TP part 3 :

HIERA :

Exercice 1 :

Pour que les serveurs de type debian/ubuntu utilise le module sudo, mais pas les autres types de serveur (Tous ces serveurs quels qu'ils soient doivent par contre utiliser le module ssh) il faut juste créer un fichier yaml dans osfamily, Debian.yaml où on met le module sudo associé à la classe et dans le common.yaml on met le module ssh :

Pour le fichier Debian.yaml :

```
classes :
  - sudo
~
```

Pour le fichier common.yaml :

```
classes :
  - ssh
~
```

Pour que ces fichiers soit pris en compte on doit bien spécifier la hiérarchie dans le fichier /etc/puppet/hiera.yaml

```
---
:backends:
  - yaml

:hierarchy:
  - fqdn/%{fqdn}
  - virtual/%{virtual}
  - osfamily/%{::osfamily}
  - common

:yaml:
  :datadir: /etc/puppet/code/hiera
~
~
```

Les variables sont exécutées dans l'ordre de cette hiérarchie. On voit bien que la variable osfamily est avant la common, la variable common est très généralement en dernier.

Exercice 2 :

Nous devons déclarer une variable de classe, on va la déclarer dans le dossier hiera fqdn pour cibler notre client.

On créer dans /etc/puppet/code/hier/fqdn le fichier rogue1.iutbeziers.fr.yaml

Vu qu'on reste dans le même module on lui attribue toujours le même module ssh et on définit la variable permitroot (attention les majuscules sont interdites pour ces variables) :

```
classes :
 - ssh

ssh::permitroot : "yes"
~
```

Par défaut le fichier sshd_config poussé sur les clients n'autorise pas les connexions en root :

```
PermitRootLogin no
<environments/production/modules/ssh/templates/sshd_config.erb" 122L, 3309C
```

On va donc changer la ligne PermitRootLogin de ce fichier :

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PermitRootLogin <%=@permitroot%>
~
```

Et on doit aussi déclarer la variable dans notre module :

```
class ssh ($permitroot="no") {
    package { 'openssh-server':
       ensure => installed,
    }
    case $::osfamily {
        'Debian': { $sshd_service = 'ssh' }
        'RedHat': { $sshd_service = 'sshd' }
         default: {fail("Invalid osfamily: ${::osfamily}")}
    }

    file { '/etc/ssh/sshd_config':
        content => template("ssh/sshd_config.erb"),
#        source => 'puppet:///modules/ssh/sshd_config',
        owner => 'root',
        group => 'root',
        mode => '640',
        notify => Service['sshd'], # sshd will restart whenever you edit this file.
        require => Package['openssh-server'],
    }
    service { 'sshd':
        name => $sshd_service,
        ensure => running,
        enable => true,
        hasstatus => true,
        hasrestart => true,
    }


}
```

La variable se déclare juste après la classe, ici on la met à « no » pour que par défaut le permitrootlogin soit sur no et uniquement quand le client avec le fqdn rogue1.iutbeziers.fr demande la config le permitrootlogin soit à yes, c'est ce qui était demandé dans l'exercice.

Test sur le client (il est par défaut à no) :

```
root@rogue1:/home/vincent# puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for rogue1.iutbeziers.fr
Info: Applying configuration version '1615309781'
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/content:
--- /etc/ssh/sshd_config        2021-03-09 11:09:21.578872678 -0600
+++ /tmp/puppet-file20210309-3052-1fc145j       2021-03-09 11:09:43.141648690 -0600
@@ -119,4 +119,4 @@
 #       AllowTcpForwarding no
 #       PermitTTY no
 #       ForceCommand cvs server
-PermitRootLogin no
+PermitRootLogin yes

Info: Computing checksum on file /etc/ssh/sshd_config
Info: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]: Filebucketed /etc/ssh/sshd_config to puppet with sum
f3ce97be0526ef53be3eae04ba68afe3
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/content: content changed '{md5}f3ce97be0526ef53be3ea
e04ba68afe3' to '{md5}24497a997fce9bc54decbbf80d32b9e4'
Info: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]: Scheduling refresh of Service[sshd]
Notice: /Stage[main]/Ssh/Service[sshd]: Triggered 'refresh' from 1 event
Notice: Applied catalog in 0.31 seconds
root@rogue1:/home/vincent#
```