Puppet Overview

June 2017

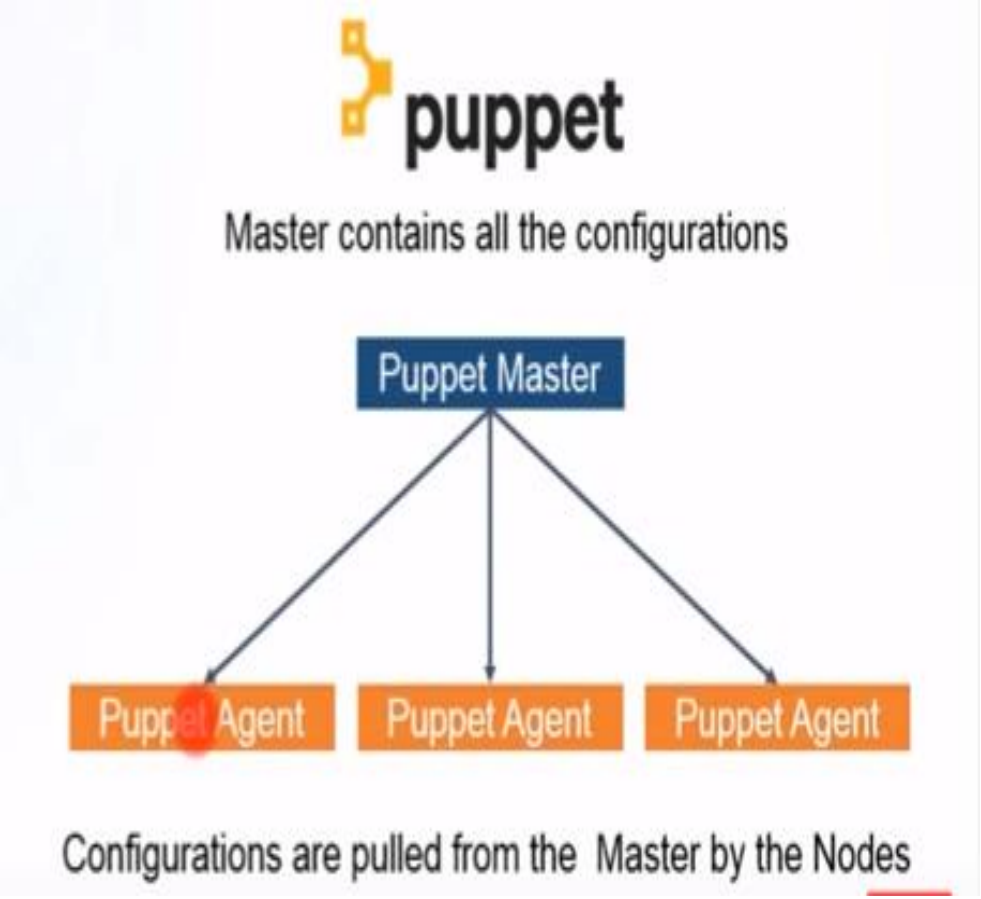
Unleash the Next

Introduction to Puppet

- Puppet is a configuration management tool developed by **Puppet Labs** in order to automate infrastructure management and configuration. Puppet is a very powerful tool which helps in the concept of Infrastructure as code. This tool is written in **Ruby DSL language** that helps in converting a complete infrastructure in code format, which can be easily managed and configured.

- Puppet follows Master-Slave model, where one machine in any cluster acts as client known as puppet master and the other acts as server known as slave on nodes. Puppet has the capability to manage any system from scratch, starting from initial configuration till end of-life of any particular machine.

- Alternatives to Puppet: Chef, CFEngine, Salt, Ansible



Puppet vs. Other Configuration Management Tools

➢ CFEngine

- CFEngine was introduced first in 1993. Its initial task was the automatic configuration of Unix workstations.
- CFEngine was written in C, making it less Ops friendly yet very friendly to developers.
- Like Puppet, CFEngine also uses an agent to enforce configuration.
- The master server is called a policy server and there could be a few policy servers.
- CFEngine also has an Enterprise version and its latest release was in December 2014. The Enterprise version can be downloaded and can be used for free for up to 25 agents.

Puppet vs. Other Configuration Management Tools

➢ Chef

- Chef is another Ops friendly tool written in Ruby. Like Puppet, Chef is declarative. Chef was first introduced in 2009 and has a paid version which offers more features such as support and high availability.

- Chef is similar to Puppet in its master/agent model. The terminology is different with Chef – having a server and a client.

- Unlike Puppet, Chef doesn't enable pushing policy. Instead, it needs to be configured to query the master periodically.

- Chef supports multiple platforms and operating systems, although Chef server needs to run at 64-bit Linux.

- Chef's latest release was in December 2014. And Chef is behind Puppet as far as tool usage goes.

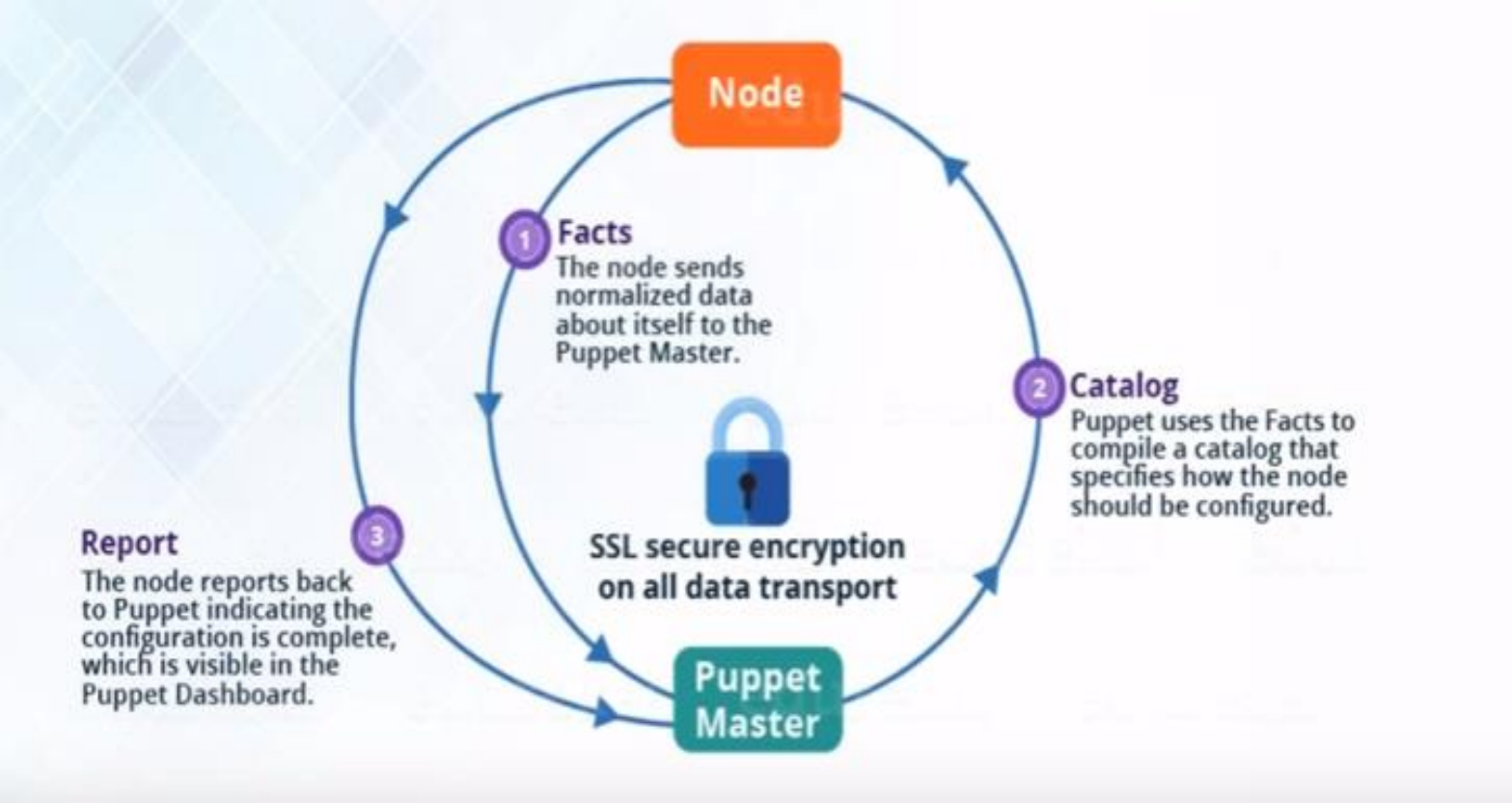
Puppet vs. Other Configuration Management Tools

➢ Ansible

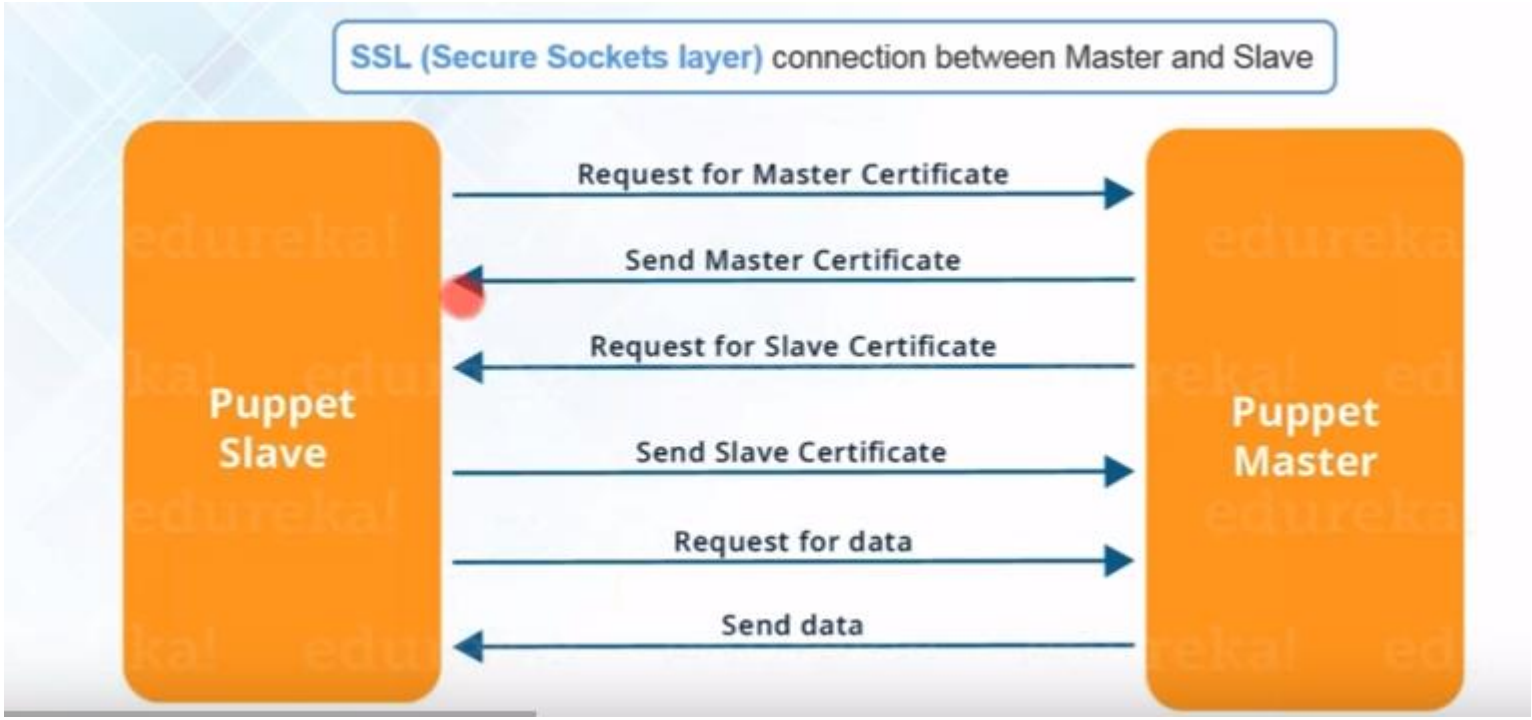
- Ansible was first introduced in 2012 and it's written in Python. It is considered to be very easy to set up and understand.
- Ansible is quite different from Puppet in its architecture as it's agentless. Also, there is no central server.
- Ansible uses preadded client SSH keys and establishes an SSH connection to populate configurations.
- Ansible is considered to be very fast for the reason that there is no master node communication.
- Ansible can also be run from the command line without the use of configuration files for simple tasks – such as reboot, yum update, and so on.

Puppet Workflow

Puppet uses the following workflow to apply configuration on the system.



Puppet Master-Slave Connection



Components of Puppet Architecture

- **Puppet master**

  Puppet master is a service runs on the main server which used to manage the entire clients to deploy, configure and maintains the infrastructures.

- **Puppet agent**

  Puppet Agents are the actual working machines which are managed by the Puppet master. They have the Puppet agent daemon service running inside them.

- **Facts**

  Facts are the details related to the node or the master machine, which are basically used for analyzing the current status of any node. On the basis of facts, changes are done on any target machine. There are pre-defined and custom facts in Puppet.

- **Facter**

  Facter is Puppet's system inventory tool. Facter reads facts about a node, such as its hostname, IP address, and operating system, and makes them available to Puppet.

Components of Puppet Architecture

- **Catalog**

A catalog is a document that describes the desired system state for one specific server. It lists all of the resources that need to be managed, as well as any dependencies between those resources.

- **Manifests**

Manifests are files with extension ".pp", where we declare all resources to be checked or to be changed. Resources may be files, packages, services and so on.

Default location : /etc/puppet/manifests

- **Resources**

Puppet resources are the key components for modeling any particular machine. These resources have their own implementation model. Puppet uses the same model to get any particular resource in the desired state.

Components of Puppet Architecture

- **Resources types**

  Resource Types are,
  a. A type (package, service, file, user, mount, exec)
  b. A title (how the resources types are called and referred)

Sample syntax:

```
type { 'title':
argument => value,
other_arg => value,
}
```

If you are looking for Resource Types reference, use the below command.
puppet describe file
For the full list of available descriptions:
puppet describe --list

Simple samples of resources

Verify the OpenSSH package

```
package { 'openssh':
  ensure => present,
}
```

Create a /etc/motd file

```
file { 'motd':
  path => '/etc/motd',
}
```

Start a httpd service

```
service { 'httpd':
  ensure => running,
  enable => true,
}
```

Components of Puppet Architecture

- **Classes**

Classes are containers or groups of different resources. These classes are defined inside Puppet manifest files which is located inside Puppet modules. The main purpose of using a class is to reduce the same code repetition inside any manifest file or any other Puppet code.

Example of a class definition:

```
class mysql (
  root_password => 'default_value',
  port          => '3306',
) {
  package { 'mysql-server':
    ensure => present,
  }
  service { 'mysql':
    ensure    => running,
```

Puppet installation

Steps to install Puppet

1. Install Puppet Master and Puppet Agent

2. Edit the host and Puppet configuration files in both Puppet Master and Puppet Agent.

3. Generate certificate signing request from Puppet Agent and it should be signed by the Puppet Master to establish secure connection between Master and Agent.

Puppet installation

Install Puppet Master

```
[root@PuppetMaster ~]# rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
Retrieving http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
warning: /var/tmp/rpm-tmp.eo15Ya: Header V4 RSA/SHA512 Signature, key ID 4bd6ec30: NOKEY
Preparing...                ########################################### [100%]
   1:puppetlabs-release     ########################################### [100%]
[root@PuppetMaster ~]#
```

```
[root@PuppetMaster ~]# yum install puppet-server
Loaded plugins: fastestmirror, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: mirrors.aluhost.com
 * extras: mirrors.nhanhoa.com
 * updates: mirrors.viethosting.vn
puppetlabs-deps                                                | 2.5 kB     00:00
puppetlabs-deps/primary_db                                     |  28 kB     00:00
puppetlabs-products                                            | 2.5 kB     00:00
puppetlabs-products/primary_db                                 | 170 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package puppet-server.noarch 0:3.8.7-1.el6 will be installed
--> Processing Dependency: puppet = 3.8.7-1.el6 for package: puppet-server-3.8.7-1.el6.noar
```

```
Installed:
  puppet-server.noarch 0:3.8.7-1.el6

Dependency Installed:
  augeas-libs.i686 0:1.0.0-10.el6     compat-readline5.i686 0:5.2-17.1.el6  facter.i386 1:2.4.6-1.el6
  hiera.noarch 0:1.3.4-1.el6          libselinux-ruby.i686 0:2.0.94-7.el6   pciutils.i686 0:3.1.10-4.el6
  puppet.noarch 0:3.8.7-1.el6         ruby.i686 0:1.8.7.374-4.el6_6         ruby-augeas.i386 0:0.4.1-3.el6
  ruby-irb.i686 0:1.8.7.374-4.el6_6   ruby-libs.i686 0:1.8.7.374-4.el6_6    ruby-rdoc.i686 0:1.8.7.374-4.el6_6
  ruby-shadow.i386 1:2.2.0-2.el6      rubygem-json.i386 0:1.5.5-3.el6       rubygems.noarch 0:1.3.7-5.el6
  virt-what.i686 0:1.11-1.2.el6
```

Puppet installation

Install Puppet Agent

```
[root@PuppetAgent ~]# rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
Retrieving http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
warning: /var/tmp/rpm-tmp.3GPZeP: Header V4 RSA/SHA512 Signature, key ID 4bd6ec30: NOKEY
Preparing...                ########################################### [100%]
   1:puppetlabs-release     ########################################### [100%]
[root@PuppetAgent ~]# cl
```

```
[root@PuppetAgent ~]# yum install puppet
Loaded plugins: fastestmirror, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: centos-hcm.viettelidc.com.vn
 * extras: mirrors.nhanhoa.com
 * updates: centos.webwerks.com
puppetlabs-deps                                                | 2.5 kB     00:00
puppetlabs-deps/primary_db                                     |  28 kB     00:01
puppetlabs-products                                            | 2.5 kB     00:00
puppetlabs-products/primary_db                                 | 170 kB     00:22
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package puppet.noarch 0:3.8.7-1.el6 will be installed
```

```
Installed:
  puppet.noarch 0:3.8.7-1.el6

Dependency Installed:
  augeas-libs.i686 0:1.0.0-10.el6      compat-readline5.i686 0:5.2-17.1.el6  facter.i386 1:2.4.6-1.el6
  hiera.noarch 0:1.3.4-1.el6           libselinux-ruby.i686 0:2.0.94-7.el6   pciutils.i686 0:3.1.10-4.el6
  ruby.i686 0:1.8.7.374-4.el6_6        ruby-augeas.i386 0:0.4.1-3.el6        ruby-irb.i686 0:1.8.7.374-4.el6_6
  ruby-libs.i686 0:1.8.7.374-4.el6_6   ruby-rdoc.i686 0:1.8.7.374-4.el6_6    ruby-shadow.i386 1:2.2.0-2.el6
  rubygem-json.i386 0:1.5.5-3.el6      rubygems.noarch 0:1.3.7-5.el6         virt-what.i686 0:1.11-1.2.el6
```

Puppet installation

Puppet Configuration File

The default puppet configuration file is located in **/etc/puppet/puppet.conf.** All Puppet related settings such as the definition of Puppet master, Puppet agent, Puppet apply and certificates are defined in this file.

Puppet uses four config sections:

• main - is the global section used by all commands and services. It can be overridden by the other sections.

• master - is used by the Puppet master service and the Puppet cert command.

• agent - is used by the Puppet agent service.

• user - is used by the Puppet apply command, as well as many of the less common Puppet subcommands.

Puppet installation

Edit the host and Puppet configuration files on both Puppet Master and Puppet Agent

1. First edit the hosts file on the Puppet Master and add the IP address and a domain name in /etc/hosts .

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.182 puppet puppet.edureka.co
~
```

2. Edit the puppet configuration file on Puppet Master. The default puppet configuration file is located in **/etc/puppet/puppet.conf.**

```
[main]
    # The Puppet log directory.
    # The default value is '$vardir/log'.
    logdir = /var/log/puppet

    # Where Puppet PID files are kept.
    # The default value is '$vardir/run'.
    rundir = /var/run/puppet

    # Where SSL certificates are kept.
    # The default value is '$confdir/ssl'.
    ssldir = $vardir/ssl
    dns_alt_names = puppet,puppet.edureka.co
    certname=puppet
```

Puppet installation

3. Start the Puppet Master service

```
[root@PuppetMaster ~]# service puppetmaster start
Starting puppetmaster:                                     [  OK  ]
[root@PuppetMaster ~]#
```

Now, Proceed with the same steps on Puppet Agent .

1. First edit the hosts file on the Puppet Agent and add the IP addresses of both Agent and the Puppet Master .

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.119 puppetagent
192.168.1.182 puppet puppet.edureka.co
```

2. Edit the puppet configuration file on Puppet Agent.

```
[agent]
    server = puppet.edureka.co
    # The file in which puppetd stores a list of the classes
    # associated with the retrieved configuratiion.  Can be loaded in
    # the separate ``puppet`` executable using the ``--loadclasses``
    # option.
    # The default value is '$confdir/classes.txt'.
    classfile = $vardir/classes.txt
```

3. Start the Puppet Agent service.

```
[root@PuppetAgent ~]# service puppet start
Starting puppet agent:                                     [  OK  ]
[root@PuppetAgent ~]#
```

Puppet installation

Generate certificate signing request

1. Stop the Puppet Master Service

```
[root@PuppetMaster ~]# service puppetmaster stop
Stopping puppetmaster:                                     [  OK  ]
[root@PuppetMaster ~]#
```

2. Now run the below command and it creates the CA certificate and the Puppet Master Certificate with the appropriate DNS names included .

```
[root@PuppetMaster ~]# sudo -u puppet puppet master --no-daemonize --verbose
Info: Creating a new SSL key for ca
Info: Creating a new SSL certificate request for ca
Info: Certificate Request fingerprint (SHA256): CA:20:F0:B7:D6:94:FA:64:1A:51:2B:65:E1:4D:30:D3:B9:56:B7:D2:96:D
3:65:3D:54:85:66:44:2D:C2:7E:BF
Notice: Signed certificate request for ca
Info: Creating a new certificate revocation list
Info: Creating a new SSL key for puppetmaster
Info: csr_attributes file loading from /var/lib/puppet/.puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for puppetmaster
Info: Certificate Request fingerprint (SHA256): 5D:47:71:9E:1B:C8:6A:B8:81:CD:6A:9C:37:BC:EB:92:FA:EC:0A:CB:47:C
0:2D:DC:37:2B:9B:B9:4C:EB:AB:78
Notice: puppetmaster has a waiting certificate request
Notice: Signed certificate request for puppetmaster
```

3. Now start the Puppet Master using the below command .

```
[root@PuppetMaster ~]# puppet resource service puppetmaster ensure=running
Notice: /Service[puppetmaster]/ensure: ensure changed 'stopped' to 'running'
service { 'puppetmaster':
  ensure => 'running',
}
[root@PuppetMaster ~]# service puppetmaster status
puppet (pid  14406) is running...
[root@PuppetMaster ~]#
```

Puppet installation

Generate certificate signing request

4. Stop the Puppet Agent Service

```
[root@PuppetAgent ~]# service puppet stop
Stopping puppet agent:                                     [  OK  ]
[root@PuppetAgent ~]#
```

5. Now run the below command , which would generate a certificate signing request to the Puppet master .

```
[root@PuppetAgent ~]# puppet agent -t
Exiting; no certificate found and waitforcert is disabled
[root@PuppetAgent ~]#
```

6. On the Puppet Master , you can see the certificates that are available , by running he below command .

```
[root@PuppetMaster ~]# puppet cert list
  "puppetagent" (SHA256) A1:B4:50:84:D3:FE:84:70:1C:72:20:7D:A6:AF:15:42:4A:83:04:09:BE:E
B9:3B:62
[root@PuppetMaster ~]#
```

7. Now we can sign the certificate as below :

```
[root@PuppetMaster ~]# puppet cert list
  "puppetagent" (SHA256) A1:B4:50:84:D3:FE:84:70:1C:72:20:7D:A6:AF:15:42:4A:83:04:09:BE:B7:45:D2:D1:01:22:C7:82:
B9:3B:62
[root@PuppetMaster ~]# puppet cert sign puppetagent
Notice: Signed certificate request for puppetagent
Notice: Removing file Puppet::SSL::CertificateRequest puppetagent at '/var/lib/puppet/ssl/ca/requests/puppetagen
t.pem'
[root@PuppetMaster ~]#
```

Puppet installation

Generate certificate signing request

8. Start the Puppet Agent now and confirm the status .

```
[root@PuppetAgent ~]# puppet resource service puppet ensure=running
Notice: /Service[puppet]/ensure: ensure changed 'stopped' to 'running'
service { 'puppet':
  ensure => 'running',
}
[root@PuppetAgent ~]# service puppet status
puppet (pid  2483) is running...
[root@PuppetAgent ~]#
```

9. To check if the correct certificate was signed by the Puppet Master , use the “fingerprint” command .

```
[root@PuppetAgent ~]# puppet agent --fingerprint
(SHA256) 5B:3B:1D:E3:7B:1A:D1:1B:5C:51:AE:72:0A:76:81:BE:F6:2B:25:7B:F8:A8:04:6B:DF:FA:F5:1D:B0:7D:F5:42
[root@PuppetAgent ~]#
```

10. Now the changes made at the Puppet Master should reflect on the Puppet Agent . This can be done manually by running the below command .

```
[root@PuppetAgent ~]# puppet agent -t
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for puppetagent
Info: Applying configuration version '1474300294'
Notice: Finished catalog run in 0.02 seconds
[root@PuppetAgent ~]#
```

Example Scenario

Lets take an example scenario that, we have 1000+ servers which are connected with our puppet master server. We have requirement to do some operation changes to manage your all 1000+ servers. In all servers, we have already installed puppet agents and connected with puppet master.

Scenario Requirements:
1. Modifying the file /etc/motd in all infrastructure servers to add a content using puppet automation.
2. Stopping Postfix service in all servers.

For file /etc/motd :
1. We will ensure whether the file present or not
2. If present, then we will do the required changes.
3. If need, We will change file permissions.

For service postfix.
1. We will ensure whether the service exists or not.
2. If exist, ensure its running or not
3. If running, stop it.

Example Scenario

Lets do these by writing the puppet manifest file under /etc/puppet/manifests/<name.pp> .

```
node default {

file {'/etc/motd':
content => 'My Testing content',
}

service {'postfix':
ensure => 'stopped',
enable => 'false',
}
```

Note : In the first line, we have defined to apply the changes for all default nodes which are connected to puppet master server. If you want to do these changes for one server, change as "node <node name>". So changes will apply only to the particular puppet agent node.

By default, Puppet agents will give request to puppet master to check any changes for every 30mins interval or manually run the below command on one server to test.

```
[root@puppet-client1 ~]# puppet agent -t
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for puppet-client1.learnitguide.net
Info: Applying configuration version '1491886477'
Notice: /Stage[main]/Main/Node[default]/File[/etc/motd]/content:
--- /etc/motd   2013-06-07 20:01:32.000000000 +0530
+++ /tmp/puppet-file20170411-2437-19shhrf       2017-04-11 10:24:40.107000000 +0530
@@ -0,0 +1 @@
+My Testing content
\ No newline at end of file

Info: Computing checksum on file /etc/motd
Info: /Stage[main]/Main/Node[default]/File[/etc/motd]: Filebucketed /etc/motd to puppet
with sum d41d8cd98f00b204e9800998ecf8427e
Notice: /Stage[main]/Main/Node[default]/File[/etc/motd]/content: content changed '{md5}d
41d8cd98f00b204e9800998ecf8427e' to '{md5}9cff20418cf6864b0c647bb677073966'
Notice: /Stage[main]/Main/Node[default]/Service[postfix]/ensure: ensure changed 'running
' to 'stopped'
Notice: Finished catalog run in 0.16 seconds
```

Example Scenario

Now we cab verify the same from the agent node if the changes reflect.

```
[root@puppet-client1 ~]# cat /etc/motd
My Testing content[root@puppet-client1 ~]#
[root@puppet-client1 ~]#
```

Market Acceptance of Puppet

- Due to its easy-to-learn nature, Puppet was widely adopted across different organizations.
- Among such organizations are Verizon, Google, Apple, Red Hat, Nike, HBO, Hershey's, Motorola, PayPal, and many more.
- Due to its large open source community and availability of prewritten modules, many organizations have adopted Puppet, not just as an automated configuration management tool rather it has been adopted and implemented at the core of an overall DevOps strategy.
- For example, Puppet is used by many organizations with third-party tools – such as Jenkins, to run automation tests. Also, Puppet is used in OpenStack projects for automation.
- At Red Hat, Puppet was integrated into a few applications such as the Foreman, Satellite, and OpenStack.
- Red Hat also uses Puppet as the automation layer in many of its cloud environments.

Market Acceptance of Puppet

- There are many other IT tasks managed by Puppet. Among them are Netapp configurations, network administration, and other areas where Puppet is not traditionally used.
- At Spiceworks, Puppet is deeply integrated to maintain infrastructure and new hardware deployment.
- Los Alamos uses Puppet to gain visibility into their deployed Macintosh environment. Los Alamos has one of the largest Puppet managed Mac environments in the world.
- Sun Oracle uses Puppet to ensure consistent web server architecture across three US data centers.
- On a bit of a side note, Puppet has an Oracle module containing all of the custom types to manage the setup of an Oracle database.

Popularity – Google Trend

- ✓ Puppet and Chef are old players, puppet has wider adoption
- ✓ SaltStack and Ansible are new players, and Ansible looks very promising with the growing trend

References

- Puppet Documentation - Main and Official reference

  https://docs.puppet.com/

- How puppet works ?

  http://www.learnitguide.net/2016/09/what-is-puppet-how-puppet-works.html

- Puppet Overview

  https://www.tutorialspoint.com/puppet/puppet_overview.htm

THANK YOU

Presentation by Rajkumar Loganathan

E-mail: Rajkumar.Loganathan@hpe.com

About Mphasis

Mphasis (an HP Company) enables chosen customers to meet the demands of an evolving market place. Mphasis fuels this by combining superior human capital with cutting edge solutions in hyper-specialized areas. Contact with us on www.mphasis.com