Puppet Configuration

Puppet client

NodeP

Master puppet

Workstation

Create a EC2 instance called as workstationP .

Install puppet server in workstation

yum install puppet3-server

Create a EC2 instance called as NodeP .

Install puppet-client in NodeP

yum install puppet3

Get host name of master i.e. workstationP

hostname -f

i.e. ip-172-31-39-79.ap-south-1.compute.internal

NodeP

vi /etc/puppet/puppet.conf

[main]

# Where SSL certificates are kept.

# The default value is '$confdir/ssl'.

ssldir = $vardir/ssl

server = ip-172-31-39-79.ap-south-1.compute.internal

start the services in both master and client

/etc/init.d/puppetmaster start master

/etc/init.d/puppet start client

Take care to start the master first

Ping both master and client

ping ip-172-31-39-79.ap-south-1.compute.internal

ping ip-172-31-12-122.ap-south-1.compute.internal

Master i.e. workstationP

puppet cert list --all lists all the nodes and master details and certificate

If the nodes are certified then + sign is showed or else no sign

To certify the unsigned nodes

puppet cert sign " hostname of client " client which needs to be certified

puppet cert sign "ip-172-31-12-122.ap-south-1.compute.internal" of nodeP

client NodeP

puppet agent --test

Master workstationP

To get the process ID of puppet

ps -ef | grep puppet

puppet 2975 1 0 16:22 ? 00:00:00 /usr/bin/ruby2.0 /usr/bin/puppe master

root 23517 3083 0 16:48 pts/2 00:00:00 grep --color=auto puppet

Process ID of puppet is 23517

To get the port of puppet

netstat -tnlp | grep 23517

The files related to puppet folder are

* puppet.conf
* fileserver.conf
* auth.conf

To write the script move to manifests of puppet folder

cd /etc/puppet/manifests/

scripts can be validated at https://validate.puppet.com/

**script to install httpd**

**vi site.pp**

package { 'httpd':

ensure => present,

}

service { 'httpd' :

ensure => running,

enable => true,

require => Package["httpd"]

}

Client NodeP

puppet agent --test –noop dry run

execution of script

puppet agent --test

**script to install tomcat7**

**vi site.pp**

package { 'httpd':

ensure => present,

}

package { 'tomcat7':

ensure => installed,

}

service { 'httpd' :

ensure => running,

enable => true,

require => Package["httpd"]

}

package { 'tomcat7-webapps': ensure => installed, }

service { 'tomcat7' :

ensure => running,

enable => true,

require => Package["tomcat7"]

}

.

To see a different page for Apache

file {'/var/www/html/index.html':

content => "welcome to apache"

}

To create template

Navigate to module level and create directories

Working path for creating directories = /etc/puppet/modules

To create templates, need to create directories e.g. apache

mkdir apache

cd apache/

mkdir manifests

cd manifests/

vi init.pp

class apache {

package { 'httpd':

ensure => present,

}

package { 'tomcat7': ensure => installed, }

service { 'httpd' :

ensure => running,

enable => true,

require => Package["httpd"]

}

package { 'tomcat7-webapps':

ensure => installed, }

service { 'tomcat7’ :

ensure => running,

enable => true,

require => Package["tomcat7"]

}

file { '/var/www/html/index.html' :

content => template("apache/index.html.erb")

}

}

Template creation

Navigate to apache level i.e Working path for template /etc/puppet/modules/apache

mkdir templates => create templates folder

cd templates

write a html template

vi index.html.erb

HTML example

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

After creating template vi index.html.erb and init.pp navigate to manifests folder of puppet

Working path for creating directories = /etc/puppet/ manifests

mv site.pp site.pp.bak **=>** Take a back of site.pp file if any

vi site.pp => create a new site.pp file

node 'ip-172-31-12-122.ap-south-1.compute.internal' {

include apache }

ip-172-31-12-122.ap-south-1.compute.internal **=> client IP address**

apache **=>** class name

Client NodeP

execution of script

puppet agent --test

Resources and Classes

**Type 1 : Install LAMP with single Manifest**

Workstation

* cd /etc/puppet/manifests/
* <https://forge.puppet.com/>

puppetlabs/stdlib => Manual installation

copy below line and paste in putty

puppet module install puppetlabs-stdlib --version 6.0.0

* vi site.pp

package {'httpd':

ensure => installed,

}

service {'httpd':

ensure => running,

}

package {'mysql-server':

ensure => installed,

}

service {'mysqld':

ensure => running,

}

package {'php':

ensure => installed,

}

file {'/var/www/html/info.php':

ensure => file,

content => '<?php phpinfo(); ?>',

require => Package['httpd'],

}

file\_line { 'DirectoryIndex':

ensure => present,

path => '/etc/httpd/conf/httpd.conf',

line => 'DirectoryIndex info.php index.html index.html.var',

match => '^DirectoryIndex',

}

Client

puppet agent --test

**Error: /Stage[main]/Main/File\_line[DirectoryIndex]: Could not evaluate: No such file or directory - /etc/httpd/conf/httpd.conf**

puppet agent –test

service httpd restart

curl ifconfig.co => to get the public IP address of client

Public ID address on web page i.e. <http://13.127.123.255/>

cd /etc/puppet/manifests/

vi date.php

<?php  
echo "Today is " . date("Y/m/d") . "<br>";  
echo "Today is " . date("Y.m.d") . "<br>";  
echo "Today is " . date("Y-m-d") . "<br>";  
echo "Today is " . date("l");

echo "The time is " . date("h:i:sa");  
?>

package {'httpd':

ensure => installed,

}

service {'httpd':

ensure => running,

}

package {'mysql-server':

ensure => installed,

}

service {'mysqld':

ensure => running,

}

package {'php':

ensure => installed,

}

file {'/var/www/html/info.php':

ensure => file,

content => '<?php date.php; ?>',

require => Package['httpd'],

}

file\_line { 'DirectoryIndex':

ensure => present,

path => '/etc/httpd/conf/httpd.conf',

line => 'DirectoryIndex info.php index.html index.html.var',

match => '^DirectoryIndex',

}