



puppet

DISCOVER
BY MATEEESH
TECHNOLOGIES
mateeesh@gmail.com

Node definition

Syntax:

```
node '<Host Name>' {  
    include <class Name>  
}  
}
```

Ex:

```
node 'mynode.satish.com' {  
    include httpd  
    include apache  
}  
}
```

```
class myapache
{
  package {'apache2':
    ensure => installed,
  }
}

class myhttpd
{
  package {'httpd':
    ensure => installed,
  }
}

node 'pup-node1.c.sathya-171302.internal'{
  include myapache
}

node 'pup-node2.c.sathya-171302.internal'{
  include myhttpd
}
```

Template

Evaluating Templates

```
$value = template ("testtemplate.erb")
```

we can also specify the full path of a template.

Ex:

```
file { '/etc/apache2/dav_svn.conf':
  ensure => "present",
  content =>
  template('/etc/puppet/templates/test.erb')
}
```

Template

```
#vi /etc/puppet/templates/test.erb
<Location "/cgi-bin/<%= name %>.cgi/login">
    AuthType Basic
    AuthName "Test"
    AuthUserFile /etc/tomcat/auth/svn
    Require valid-user
</Location>
```

User defined Modules

Example-Module1:

```
#cd /etc/puppet/modules  
#mkdir -p nginx/manifests/  
#vi nginx/manifests/init.pp  
  
class nginx {  
    exec { 'yum update':  
        command => '/bin/yum update'  
    }  
    package { 'nginx':  
        ensure => installed,  
    }  
    service { 'nginx':  
        ensure => running,  
        require => package['nginx'],  
    }  
}
```

User defined Modules

Example-Module2:

```
#cd /etc/puppet/modules  
#mkdir user/manifests  
#vi user/manifests/init.pp  
  
class user {  
    user { 'testuser':  
        ensure => present,  
        comment => ' user',  
        home => '/home/testuser',  
        managehome => true  
    }  
}  
}
```

To Run the Modules :

```
#vi /etc/puppet/manifests/site.pp
```

```
node 'pupnode1' {
```

```
    include nginx
```

```
}
```

```
node 'pupnode2' {
```

```
    include user
```

```
}
```

```
Node1# puppet agent -t
```

```
Node2# puppet agent -t
```

Copy File to Nodes

```
# cd /etc/puppet  
# mkdir -p modules/mymodule/manifests  
# mkdir -p modules/mymodule/files  
# vi modules/mymodule/manifests/init.pp  
class myfile {  
    file { "filecopy":  
        path => "/var/www/html/index.html",  
        source =>  
        'puppet:///modules/mymodule/index.html',  
    }  
}
```

Copy File to Nodes

```
# vi modules/mymodule/files/index.html
```

```
<h1> Hello from Puppet Master </h1>
```

```
# vi /etc/puppet/manifests/site.pp
```

```
import 'mymodule'
```

```
include myfile
```

Node1# puppet agent -t

node2# puppet agent -t

```
*  
├── auth.conf  
├── etckeeper-commit-post  
├── etckeeper-commit-pre  
├── files  
├── fileserver.conf  
├── manifests  
│   └── site.pp  
└── modules  
    ├── exec  
    │   └── manifests  
    │       └── init.pp  
    └── user  
        └── manifests  
            └── init.pp  
├── puppet.conf  
└── templates
```



```
class webserver
{
  if $operatingsystem == 'Ubuntu'{
    package {'apache2':
      ensure => installed,
    }
    service {'apache2':
      ensure => running,
      require => Package['apache2'],
    }
  }
}
```

```
elsif $operatingsystem == 'CentOS'{  
    package {'httpd':  
        ensure => installed,  
    }  
    service {'httpd':  
        ensure => running,  
        require => Package['httpd'],  
    }  
}  
file{'copyfile':  
    path => '/var/www/html/index.html',  
    source =>  
        'puppet:///modules/copymodule/index.html',  
}  
}
```

```
class appserver
{
  if $operatingsystem == 'Ubuntu'
    package {'openjdk-7-jdk':
      ensure => installed,
    }
    package {'tomcat7':
      ensure => installed,
    }
    file{'copyfile':
      path => '/var/lib/tomcat7/webapps/sample.war'
      source => 'puppet:///modules/copymodule/sample.war',
    }
    service {'tomcat7':
      ensure => running,
      require => Package['tomcat7'],
    } }
}
```

```
elsif $operatingsystem == 'CentOS'{  
    package {'java':  
        ensure => installed,  
    }  
    package {'tomcat':  
        ensure => installed,  
    }  
    file{'copyfile':  
        path => '/var/lib/tomcat/webapps/sample.war',  
        source => 'puppet:///modules/copymodule/sample.war',  
    }  
    service {'tomcat':  
        ensure => running,  
        require => Package['tomcat'],  
    } }  
}
```

Puppet Forge modules

- The modules in the Puppet Forge repository is publically-available modules, and they can be useful when we develop our own infrastructure. The Puppet Forge modules can be quickly installed with built-in **puppet module** command.

The Pre-defined modules can get from

<https://forge.puppet.com/modules>

Ex:

```
#puppet module install puppetlabs-apache
```

Master# puppet module install puppetlabs-mysql --version 3.10.0

Master# puppet module install mayflower-php --version 4.0.0-beta1

#cd /etc/puppet/modules

#vi /etc/puppet/manifest/site.pp

include '::mysql::server'

include '::php'

Agent# puppet agent -t

Agent# mysql

Agent# php -v

Module Generate

```
#cd /etc/puppet/modules  
#puppet module generate sathya-httpd  
#cd /etc/puppet/modules/sathya-httpd  
/manifests/install.pp  
  
class httpd::install inherits httpd {  
  package { 'httpd':  
    ensure => installed,  
  }  
}  
}
```

```
#cd /etc/puppet/modules/sathya-httpd  
/manifests/start.pp  
  
class httpd::start inherits httpd{  
    service { 'httpd':  
        ensure     => running,  
        enable     => true,  
        hasstatus  => true,  
        hasrestart => true,  
        require   => Package['httpd'],  
    }  
}
```

```
#cd /etc/puppet/modules/sathya-httpd  
/manifests/init.pp  
  
class httpd {  
  include httpd::install  
  include httpd::start  
}  
  
#cd /etc/puppet/manifests/site.pp  
node default {  
  include httpd  
}  
  
node# puppet agent -t
```

