PUPPET BASE TORTEL Vincent Asur

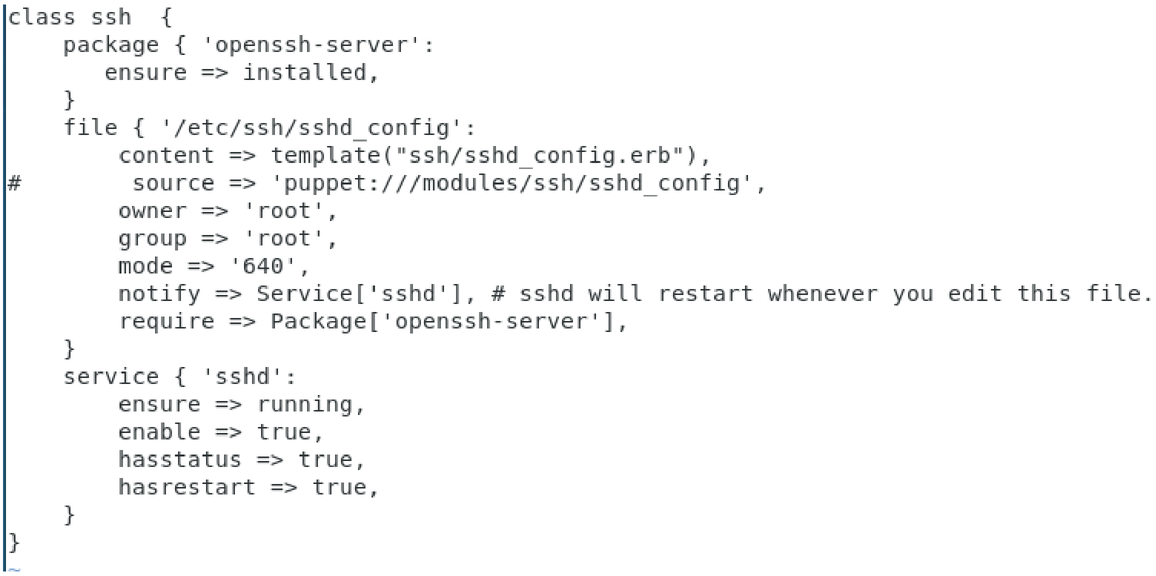
TP part 2 :

Module ssh – type files

Exercice 1 :

Pour le module ssh, création d’un dossier ssh à la racine de /etc/puppet/code/environments/production/modules dans ce dossier ssh création de trois dossiers templates, files, manifests .

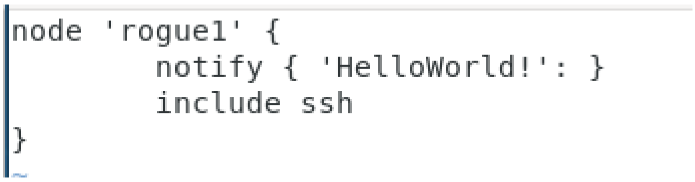
Dans le dossier manifests on met le init.pp :



```
class ssh  {
    package { 'openssh-server':
       ensure => installed,
    }
    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':
        ensure => running,
        enable => true,
        hasstatus => true,
        hasrestart => true,
    }
}
~
```

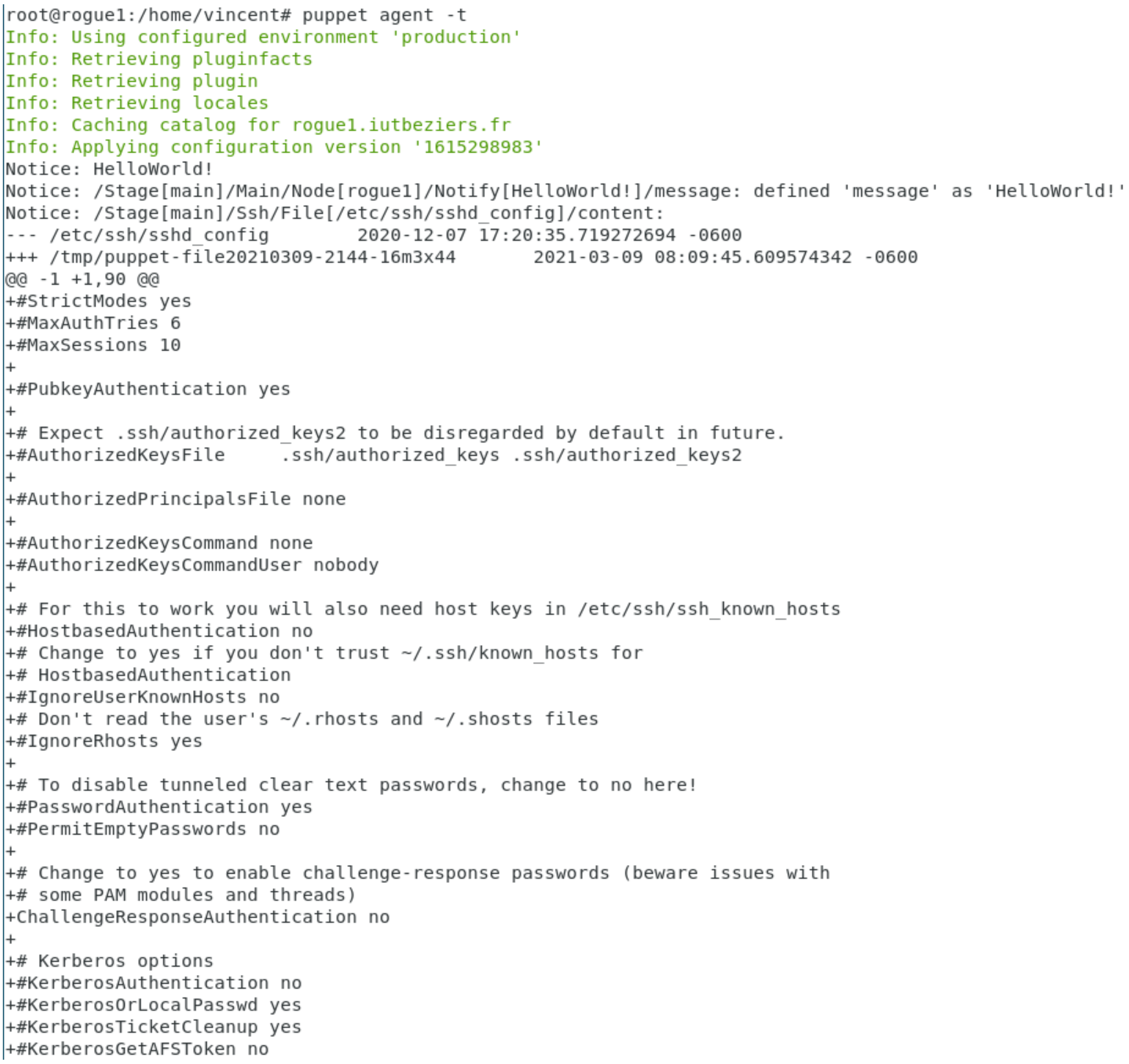
Dans le files on met la config de notre daemon sshd.

On modifie le site.pp

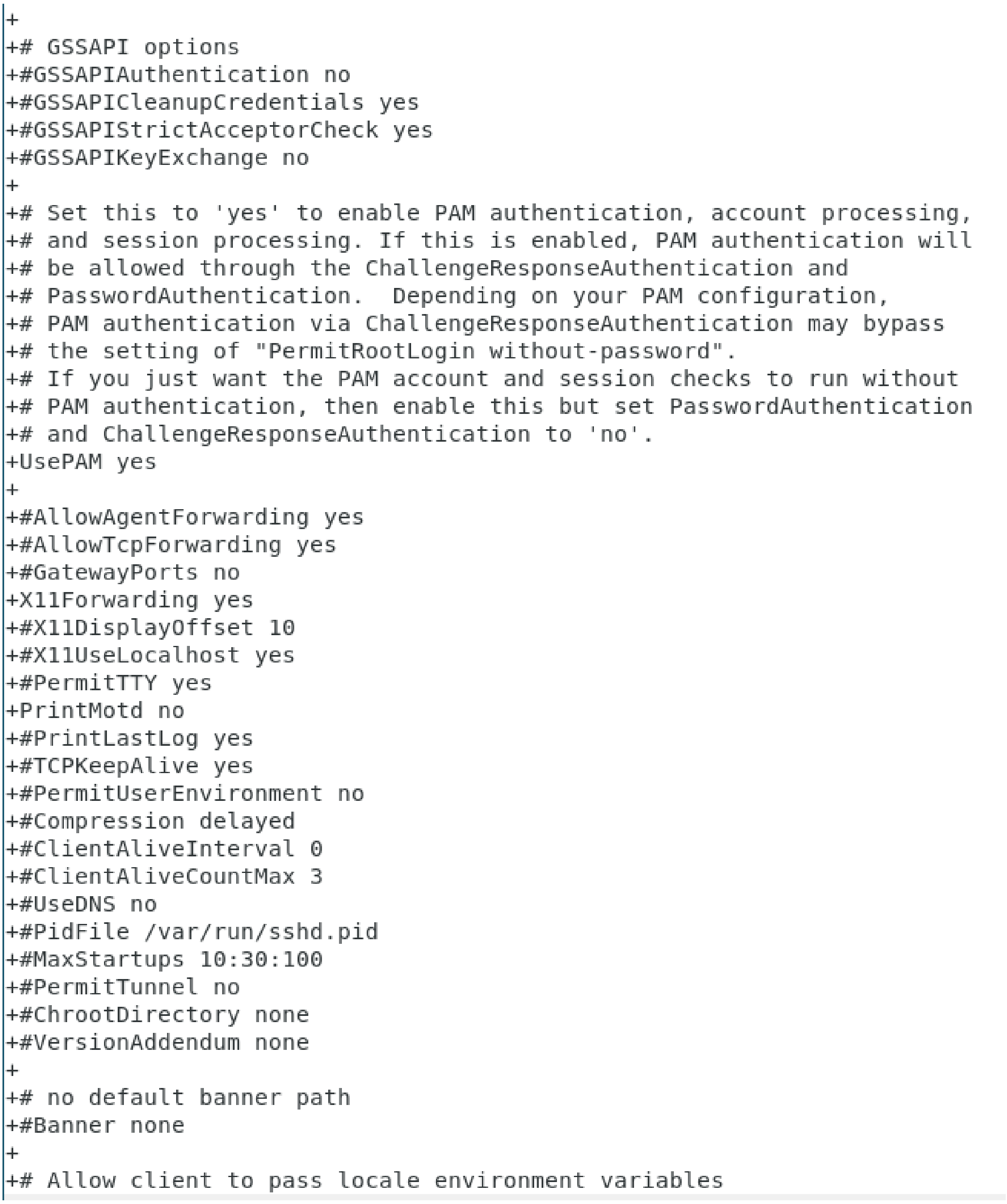


```
node 'rogue1' {
          notify { 'HelloWorld!': }
          include ssh
}
~
```

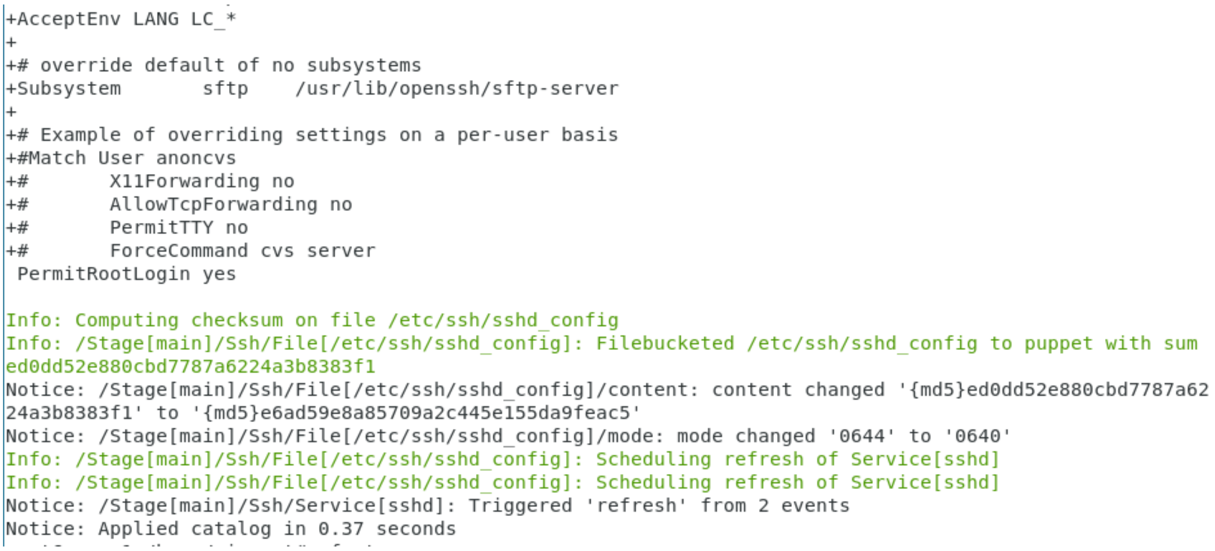
Plus qu'à tester sur le client :



```
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 '1615298983'
Notice: HelloWorld!
Notice: /Stage[main]/Main/Node[rogue1]/Notify[HelloWorld!]/message: defined 'message' as 'HelloWorld!'
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/content:
--- /etc/ssh/sshd_config        2020-12-07 17:20:35.719272694 -0600
+++ /tmp/puppet-file20210309-2144-16m3x44       2021-03-09 08:09:45.609574342 -0600
@@ -1 +1,90 @@
+#StrictModes yes
+#MaxAuthTries 6
+#MaxSessions 10
+
+#PubkeyAuthentication yes
+
+# Expect .ssh/authorized_keys2 to be disregarded by default in future.
+#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2
+
+#AuthorizedPrincipalsFile none
+
+#AuthorizedKeysCommand none
+#AuthorizedKeysCommandUser nobody
+
+# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
+#HostbasedAuthentication no
+# Change to yes if you don't trust ~/.ssh/known_hosts for
+# HostbasedAuthentication
+#IgnoreUserKnownHosts no
+# Don't read the user's ~/.rhosts and ~/.shosts files
+#IgnoreRhosts yes
+
+# To disable tunneled clear text passwords, change to no here!
+#PasswordAuthentication yes
+#PermitEmptyPasswords no
+
+# Change to yes to enable challenge-response passwords (beware issues with
+# some PAM modules and threads)
+ChallengeResponseAuthentication no
+
+# Kerberos options
+#KerberosAuthentication no
+#KerberosOrLocalPasswd yes
+#KerberosTicketCleanup yes
+#KerberosGetAFSToken no
```



```
+
+# GSSAPI options
+#GSSAPIAuthentication no
+#GSSAPICleanupCredentials yes
+#GSSAPIStrictAcceptorCheck yes
+#GSSAPIKeyExchange no
+
+# Set this to 'yes' to enable PAM authentication, account processing,
+# and session processing. If this is enabled, PAM authentication will
+# be allowed through the ChallengeResponseAuthentication and
+# PasswordAuthentication.  Depending on your PAM configuration,
+# PAM authentication via ChallengeResponseAuthentication may bypass
+# the setting of "PermitRootLogin without-password".
+# If you just want the PAM account and session checks to run without
+# PAM authentication, then enable this but set PasswordAuthentication
+# and ChallengeResponseAuthentication to 'no'.
+UsePAM yes
+
+#AllowAgentForwarding yes
+#AllowTcpForwarding yes
+#GatewayPorts no
+X11Forwarding yes
+#X11DisplayOffset 10
+#X11UseLocalhost yes
+#PermitTTY yes
+PrintMotd no
+#PrintLastLog yes
+#TCPKeepAlive yes
+#PermitUserEnvironment no
+#Compression delayed
+#ClientAliveInterval 0
+#ClientAliveCountMax 3
+#UseDNS no
+#PidFile /var/run/sshd.pid
+#MaxStartups 10:30:100
+#PermitTunnel no
+#ChrootDirectory none
+#VersionAddendum none
+
+# no default banner path
+#Banner none
+
+# Allow client to pass locale environment variables
```



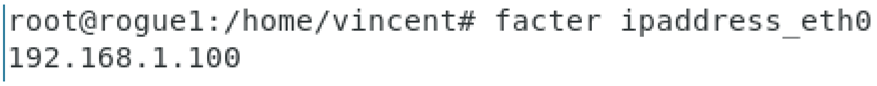
```
+AcceptEnv LANG LC_*
+
+# override default of no subsystems
+Subsystem       sftp    /usr/lib/openssh/sftp-server
+
+# Example of overriding settings on a per-user basis
+#Match User anoncvs
+#       X11Forwarding no
+#       AllowTcpForwarding no
+#       PermitTTY no
+#       ForceCommand cvs server
 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
ed0dd52e880cbd7787a6224a3b8383f1
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/content: content changed '{md5}ed0dd52e880cbd7787a62
24a3b8383f1' to '{md5}e6ad59e8a85709a2c445e155da9feac5'
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/mode: mode changed '0644' to '0640'
Info: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]: Scheduling refresh of Service[sshd]
Info: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]: Scheduling refresh of Service[sshd]
Notice: /Stage[main]/Ssh/Service[sshd]: Triggered 'refresh' from 2 events
Notice: Applied catalog in 0.37 seconds
```

La config du sshd_config est bien poussé notre module fonctionne.

Exercice 2 :

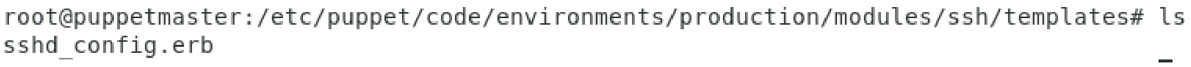
La commande facter nous liste toutes les variables du rogue1, la variable qu'on recherche est ipaddress_eth0 on peut le vérifier avec facter ipaddress_eth0 :



```
root@rogue1:/home/vincent# facter ipaddress_eth0
192.168.1.100
```

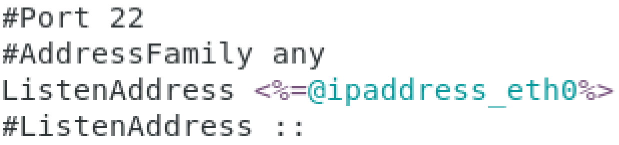
Exercice 3 :

On copie le sshd_config dans le dossier template en .erb :



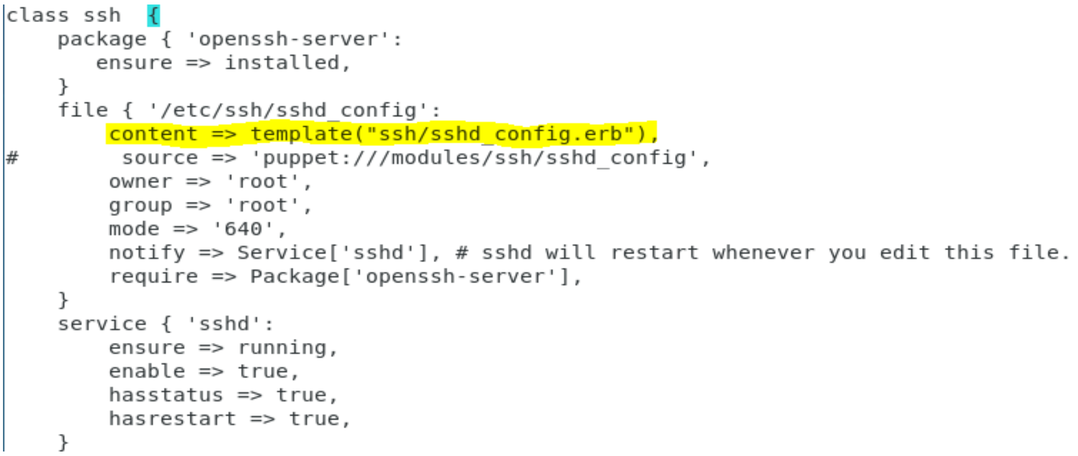
```
root@puppetmaster:/etc/puppet/code/environments/production/modules/ssh/templates# ls
sshd_config.erb
```

Dedans on dé commente la ligne list address et on met la variable adéquate :



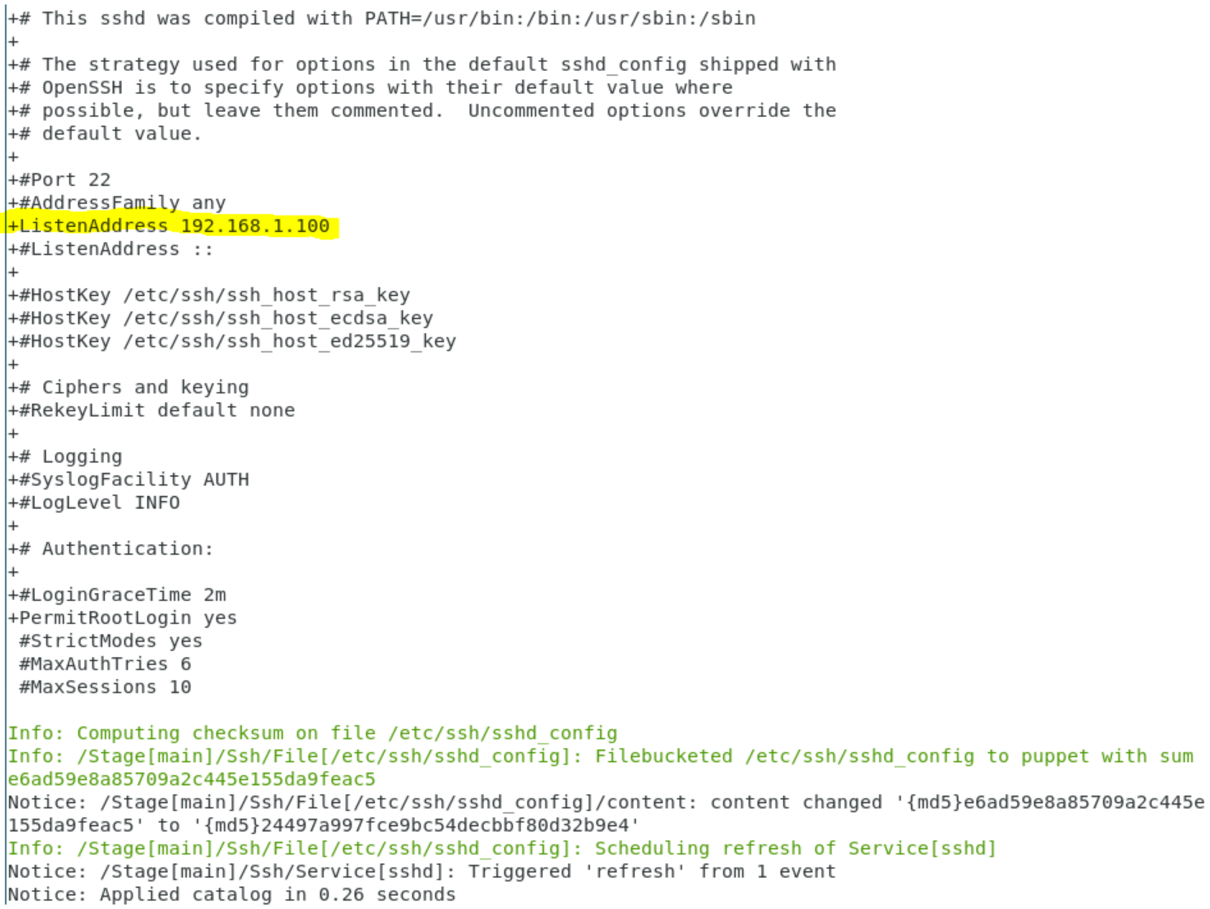
```
#Port 22
#AddressFamily any
ListenAddress <%=@ipaddress_eth0%>
#ListenAddress ::
```

On modifie notre init.pp en conséquence :



```
class ssh  {
    package { 'openssh-server':
       ensure => installed,
    }
    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':
        ensure => running,
        enable => true,
        hasstatus => true,
        hasrestart => true,
    }
```

On a plus qu’a retester sur le client :



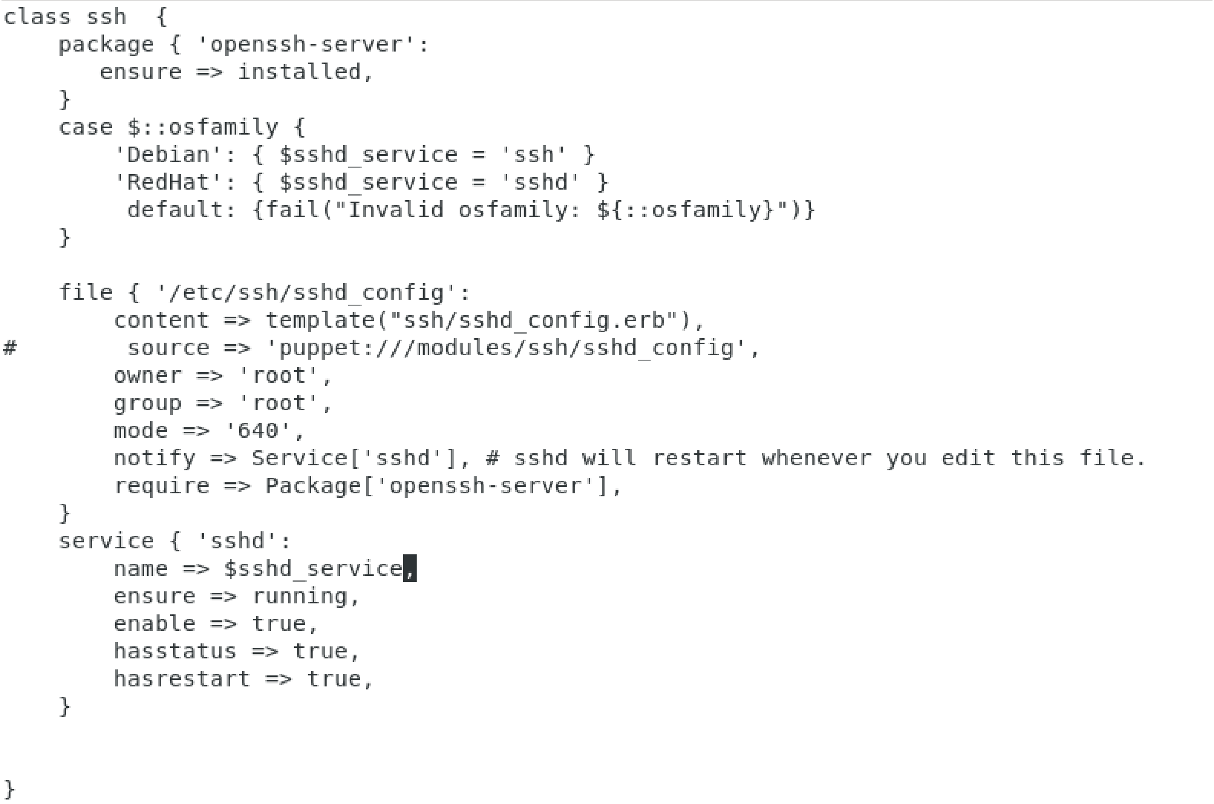
```
+# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
+
+# The strategy used for options in the default sshd_config shipped with
+# OpenSSH is to specify options with their default value where
+# possible, but leave them commented.  Uncommented options override the
+# default value.
+
+#Port 22
+#AddressFamily any
+ListenAddress 192.168.1.100
+#ListenAddress ::
+
+#HostKey /etc/ssh/ssh_host_rsa_key
+#HostKey /etc/ssh/ssh_host_ecdsa_key
+#HostKey /etc/ssh/ssh_host_ed25519_key
+
+# Ciphers and keying
+#RekeyLimit default none
+
+# Logging
+#SyslogFacility AUTH
+#LogLevel INFO
+
+# Authentication:
+
+#LoginGraceTime 2m
+PermitRootLogin yes
 #StrictModes yes
 #MaxAuthTries 6
 #MaxSessions 10

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
e6ad59e8a85709a2c445e155da9feac5
Notice: /Stage[main]/Ssh/File[/etc/ssh/sshd_config]/content: content changed '{md5}e6ad59e8a85709a2c445e
155da9feac5' 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 0.26 seconds
```

Notre variable à listen address a bien été poussée.

Exercice 4 :

On copie le case dans notre init.pp mais pour qu'il soit prit en compte dans notre service il faut rajouter une ligne name qui ira chercher la variable $sshd_service du bon os que la variable $osfamily du client nous poussera, ici notre rogue1 est un debian, le service poussé sera ssh.

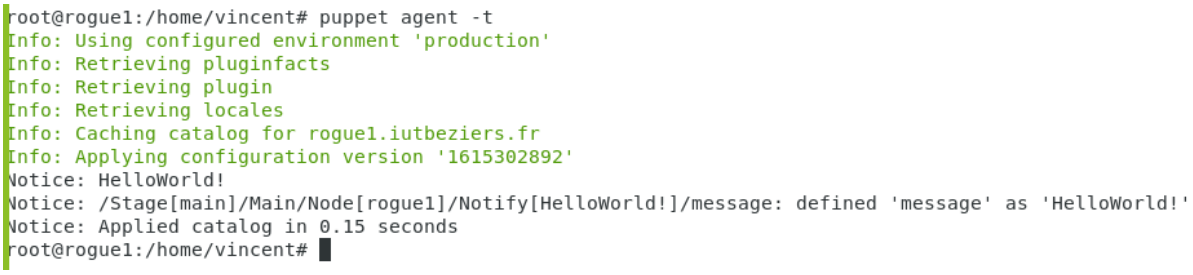


```
class ssh  {
    package { 'openssh-server':
       ensure => installed,
    }
    case $::osfamily {
        'Debian': { $sshd_service = 'ssh' }
        'RedHat': { $sshd_service = 'sshd' }
         default: {fail("Invalid osfamily: ${::osfamily}")}
    }

    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,
    }


}
```

Test sur le client, ça fonctionne :



```
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 '1615302892'
Notice: HelloWorld!
Notice: /Stage[main]/Main/Node[rogue1]/Notify[HelloWorld!]/message: defined 'message' as 'HelloWorld!'
Notice: Applied catalog in 0.15 seconds
root@rogue1:/home/vincent#
```