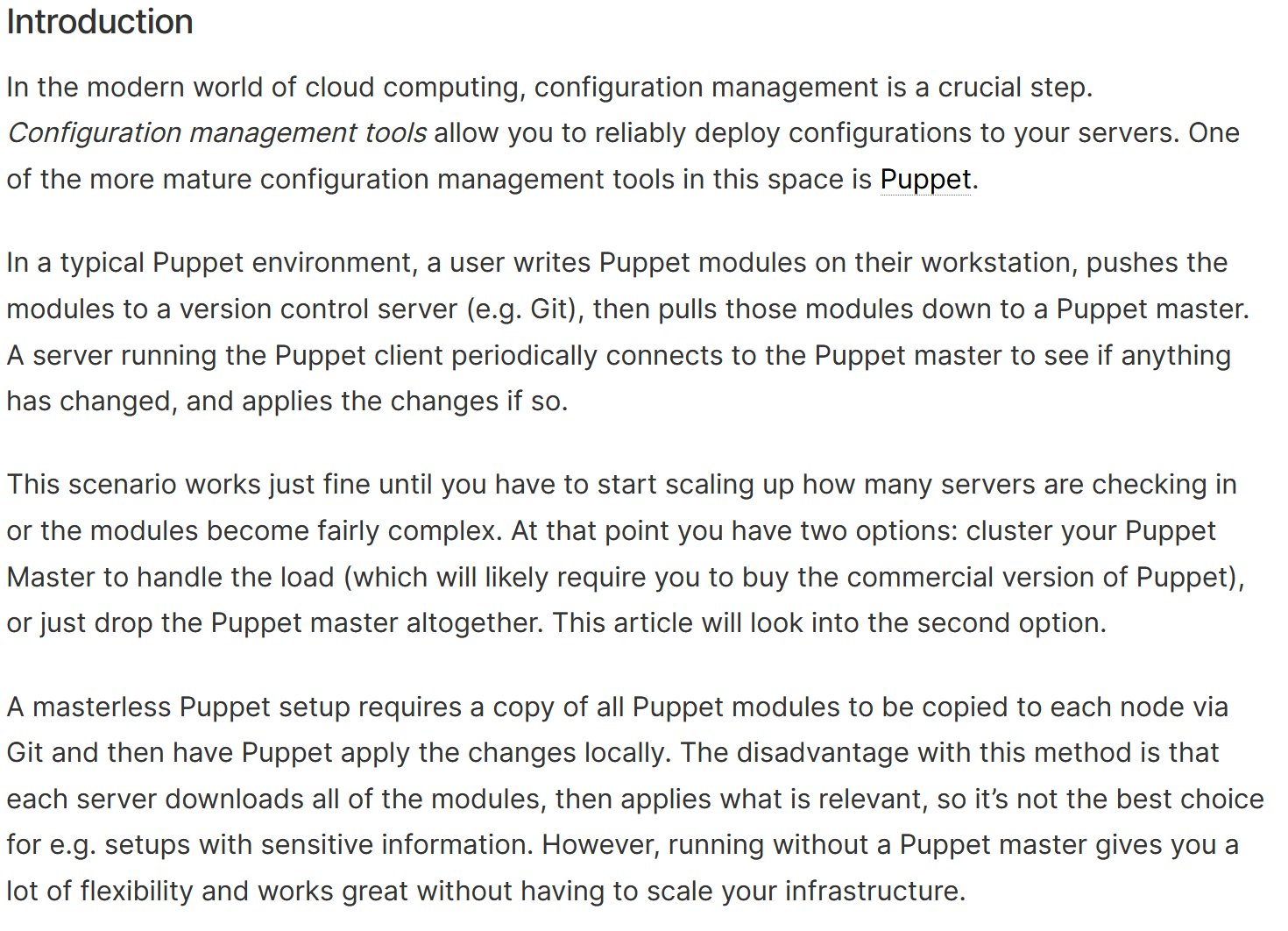
}

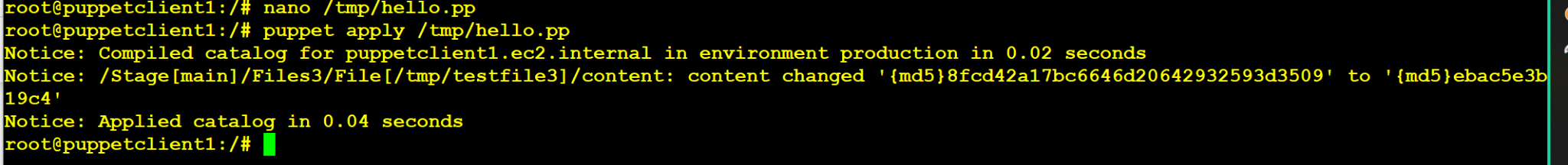
node /^web/ {

include mynginx

include myfiles

}





**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 -p 8000:80 --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

To remove and add the agent with the same name:

On puppet server run the following command:

puppetserver ca clean --certname puppetclient1.ec2.internal

On the agent level, go to the directory /etc/puppetlabs/puppet/ssl/certs and delete all files.

docker run --rm -dit --name puppetclient1 --hostname puppetclient1 -p 8100:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient1 puppet agent -t &&

docker run --rm -dit --name puppetclient2 --hostname puppetclient2 -p 8200:80 --network puppet\_nw sreeharshav/puppetagent:V2 && docker exec -it puppetclient2 puppet agent -t

docker run --rm -dit --name puppetclient3 --hostname puppetclient3 -p 8300:80 --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

Using puppet agent V3:

docker run --rm -dit --name webserver11 --hostname webserver11 -p 9000:80 --network puppet\_nw sreeharshav/puppetagent:V3 && docker exec -it webserver11 puppet agent -t

docker run --rm -dit --name webserver22 --hostname webserver22 -p 9001:80 --network puppet\_nw sreeharshav/puppetagent:V3 && docker exec -it webserver22 puppet agent -t

docker run --rm -dit --name webserver33 --hostname webserver33 -p 9002:80 --network puppet\_nw sreeharshav/puppetagent:V3 && docker exec -it webserver33 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 {

notify {'Default\_Node\_Config':}

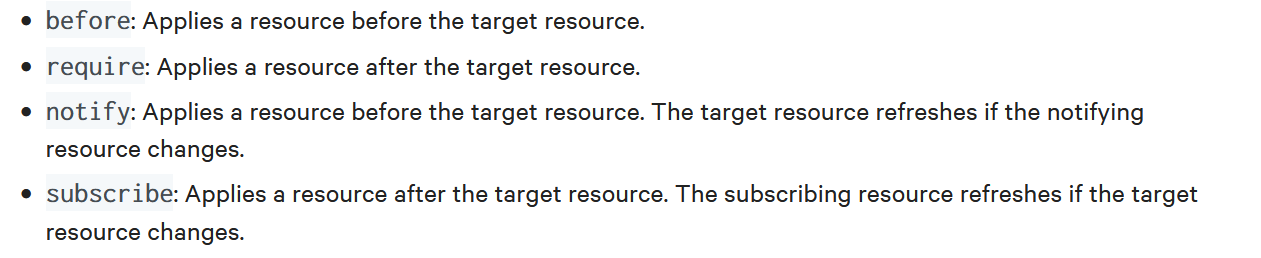
}

node /^web/ {

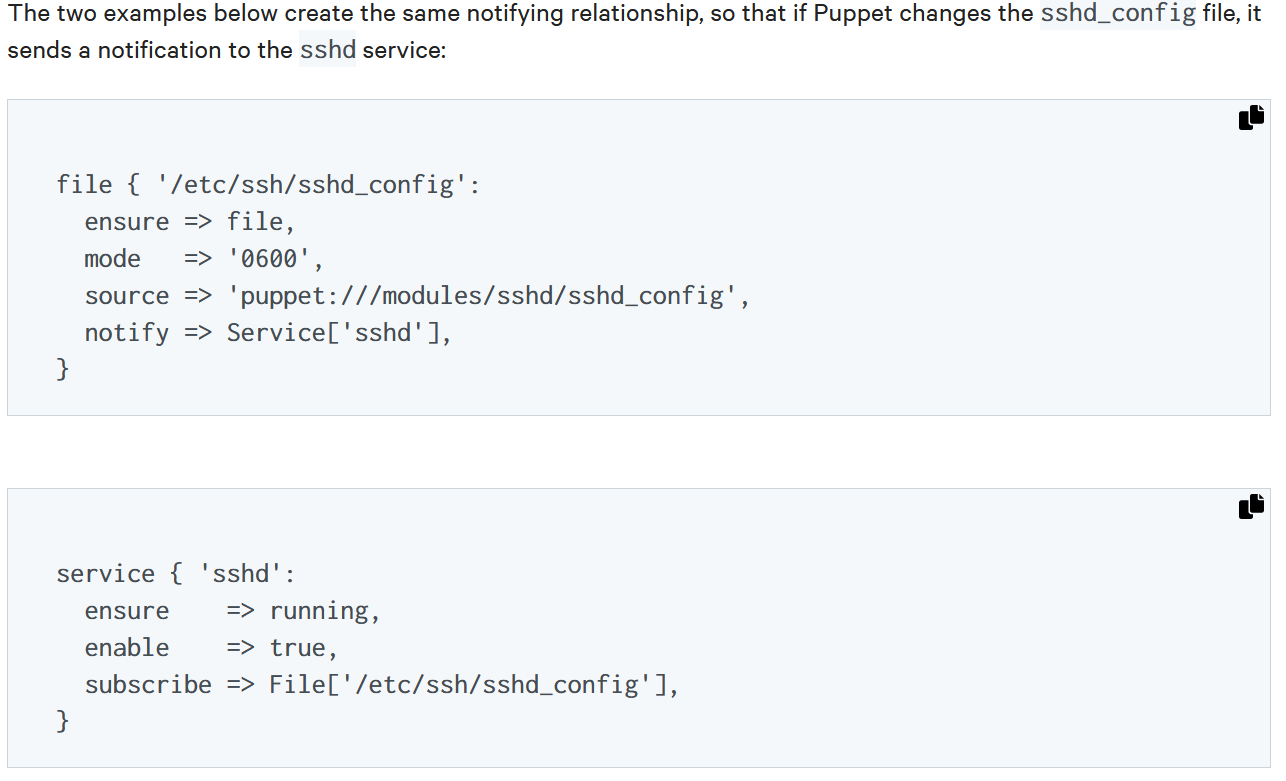
include mynginx

include myfiles

}







class myfiles {

file { '/tmp/file1':

ensure => 'file',

mode => '0600',

content => "Welcome. This Is File1",

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

}

file { '/tmp/file2':

ensure => 'file',

mode => '0774',

content => "Welcome. This Is File2",

#require => File['/tmp/file3'],

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

}

file { '/tmp/file3':

ensure => 'file',

mode => '0776',

content => "Welcome. This Is File3",

#require => File['/tmp/file4'],

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

}

file { '/tmp/file4':

ensure => 'file',

mode => '0777',

content => "Welcome. This Is File4",

before => File['/tmp/file3'],

}

}

**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 PDK:

<https://puppet.com/docs/pdk/1.x/pdk.html>

wget https://apt.puppet.com/puppet-tools-release-bionic.deb

dpkg -i puppet-tools-release-bionic.deb

apt-get update

apt-get install pdk -y

Puppet Pamaterized Clases:

class users ($username, $ensure) {

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':}

}

dusers {'p1':

ensure => 'present',

username => 'produser001',

}

dusers {'p2':

ensure => 'present',

username => 'produser002',

}

define users::testusers {

user {$name:

ensure => 'present',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

class testing {}

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

Define type Module User:

define defineusers::dusers ($username, $ensure) {

user {$username:

ensure => $ensure,

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

Define type Module Package:

define installapp::app ($application)

{

package { $application:

ensure => installed,

}

}

In Node Def:

installapp::app {'unzip':

application => 'unzip'

}

installapp::app {'tree':

application => 'tree'

}

installapp::app {'nginx':

application => 'nginx'

}

Define type Node Definitions:

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

node 'webserver3.ec2.internal','webserver2.ec2.internal','webserver1.ec2.internal' {

defineusers::dusers{'Creating Munanbhai':

username => 'munnabhai',

ensure => 'present',

}

defineusers::dusers{'Creating Cirtcuit':

username => 'circuit',

ensure => 'present',

}

defineusers::dusers{'Creating xyz':

username => 'xyz',

ensure => 'present',

}

defineusers::dusers{'Creating abc':

username => 'abc',

ensure => 'present',

}

}

==

pdk new module mydefine --skip-interview

cd mydefine/

pdk new class users

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

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

#

# A description of what this class does

#

# @example

# include mydefine::users

define mydefine::users {

user {$name:

ensure => 'present',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

Node-Def File for Define Classes:

node /^puppetclient\d+$/ {

include users::myusers1

include myapps::applications

include awscli

mydefine::users {'amar': }

mydefine::users {'akbar': }

mydefine::users {'anthony': }

}

#Calling module for each user

defineusers::createusers {'amar': }

defineusers::createusers {'akbar': }

defineusers::createusers {'anthony': }

defineusers::createusers {'megastar': }

#Calling module for multiple users

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

defineusers::createusers {[$myusers]:}

define defineusers::createusers{

user {$name:

ensure => 'present',

home => '/home/$username',

password => 'India@123456',

password\_max\_age => '99999',

password\_min\_age => '0',

shell => '/bin/bash',

}

}

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# cat data/common.yaml

---

dns\_servers:

- "8.8.8.8"

- "8.8.4.4"

ldap\_servers:

- "10.132.17.196"

- "10.132.17.195"

mon\_servers:

eastus:

- "10.132.17.200"

- "10.132.17.201"

westus:

- "10.132.17.205"

- "10.132.17.206"

india:

- "10.132.1.205"

- "10.132.2.205"

users:

jenkins:

password: mysecret

joe:

home: /home/joe

root@puppet:/etc/puppetlabs/code/environments/production# puppet lookup mon\_servers.india

---

- 10.132.1.205

- 10.132.2.205

root@puppet:/etc/puppetlabs/code/environments/production# puppet lookup mon\_servers.eastus

---

- 10.132.17.200

- 10.132.17.201

root@puppet:/etc/puppetlabs/code/environments/production# puppet lookup mon\_servers.westus

---

- 10.132.17.205

- 10.132.17.206

node default {

include files

}

class files {

file { '/tmp/file1':

ensure => file,

mode => '0600',

content => lookup('mon\_servers.eastus'),

}

file { '/tmp/file2':

ensure => file,

mode => '0600',

content => lookup('mon\_servers.westus'),

}

file { '/tmp/file3':

ensure => file,

mode => '0600',

content => lookup('mon\_servers.india'),

}

}

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

}

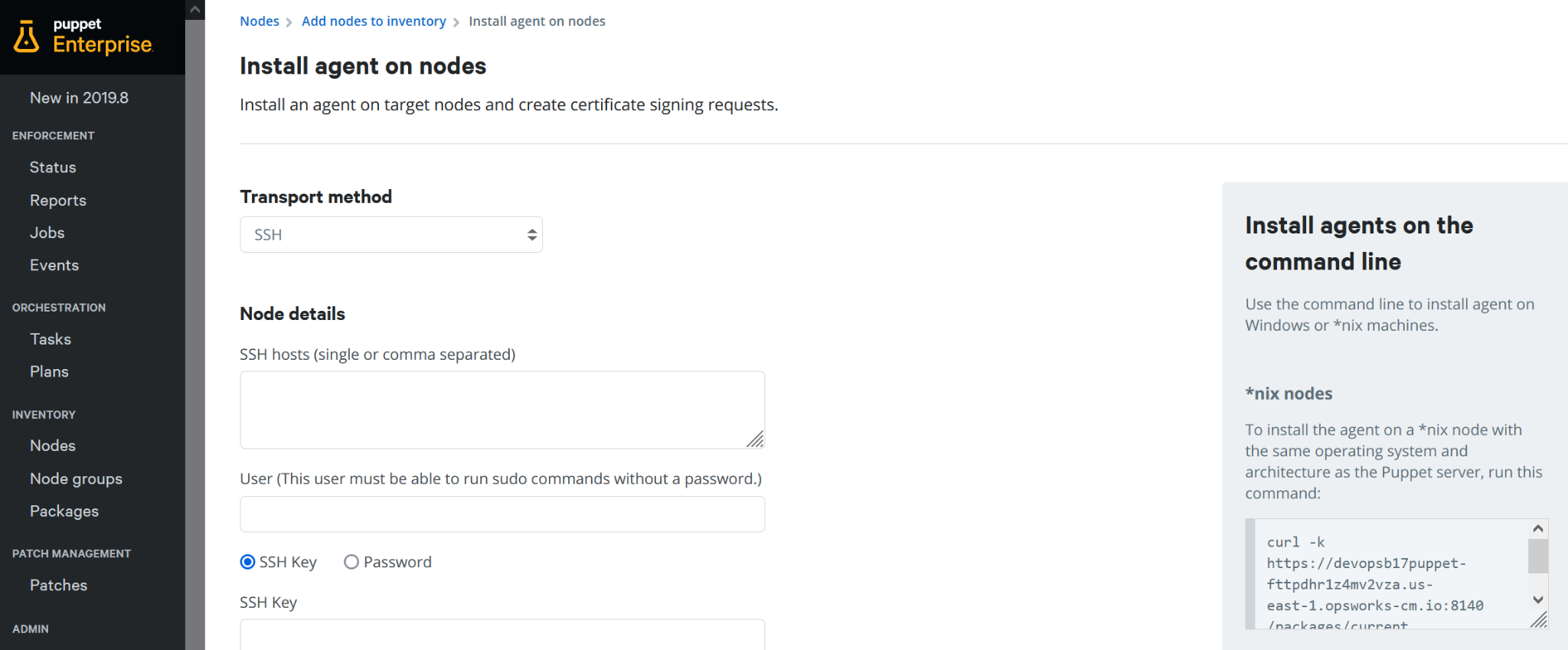
node default {

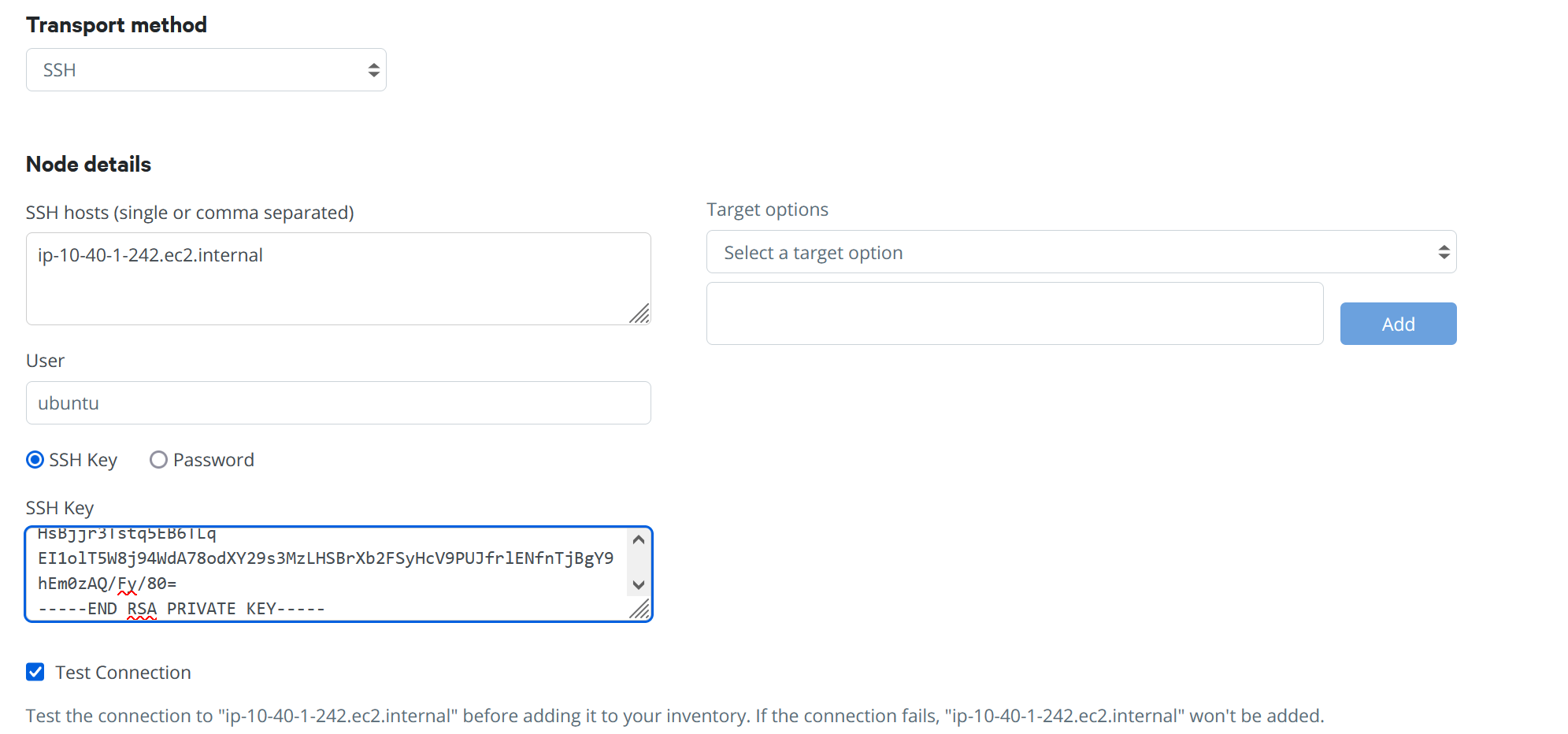
include role::main

}

Puppet Dashboard:

1. Deploy Opsworks puppet enterprise.
2. Deploy two ubuntu servers.
3. Install agent on the node by using the command on the right of the screenshot.



1. Once the agent is installed, add the node using the pem key as below.  
   
2. Checking the nodes

