

Dernière mise-à-jour : 2018/06/05 14:33

# PPT101 - Gestion de l'Administration de Puppet

## Qu'est-ce Puppet ?

Puppet est :

- un logiciel libre écrit partiellement en Ruby permettant la gestion de la configuration de serveurs esclaves ( GNU/Linux, Mac OS X et Windows ),
- diffusé sous licence Apache 2.0 pour les versions récentes de Puppet.

La version libre :

- permet de gérer les déploiements système et applicatif,
- accepte les machines virtuelles de type Amazon EC2.

La version commerciale de Puppet permet aussi :

- de gérer les machines virtuelles VMware,
- d'avoir une interface graphique de gestion,
- d'automatiser et d'orchestrer les déploiements,
- d'avoir une plateforme de développement pour tous les environnements,
- de gérer individuellement les droits utilisateurs.

## Que couvre ce cours

Ce cours couvre les sujets suivants :

- Démarrer avec Puppet,
  - Utiliser des manifests,
-

- Gérer du code Puppet avec Git,
- Ressources Puppet,
- Variables, Expressions et Expressions Conditionnelles,
- Facter et les Facts,
- Hiera,
- Modules Puppet.

## Démarrer avec Puppet

Puppet est un outil de gestion de la configuration de systèmes. Pour résumer le flux de travail avec Puppet, on peut dire que :

- vous spécifiez l'état de configuration voulu en éditant des fichiers textes et un modèle des ressources du système,
- Puppet compare l'état actuel avec l'état voulu et procède aux modifications nécessaires.

Plus particulièrement Puppet est :

- un langage pour spécifier l'état désiré,
- un moteur qui interprète le code écrit dans ce langage et qui l'applique aux nœuds pour arriver à l'état désiré.

Par exemple :

```
package { 'curl':  
  ensure => installed,  
}
```

Ce **manifest** indique que le paquet **curl** doit être installé. Quand ce code est appliqué, Puppet vérifie si le paquet **curl** est installé dans le nœud puis :

- si c'est le cas, ne fait rien,
- sinon l'installe.

Voici un autre exemple :

```
user { 'trainee':
```

```
ensure => present,  
}
```

Dans ce cas, l'utilisateur doit être présent dans le noeud concerné, sinon Puppet va créer l'utilisateur.

La force de Puppet est qu'il est capable d'implémenter ce code quelque soit le système d'exploitation du noeud, parmi les systèmes d'exploitation compatibles :

- vous décrivez la configuration désirée en termes de ressources et d'attributs,
- Puppet crée les ressources dans les différents systèmes d'exploitation en utilisant le même **manifest**.

Il existe deux façons d'utiliser Puppet :

- Architecture Agent/Maître,
  - un noeud est dédié à l'exécution de Puppet et tous les autres noeuds doivent le contacter pour connaître la configuration à appliquer,
- Stand-alone Puppet ou "Sans Maître",
  - Puppet est exécuté sur **chaque** noeud et n'a pas donc besoin de contacter un Maître. Cette configuration utilise Git, SFTP ou rsync pour mettre à jour les manifests sur chaque noeud.



**Important** - Ce cours se concentre sur l'utilisation de Puppet en mode **Sans Maître** en utilisant Git.

## LAB #1 - Installation

### Installer Git

Si ce n'est pas déjà fait, commencez par créer votre compte sur GitHub <https://github.com/>.

Si ce n'est pas déjà fait, naviguez à l'URL <https://git-scm.com/download> et suivez les instructions pour installer Git sur votre système d'exploitation.

Clonez ensuite les fichiers pour ce cours :

### Sous Debian/Ubuntu

```
hnorris@hnorris-E200HA:~$ git clone https://github.com/bitfield/puppet-beginners-guide-3.git
Clonage dans 'puppet-beginners-guide-3'...
remote: Counting objects: 854, done.
remote: Total 854 (delta 0), reused 0 (delta 0), pack-reused 854
Réception d'objets: 100% (854/854), 98.47 KiB | 320.00 KiB/s, fait.
Résolution des deltas: 100% (443/443), fait.
```

### Sous Windows

```
C:\Users\hnorris>git clone https://github.com/bitfield/puppet-beginners-guide-3.git
Cloning into 'puppet-beginners-guide-3'...
remote: Counting objects: 854, done.
remote: Total 854 (delta 0), reused 0 (delta 0), pack-reused 854R
Receiving objects: 100% (854/854), 98.47 KiB | 202.00 KiB/s, done.
Resolving deltas: 100% (443/443), done.
```

### Installer VirtualBox

Si ce n'est pas déjà fait, naviguez à l'URL <https://www.virtualbox.org/wiki/Downloads> et suivez les instructions pour installer VirtualBox sur votre système d'exploitation.

### Installer Vagrant

Si ce n'est pas déjà fait, naviguez à l'URL <https://www.vagrantup.com/downloads.html> et suivez les instructions pour installer Vagrant sur votre

système d'exploitation.

## Installer la Machine Virtuelle Puppet

### Sous Debian/Ubuntu

```
hnorris@hnorris-E200HA:~$ cd puppet-beginners-guide-3
hnorris@hnorris-E200HA:~/puppet-beginners-guide-3$ scripts/start_vagrant.sh
```

### Sous Windows

Ajoutez le chemin **C:\Program Files\Git\bin** à votre PATH et au PATH système.

Par exemple, sous Windows™ 10 :

```
Panneau de configuration > Système et sécurité > Système > Modifier les paramètres > Paramètres système avancés >
Variables d'environnement > Path > Modifier (Pour votre utilisateur ET le système) > Nouveau > C:\Program
Files\Git\bin > OK > OK > OK
```

Exécutez **cmd** et saisissez les commandes suivantes :

```
Microsoft Windows [version 10.0.16299.431]
(c) 2017 Microsoft Corporation. Tous droits réservés.

C:\Users\hnorris>cd puppet-beginners-guide-3

C:\Users\hnorris\puppet-beginners-guide-3>cd scripts

C:\Users\hnorris\puppet-beginners-guide-3\scripts>start_vagrant.sh
```

```
C:\Users\hnorris\puppet-beginners-guide-3\scripts>
```

**Git bash** sera lancé pour exécuter le script.

## Se connecter à la Machine Virtuelle Puppet

A l'issue du processus d'installation, vous obtiendrez :

```
...
default: The `minitar` executable is no longer bundled with `minitar`. If you are
default: expecting this executable, make sure you also install `minitar-cli`.
default: Successfully installed minitar-0.6.1
default: Successfully installed puppet_forge-2.2.9
default: Successfully installed r10k-2.6.2
default: 10 gems installed
```

Connectez-vous à la machine virtuelle en utilisant la commande **vagrant ssh** :

### Sous Debian/Ubuntu

```
hnorris@hnorris-E200HA:~/puppet-beginners-guide-3$ vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
```

```
0 updates are security updates.
```

```
vagrant@ubuntu-xenial:~$
```

Vérifiez que Puppet a bien été installé :

```
vagrant@ubuntu-xenial:~$ puppet --version
5.5.1
vagrant@ubuntu-xenial:~$ exit
logout
Connection to 127.0.0.1 closed.
```

#### Sous Windows

```
C:\Users\hnmorris\puppet-beginners-guide-3\scripts>vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

```
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
```

```
0 packages can be updated.
0 updates are security updates.
```

```
vagrant@ubuntu-xenial:~$
```

Vérifiez que Puppet a bien été installé :

```
vagrant@ubuntu-xenial:~$ puppet --version
5.5.1
vagrant@ubuntu-xenial:~$ exit
logout
Connection to 127.0.0.1 closed.

C:\Users\hnnorris\puppet-beginners-guide-3\scripts>
```

## Utiliser des mainifests

### LAB #2 - Gestion des Fichiers

Re-connectez-vous à votre machine virtuelle et créez le fichier **file\_hello.pp** :

```
hnnorris@hnnorris-E200HA:~/puppet-beginners-guide-3$ vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Sat May 26 12:23:54 2018 from 10.0.2.2

vagrant@ubuntu-xenial:~$ vi file_hello.pp
```



```
vagrant@ubuntu-xenial:~$ cat file_hello.pp
file { ['/tmp/hello.txt']:
  ensure => file,
  content => "hello, world\n",
}
```

```
C:\Users\hnnorris\puppet-beginners-guide-3\scripts>vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

```
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
```

```
0 packages can be updated.
0 updates are security updates.
```

```
Last login: Sat May 26 12:25:41 2018 from 10.0.2.2
```

```
vagrant@ubuntu-xenial:~$ vi file_hello.pp
vagrant@ubuntu-xenial:~$ cat file_hello.pp
file { ['/tmp/hello.txt']:
  ensure => file,
  content => "hello, world\n",
}
```

Dans ce fichier nous pouvons constater la déclaration d'une ressource de type fichier avec **file**. Une ressource est une configuration que vous souhaitez être gérée par Puppet par exemple, un fichier, un utilisateur, un compte ou bien un paquet. Le format de notre manifest est donc :

```
TYPE_RESSOURCE { TITRE:
  ATTRIBUT => VALEUR,
```

```
...  
}
```

La ressource est identifiée par un titre. Chaque ressource doit avoir un titre unique. Dans le cas d'un fichier c'est le chemin complet vers le fichier - **/tmp/hello.txt**.

Le reste du code est une liste d'attributs pour la ressource. Dans le cas d'un fichier les attributs disponibles pour sont, par exemple :

- le contenu,
- le propriétaire,
- le groupe,
- le mode,
- etc ...



**Important** - Bien que les attributs soient différents selon le type de ressource, l'attribut **ensure** est commun à toutes les ressources. Par contre, la valeur de cet attribut diffère selon le type de ressource. Dans le cas de notre code, il est stipulé un fichier et non un répertoire ou un liens symbolique.

Le contenu de ce fichier est stipulé par l'attribut **content** qui est ici une chaîne **hello, world** suivie par un caractère de nouvelle ligne **\n**.

L'application du manifest par Puppet se résume ainsi :

- Puppet lit le manifest ainsi que la liste des ressources,
- Puppet compile les ressources en un **catalogue**,
- Puppet lit le catalogue et applique chaque ressource à tour de rôle.

Le nom du manifest n'est pas important, par contre l'extention doit être **.pp**

Appliquez ce fichier avec la commande suivante :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_hello.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/tmp/hello.txt]/ensure: defined content as '{md5}22c3683b094136c3398391ae71b20f04'
Notice: Applied catalog in 0.06 seconds
```

Vérifiez que Puppet a écrit le fichier **/tmp/hello.txt** :

```
vagrant@ubuntu-xenial:~$ cat /tmp/hello.txt
hello, world
```

### Modification d'un Fichier Existant sur le Serveur

Dans le cas où le fichier existe déjà et son contenu est différent, Puppet écrasera son contenu avec celui du manifest :

```
vagrant@ubuntu-xenial:~$ sudo sh -c 'echo "goodbye, world" > /tmp/hello.txt'
vagrant@ubuntu-xenial:~$ cat /tmp/hello.txt
goodbye, world
vagrant@ubuntu-xenial:~$ sudo puppet apply file_hello.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/tmp/hello.txt]/content: content changed '{md5}767887814e925822027f4fe63fb69ce2'
to '{md5}22c3683b094136c3398391ae71b20f04'
Notice: Applied catalog in 0.11 seconds
vagrant@ubuntu-xenial:~$ cat /tmp/hello.txt
hello, world
```



**Important** - Des modifications manuelles faites donc à un fichier géré par Puppet seront perdues la prochaine fois que Puppet s'exécute, sauf dans le cas où le manifest reflète les mêmes modifications. Pour cette raison, il est une bonne pratique d'insérer un commentaire en début de fichier comme avertissement.

## Effectuer un Dry Run avec Puppet

Il est possible de demander à Puppet, grâce à l'utilisation de l'option **-noop**, de nous informer des modifications qu'il aurait fait en appliquant un manifest, sans que ces modifications soient réellement effectuées :

```
vagrant@ubuntu-xenial:~$ sudo sh -c 'echo "goodbye, world" > /tmp/hello.txt'
vagrant@ubuntu-xenial:~$ sudo puppet apply --noop file_hello.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: /Stage[main]/Main/File[/tmp/hello.txt]/content: current_value '{md5}767887814e925822027f4fe63fb69ce2',
should be '{md5}22c3683b094136c3398391ae71b20f04' (noop)
Notice: Class[Main]: Would have triggered 'refresh' from 1 event
Notice: Stage[main]: Would have triggered 'refresh' from 1 event
Notice: Applied catalog in 0.07 seconds
vagrant@ubuntu-xenial:~$ cat /tmp/hello.txt
goodbye, world
```

Comme vous pouvez constater, Puppet décide si un fichier doit être modifié en fonction de la valeur du hash md5.

Pour constater les modifications que Puppet aurait effectué, utilisez l'option **-show\_diff** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --noop --show_diff file_hello.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: /Stage[main]/Main/File[/tmp/hello.txt]/content:
--- /tmp/hello.txt  2018-05-26 13:31:30.480333595 +0000
+++ /tmp/puppet-file20180526-27193-1pjhkk9  2018-05-26 13:36:45.039195308 +0000
@@ -1 +1 @@
-goodbye, world
+hello, world

Notice: /Stage[main]/Main/File[/tmp/hello.txt]/content: current_value '{md5}767887814e925822027f4fe63fb69ce2',
should be '{md5}22c3683b094136c3398391ae71b20f04' (noop)
Notice: Class[Main]: Would have triggered 'refresh' from 1 event
Notice: Stage[main]: Would have triggered 'refresh' from 1 event
```

Notice: Applied catalog in 0.11 seconds

## LAB #3 - Gestion des Paquets

Puppet est capable de gérer les paquets grâce à la ressource **package**. Dans le manifest nous trouvons donc cette ressource, le nom du paquet ainsi que l'attribut **ensure** :

```
package { 'package_name':  
  ensure => installed,  
}
```

Ce manifest aura comme résultat d'assurer que le package **package\_name** soit installé en utilisant le gestionnaire des paquets du système d'exploitation.

Créez le fichier **package.pp** suivant :

```
vagrant@ubuntu-xenial:~$ vi package.pp  
vagrant@ubuntu-xenial:~$ cat package.pp  
package { 'cowsay':  
  ensure => installed,  
}
```



**Important** - Le titre de la ressource de type paquet est **cowsay**.

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply package.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.51 seconds  
Notice: /Stage[main]/Main/Package[cowsay]/ensure: created
```

Notice: Applied catalog in 8.76 seconds

Le résultat de l'application de ce manifest est l'installation du paquet **cowsay** :

```
vagrant@ubuntu-xenial:~$ dpkg --get-selections | grep cowsay
cowsay                install
cowsay-off            install
vagrant@ubuntu-xenial:~$ dpkg -s cowsay
Package: cowsay
Status: install ok installed
Priority: optional
Section: games
Installed-Size: 90
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: all
Version: 3.03+dfsg1-15
Depends: libtext-charwidth-perl, perl
Recommends: cowsay-off
Suggests: filters
Description: configurable talking cow
 Cowsay (or cowthink) will turn text into happy ASCII cows, with
 speech (or thought) balloons. If you don't like cows, ASCII art is
 available to replace it with some other creatures (Tux, the BSD
 daemon, dragons, and a plethora of animals, from a turkey to
 an elephant in a snake).
Original-Maintainer: Tony Maillefaud <maltouzes@gmail.com>
Homepage: https://web.archive.org/web/20120527202447/http://www.nog.net/~tony/warez/cowsay.shtml
```

Pour voir la version du paquet que Puppet pense être installé, utilisez la commande **puppet resource** en spécifiant le paquet concerné :

```
vagrant@ubuntu-xenial:~$ sudo puppet resource package openssl
package { 'openssl':
  ensure => '1.0.2g-1ubuntu4.12',
}
```

```
vagrant@ubuntu-xenial:~$ dpkg -s openssl
Package: openssl
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 934
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Version: 1.0.2g-1ubuntu4.12
Depends: libc6 (>= 2.15), libssl1.0.0 (>= 1.0.2g)
Suggests: ca-certificates
Conffiles:
 /etc/ssl/openssl.cnf 7df26c55291b33344dc15e3935dabaf3
Description: Secure Sockets Layer toolkit - cryptographic utility
 This package is part of the OpenSSL project's implementation of the SSL
 and TLS cryptographic protocols for secure communication over the
 Internet.
.
It contains the general-purpose command line binary /usr/bin/openssl,
useful for cryptographic operations such as:
 * creating RSA, DH, and DSA key parameters;
 * creating X.509 certificates, CSRs, and CRLs;
 * calculating message digests;
 * encrypting and decrypting with ciphers;
 * testing SSL/TLS clients and servers;
 * handling S/MIME signed or encrypted mail.
Original-Maintainer: Debian OpenSSL Team <pkg-openssl-devel@lists.alioth.debian.org>
```

L'utilisation de cette commande sans spécifier un paquet permet de voir la liste de tous les paquets :

```
vagrant@ubuntu-xenial:~$ sudo puppet resource package | more
package { 'accountsservice':
  ensure => '0.6.40-2ubuntu11.3',
}
```

```
package { 'acl':  
  ensure => '2.2.52-3',  
}  
package { 'acpid':  
  ensure => '1:2.0.26-1ubuntu2',  
}  
package { 'adduser':  
  ensure => '3.113+nmu3ubuntu4',  
}  
package { 'apparmor':  
  ensure => '2.10.95-0ubuntu2.9',  
}  
package { 'apport':  
  ensure => '2.20.1-0ubuntu2.17',  
}  
package { 'apport-symptoms':  
  ensure => '0.20',  
}  
package { 'apt':  
  ensure => '1.2.26',  
  --More--
```

Puppet resource a aussi un mode interactif :

```
vagrant@ubuntu-xenial:~$ sudo puppet resource -e package openssl
```

Cette commande génère un manifest pour l'état actuel de la resource et l'ouvre dans un éditeur :

```
package { 'openssl':  
  ensure => '1.0.2g-1ubuntu4.12',  
}
```

Si vous modifiez ce manifest, lors de l'enregistrement du fichier, Puppet appliquera les modifications.



## LAB #4 - Gestion des Services

Le manifest pour la gestion d'un service contient la ressource **service**. Créez le fichier suivant :

```
vagrant@ubuntu-xenial:~$ vi service.pp
vagrant@ubuntu-xenial:~$ cat service.pp
service { 'sshd':
    ensure => running,
    enable => true,
}
```

Les attributs d'une ressource peuvent être consultés par la commande **puppet describe** :

```
vagrant@ubuntu-xenial:~$ sudo puppet describe service

service
=====
Manage running services.  Service support unfortunately varies
widely by platform --- some platforms have very little if any concept of a
running service, and some have a very codified and powerful concept.
Puppet's service support is usually capable of doing the right thing, but
the more information you can provide, the better behaviour you will get.
Puppet 2.7 and newer expect init scripts to have a working status command.
If this isn't the case for any of your services' init scripts, you will
need to set `hasstatus` to false and possibly specify a custom status
command in the `status` attribute. As a last resort, Puppet will attempt to
search the process table by calling whatever command is listed in the `ps`
fact. The default search pattern is the name of the service, but you can
specify it with the `pattern` attribute.
**Refresh:** `service` resources can respond to refresh events (via
`notify`, `subscribe`, or the `~>` arrow). If a `service` receives an
event from another resource, Puppet will restart the service it manages.
The actual command used to restart the service depends on the platform and
```

can be configured:

- \* If you set ``hasrestart`` to true, Puppet will use the init script's restart command.
- \* You can provide an explicit command for restarting with the ``restart`` attribute.
- \* If you do neither, the service's stop and start commands will be used.

## Parameters

-----

### - **`**binary**`**

The path to the daemon. This is only used for systems that do not support init scripts. This binary will be used to start the service if no ``start`` parameter is provided.

### - **`**control**`**

The control variable used to manage services (originally for HP-UX). Defaults to the upcased service name plus ``START`` replacing dots with underscores, for those providers that support the ``controllable`` feature.

### - **`**enable**`**

Whether a service should be enabled to start at boot. This property behaves quite differently depending on the platform; wherever possible, it relies on local tools to enable or disable a given service. Valid values are ``true``, ``false``, ``manual``, ``mask``. Requires features enableable.

### - **`**ensure**`**

Whether a service should be running. Valid values are ``stopped`` (also called ``false``), ``running`` (also called

`true`).

- **\*\*flags\*\***

Specify a string of flags to pass to the startup script.  
Requires features flaggable.

- **\*\*hasrestart\*\***

Specify that an init script has a `restart` command. If this is false and you do not specify a command in the `restart` attribute, the init script's `stop` and `start` commands will be used.  
Defaults to false.

Valid values are `true`, `false`.

- **\*\*hasstatus\*\***

Declare whether the service's init script has a functional status command; defaults to `true`. This attribute's default value changed in Puppet 2.7.0.

The init script's status command must return 0 if the service is running and a nonzero value otherwise. Ideally, these exit codes should conform to [the LSB's specification][lsb-exit-codes] for init script status actions, but Puppet only considers the difference between 0 and nonzero to be relevant.

If a service's init script does not support any kind of status command, you should set `hasstatus` to false and either provide a specific command using the `status` attribute or expect that Puppet will look for the service name in the process table. Be aware that 'virtual' init scripts (like 'network' under Red Hat systems) will respond poorly to refresh events from other resources if you override the default behavior without providing a status command.

Valid values are `true`, `false`.

- **\*\*manifest\*\***

Specify a command to config a service, or a path to a manifest to do so.

- **\*\*name\*\***

The name of the service to run.

This name is used to find the service; on platforms where services have short system names and long display names, this should be the short name. (To take an example from Windows, you would use "wuauserv" rather than "Automatic Updates.")

- **\*\*path\*\***

The search path for finding init scripts. Multiple values should be separated by colons or provided as an array.

- **\*\*pattern\*\***

The pattern to search for in the process table.

This is used for stopping services on platforms that do not support init scripts, and is also used for determining service status on those service whose init scripts do not include a status command.

Defaults to the name of the service. The pattern can be a simple string or any legal Ruby pattern, including regular expressions (which should be quoted without enclosing slashes).

- **\*\*restart\*\***

Specify a *\*restart\** command manually. If left unspecified, the service will be stopped and then started.

- **\*\*start\*\***

Specify a *\*start\** command manually. Most service subsystems support a *`start`* command, so this will not need to be specified.

- **\*\*status\*\***

Specify a *\*status\** command manually. This command must return 0 if the service is running and a nonzero value otherwise. Ideally, these exit codes should conform to [the LSB's

specification][lsb-exit-codes] for init script status actions, but Puppet only considers the difference between 0 and nonzero to be relevant.

If left unspecified, the status of the service will be determined automatically, usually by looking for the service in the process table.

[lsb-exit-codes]:

[http://refspecs.linuxfoundation.org/LSB\\_4.1.0/LSB-Core-generic/LSB-Core-generic/iniscrptact.html](http://refspecs.linuxfoundation.org/LSB_4.1.0/LSB-Core-generic/LSB-Core-generic/iniscrptact.html)

- **\*\*stop\*\***

Specify a **\*stop\*** command manually.

Providers

-----

base, bsd, daemontools, debian, freebsd, gentoo, init, launchd, openbsd, openrc, openwrt, rcng, redhat, runit, service, smf, src, systemd, upstart, windows

Pour consulter la liste des types de ressources, utilisez la commande suivante :

```
vagrant@ubuntu-xenial:~$ sudo puppet describe --list
```

These are the types known to puppet:

augeas	- Apply a change or an array of changes to the ...
computer	- Computer object management using DirectorySer ...
cron	- Installs and manages cron jobs
exec	- Executes external commands
file	- Manages files, including their content, owner ...
filebucket	- A repository for storing and retrieving file ...
group	- Manage groups
host	- Installs and manages host entries
interface	- This represents a router or switch interface
k5login	- Manage the <code>`.k5login`</code> file for a user
macauthorization	- Manage the Mac OS X authorization database

```
mailalias      - .. no documentation ..
maillist       - Manage email lists
mcx            - MCX object management using DirectoryService ...
mount         - Manages mounted filesystems, including puttin ...
nagios_command - The Nagios type command
nagios_contact - The Nagios type contact
nagios_contactgroup - The Nagios type contactgroup
nagios_host     - The Nagios type host
nagios_hostdependency - The Nagios type hostdependency
nagios_hostescalation - The Nagios type hostescalation
nagios_hostextinfo - The Nagios type hostextinfo
nagios_hostgroup - The Nagios type hostgroup
nagios_service - The Nagios type service
nagios_servicedependency - The Nagios type servicedependency
nagios_serviceescalation - The Nagios type serviceescalation
nagios_serviceextinfo - The Nagios type serviceextinfo
nagios_servicegroup - The Nagios type servicegroup
nagios_timeperiod - The Nagios type timeperiod
notify        - .. no documentation ..
package       - Manage packages
resources     - This is a metatype that can manage other reso ...
router        - .. no documentation ..
schedule      - Define schedules for Puppet
scheduled_task - Installs and manages Windows Scheduled Tasks
selboolean    - Manages SELinux booleans on systems with SELi ...
selmodule     - Manages loading and unloading of SELinux poli ...
service       - Manage running services
ssh_authorized_key - Manages SSH authorized keys
sshkey        - Installs and manages ssh host keys
stage         - A resource type for creating new run stages
tidy          - Remove unwanted files based on specific crite ...
user          - Manage users
vlan          - .. no documentation ..
whit          - Whits are internal artifacts of Puppet's curr ...
```

yumrepo	- The client-side description of a yum repository ...
zfs	- Manage zfs
zone	- Manages Solaris zones
zpool	- Manage zpools

Les différents types de ressources peuvent être regroupés dans le même manifest. Créez le fichier **package\_file\_service.pp** :

```
vagrant@ubuntu-xenial:~$ vi package_file_service.pp
vagrant@ubuntu-xenial:~$ cat package_file_service.pp
package { 'mysql-server':
  ensure => installed,
  notify => Service['mysql'],
}

file { ['/etc/mysql/mysql.cnf':
  source => '/examples/files/mysql.cnf',
  notify => Service['mysql'],
}

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

Notez que dans ce manifest se trouve l'attribut **notify**. Cet attribut notifie le service mysql lors d'un changement de son fichier de configuration **mysql.cnf**. L'action par défaut de Puppet dans ce cas est de re-démarrer le service.

L'ordre de la déclaration des ressources dans ce manifest est suivi par Puppet :

- Puppet installe le paquet **mysql-server**,
- Puppet copie le fichier **/examples/files/mysql.cnf** vers **/etc/mysql/mysql.cnf**,
- Puppet démarre le service **mysql**.

Cet ordre est logique. Il est évident que le manifest suivant ne donnera pas un résultat satisfaisant :

```
package { 'mysql-server':  
  ensure => installed,  
  notify => Service['mysql'],  
}  
  
service { 'mysql':  
  ensure => running,  
  enable => true,  
}  
  
file { '/etc/mysql/mysql.cnf':  
  source => '/examples/files/mysql.cnf',  
  notify => Service['mysql'],  
}
```

L'ordre de l'application des ressources dans un manifest peut cependant être fixé en utilisant l'attribut **require**. Voici le même manifest avec l'utilisation de cet attribut :

```
package { 'mysql-server':  
  ensure => installed,  
}  
  
file { '/etc/mysql/mysql.cnf':  
  source  => '/examples/files/mysql.cnf',  
  notify  => Service['mysql'],  
  require => Package['mysql-server'],  
}  
  
service { 'mysql':  
  ensure  => running,  
  enable  => true,  
  require => [Package['mysql-server'], File['/etc/mysql/mysql.cnf']],  
}
```



# Gérer du code Puppet avec Git

Git est :

- un logiciel de **gestion de versions** décentralisé,
- un logiciel libre créé par Linus Torvalds,
- distribué selon les termes de la licence publique générale GNU version 2.

Quand le code d'un projet est modifié par un développeur, celui-ci procède à un **commit** pour rendre le code disponible aux autres. Un commit est un **snapshot** ou instantanée du repo qui est gardé pour toujours ce qui implique la possibilité de **rollbacks**.



**Important** - Pour apprendre comment écrire un message de commit Git, consultez ce lien : <https://chris.beams.io/posts/git-commit> (en anglais).

## LAB #5 - Créer un Repository Local

Un jeu de fichiers sous Git est appelé un **repository** ou simplement un **repo**. Pour créer un repo local pour puppet, utilisez les commandes suivantes :

```
vagrant@ubuntu-xenial:~$ mkdir puppet
vagrant@ubuntu-xenial:~$ cd puppet
vagrant@ubuntu-xenial:~$ git init
Initialized empty Git repository in /home/vagrant/.git/
</CODE>
```

Git a besoin de savoir qui vous êtes. Saissiez donc les commandes suivantes en remplaçant les coordonnées d'identification avec les vôtres :

```
<code>
vagrant@ubuntu-xenial:~/puppet$ git config --global user.name "ittrainingdev"
```

```
vagrant@ubuntu-xenial:~/puppet$ git config --global user.email "infos@i2tch.co.uk"
```

Un repo doit contenir un fichier README qui contient des informations concernant le repo. Créez donc ce fichier avec un minimum d'informations :

```
vagrant@ubuntu-xenial:~/puppet$ echo "Coming soon!" > README.md
```

Consultez le statut du repo avec la commande **git status** :

```
vagrant@ubuntu-xenial:~/puppet$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README.md

nothing added to commit but untracked files present (use "git add" to track)
```

Pour informer Git de la présence du fichier **README.md**, il convient d'utiliser la commande **git add** :

```
vagrant@ubuntu-xenial:~/puppet$ git add README.md
vagrant@ubuntu-xenial:~/puppet$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   README.md
```

Notez que le fichier est maintenant sous la ligne **Changes to be committed:**. Procédez donc à un commit :

```
vagrant@ubuntu-xenial:~/puppet$ git commit -m 'Add README file'
[master (root-commit) 9139685] Add README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Pour voir l'historique des commits, utilisez la commande **git log** :

```
vagrant@ubuntu-xenial:~/puppet$ git log
commit 913968526d9748f8d92e6eacea03ac0a6d1ac901
Author: ittrainingdev <infos@i2tch.co.uk>
Date:   Sun May 27 04:53:07 2018 +0000
```

```
    Add README file
```

## Les Branches avec Git

Une Branche sous Git est une copie parallèle du code où les modifications sont indépendantes. Ces modification peuvent ensuite être fusionner avec la Branche Maître. Ceci permet :

- de connecter un noeud à une branche spécifique pour tester du code, sans l'implémenter sur tous le noeuds,
- à deux ou plusieurs développeurs de travailler d'une manière indépendante puis d'échanger des commits entre les Branches.

Vous pouvez trouver d'avantage d'informations concernant les Branches de Git à cet URL

<https://git-scm.com/book/fr/v1/Les-branches-avec-Git-Brancher-et-fusionner%C2%A0%3A-les-bases> ainsi que l'ebook gratuit ici - <https://git-scm.com/book/fr/v2>.

## LAB #6 - Créer un Repository Distant

Commencez par vous connecter à votre compte sur <https://github.com>. Créez ensuite un nouveau repo **puppet** à l'adresse <https://github.com/new>.

Copiez l'URL du repo, par exemple : <https://github.com/ittrainingdev/puppet.git>.

Vous devez maintenant pousser votre repo local vers Github. Placez-vous dans le répertoire puppet de votre machine virtuelle et saisissez la commande suivante en collant **à la place** de l'URL ci-dessous, l'URL que vous avez copié ci-dessus et **en utilisant vos coordonnées de connexion** :

```
vagrant@ubuntu-xenial:~/puppet$ git remote add origin https://github.com/ittrainingdev/puppet.git
vagrant@ubuntu-xenial:~/puppet$ git push -u origin master
Username for 'https://github.com': ittrainingdev
Password for 'https://ittrainingdev@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 232 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ittrainingdev/puppet.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Vérifiez maintenant votre repo sur Github en visitant l'URL de votre repo. Vous devez voir le fichier README.md :

```
vagrant@ubuntu-xenial:~/puppet$ sudo apt-get install lynx

vagrant@ubuntu-xenial:~/puppet$ lynx --dump https://github.com/ittrainingdev/puppet.git
...

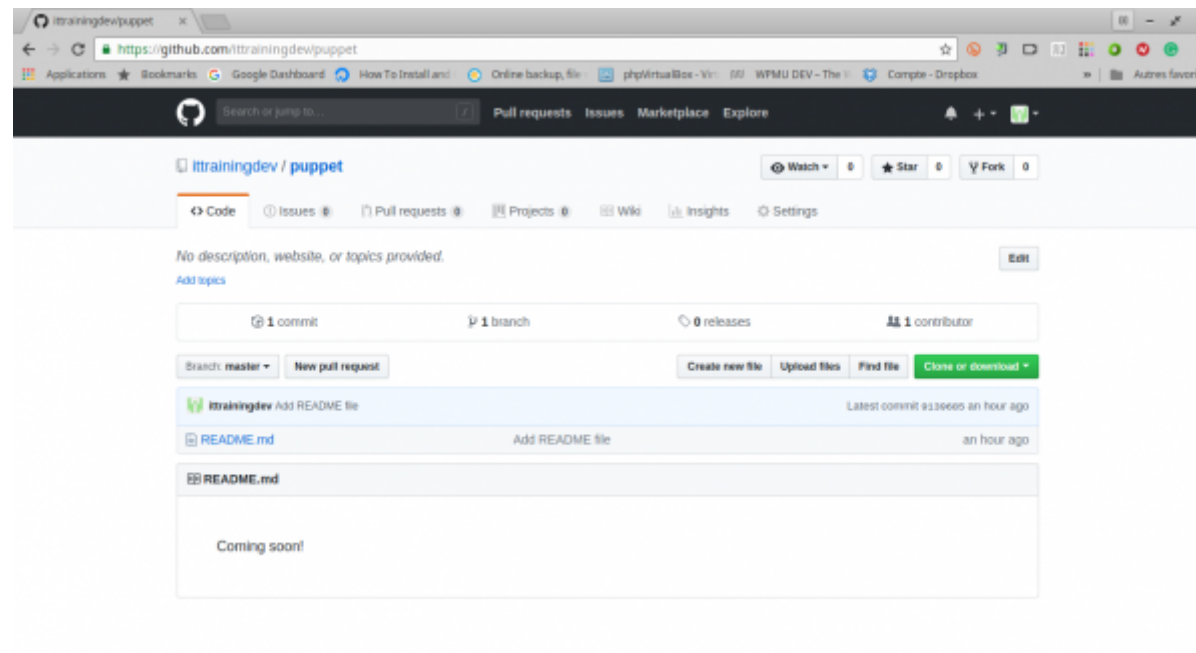
(BUTTON) Go back
Fetching latest commit...
Cannot retrieve the latest commit at this time.
[39]Permalink
Failed to load latest commit information.
[40]README.md [41]Add README file May 27, 2018

README.md

Coming soon!
```

- \* © 2018 GitHub, Inc.
- \* [42]Terms
- \* [43]Privacy
- \* [44]Security
- \* [45]Status
- \* [46]Help
  
- \* [47]Contact GitHub
- \* [48]API
- \* [49]Training
- \* [50]Shop
- \* [51]Blog
- \* [52]About

...



## LAB #7 - Cloner un Repository

Le répertoire par défaut de stockage des manifests de Puppet est **/etc/puppetlabs/code/environments/**.

Placez-vous donc dans ce répertoire :

```
vagrant@ubuntu-xenial:~/puppet$ cd /etc/puppetlabs/code/environments/
```

Consultez son contenu :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments$ ls  
pbg  production
```

Le répertoire **production** contient des exemples de manifests de production installés par défaut :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments$ cd production/  
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ ls  
data  environment.conf  hiera.yaml  manifests  modules
```

Git refuse de cloner vers un répertoire non-vidé. Renommmez donc le répertoire production :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ cd ..  
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments$ sudo mv production production.sample
```

Cloner maintenant le repo sur Github vers un **nouveau** répertoire **/etc/puppetlabs/code/environments/production** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments$ sudo git clone  
https://github.com/ittrainingdev/puppet.git production  
Cloning into 'production'...  
remote: Counting objects: 3, done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

## LAB #8 - Appliquer des Modifications Automatiquement

En production, chaque neoud doit automatiquement télécharger les modifications du code dans le **repo** distant puis les appliquer avec Puppet.

Pour arriver à cette fin, il convient de créer un script bash dans le répertoire **/home/vagrant/puppet/files/** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments$ cd /home/vagrant/puppet
vagrant@ubuntu-xenial:~/puppet$ mkdir manifests files
vagrant@ubuntu-xenial:~/puppet$ vi files/run-puppet.sh
vagrant@ubuntu-xenial:~/puppet$ cat files/run-puppet.sh
#!/bin/bash
cd /etc/puppetlabs/code/environments/production && git pull
/opt/puppetlabs/bin/puppet apply manifests/
```

Créez ensuite le manifest **run-puppet.pp** dans le répertoire **/home/vagrant/puppet/manifests/** :

```
vagrant@ubuntu-xenial:~/puppet$ vi manifests/run-puppet.pp
vagrant@ubuntu-xenial:~/puppet$ cat manifests/run-puppet.pp
# Set up regular Puppet runs
file { ['/usr/local/bin/run-puppet']:
  source => '/etc/puppetlabs/code/environments/production/files/run-puppet.sh',
  mode   => '0755',
}

cron { 'run-puppet':
  command => '/usr/local/bin/run-puppet',
  hour    => '*',
  minute  => '*/15',
}
```



**Important** - Ce manifest copie le script **/etc/puppetlabs/code/environments/production/files/run-puppet.sh** dans le répertoire **/usr/local/bin/** du noeud en le nommant **run-puppet** puis crée un cron job qui appelle ce script toutes les 15 minutes.

Ajoutez maintenant les fichiers à Git :

```
vagrant@ubuntu-xenial:~/puppet$ git add manifests files
vagrant@ubuntu-xenial:~/puppet$ git commit -m 'Add run-puppet script and cron job'
[master 756859c] Add run-puppet script and cron job
 2 files changed, 14 insertions(+)
 create mode 100644 files/run-puppet.sh
 create mode 100644 manifests/run-puppet.pp
vagrant@ubuntu-xenial:~/puppet$ git push origin master
Username for 'https://github.com': ittrainingdev
Password for 'https://ittrainingdev@github.com':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 688 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/ittrainingdev/puppet.git
 9139685..756859c  master -> master
```

## LAB #9 - Mise en Place sur un Noeud

Votre noeud se trouve dans le répertoire **/etc/puppetlabs/code/environments/production** :

```
vagrant@ubuntu-xenial:~/puppet$ cd /etc/puppetlabs/code/environments/production
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ ls
```



## README.md

Commencez par faire un **git pull** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ sudo git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/ittrainingdev/puppet
   9139685..756859c  master    -> origin/master
Updating 9139685..756859c
Fast-forward
   files/run-puppet.sh      | 3 +++
   manifests/run-puppet.pp | 11 ++++++++
 2 files changed, 14 insertions(+)
 create mode 100644 files/run-puppet.sh
 create mode 100644 manifests/run-puppet.pp
```

Vérifiez le résultat :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ ls
files  manifests  README.md
```

Appliquez maintenant le manifest sur le noeud :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ sudo puppet apply manifests/
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.20 seconds
Notice: /Stage[main]/Main/File[/usr/local/bin/run-puppet]/ensure: defined content as '{md5}dbfba978957e90ebb47e3a266b89231b'
Notice: /Stage[main]/Main/Cron[run-puppet]/ensure: created
Notice: Applied catalog in 0.22 seconds
```

Vérifiez que le script a été créé sur le noeud :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ cat /usr/local/bin/run-puppet
#!/bin/bash
cd /etc/puppetlabs/code/environments/production && git pull
/opt/puppetlabs/bin/puppet apply manifests/
```

ainsi que le cron job :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production$ sudo crontab -l
# HEADER: This file was autogenerated at 2018-05-27 07:26:05 +0000 by puppet.
# HEADER: While it can still be managed manually, it is definitely not recommended.
# HEADER: Note particularly that the comments starting with 'Puppet Name' should
# HEADER: not be deleted, as doing so could cause duplicate cron jobs.
# Puppet Name: run-puppet
*/15 * * * * /usr/local/bin/run-puppet
```

## Les ressources Puppet

### LAB #10 - La Ressource File

#### L'Attribut Source

Créez le fichier **file\_source.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_source.pp
vagrant@ubuntu-xenial:~$ cat file_source.pp
file { '/etc/motd':
  source => '/home/vagrant/files/motd.txt',
}
```

Créez le répertoire **files** ainsi que le fichier **files/motd.txt** :

```
vagrant@ubuntu-xenial:~$ mkdir files
vagrant@ubuntu-xenial:~$ vi files/motd.txt
vagrant@ubuntu-xenial:~$ cat files/motd.txt
The best software in the world only sucks. The worst software is significantly worse than that.
-Luke Kanies
```

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_source.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/motd]/ensure: defined content as '{md5}2400ab50861ff84ea5a3e43126d162d5'
Notice: Applied catalog in 0.16 seconds
vagrant@ubuntu-xenial:~$ cat /etc/motd
The best software in the world only sucks. The worst software is significantly worse than that.
-Luke Kanies
```



**Important** - Notez que le contenu du fichier **/etc/motd** est le contenu du fichier **/home/vagrant/files/motd.txt** indiqué par l'attribut source.

L'attribut source peut aussi utilisé la valeur d'un URL. Créez le fichier **file\_http.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_http.pp
vagrant@ubuntu-xenial:~$ cat file_http.pp
file { '/tmp/README.md':
  source => 'https://raw.githubusercontent.com/puppetlabs/puppet/master/README.md',
}
```

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_http.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/tmp/README.md]/ensure: defined content as '{mtime}2018-05-27 09:16:00 +0000'
Notice: Applied catalog in 1.39 seconds
```

```
vagrant@ubuntu-xenial:~$ cat /tmp/README.md
Puppet
=====

[![Travis
Status](https://travis-ci.org/puppetlabs/puppet.svg?branch=master)](https://travis-ci.org/puppetlabs/puppet)
[![Appveyor
Status](https://ci.appveyor.com/api/projects/status/cvhpypd4504sevqq/branch/master?svg=true)](https://ci.appveyor
.com/project/puppetlabs/puppet/branch/master)
[![Gem Version](https://badge.fury.io/rb/puppet.svg)](https://badge.fury.io/rb/puppet)
[![Inline docs](https://inch-ci.org/github/puppetlabs/puppet.svg)](https://inch-ci.org/github/puppetlabs/puppet)

Puppet, an automated administrative engine for your Linux, Unix, and Windows systems, performs
administrative tasks (such as adding users, installing packages, and updating server
configurations) based on a centralized specification.

Documentation
-----

Documentation for Puppet and related projects can be found online at the
[Puppet Docs site](https://puppet.com/docs).

...
```

## L'Attribut owner

Dans l'exemple ci-dessus, l'utilisateur root a été utilisé pour exécuter les commandes puppet. Pour cette raison, les fichiers ainsi créés appartiennent à l'utilisateur **root** et sont associés avec le groupe **root** :

```
vagrant@ubuntu-xenial:~$ ls -l /etc/motd
-rw-r--r-- 1 root root 109 May 27 09:07 /etc/motd
```

L'appartenance d'un fichier peut être fixée grâce à l'utilisation de l'attribut **owner**. Créez le fichier **file\_owner.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_owner.pp
vagrant@ubuntu-xenial:~$ cat file_owner.pp
file { '/etc/owned_by_ubuntu':
  ensure => present,
  owner  => 'ubuntu',
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_owner.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/owned_by_ubuntu]/ensure: created
Notice: Applied catalog in 0.07 seconds
```

Vérifiez maintenant le propriétaire du fichier **/etc/owned\_by\_ubuntu** :

```
vagrant@ubuntu-xenial:~$ ls -l /etc/owned_by_ubuntu
-rw-r--r-- 1 ubuntu root 0 May 27 09:25 /etc/owned_by_ubuntu
```

## L'Attribut group

Le groupe d'un fichier peut être fixé grâce à l'utilisation de l'attribut **group**. Créez le fichier **file\_group.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_group.pp
vagrant@ubuntu-xenial:~$ cat file_group.pp
file { '/etc/owned_by_ubuntu':
  ensure => present,
  owner  => 'ubuntu',
  group  => 'ubuntu',
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_group.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/owned_by_ubuntu]/group: group changed 'root' to 'ubuntu'
Notice: Applied catalog in 0.06 seconds
```

Vérifiez maintenant le groupe du fichier **/etc/owned\_by\_ubuntu** :

```
vagrant@ubuntu-xenial:~$ ls -l /etc/owned_by_ubuntu
-rw-r--r-- 1 ubuntu ubuntu 0 May 27 09:25 /etc/owned_by_ubuntu
```

## L'Attribut mode

Les permission d'un fichier peuvent être fixées grâce à l'utilisation de l'attribut **mode**. Créez le fichier **file\_mode.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_mode.pp
vagrant@ubuntu-xenial:~$ cat file_mode.pp
file { '/etc/owned_by_ubuntu':
  ensure => present,
  owner  => 'ubuntu',
  mode   => '0600',
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_mode.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: Applied catalog in 0.05 seconds
```

Vérifiez maintenant les permissions du fichier **/etc/owned\_by\_ubuntu** :

```
vagrant@ubuntu-xenial:~$ ls -l /etc/owned_by_ubuntu
-rw----- 1 ubuntu ubuntu 0 May 27 09:25 /etc/owned_by_ubuntu
```

## L'Attribut ensure

Si la valeur de l'attribut **ensure** est **directory**, la ressource **file** fait référence à un répertoire et non à un fichier :

```
vagrant@ubuntu-xenial:~$ vi file_directory.pp
vagrant@ubuntu-xenial:~$ cat file_directory.pp
file { ['/etc/config_dir']:
    ensure => directory,
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_directory.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/config_dir]/ensure: created
Notice: Applied catalog in 0.07 seconds
```

Vérifiez maintenant la création du répertoire **/etc/config\_dir** :

```
vagrant@ubuntu-xenial:~$ ls -l /etc | grep config_
drwxr-xr-x 2 root  root  4096 May 27 12:17 config_dir
```

Si la valeur de l'attribut **ensure** est **link**, la ressource **file** fait référence à **lien symbolique** :

```
vagrant@ubuntu-xenial:~$ vi file_symlink.pp
vagrant@ubuntu-xenial:~$ cat file_symlink.pp
file { ['/etc/this_is_a_link']:
    ensure => link,
    target => '/etc/motd',
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_symlink.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/this_is_a_link]/ensure: created
Notice: Applied catalog in 0.07 seconds
```

Vérifiez maintenant la création du lien :

```
vagrant@ubuntu-xenial:~$ ls -l /etc/this_is_a_link
lrwxrwxrwx 1 root root 9 May 27 12:39 /etc/this_is_a_link -> /etc/motd
```

## L'Attribut recurse

Cet attribut permet de copier une arborescence complète. Créez le répertoire **files/config** contenant trois fichiers **a**, **b** et **c** :

```
vagrant@ubuntu-xenial:~$ mkdir files/config
vagrant@ubuntu-xenial:~$ touch files/config/a files/config/b files/config/c
```

Créez le manifest **file\_tree.pp** :

```
vagrant@ubuntu-xenial:~$ vi file_tree.pp
vagrant@ubuntu-xenial:~$ cat file_tree.pp
file { ['/etc/config_dir':
  source => '/home/vagrant/files/config',
  recurse => true,
}]
```

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply file_tree.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/File[/etc/config_dir/a]/ensure: defined content as
'{md5}d41d8cd98f00b204e9800998ecf8427e'
```



```
Notice: /Stage[main]/Main/File[/etc/config_dir/b]/ensure: defined content as
'{md5}d41d8cd98f00b204e9800998ecf8427e'
Notice: /Stage[main]/Main/File[/etc/config_dir/c]/ensure: defined content as
'{md5}d41d8cd98f00b204e9800998ecf8427e'
Notice: Applied catalog in 0.18 seconds
```

Vérifiez maintenant le contenu du répertoire **/etc/config\_dir** :

```
vagrant@ubuntu-xenial:~$ ls -lR /etc/config_dir
/etc/config_dir:
total 0
-rw-r--r-- 1 root root 0 May 27 12:30 a
-rw-r--r-- 1 root root 0 May 27 12:30 b
-rw-r--r-- 1 root root 0 May 27 12:30 c
```



**Important** - Si le répertoire cible existe et il contient déjà des fichiers, Puppet ne fera rien. Ceci peut être modifié en utilisant l'attribut **purge** au quel cas Puppet supprimera les fichiers et sous-répertoires dans le répertoire cible qui ne sont pas présents dans le répertoire source.

## LAB #11 - La Ressource package

### L'Attribut ensure

#### Installation de Paquets

Normalement, la valeur de cet attribut est **installed**, comme vu dans l'exemple précédent :

```
vagrant@ubuntu-xenial:~$ cat package.pp
package { 'cowsay':
  ensure => installed,
}
```

==

Il est cependant possible de spécifier une version spécifique d'un paquet, comme démontre l'exemple ci-dessous :

```
package { 'openssl':
  ensure => '1.0.2g-1ubuntu4.8',
}
```



**Important** - Si de multiple versions d'un paquet existe, en spécifiant **ensure => to install** Puppet installera la version par défaut. Par contre, en spécifiant **ensure => latest**, Puppet installera la version la plus récente.

### Le Cas Spécifique des Gems de Ruby

Installez Ruby :

```
vagrant@ubuntu-xenial:~$ vi package_gem.pp
vagrant@ubuntu-xenial:~$ cat package_gem.pp
package { 'ruby':
  ensure => installed,
}

package { 'puppet-lint':
  ensure  => installed,
```

```
provider => gem,  
}
```



**Important** - Un Ruby Gem est une bibliothèque Ruby qui étend les fonctionnalités de ce premier. Dans le fichier précédent, **puppet-lint** est un Gem et ne peut pas être installé par le gestionnaire de paquet. Puppet est informé de ceci par la ligne **provider => gem**.

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply package_gem.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.50 seconds  
Notice: /Stage[main]/Main/Package[ruby]/ensure: created  
Notice: /Stage[main]/Main/Package[puppet-lint]/ensure: created  
Notice: Applied catalog in 44.94 seconds
```

L'outil **puppet-lint** permet de vérifier les manifests pour des erreurs, par exemple :

```
vagrant@ubuntu-xenial:~$ vi lint_test.pp  
vagrant@ubuntu-xenial:~$ cat lint_test.pp  
file { '/tmp/lint.txt':  
  ensure => file,  
  content => "puppet-lint is your friend\n",  
}  
  
vagrant@ubuntu-xenial:~$ puppet-lint lint_test.pp  
WARNING: indentation of => is not properly aligned (expected in column 11, but found it in column 10) on line 2
```

Bien que Puppet soit partiellement écrit en Ruby, il ne peut pas utiliser des Gems du **provider => gem**. En effet, il convient d'utiliser le **provider => puppet\_gem** afin d'installer un Gem dans le contexte de Puppet et non dans le contexte du système :

```
vagrant@ubuntu-xenial:~$ vi package_puppet_gem.pp
vagrant@ubuntu-xenial:~$ cat package_puppet_gem.pp
package { 'r10k':
  ensure => installed,
  provider => puppet_gem,
}
vagrant@ubuntu-xenial:~$ sudo puppet apply package_puppet_gem.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.48 seconds
Notice: Applied catalog in 1.11 seconds
```

Pour vérifier si le Gem a été installé dans le contexte de Puppet, utilisez la commande suivante :

```
vagrant@ubuntu-xenial:~$ /opt/puppetlabs/puppet/bin/gem list | grep r10k
r10k (2.6.2)
```

### Suppression de Paquets

Il est aussi possible de désinstaller un paquet :

```
vagrant@ubuntu-xenial:~$ vi package_remove.pp
vagrant@ubuntu-xenial:~$ cat package_remove.pp
package { 'apparmor':
  ensure => absent,
}

vagrant@ubuntu-xenial:~$ dpkg -l | grep apparmor
ii  apparmor                2.10.95-0ubuntu2.9      amd64      user-space parser
utility for AppArmor
ii  libapparmor-perl        2.10.95-0ubuntu2.9      amd64      AppArmor library
Perl bindings
ii  libapparmor1:amd64      2.10.95-0ubuntu2.9      amd64      changehat AppArmor
library
```

Appliquez les manifest avec Puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply package_remove.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.50 seconds
Notice: /Stage[main]/Main/Package[apparmor]/ensure: removed
Notice: Applied catalog in 11.91 seconds
vagrant@ubuntu-xenial:~$ dpkg -l | grep apparmor
rc  apparmor                2.10.95-0ubuntu2.9      amd64      user-space parser
utility for AppArmor
ii  libapparmor-perl          2.10.95-0ubuntu2.9      amd64      AppArmor library
Perl bindings
ii  libapparmor1:amd64        2.10.95-0ubuntu2.9      amd64      changehat AppArmor
library
```



**Important** - Par défaut, quand Puppet supprime un paquet, il laisse les fichiers associés avec le paquet en place. Pour purger ces fichiers, il convient d'utiliser **purge**.

## LAB #12 - La Ressource service

Puppet a besoin de vérifier si un service a été démarré :

```
vagrant@ubuntu-xenial:~$ vi service_hasstatus.pp
vagrant@ubuntu-xenial:~$ cat service_hasstatus.pp
package { 'ntp':
  ensure => installed,
}

service { 'ntp':
  ensure  => running,
```

```
enable    => true,  
hasstatus => false,  
}
```

## L'Attribut **hasstatus**

La façon que cette vérification a lieu dépend du système d'exploitation du noeud. Par exemple dans le cas d'Ubuntu 16.04, Puppet utilisera la commande **systemctl**.

Dans certains cas Puppet essayera de démarrer un service chaque fois qu'il est exécuté. Ceci implique en règle générale que Puppet ne dispose pas de façon à vérifier si le service a déjà démarré. Dans ce cas, il convient d'utiliser l'attribut **hasstatus ⇒ false**. Dans ce cas, Puppet n'utilise pas les outils du système d'exploitation mais regarde dans le tableau des processus pour un processus dont le nom est le même que le service :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply service_hasstatus.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 2.11 seconds  
Notice: /Stage[main]/Main/Package[ntp]/ensure: created  
Notice: Applied catalog in 13.45 seconds  
  
vagrant@ubuntu-xenial:~$ systemctl status ntp  
● ntp.service - LSB: Start NTP daemon  
   Loaded: loaded (/etc/init.d/ntp; bad; vendor preset: enabled)  
   Active: active (running) since Sun 2018-05-27 13:52:05 UTC; 23s ago  
     Docs: man:systemd-sysv-generator(8)  
   CGroup: /system.slice/ntp.service  
           └─8582 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 112:116
```

## L'Attribut **pattern**

Dans le cas où le processus n'a pas le même nom que le processus, il convient d'utiliser l'attribut **pattern** pour aide Puppet a trouvé la correspondance :

```
vagrant@ubuntu-xenial:~$ ps aux | grep ntp
ntp      8582  0.1  0.4 110032  5020 ?        Ss   13:52   0:00 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u
112:116
vagrant  9328  0.0  0.0  12944   920 pts/0    S+   13:57   0:00 grep --color=auto ntp
```

Par exemple :

```
service { 'ntp':
  ensure    => running,
  enable    => true,
  hasstatus => false,
  pattern   => 'ntpd',
}
```

## Les Attributs **hasrestart** et **restart**

Par défaut, quand une ressource **file** utilise l'attribut **notify** pour informer Puppet que son fichier de configuration a été modifié, Puppet arrête puis démarre les service.

Cependant, certains services acceptent la commande **restart**. Dans ce cas, il convient d'informer Puppet en incluant dans le manifest l'attribut **hasrestart** :

```
vagrant@ubuntu-xenial:~$ vi service_hasrestart.pp
vagrant@ubuntu-xenial:~$ cat service_hasrestart.pp
service { 'ntp':
  ensure    => running,
  enable    => true,
  hasrestart => true,
}

vagrant@ubuntu-xenial:~$ sudo puppet apply service_hasrestart.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.69 seconds
```

Notice: Applied catalog in 0.14 seconds

Par contre, que se passe-t-il si la commande restart par défaut du système d'exploitation ne fonctionne pas ? Dans ce cas, il est possible de stipuler une commande avec l'attribut **restart** :

```
service { 'ntp':  
  ensure => running,  
  enable => true,  
  restart => '/bin/echo Restarting >>/tmp/debug.log && systemctl restart ntp',  
}
```

## LAB #13 - La Ressource user

Un utilisateur est un objet qui peut :

- posséder des fichiers,
- exécuter des commandes,
- éventuellement peut lire ou modifier les fichiers d'autres utilisateurs.

### Créer un Utilisateur

Créez le fichier **user.pp** :

```
vagrant@ubuntu-xenial:~$ vi user.pp  
vagrant@ubuntu-xenial:~$ cat user.pp  
group { 'devs':  
  ensure => present,  
  gid    => 3000,  
}  
  
user { 'trainee':  
  ensure => present,
```



```
uid    => '3002',
home   => '/home/trainee',
shell  => '/bin/bash',
groups => ['devs'],
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply user.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: /Stage[main]/Main/User[trainee]/ensure: created
Notice: Applied catalog in 0.19 seconds
```



**Important** - Le type de la ressource est le login de l'utilisateur spécifié par l'attribut **user**. Notez que l'UID est spécifié de façon à ce que l'utilisateur possède le même UID sur tous les nœuds.

Contrôlez la présence de **trainee** dans le fichier **/etc/passwd** :

```
vagrant@ubuntu-xenial:~$ tail /etc/passwd
uidd:x:108:112::/run/uidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000::,/home/vagrant:/bin/bash
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
ntp:x:112:116::/home/ntp:/bin/false
trainee:x:3002:3002::/home/trainee:/bin/bash
```

Notez cependant que le répertoire personnel n'a pas été créé :

```
vagrant@ubuntu-xenial:~$ ls -l /home
total 8
drwxr-xr-x 3 ubuntu  ubuntu  4096 May 26 09:55 ubuntu
drwxr-xr-x 8 vagrant  vagrant 4096 May 27 14:35 vagrant
```

## Supprimer un Utilisateur

Créez le fichier **user\_remove.pp** :

```
vagrant@ubuntu-xenial:~$ vi user_remove.pp
vagrant@ubuntu-xenial:~$ cat user_remove.pp
user { 'trainee':
  ensure => absent,
}
```

Appliquez ce manifest avec puppet :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply user_remove.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: /Stage[main]/Main/User[trainee]/ensure: removed
Notice: Applied catalog in 0.19 seconds
```

Contrôlez la présence de **trainee** dans le fichier **/etc/passwd** :

```
vagrant@ubuntu-xenial:~$ tail /etc/passwd
uidd:x:108:112::/run/uidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000:,,,:/home/vagrant:/bin/bash
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

```
ntp:x:112:116::/home/ntp:/bin/false
```

## LAB #14 - La Ressource cron

Consultez le fichier `~/puppet/manifests/run-puppet.pp` :

```
vagrant@ubuntu-xenial:~$ cat puppet/manifests/run-puppet.pp
# Set up regular Puppet runs
file { ['/usr/local/bin/run-puppet':
  source => '/etc/puppetlabs/code/environments/production/files/run-puppet.sh',
  mode   => '0755',
}

cron { 'run-puppet':
  command => '/usr/local/bin/run-puppet',
  hour    => '*',
  minute  => '*/15',
}
```

Rappelez-vous que ce manifest avait créé le fichier crontab de **root** :

```
vagrant@ubuntu-xenial:~$ sudo crontab -l
# HEADER: This file was autogenerated at 2018-05-27 07:26:05 +0000 by puppet.
# HEADER: While it can still be managed manually, it is definitely not recommended.
# HEADER: Note particularly that the comments starting with 'Puppet Name' should
# HEADER: not be deleted, as doing so could cause duplicate cron jobs.
# Puppet Name: run-puppet
*/15 * * * * /usr/local/bin/run-puppet
```

Consultez l'exemple suivant :

```
cron { 'cron example':
```

```
command    => '/bin/date +%F',  
user       => 'ubuntu',  
environment => ['MAILTO=admin@example.com', 'PATH=/bin'],  
hour       => '0',  
minute     => '0',  
weekday    => ['Saturday', 'Sunday'],  
}
```

### **L'Attribut user**

Cet attribut spécifie qui exécute le cron job. Si ce n'est pas spécifié, le job est exécuté par root.

### **L'Attribut environment**

Cet attribut permet de définir des variables système pour cron.

### **L'Attribut weekday**

Cet attribut permet de spécifier les jours de la semaine.

### **L'Attribut monthday**

Cet attribut permet de spécifier le jour du mois entre 1 et 31.

### **La Fonction fqdn\_rand**

Afin d'éviter à ce que les mêmes cron jobs ne s'exécutent sur tous les noeuds en même temps, Puppet utilise la fonction **fqdn\_rand**.

---

Cette fonction fournit un nombre aléatoire de 0 à 23 dans le cas de l'exemple ci-dessous :

```
cron { 'run daily backup':  
  command => '/usr/local/bin/backup',  
  minute  => '0',  
  hour    => fqdn_rand(24, 'run daily backup'),  
}  
  
cron { 'run daily backup sync':  
  command => '/usr/local/bin/backup_sync',  
  minute  => '0',  
  hour    => fqdn_rand(24, 'run daily backup sync'),  
}
```

## LAB #15 - La Ressource exec

La ressource **exec** permet d'exécuter toute commande sur un noeud. Ce qui peut être exécuté en ligne de commande peut être exécuté en utilisant **exec**.

Considérez l'exemple suivant concernant la compilation d'un logiciel imaginaire :

```
exec { 'install-cat-picture-generator':  
  cwd      => '/tmp/cat-picture-generator',  
  command  => '/tmp/cat-picture-generator/configure && /usr/bin/make install',  
  creates  => '/usr/local/bin/cat-picture-generator',  
}
```

### L'Attribut exec

Cet attribut peut être ce que vous voulez mais doit être unique.

## L'Attribut `cwd`

Cet attribut fixe le répertoire de travail.

## L'Attribut `command`

Cet attribut indique la ou les commandes à exécuter. Il faut indiquer le chemin complet.

## L'Attribut `creates`

Cet attribut indique un fichier qui sera présent après l'exécution de la commande. Si ce fichier existe, Puppet n'exécutera pas la commande une deuxième fois.

## L'Attribut `user`

Créez maintenant le fichier **exec\_user.pp** :

```
vagrant@ubuntu-xenial:~$ vi exec_user.pp
vagrant@ubuntu-xenial:~$ cat exec_user.pp
exec { 'say-hello':
  command => '/bin/echo Hello, this is `whoami` >/tmp/hello-ubuntu.txt',
  user     => 'ubuntu',
  creates  => '/tmp/hello-ubuntu.txt',
}
```

Dans le cas de ce manifest, la commande sera exécutée par l'utilisateur **ubuntu** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply exec_user.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.20 seconds
```

```
Notice: /Stage[main]/Main/Exec[say-hello]/returns: executed successfully
Notice: Applied catalog in 0.15 seconds
```

```
vagrant@ubuntu-xenial:~$ cat /tmp/hello-ubuntu.txt
Hello, this is ubuntu
```



**Important** - Si l'utilisateur n'est pas spécifié, la commande sera exécutée par **root**.

## L'Attribut **onlyif**

Certaines commandes ne doivent être exécutées que dans certaines conditions. Pour cette situation, Puppet possède l'attribut **onlyif** :

```
exec { 'process-incoming-cat-pictures':
  command => '/usr/local/bin/cat-picture-generator --import /tmp/incoming/*',
  onlyif  => '/bin/ls /tmp/incoming/*',
}
```



**Important** - L'attribut **onlyif** spécifie la commande qui est exécuté en premier. Dans le manifest ci-dessus, si le code retour de la commande **/bin/ls /tmp/incoming/\*** est autre que 0, la commande spécifiée par l'attribut **command** ne sera **pas** exécutée.

## L'Attribut **unless**

L'attribut **unless** fait l'opposé de l'attribut **onlyif** à savoir, la commande sera toujours exécutée sauf si le code retour de la commande spécifiée par l'attribut **unless** est un 0.

### L'Attribut refreshonly

Considérez l'exemple suivant :

```
package { 'postfix':  
  ensure => installed,  
}  
  
file { ['/etc/aliases':  
  content => 'root: admin@example.com',  
  notify  => Exec['newaliases'],  
}  
  
exec { 'newaliases':  
  command      => '/usr/bin/newaliases',  
  refreshonly => true,  
}
```

Dans les cas de ce manifest, la commande **/usr/bin/newaliases** n'est exécutée que dans le cas où le fichier **/etc/aliases** a été modifié.

### L'Attribut logoutput

Cet attribut permet de modifier la journalisation de la commande.

L'attribut peut prendre trois valeurs :

- **true**,
  - la sortie de la commande est toujours journalisée,



- **false**,
  - la sortie de la commande n'est jamais journalisée,
- **on failure**,
  - la sortie de la commande n'est journalisée que dans le cas d'une erreur.

Par exemple :

```
exec { 'newaliases':  
  command => '/usr/bin/newaliases',  
  logoutput => true,  
}
```

### L'Attribut timeout

Par défaut, Puppet attend 300 secondes pour qu'une commande se termine. Une valeur de 0 permet la commande de s'exécuter à l'infini.

Pour toute autre valeur, il convient d'utiliser l'attribut **timeout**.

## Variables, Expressions, Expressions Conditionnelles et Facts

### LAB #16 - Variables

#### Variables Simples

Une variable sous Puppet est une façon de donner un nom à une valeur. Par exemple :

```
$php_package = 'php7.0-cli'  
  
package { $php_package:  
  ensure => installed,
```

```
}
```



**Important** - Le nom d'une variable doit commencer par le caractère **\$** puis par une lettre miniscule ou un underscore.

Le contenu de la variable peut être :

- une chaîne,
- un chiffre,
- un booléen.

```
$my_name = 'Zaphod Beeblebrox'  
$answer = 42  
$scheduled_for_demolition = true
```



**Important** - La valeur d'un variable booléenne doit être **true** ou **false**.

Les variables peuvent ensuite être appelées ainsi :

```
vagrant@ubuntu-xenial:~$ vi string_interpolation.pp  
vagrant@ubuntu-xenial:~$ cat string_interpolation.pp  
$my_name = 'John'  
notice("Hello, ${my_name}! It's great to meet you!")  
vagrant@ubuntu-xenial:~$ sudo puppet apply string_interpolation.pp  
Notice: Scope(Class[main]): Hello, John! It's great to meet you!  
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds  
Notice: Applied catalog in 0.07 seconds
```



**Important** - Notez l'utilisation de l'attribut **notice** qui imprime la chaîne et le contenu de la variable à l'écran.

## Tableaux

Un tableau est une groupe de valeurs. Par exemple :

```
$heights = [193, 120, 181, 164, 172]

$first_height = $heights[0]
```

Il est possible de référencer une des valeurs en utilisant son index. Par exemple **\$heights[0]=193** et **\$heights[4]=172**.

Dans l'exemple de l'utilisation d'un tableaux ci-dessous, le tableau contient une liste de dépendances qui doivent être installées sur le système :

```
vagrant@ubuntu-xenial:~$ vi resource_array.pp
vagrant@ubuntu-xenial:~$ cat resource_array.pp
$dependencies = [
  'php7.0-cgi',
  'php7.0-cli',
  'php7.0-common',
  'php7.0-gd',
  'php7.0-json',
  'php7.0-mcrypt',
  'php7.0-mysql',
  'php7.0-soap',
]

package { $dependencies:
  ensure => installed,
```

```
}
```

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [107 kB]
Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Ign:4 http://apt.puppetlabs.com xenial InRelease
Hit:5 http://apt.puppetlabs.com xenial Release
Get:6 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial-updates/main Sources [306 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [783 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [631 kB]
Fetched 2,042 kB in 3s (533 kB/s)
Reading package lists... Done

vagrant@ubuntu-xenial:~$ sudo puppet apply resource_array.pp
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.52 seconds
Notice: /Stage[main]/Main/Package[php7.0-cgi]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-gd]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-mcrypt]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-mysql]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-soap]/ensure: created
Notice: Applied catalog in 91.09 seconds
```

Comme vous pouvez constater, Puppet a créé une ressource pour chaque paquet qui doit être installé :

```
Notice: /Stage[main]/Main/Package[php7.0-cgi]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-gd]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-mcrypt]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-mysql]/ensure: created
Notice: /Stage[main]/Main/Package[php7.0-soap]/ensure: created
```

## Hashes

Un **Hash** est comme un Tableau, à l'exception du fait que chaque élément a un nom appelé une **clef** :

```
vagrant@ubuntu-xenial:~$ cat variable_hash.pp
$heights = {
  'john'    => 193,
  'rabiah'  => 120,
  'abigail' => 181,
  'melina'  => 164,
  'sumiko'  => 172,
}

notice("John's height is ${heights['john']}cm.")
```

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply variable_hash.pp
Notice: Scope(Class[main]): John's height is 193cm.
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.07 seconds
```

Un Hash peut être utilisé pour définir les attributs d'une ressource :

```
vagrant@ubuntu-xenial:~$ vi hash_attributes.pp
vagrant@ubuntu-xenial:~$ cat hash_attributes.pp
$attributes = {
  'owner' => 'ubuntu',
  'group' => 'ubuntu',
  'mode'  => '0644',
}

file { ['/tmp/test':
```

```
ensure => present,  
*      => $attributes,  
}
```



**Important** - Le caractère \* informe Puppet d'utiliser ce Hash comme une liste d'attributs.

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply hash_attributes.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds  
Notice: /Stage[main]/Main/File[/tmp/test]/ensure: created  
Notice: Applied catalog in 0.08 seconds  
  
vagrant@ubuntu-xenial:~$ ls -l /tmp/test  
-rw-r--r-- 1 ubuntu ubuntu 0 May 28 15:59 /tmp/test
```

## LAB #17 - Expressions

### Expressions Mathématiques

Puppet peut gérer des expressions mathématiques. Créez donc le fichier **expression\_numeric.pp** :

```
vagrant@ubuntu-xenial:~$ vi expression_numeric.pp  
vagrant@ubuntu-xenial:~$ cat expression_numeric.pp  
$value = (17 * 8) + (12 / 4) - 1  
notice($value)
```

L'application de ce manifest affichera le résultat de l'expression :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply expression_numeric.pp
Notice: Scope(Class[main]): 138
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.08 seconds
Notice: Applied catalog in 0.07 seconds
```

## Expression Booléennes

Puppet sait aussi gérer des expression booléennes, c'est-à-dire des expressions qui produisent come résultat soit **vrai** soit **faux**. Créez donc le fichier **expression\_boolean.pp** :

```
vagrant@ubuntu-xenial:~$ vi expression_boolean.pp
vagrant@ubuntu-xenial:~$ cat expression_boolean.pp
notice(9 < 10)
notice(11 > 10)
notice(10 >= 10)
notice(10 <= 10)
notice('foo' == 'foo')
notice('foo' in 'foobar')
notice('foo' in ['foo', 'bar'])
notice('foo' in { 'foo' => 'bar' })
notice('foo' =~ /oo/)
notice('foo' =~ String)
notice(1 != 2)
```

Les opérateurs utilisés dans le fichier **expression\_boolean.pp** sont les suivants :

Opérateur	Description
==	Egal
!=	Pas égal
>, >=, < et <=	Sauvegarde le document entier et non uniquement le script
A in B	A est une sous-chaîne de B, A est un élément du tableau B ou A est une clef du Hash B

Opérateur	Description
A =~ B	A correspond à l'expression régulière B ou A est une valeur qui correspond au type de données de B ( p.e. <b>'trainee' =~ String</b> est vrai )
Valeur =~ ER	Si valeur égal l'ER alors vrai
/a+/	Correspond à a aa aaa etc

Les expressions dans le fichier **expression\_boolean.pp** donnent toutes un résultat de **vrai**, comme vous pouvez voir en appliquant le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply expression_boolean.pp
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.09 seconds
Notice: Applied catalog in 0.08 seconds
```

## Expressions Régulières

Les expression régulières dans Puppet doivent être entourées du caractère /. Créez don le fichier **regex.pp** :

```
vagrant@ubuntu-xenial:~$ vi regex.pp
vagrant@ubuntu-xenial:~$ cat regex.pp
$candidate = 'foo'
notice($candidate =~ /foo/) # literal
notice($candidate =~ /f/)   # substring
```



```
notice($candidate =~ /f.*/) # f followed by zero or more characters
notice($candidate =~ /f.o/) # f, any character, o
notice($candidate =~ /fo+/) # f followed by one or more 'o's
notice($candidate =~ /[fgh]oo/) # f, g, or h followed by 'oo'
```

Comme la valeur contenue dans **\$candidate** correspond à chaque expression régulière dans le manifest, le résultat est toujours **vrai** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply regex.pp
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Scope(Class[main]): true
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.08 seconds
Notice: Applied catalog in 0.08 seconds
```



**Important** - La syntaxe des expressions régulières Puppet est la même que celle de **Ruby**. Vous pouvez trouver plus d'informations concernant cette syntaxe à l'adresse suivante :

<http://ruby-doc.org/core/Regexp.html>.

## Expressions Conditionnelles

Puppet peut utiliser des expressions. Créez le fichier **if.pp** :

```
vagrant@ubuntu-xenial:~$ vi if.pp
vagrant@ubuntu-xenial:~$ cat if.pp
$install_perl = true
if $install_perl {
```

```
package { 'perl':  
  ensure => installed,  
}  
} else {  
  package { 'perl':  
    ensure => absent,  
  }  
}
```

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply if.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.55 seconds  
Notice: Applied catalog in 0.17 seconds
```

Contrôlez maintenant si la commande **dpkg --get-selections** retourne une valeur d'**install** pour le paquet **perl** :

```
vagrant@ubuntu-xenial:~$ dpkg --get-selections | grep perl  
libapparmor-perl          install  
liberror-perl             install  
liblocale-gettext-perl    install  
libperl5.22:amd64         install  
libtext-charwidth-perl    install  
libtext-iconv-perl        install  
libtext-wrap18n-perl      install  
perl                      install  
perl-base                 install  
perl-modules-5.22         install
```

Modifiez maintenant la valeur de **\$install\_perl** dans le fichier **if.pp** :

```
vagrant@ubuntu-xenial:~$ vi if.pp  
vagrant@ubuntu-xenial:~$ cat if.pp  
$install_perl = false
```

```
if $install_perl {  
  package { 'perl':  
    ensure => installed,  
  }  
} else {  
  package { 'perl':  
    ensure => absent,  
  }  
}
```

Appliquez maintenant le manifest modifié :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply if.pp  
Notice: Compiled catalog for ubuntu-xenial in environment production in 1.56 seconds  
Notice: /Stage[main]/Main/Package[perl]/ensure: removed  
Notice: Applied catalog in 11.00 seconds
```

Contrôlez maintenant si la commande **dpkg --get-selections** retourne une valeur d'**deinstall** pour le paquet **perl** :

```
vagrant@ubuntu-xenial:~$ dpkg --get-selections | grep perl  
libapparmor-perl          install  
liblocale-gettext-perl    install  
libperl5.22:amd64         install  
libtext-charwidth-perl    install  
libtext-iconv-perl        install  
libtext-wrap18n-perl      install  
perl                      deinstall  
perl-base                 install  
perl-modules-5.22         install
```

Une expression conditionnelle utilisant un **if** ne permet que deux choix. S'il faut plus de choix, il convient d'utiliser **case**. Créez donc le fichier **case.pp** :

```
vagrant@ubuntu-xenial:~$ vi case.pp  
vagrant@ubuntu-xenial:~$ cat case.pp
```

```
$webserver = 'nginx'
case $webserver {
  'nginx': {
    notice("Looks like you're using Nginx! Good choice!")
  }
  'apache': {
    notice("Ah, you're an Apache fan, eh?")
  }
  'IIS': {
    notice('Well, somebody has to.')
  }
  default: {
    notice("I'm not sure which webserver you're using!")
  }
}
```

En appliquant ce manifest, Puppet retourne la chaîne **Looks like you're using Nginx! Good choice!** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply case.pp
Notice: Scope(Class[main]): Looks like you're using Nginx! Good choice!
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.07 seconds
```

## LAB #18 - Facts

Les manifests de Puppet ont souvent besoin de connaître quelque chose concernant le système, par exemple :

- Le nom d'hôte,
- L'adresse IP,
- La version du système d'exploitation.

Sous Puppet, le mécanisme qui permet d'obtenir ces informations s'appelle **Facteur** et chaque information fournie par Facteur s'appelle un **Fact**.

## Facts dans un Hash

Créez donc le fichier **facts\_hash.pp** :

```
vagrant@ubuntu-xenial:~$ vi facts_hash.pp
vagrant@ubuntu-xenial:~$ cat facts_hash.pp
notice($facts['kernel'])
```



**Important** - Dans ce manifest est utilisé la variable **\$facts**. Le fact recherché est fourni en tant que clef - dans notre exemple **kernel**.

En appliquant ce manifest, vous obtiendrez donc le type de noyau du système d'exploitation :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply facts_hash.pp
Notice: Scope(Class[main]): Linux
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: Applied catalog in 0.08 seconds
```

L'utilisation de la commande **facter** permet de se renseigner sur les Facts disponibles, par exemple ceux concernant le système d'exploitation :

```
vagrant@ubuntu-xenial:~$ facter os
...
"operatingsystem": "Ubuntu",
"operatingsystemmajrelease": "16.04",
"operatingsystemrelease": "16.04",
"os": {
  "architecture": "amd64",
  "distro": {
    "codename": "xenial",
    "description": "Ubuntu 16.04.4 LTS",
```

```
    "id": "Ubuntu",
    "release": {
      "full": "16.04",
      "major": "16.04"
    },
    "family": "Debian",
    "hardware": "x86_64",
    "name": "Ubuntu",
    "release": {
      "full": "16.04",
      "major": "16.04"
    },
    "selinux": {
      "enabled": false
    }
  },
  ...
```



**Important** - Comme vous pouvez constater, la sortie est sous forme de Hashs multiples.

Pour accéder à une valeur dans un Hash, il convient de spécifier la clef contenant la valeur entre les caractères [ ]. Pour spécifier une clef dans une clef, il convient de nouveau à la spécifier entre des caractères [ ]. Créez le fichier **facts\_architecture.pp** :

```
vagrant@ubuntu-xenial:~$ vi facts_architecture.pp
vagrant@ubuntu-xenial:~$ cat facts_architecture.pp
notice($facts['os']['architecture'])
```

Appliquez le manifest et vous obtenez la valeur d'**architecture** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply facts_architecture.pp
Notice: Scope(Class[main]): amd64
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.08 seconds
```

Cette même technique est applicable quelque soit le profondeur du Hash. Pour obtenir la valeur du **codename**, créez le fichier **facts\_distro\_codename.pp** :

```
vagrant@ubuntu-xenial:~$ vi facts_distro_codename.pp
vagrant@ubuntu-xenial:~$ cat facts_distro_codename.pp
notice($facts['os']['distro']['codename'])
```

Appliquez le manifest et vous obtenez la valeur **xenial** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply facts_distro_codename.pp
Notice: Scope(Class[main]): xenial
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: Applied catalog in 0.07 seconds
```

Pour obtenir la valeur du numéro de version de la distribution, créez le fichier **facts\_major** :

```
vagrant@ubuntu-xenial:~$ vi facts_major.pp
vagrant@ubuntu-xenial:~$ cat facts_major.pp
notice($facts['os']['release']['major'])
```

Appliquez le manifest et vous obtenez la valeur **16.04** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply facts_major.pp
Notice: Scope(Class[main]): 16.04
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: Applied catalog in 0.07 seconds
```

Pour obtenir le nom d'hôte, le FQDN et l'adresse IP de la machine, créez le fichier **fact\_networking.pp** :

```
vagrant@ubuntu-xenial:~$ cat fact_networking.pp
notice("My hostname is ${facts['hostname']}")
notice("My FQDN is ${facts['fqdn']}")
notice("My IP is ${facts['networking']['ip']}")
```

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply fact_networking.pp
Notice: Scope(Class[main]): My hostname is ubuntu-xenial
Notice: Scope(Class[main]): My FQDN is ubuntu-xenial
Notice: Scope(Class[main]): My IP is 10.0.2.15
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.08 seconds
```

## Facts dans une Expression

Il est aussi possible de référencer un Fact dans une expression. Créez don le fichier **fact\_ip.pp** :

```
vagrant@ubuntu-xenial:~$ vi fact_ip.pp
vagrant@ubuntu-xenial:~$ cat fact_ip.pp
if $facts['os']['selinux']['enabled'] {
  notice('SELinux is enabled')
} else {
  notice('SELinux is disabled')
}
```

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply fact_ip.pp
Notice: Scope(Class[main]): SELinux is disabled
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.06 seconds
Notice: Applied catalog in 0.07 seconds
```



Facts permettent la configuration dynamique de variables pour les applications. Dans le cas suivant, on va configurer la variable MariaDB **innodb\_buffer\_pool\_size**. Créez donc le fichier **fact\_memory.pp** :

```
vagrant@ubuntu-xenial:~$ vi fact_memory.pp
vagrant@ubuntu-xenial:~$ cat fact_memory.pp
$buffer_pool = $facts['memory']['system']['total_bytes'] * 3/4
notice("innodb_buffer_pool_size=${buffer_pool}")
```

En appliquant ce manifest la valeur de la directive **innodb\_buffer\_pool\_size** est fixée à **75%** de la mémoire RAM totale :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply fact_memory.pp
Notice: Scope(Class[main]): innodb_buffer_pool_size=780174336
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.08 seconds
```

## Facts Externes

Facts Externes sont des Facts créés par l'Administrateur. Puppet cherche des Facts Externes dans le répertoire **/opt/puppetlabs/facter/facts.d/**. Créez le fichier **fact\_external.txt** et copiez-le dans le répertoire **/opt/puppetlabs/facter/facts.d/** :

```
vagrant@ubuntu-xenial:~$ vi fact_external.txt
vagrant@ubuntu-xenial:~$ cat fact_external.txt
cloud=aws
vagrant@ubuntu-xenial:~$ sudo cp fact_external.txt /opt/puppetlabs/facter/facts.d/
```

Utilisez ensuite la commande **facter** pour vérifier que le Fact fonctionne :

```
vagrant@ubuntu-xenial:~$ sudo facter cloud
aws
```

Dans un manifest, un Fact Externe est référencé exactement de la même façon qu'un Fact Interne. Créez le fichier **fact\_cloud.pp** :

```
vagrant@ubuntu-xenial:~$ vi fact_cloud.pp
vagrant@ubuntu-xenial:~$ cat fact_cloud.pp
case $facts['cloud'] {
  'aws': {
    notice('This is an AWS cloud server ')
  }
  'gcp': {
    notice('This is a Google cloud server')
  }
  default: {
    notice("I'm not sure which cloud I'm in!")
  }
}
```

Appliquez le manifest pour obtenir le résultat :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply fact_cloud.pp
Notice: Scope(Class[main]): This is an AWS cloud server
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.07 seconds
Notice: Applied catalog in 0.08 seconds
```

## Facts Exécutables

Facts Externes ne sont pas uniquement des fichiers textes statiques. Ils peuvent aussi être la sortie d'un script ou d'une commande. Créez le script **date.sh** et copiez-le dans le répertoire **/opt/puppetlabs/facter/facts.d/** :

```
vagrant@ubuntu-xenial:~$ cat date.sh
#!/bin/bash
echo "date=`date +%F`"
vagrant@ubuntu-xenial:~$ sudo cp date.sh /opt/puppetlabs/facter/facts.d/
```

Rendez le script exécutable et vérifiez avec la commande **facter** que le Fact fonctionne :

```
vagrant@ubuntu-xenial:~$ sudo chmod a+x /opt/puppetlabs/facter/facts.d/date.sh
vagrant@ubuntu-xenial:~$ sudo facter date
2018-06-01
```

## LAB #19 - Iteration

Iteration permet d'économiser du code.

### Iteration et Tableaux

Créez le fichier **iteration\_simple.pp**

```
vagrant@ubuntu-xenial:~$ vi iteration_simple.pp
vagrant@ubuntu-xenial:~$ cat iteration_simple.pp
file { ['/usr/local/bin/task1':
  content => "echo I am task1\n",
  mode    => '0755',
}

file { ['/usr/local/bin/task2':
  content => "echo I am task2\n",
  mode    => '0755',
}

file { ['/usr/local/bin/task3':
  content => "echo I am task3\n",
  mode    => '0755',
}
```



**Important** - Dans ce manifest il y a trois ressources presque identiques,



différenciées uniquement par le numéro de tâche ( **task1**, **task2**, **task3** ).

Créez maintenant le fichier **iteration\_each.pp** :

```
$tasks = ['task1', 'task2', 'task3']
$tasks.each | $task | {
  file { ["/usr/local/bin/${task}"]:
    content => "echo I am ${task}\n",
    mode     => '0755',
  }
}
```



**Important** - Dans ce manifest la fonction **each** crée une boucle qui prend pour chaque exécution une des valeurs du tableau **\$tasks** ( **task1**, **task2**, **task3** ).

La forme du boucle **each** est :

```
ARRAY.each | ELEMENT | {
  BLOCK
}
```



**Important** - Ici ARRAY est un tableau et ELEMENT est le nom d'une variable qui contiendra chaque valeur du tableau à tour de rôle. BLOCK est du code Puppet qui appelle l'ELEMENT.

## Iteration et Hashes

Créez le fichier suivant :

```
vagrant@ubuntu-xenial:~$ vi iteration_hash.pp
vagrant@ubuntu-xenial:~$ cat iteration_hash.pp
$nics = $facts['networking']['interfaces']
$nics.each | String $interface, Hash $attributes | {
  notice("Interface ${interface} has IP ${attributes['ip']}")
}
```

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply iteration_hash.pp
Notice: Scope(Class[main]): Interface enp0s3 has IP 10.0.2.15
Notice: Scope(Class[main]): Interface lo has IP 127.0.0.1
Notice: Compiled catalog for ubuntu-xenial in environment production in 0.10 seconds
Notice: Applied catalog in 0.08 seconds
```

La liste des interfaces réseau retournée est un Hash nommé **\$nics**.

## Hiera

### Préparation

#### Sous Debian/Ubuntu

```
hnorris@hnorris-E200HA:~$ cd puppet-beginners-guide-3
hnorris@hnorris-E200HA:~/puppet-beginners-guide-3$ vagrant destroy
hnorris@hnorris-E200HA:~/puppet-beginners-guide-3$ scripts/start_vagrant.sh
```

```
hnorris@hnorris-E200HA:~/puppet-beginners-guide-3$ vagrant ssh
```

## Sous Windows

Exécutez **cmd** et saisissez les commandes suivantes :

```
Microsoft Windows [version 10.0.16299.431]
(c) 2017 Microsoft Corporation. Tous droits réservés.

C:\Users\hnorris>cd puppet-beginners-guide-3

C:\Users\hnorris\puppet-beginners-guide-3>vagrant destroy

C:\Users\hnorris\puppet-beginners-guide-3>cd scripts

C:\Users\hnorris\puppet-beginners-guide-3\scripts>start_vagrant.sh

C:\Users\hnorris\puppet-beginners-guide-3\scripts>vagrant ssh
```

## Présentation

Le manifest suivant stipule la version de l'agent Puppet qui doit être installé sur un noeud :

```
package { 'puppet-agent':
  ensure => '5.2.0-1xenial',
}
```

Quand une mise à jour devient disponible, ce code doit être trouvé et modifié sur l'ensemble des noeuds. La multiplication de cette tâche par tous les paquets référencés par les manifests mène à un travail titanesque, compliqué et fastidieux.

Qui plus est, la mise à jour des données dans les manifests ne concerne pas uniquement les paquets mais aussi, par exemple, les :

- cron jobs,
- adresses email,
- URLs des fichiers,
- données des monitorings,
- valeurs des configurations telles la quantité de mémoire allouée à un serveur de base de données.

Hiera permet de gérer les données indépendamment du code Puppet et de spécifier des valeurs différentes par noeud en fonction, par exemple, du nom d'hôte ou du système d'exploitation. Hiera stocke ces informations dans de simples fichiers texte ayant une extension **.yaml** (YAML Ain't Markup Language).

Puppet peut ensuite consulter les informations en utilisant **lookup** :

```
file { lookup('backup_path', String):  
  ensure => directory,  
}
```



**Important - lookup** a besoin du nom de la clef Hiera, par exemple **backup\_path**, ainsi que le type de données, soit **String** dans l'exemple ci-dessus.

Hiera est configuré par un fichier par **environnement**, par exemple **/etc/puppetlabs/environments/production/hiera.yaml** ainsi qu'un fichier global **/etc/puppetlabs/puppet/hiera.yaml**

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/puppet/hiera.yaml  
---  
# Hiera 5 Global configuration file  
  
version: 5  
  
# defaults:  
#   data_hash: yaml_data
```

```
# hierarchy:
#   - name: Common
#     data_hash: yaml_data
hierarchy: []
```

Prenons le cas d'un fichier de configuration hiera.yaml minimalist :

```
---
version: 5

defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:
  - name: "Common defaults"
    path: "common.yaml"
```

Ce fichier commence par un ligne contenant les caractères `---`. Cette syntaxe indique le début d'un nouveau document YAML. La ligne la plus importante dans ce fichier est **datadir: data**. Cette ligne indique à Hiera où se trouvent ses fichiers de données. La section **hierarchy** stipule que Hiera doit rechercher ses données dans le fichier **common.yaml**.

Par convention, le répertoire **data** est un sous-répertoire du répertoire où sont stockés les manifests de Puppet :

```
vagrant@ubuntu-xenial:~$ ls /etc/puppetlabs/code/environments/production
data  environment.conf  hiera.yaml  manifests  modules
```

## LAB #20 - Environnements

Un environnement est un répertoire contenant :

- un fichier de configuration Hiera,
- un jeu de manifests Puppet.



Les répertoires d'environnement se trouvent dans le répertoire **/etc/puppetlabs/code/environments/**. L'environnement par défaut est **production** mais il est possible de stipuler un autre environnement en utilisant l'option **-environment** :

```
vagrant@ubuntu-xenial:~$ sudo puppet lookup --environment pbg test
--- This is a test
...
```

Créez le fichier **lookup2.pp** :

```
vagrant@ubuntu-xenial:~$ vi lookup2.pp
vagrant@ubuntu-xenial:~$ cat lookup2.pp
notice("Apache is set to use ${lookup('apache_worker_factor', Integer)} workers")

unless lookup('apparmor_enabled', Boolean) {
  exec { 'apt-get -y remove apparmor': }
}

notice('dns_allow_query enabled: ', lookup('dns_allow_query', Boolean))
```

Appliquez ce manifest dans l'environnement **pbg** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment pbg lookup2.pp
Notice: Scope(Class[main]): Apache is set to use 100 workers
Notice: Scope(Class[main]): dns_allow_query enabled: true
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.18 seconds
Notice: Applied catalog in 0.03 seconds
```

Notez que la valeur retournée pour le nombre de workers d'apache est **100**. Cette valeur est stipulée dans le fichier **/etc/puppetlabs/code/environments/pbg/data/common.yaml** :

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/code/environments/pbg/data/common.yaml
---
test: 'This is a test'
consul_node: true
```

```
apache_worker_factor: 100
apparmor_enabled: true
syslog_server: '10.170.81.32'
monitor_ips:
  - '10.179.203.46'
  - '212.100.235.160'
  - '10.181.120.77'
  - '94.236.56.148'
cobbler_config:
  manage_dhcp: true
  pxe_just_once: true
domain: 'bitfieldconsulting.com'
servername: 'www.bitfieldconsulting.com'
port: 80
docroot: '/var/www/bitfieldconsulting.com'
dns_allow_query: true
backup_retention_days: 10
backup_path: "/backup/%{facts.hostname}"
ips:
  home: '130.190.0.1'
  office1: '74.12.203.14'
  office2: '95.170.0.75'
firewall_allow_list:
  - "%{lookup('ips.home')}}"
  - "%{lookup('ips.office1')}}"
  - "%{lookup('ips.office2')}}"
vpn_allow_list: "%{alias('firewall_allow_list')}}"
cms_parameters:
  static:
    sites_root: '/var/www/sites'
    assets_root: 'files'
    web_root: 'public_html'
  laravel:
    sites_root: '/var/www/sites'
```

```
assets_root: 'public_html/files'
web_root: 'current/public'
force_www_rewrite:
  comment: "Force WWW"
  rewrite_cond: "%{literal('%')}{HTTP_HOST} !^www\\. [NC]"
  rewrite_rule: "^(.*)$ https://www.%{literal('%')}{HTTP_HOST}%{literal('%')}{REQUEST_URI} [R=301,L]"
users:
  - 'katy'
  - 'lark'
  - 'bridget'
  - 'hsing-hui'
  - 'charles'
users2:
  'katy':
    ensure: present
    uid: 1900
    shell: '/bin/bash'
  'lark':
    ensure: present
    uid: 1901
    shell: '/bin/sh'
  'bridget':
    ensure: present
    uid: 1902
    shell: '/bin/bash'
  'hsing-hui':
    ensure: present
    uid: 1903
    shell: '/bin/sh'
  'charles':
    ensure: present
    uid: 1904
    shell: '/bin/bash'
mysql::server::root_password: 'hairline-quotient-inside-tableful'
```

```
mysql::server::remove_default_accounts: true
apache::default_vhost: false
pbg_ntp_params::version: 'latest'
pbg_ntp_params2::start_at_boot: true
pbg_ntp_params2::version: 'latest'
pbg_ntp_params2::service_state: 'running'
```

Appliquez ce manifest dans l'environnement **production** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply lookup2.pp
Error: Function lookup() did not find a value for the name 'apache_worker_factor'
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment production lookup2.pp
Error: Function lookup() did not find a value for the name 'apache_worker_factor'
```

Cette erreur est due au fait que la donnée n'est pas définie dans l'environnement **production** :

```
vagrant@ubuntu-xenial:~$ ls /etc/puppetlabs/code/environments/production/
data  environment.conf  hiera.yaml  manifests  modules
vagrant@ubuntu-xenial:~$ ls /etc/puppetlabs/code/environments/production/data
vagrant@ubuntu-xenial:~$
```

## LAB #21 -Les Types de Données Hiera

Dans le fichier **/etc/puppetlabs/code/environments/pbg/data/common.yaml**, on peut constater trois types de données :

- les valeurs singulières,
- les valeurs booléennes,
- les Tableaux et les Hash.

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/code/environments/pbg/data/common.yaml
---
test: 'This is a test'
consul_node: true
```

```
apache_worker_factor: 100
apparmor_enabled: true
syslog_server: '10.170.81.32'
monitor_ips:
  - '10.179.203.46'
  - '212.100.235.160'
  - '10.181.120.77'
  - '94.236.56.148'
cobbler_config:
  manage_dhcp: true
  pxe_just_once: true
domain: 'bitfieldconsulting.com'
servername: 'www.bitfieldconsulting.com'
port: 80
docroot: '/var/www/bitfieldconsulting.com'
dns_allow_query: true
backup_retention_days: 10
backup_path: "/backup/%{facts.hostname}"
...
```

Un exemple d'une **valeur singulière** au format **chaîne** ( String ) est la ligne suivante :

```
...
syslog_server: '10.170.81.32'
...
```

Tandis qu'un exemple d'une **valeur singulière** au format **entier** ( Integer ) est la ligne suivante :

```
...
apache_worker_factor: 100
...
```

Un exemple d'une **valeur booléenne** est la ligne suivante :

```
...  
consul_node: true  
...
```



**Important** - Une valeur booléenne doit être soit **true**, soit **false**.

Un exemple d'un **tableau** est :

```
...  
monitor_ips:  
  - '10.179.203.46'  
  - '212.100.235.160'  
  - '10.181.120.77'  
  - '94.236.56.148'  
...
```

Ici la clef **monitor\_ips** contient plusieurs valeurs, chacune sur une ligne et précédée par le caractère -.

Un exemple d'un Hash est :

```
...  
cobbler_config:  
  manage_dhcp: true  
  pxe_just_once: true  
...
```

Chaque clef du Hash possède son propre nom et est indentée par rapport à la première ligne.

Pour mieux comprendre, créez le fichier **lookup\_hash.pp** :

```
vagrant@ubuntu-xenial:~$ vi lookup_hash.pp
```

```
vagrant@ubuntu-xenial:~$ cat lookup_hash.pp
$cobbler_config = lookup('cobbler_config', Hash)
$manage_dhcp = $cobbler_config['manage_dhcp']
$pxe_just_once = $cobbler_config['pxe_just_once']
if $pxe_just_once {
  notice('pxe_just_once is enabled')
} else {
  notice('pxe_just_once is disabled')
}
```

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment pbgi lookup_hash.pp
Notice: Scope(Class[main]): pxe_just_once is enabled
Notice: Compiled catalog for ubuntu-xenial in environment pbgi in 0.20 seconds
Notice: Applied catalog in 0.04 seconds
```

Pour obtenir des données en profondeur dans un Hash, il convient d'utiliser la syntaxe "par points". Créez le fichier **lookup\_hash\_dot.pp** :

```
vagrant@ubuntu-xenial:~$ vi lookup_hash_dot.pp
vagrant@ubuntu-xenial:~$ cat lookup_hash_dot.pp
$web_root = lookup('cms_parameters.static.web_root', String)
notice("web_root is ${web_root}")
```

Appliquez ce manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment pbgi lookup_hash_dot.pp
Notice: Scope(Class[main]): web_root is public_html
Notice: Compiled catalog for ubuntu-xenial in environment pbgi in 0.15 seconds
Notice: Applied catalog in 0.04 seconds
```

La donnée obtenue se trouve dans le fichier **/etc/puppetlabs/code/environments/pbgi/data/common.yaml** dans **cms\_parameters.static.web\_root** :

```
...
cms_parameters:
  static:
    sites_root: '/var/www/sites'
    assets_root: 'files'
    web_root: 'public_html'
...
```

## LAB #22 - Interpolation

L'interpolation est l'action d'introduire dans un texte un élément qui n'était pas dans l'original. Avec Puppet et Hiera ceci concerne :

- les valeurs multiples,
- les comportements de fusion,
- les sources de données.

Jusqu'à maintenant nous n'avons utilisé qu'une seule source de données Hiera à savoir le fichier

**/etc/puppetlabs/code/environments/pbg/data/common.yaml**. En fait il est possible d'utiliser plusieurs sources de données sous la forme de fichiers \*.yaml multiples. Ces sources sont listées dans la section **hierarchy** du fichier **hiera.yaml** de l'environnement :

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/code/environments/pbg/hiera.yaml
---
version: 5

defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:
  - name: "Secret data (encrypted)"
    lookup_key: eyaml_lookup_key
    path: "secret.eyaml"
    options:
```



```
  gpg_gnupghome: '/home/ubuntu/.gnupg'
- name: "AWS resources"
  path: "aws.yaml"
- name: "Host-specific data"
  path: "nodes/%{facts.hostname}.yaml"
- name: "OS release-specific data"
  path: "os/%{facts.os.release.major}.yaml"
- name: "OS distro-specific data"
  path: "os/%{facts.os.distro.codename}.yaml"
- name: "Common defaults"
  path: "common.yaml"
```



**Important** - La priorité des sources est descendante. Si la valeur d'une clef est spécifiée dans deux sources et les valeurs sont différentes, Hiera recherche dans les sources de données dans l'ordre de leur apparition dans le fichier **hiera.yaml** et retourne par défaut la **première** valeur retrouvée.

Si on souhaite qu'Hiera agit autrement que par défaut, il convient de spécifier la méthode de fusion en tant que troisième argument à **lookup**, après le type de données. Créez le fichier **lookup\_merge.pp** :

```
vagrant@ubuntu-xenial:~$ vi lookup_merge.pp
vagrant@ubuntu-xenial:~$ cat lookup_merge.pp
notice(lookup('firewall_allow_list', Array, 'unique'))
```

Dans ce cas, Hiera retourne un tableau de toutes les clefs et les valeurs **uniques** retrouvées.

Quand il concerne des données dans un Hash, les possibilités sont :

- **Hash Merge** - retourne un Hash contenant toutes les clefs et toutes les valeurs correspondantes qui correspondent à la recherche,
- **Shallow Merge** - si Hiera retrouve deux Hash avec le même nom, il retourne uniquement le premier Hash,
- **Deep merge** - permet de considerer le Hash entier et non seulement que le premier niveau.

## Les Sources de Données basées sur des Facts

Revenons au fichier **/etc/puppetlabs/code/environments/pbg/hiera.yaml** :

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/code/environments/pbg/hiera.yaml
---
version: 5

defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:
  - name: "Secret data (encrypted)"
    lookup_key: eyaml_lookup_key
    path: "secret.eyaml"
    options:
      gpg_gnupghome: '/home/ubuntu/.gnupg'
  - name: "AWS resources"
    path: "aws.yaml"
  - name: "Host-specific data"
    path: "nodes/%{facts.hostname}.yaml"
  - name: "OS release-specific data"
    path: "os/%{facts.os.release.major}.yaml"
  - name: "OS distro-specific data"
    path: "os/%{facts.os.distro.codename}.yaml"
  - name: "Common defaults"
    path: "common.yaml"
```

Dans ce fichier, on peut constater les lignes suivantes :

```
...
- name: "Host-specific data"
```

```
    path: "nodes/%{facts.hostname}.yaml"
    ...
```

Ces deux lignes permettent d'avoir une configuration différente par noeud contenue dans un fichier dénommé <nom d'hôte>.yaml.

De même les deux lignes suivantes :

```
    ...
    - name: "OS release-specific data"
      path: "os/%{facts.os.release.major}.yaml"
    ...
```

permettent une configuration différente par version du système d'exploitation.

## LAB #23 -Créer des Ressources avec le Données d'Hiera

Commencez par créer le fichier **hiera\_users.pp** :

```
vagrant@ubuntu-xenial:~$ vi hiera_users.pp
vagrant@ubuntu-xenial:~$ cat hiera_users.pp
lookup('users', Array[String]).each | String $username | {
  user { $username:
    ensure => present,
  }
}
```

Les données utilisées dans ce manifest se trouvent dans le fichier **/etc/puppetlabs/code/environments/pbg/data/common.yaml** :

```
...
users:
  - 'katy'
  - 'lark'
```

```
- 'bridget'  
- 'hsing-hui'  
- 'charles'  
...
```

Appliquez le manifest pour vérifier :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment pbk hiera_users.pp  
Notice: Compiled catalog for ubuntu-xenial in environment pbk in 0.19 seconds  
Notice: /Stage[main]/Main/User[katy]/ensure: created  
Notice: /Stage[main]/Main/User[lark]/ensure: created  
Notice: /Stage[main]/Main/User[bridget]/ensure: created  
Notice: /Stage[main]/Main/User[hsing-hui]/ensure: created  
Notice: /Stage[main]/Main/User[charles]/ensure: created  
Notice: Applied catalog in 0.66 seconds
```

Les données Hiera utilisées par le manifest **hiera\_users.pp** ne sont pas complètes. Créez donc le fichier **hiera\_users2.pp** :

```
vagrant@ubuntu-xenial:~$ vi hiera_users2.pp  
vagrant@ubuntu-xenial:~$ cat hiera_users2.pp  
lookup('users2', Hash, 'hash').each | String $username, Hash $attrs | {  
  user { $username:  
    * => $attrs,  
  }  
}
```

Dans ce cas les données utilisées dans ce manifest se trouvent aussi dans le fichier **/etc/puppetlabs/code/environments/pbk/data/common.yaml** :

```
...  
users2:  
  'katy':  
    ensure: present  
    uid: 1900
```

```
    shell: '/bin/bash'
  'lark':
    ensure: present
    uid: 1901
    shell: '/bin/sh'
  'bridget':
    ensure: present
    uid: 1902
    shell: '/bin/bash'
  'hsing-hui':
    ensure: present
    uid: 1903
    shell: '/bin/sh'
  'charles':
    ensure: present
    uid: 1904
    shell: '/bin/bash'
  ...
```

Appliquez ce manifest pour examiner le résultat :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment pbg hiera_users2.pp
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.18 seconds
Notice: /Stage[main]/Main/User[katy]/uid: uid changed 1002 to 1900
Notice: /Stage[main]/Main/User[katy]/shell: shell changed '' to '/bin/bash'
Notice: /Stage[main]/Main/User[lark]/uid: uid changed 1003 to 1901
Notice: /Stage[main]/Main/User[lark]/shell: shell changed '' to '/bin/sh'
Notice: /Stage[main]/Main/User[bridget]/uid: uid changed 1004 to 1902
Notice: /Stage[main]/Main/User[bridget]/shell: shell changed '' to '/bin/bash'
Notice: /Stage[main]/Main/User[hsing-hui]/uid: uid changed 1005 to 1903
Notice: /Stage[main]/Main/User[hsing-hui]/shell: shell changed '' to '/bin/sh'
Notice: /Stage[main]/Main/User[charles]/uid: uid changed 1006 to 1904
Notice: /Stage[main]/Main/User[charles]/shell: shell changed '' to '/bin/bash'
```

Notice: Applied catalog in 0.80 seconds

Consultez maintenant le fichier **/etc/passwd** :

```
vagrant@ubuntu-xenial:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108:./home/syslog:/bin/false
_apt:x:105:65534:./nonexistent:/bin/false
lxd:x:106:65534:./var/lib/lxd:/bin/false
messagebus:x:107:111:./var/run/dbus:/bin/false
uuidd:x:108:112:./run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534:./var/run/sshd:/usr/sbin/nologin
```

```
pollinate:x:111:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000:::/home/vagrant:/bin/bash
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
katy:x:1900:1002::/home/katy:/bin/bash
lark:x:1901:1003::/home/lark:/bin/sh
bridget:x:1902:1004::/home/bridget:/bin/bash
hsing-hui:x:1903:1005::/home/hsing-hui:/bin/sh
charles:x:1904:1006::/home/charles:/bin/bash
```

## LAB #24 - Gérer des Données Secrètes

Souvent Puppet a besoin d'informations sensibles telles :

- mots de passe,
- clefs privées.

Ces informations ne peuvent pas être stockées dans un repo de GIT car tout le monde y aurait accès !!! La solution à ce problème consiste en crypter les données en utilisant **GnuGP**.

Installez donc **gnupg** et **rng-tools** :

```
vagrant@ubuntu-xenial:~$ sudo apt-get install -y gnupg rng-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (1.4.20-1ubuntu3.1).
The following NEW packages will be installed:
  rng-tools
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 21.9 kB of archives.
After this operation, 139 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial/universe amd64 rng-tools amd64 5-0ubuntu3 [21.9 kB]
```

```

Fetched 21.9 kB in 0s (64.2 kB/s)
Selecting previously unselected package rng-tools.
(Reading database ... 67016 files and directories currently installed.)
Preparing to unpack .../rng-tools_5-0ubuntu3_amd64.deb ...
Unpacking rng-tools (5-0ubuntu3) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for systemd (229-4ubuntu21.2) ...
Processing triggers for ureadahead (0.100.0-19) ...
Setting up rng-tools (5-0ubuntu3) ...
Processing triggers for systemd (229-4ubuntu21.2) ...
Processing triggers for ureadahead (0.100.0-19) ...

```

Générez maintenant une paire de clefs :

```

vagrant@ubuntu-xenial:~$ gpg --gen-key
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/home/vagrant/.gnupg' created
gpg: new configuration file `/home/vagrant/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/vagrant/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/vagrant/.gnupg/secring.gpg' created
gpg: keyring `/home/vagrant/.gnupg/pubring.gpg' created
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.

```



```
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
```

Key is valid for? (0) 0

Key does not expire at all

Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

```
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

Real name: puppet

Email address: puppet@i2tch.co.uk

Comment:

You selected this USER-ID:

```
"puppet <puppet@i2tch.co.uk>"
```

Change (N)ame, (C)omment, (E)mail or (0)key/(Q)uit? o

You need a Passphrase to protect your secret key.

You don't want a passphrase - this is probably a \*bad\* idea!

I will do it anyway. You can change your passphrase at any time, using this program with the option "--edit-key".

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

```
+++++
```

```
.+++++
```

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the

```
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

```
.....+++++
```

```
+++++
```

```
gpg: /home/vagrant/.gnupg/trustdb.gpg: trustdb created
```

```
gpg: key 05205AF4 marked as ultimately trusted
```

```
public and secret key created and signed.
```

```
gpg: checking the trustdb
```

```
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
```

```
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
```

```
pub   2048R/05205AF4 2018-06-03
```

```
      Key fingerprint = 804C 7675 2822 6E73 6C4E  E0D4 1280 D1A2 0520 5AF4
```

```
uid           puppet <puppet@i2tch.co.uk>
```

```
sub   2048R/F72F4F77 2018-06-03
```

La clef est visible :

```
pub   2048R/05205AF4 2018-06-03
```

```
      Key fingerprint = 804C 7675 2822 6E73 6C4E  E0D4 1280 D1A2 0520 5AF4
```

```
uid           puppet <puppet@i2tch.co.uk>
```

```
sub   2048R/F72F4F77 2018-06-03
```

Editez le fichier **/etc/puppetlabs/code/environments/pbg/hiera.yaml** ainsi :

```
...
  gpg_gnupghome: '/home/vagrant/.gnupg'
...
```

Vous obtiendrez :

```
vagrant@ubuntu-xenial:~$ vi /etc/puppetlabs/code/environments/pbg/hiera.yaml
```

```
---
```

```
version: 5
```

```
defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:
  - name: "Secret data (encrypted)"
    lookup_key: eyaml_lookup_key
    path: "secret.eyaml"
    options:
      gpg_gnupghome: '/home/vagrant/.gnupg'
  - name: "AWS resources"
    path: "aws.yaml"
  - name: "Host-specific data"
    path: "nodes/{facts.hostname}.yaml"
  - name: "OS release-specific data"
    path: "os/{facts.os.release.major}.yaml"
  - name: "OS distro-specific data"
    path: "os/{facts.os.distro.codename}.yaml"
  - name: "Common defaults"
    path: "common.yaml"
```

La section de fichier qui permet à Hiera d'utiliser GnuGP est la suivante :

```
...
  - name: "Secret data (encrypted)"
    lookup_key: eyaml_lookup_key
    path: "secret.eyaml"
    options:
      gpg_gnupghome: '/home/vagrant/.gnupg'
...
```

Créez maintenant un fichier pour contenir des données secrètes :

```
vagrant@ubuntu-xenial:~$ sudo touch /etc/puppetlabs/code/environments/pbg/data/secret.eyaml
```

Lors de l'édition de la commande, utilisez la commande **eyaml** qui crypte le contenu lors de l'enregistrement du fichier :

```
vagrant@ubuntu-xenial:~$ sudo /opt/puppetlabs/puppet/lib/ruby/vendor_gems/gems/hiera-eyaml-2.1.0/bin/eyaml edit -  
-gpg-always-trust --gpg-recipients=puppet@i2tch.co.uk /etc/puppetlabs/code/environments/pbg/data/secret.eyaml  
[hiera-eyaml-core] /etc/puppetlabs/code/environment/pbg/data/secret.yaml doesn't exist, editing new file
```

Select an editor. To change later, run 'select-editor'.

1. /bin/ed
2. /bin/nano <---- easiest
3. /usr/bin/vim.basic
4. /usr/bin/vim.tiny

Choose 1-4 [2]: 3



**Important** - Utilisez l'adresse email saisie lors de l'exécution de la commande **gpg -gen-key**.

Vous obtiendrez un fichier comme celui-ci :

```
#| This is eyaml edit mode. This text (lines starting with #| at the top of the  
#| file) will be removed when you save and exit.  
#| - To edit encrypted values, change the content of the DEC(<num>)::PKCS7[!]  
#| block (or DEC(<num>)::GPG[!]).  
#| WARNING: DO NOT change the number in the parentheses.  
#| - To add a new encrypted value copy and paste a new block from the  
#| appropriate example below. Note that:  
#| * the text to encrypt goes in the square brackets  
#| * ensure you include the exclamation mark when you copy and paste  
#| * you must not include a number when adding a new block  
#| e.g. DEC::PKCS7[!] -or- DEC::GPG[!]
```

Editez ce fichier ainsi et sauvegardez-le :

```
#| This is eyaml edit mode. This text (lines starting with #| at the top of the
#| file) will be removed when you save and exit.
#| - To edit encrypted values, change the content of the DEC(<num>)::PKCS7[]!
#|   block (or DEC(<num>)::GPG[]!).
#|   WARNING: DO NOT change the number in the parentheses.
#| - To add a new encrypted value copy and paste a new block from the
#|   appropriate example below. Note that:
#|   * the text to encrypt goes in the square brackets
#|   * ensure you include the exclamation mark when you copy and paste
#|   * you must not include a number when adding a new block
#|   e.g. DEC::PKCS7[]! -or- DEC::GPG[]!
---
test_secret: DEC::GPG[This is a test secret]!
---
```

Vérifiez que le fichier a été crypté :

```
vagrant@ubuntu-xenial:~$ cat /etc/puppetlabs/code/environments/pbg/data/secret.eyaml - -
test_secret:
ENC[GPG,hQEMAwMDMyr3L093AQf/ZVi+EVd2w0bVl76p1WA7wTwFnKx47jE7smq2DI/r/KmDzKisukiSxtbRm4A60FI0f5aZIKuXLHi0KWE64w4Gk
1riBn7GECLRNcKKZjvwZEebsh6qvnPA41kbTNYNNbNIhjDTlTzrIGjM/YvPVi8Fb4LPFJNJtZtXV8l0vYw0YBJZ8jDazFI+VNSLaH8ttiU3LuUZlW
MTHtB3coSCOAUEPXGKYxIrW0c0SUAFHtNSEGAzbikMSEMMeaoHfJCEKHQ5DyitGFL8o1Esl6A0df1/LJiH0YiDtHx/xjvVBtz6uK0MNHUP10eJi6l
KmtxBR1N0wactfqQVvQnbI9chcrbACNJSAch1zZb3AsU5afdJJQ0N5200yDdGDhdetHu2jx6pZKqQUb3i8BcD9X6WVAiva00gcFSEmKigFpdLRpAMV
RNQ0yej10sYLJaQTWgRT3ddEySl4Lw==]
---
```



**Important** - **ENC** indique à Hiera que ce fichier est crypté. GPG indique à Hiera quel type de cryptage.

Vérifiez que Hiera peut lire le secret avec la commande suivante :

```
vagrant@ubuntu-xenial:~$ sudo puppet lookup --environment pbgi test_secret
--- This is a test secret
...
```

Créez maintenant le script **eyaml\_edit.sh** :

```
vagrant@ubuntu-xenial:~$ vi eyaml_edit.sh
vagrant@ubuntu-xenial:~$ cat eyaml_edit.sh
#!/bin/bash
/opt/puppetlabs/puppet/lib/ruby/vendor_gems/gems/hiera-eyaml-2.1.0/bin/eyaml edit --gpg-always-trust --gpg-
recipients=puppet@i2tch.co.uk /etc/puppetlabs/code/environments/pbgi/data/secret.eyaml
```



**Important** - Utilisez l'adresse email saisie lors de l'exécution de la commande **gpg -gen-key**.

Copiez maintenant le fichier **eyaml\_edit.sh** dans le répertoire **/usr/local/bin** :

```
vagrant@ubuntu-xenial:~$ sudo cp eyaml_edit.sh /usr/local/bin
```

Rendez ce script exécutable :

```
vagrant@ubuntu-xenial:~$ ls -l /usr/local/bin/eyaml_edit.sh
-rw-r--r-- 1 root root 205 Jun  3 12:04 /usr/local/bin/eyaml_edit.sh
vagrant@ubuntu-xenial:~$ sudo chmod u+x /usr/local/bin/eyaml_edit.sh
vagrant@ubuntu-xenial:~$ ls -l /usr/local/bin/eyaml_edit.sh
-rwxr--r-- 1 root root 205 Jun  3 12:04 /usr/local/bin/eyaml_edit.sh
```

Pour ajouter un nouveau secret, exécutez simplement le script **eyaml\_edit.sh** :

```
vagrant@ubuntu-xenial:~$ sudo eyaml_edit.sh
```

Vous obtiendrez un résultat comme celui-ci :

```
#| This is eyaml edit mode. This text (lines starting with #| at the top of the
#| file) will be removed when you save and exit.
#| - To edit encrypted values, change the content of the DEC(<num>)::PKCS7[]!
#|   block (or DEC(<num>)::GPG[]!).
#|   WARNING: DO NOT change the number in the parentheses.
#| - To add a new encrypted value copy and paste a new block from the
#|   appropriate example below. Note that:
#|   * the text to encrypt goes in the square brackets
#|   * ensure you include the exclamation mark when you copy and paste
#|   * you must not include a number when adding a new block
#|   e.g. DEC::PKCS7[]! -or- DEC::GPG[]!
---
test_secret: DEC(1)::GPG[This is a test secret]!
---
```



**Important** - Notez la modification de la ligne **test\_secret:**  
**DEC::GPG[This is a test secret]!** en **test\_secret: DEC(1)::GPG[This is**  
**a test secret]!**. Le numéro indique que le secret est existant et non un  
nouveau secret.

Ajoutez maintenant un dixième secret :

```
#| This is eyaml edit mode. This text (lines starting with #| at the top of the
#| file) will be removed when you save and exit.
#| - To edit encrypted values, change the content of the DEC(<num>)::PKCS7[]!
#|   block (or DEC(<num>)::GPG[]!).
#|   WARNING: DO NOT change the number in the parentheses.
```

```
#| - To add a new encrypted value copy and paste a new block from the
#| appropriate example below. Note that:
#| * the text to encrypt goes in the square brackets
#| * ensure you include the exclamation mark when you copy and paste
#| * you must not include a number when adding a new block
#| e.g. DEC::PKCS7[!] -or- DEC::GPG[!]
---
test_secret: DEC(1)::GPG[This is a test secret]!
new_secret: DEC::GPG[Somebody wakes up]!
```

Vérifiez que Hiera peut lire le nouveau secret avec la commande suivante :

```
vagrant@ubuntu-xenial:~$ sudo puppet lookup --environment pbgs new_secret--- Somebody wakes up
...
```

Dernièrement il faut copier la clef GPG a chaque noeud. Pour exporter la clef, utilisez la commande suivante :

```
vagrant@ubuntu-xenial:~$ sudo sh -c 'gpg --export-secret-key -a puppet@i2tch.co.uk > key.txt'
gpg: WARNING: unsafe ownership on configuration file `/home/vagrant/.gnupg/gpg.conf'
```

Vérifiez le contenu du fichier **key.txt** :

```
vagrant@ubuntu-xenial:~$ cat key.txt
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v1

lQ0YBFsTw6ABCADdkb6Ev1lqztqgNGBebDq0frW79ncPT6n4fqUFT70BBiQLdBeY
Ysd7kmaxBK0cWxIopBHUme5KA/BZ/t02Wr6VzKBnvW7xxy++TFTsK3Iza9QVXqZR
aAJR1NEA5KqZo3iSeMIeq8E60hs1cX4EKwi+5+4zAph2K1HPkLl+S+R7JKazEIVR
MFPt9eZmKFCtEHYw0gdjvVSkcUJs5nVXKMqL9kAVOUfh3WrYDUrahPzk0Q4Nr5u0
KRMunnsIrbhTsh/wfoZ6FtLeaBuwdrnAEWnKhVxxKt57H0Inx08COM5nUxLz3L/
c/dz7CYErRmHbyRmFe01L/s/QMDIiUmZgKtFABEBAAEAB/4wVYFaowVmC97cc0C0
6Iw74KJTCrYS5H1/JF1qVbMiU3PHrRkSmIXMuqil0BBtbZ59+CCdoMC/K7HMaA8W
KiCGPB/H0aa4nouXNz6VUHR2pbCbwsyEWKyc3lCDod5naqY8SZSLLUuXQctGss3b
```



Keob/ytSvF+284dCwN7+Ip8ij3nI94TB1WxIBWj70Yp0d00+AgCNUrYAdFriPEZ2  
NcB050AnFCl6dRXfKmIQakvRKM7uxPF09gGomhIENcVSsYChvgmlqfkSxT67CpT+  
3aK/60aJZYqj0D8jy0B1DqaT9tUWanAZK29KA8IZkJPnp3A0Hbta2BQ4GenZW+od  
IBZhBADmgNYnLLRr1qa1wjm/2Mn/Ibrep7q5tpb7bKyxLVGcJE3tQ0nlil0uRnq4  
hph+6m0FZUzxNgdUguikBapUJtiigs/Cso0ZhBwKh3P++fwEgVtqM7nf5ci07pDD  
Onewda6Av61a6ynVNGs1eLeNg0danFaaqiD0q6xsWysedCAcQQA9hPsv4L7c9D6  
a7GKWVGpVPyrVYxrQ0PVSxT59kHQqeb8ngV5J9ST7a5+83AzXJ3vkXUYMqV2zGcv  
42p7D0pIM0IACQnBBiGgWLIpwY+ubCAF9HvzlfGSRoomMjpRboyEQ4Z7tPAnWeb9  
uAl6/QnCgM30uyzL8JN850Cfi+NZQhUD/i6zKx7ly9YfXqZiEycsWYwFfluPCXIS  
p1qcVJg8vK44FRoLWpuchC/LWhhP++XwKKva6LRJRSLR23ZBe6VK5S3ZGjdZQ7XV  
pz+sqaYCrN9fZ477SLPuLAAR5g6fBY7k6Gm78b5cPFWPh5p04/Kw6PvcnEh2YMSI  
VHmCRbGk9xYTRTC0G3B1cHBlDCA8cHVwcGV0QGkydGNoLmNvLnVrPokB0AQTAAQIA  
IgUCWxPD0AiBawYLCQgHawIGFQgCCQoLBBYCAwECHgECF4AACgkQEoDRogUgWvTR  
agf/c+0wKPjryHzYxNbRN54/0dvkutdlHj1P0VELtWvjAG8rtknpYXEY2aotarlr  
Bl5bHIP2QzF6ozwEXlktVHcweHji3BHbyBNqgxCxWm+DG5yk55TH9+RB5da6Ud3V  
FXIyCf2Kwwaub4Imki29o4aFy82Z1kzN2+hxoZshKR+82duGAV/KCZwqPRW3rrvV  
HgaFGjFR66JZ3x4VPJk8dg4uCC6D1BEVSqBay8P2QR/KxqgqpIZ9bJlMVGDkzBq  
P+0fZsknd2MsuITD0YN6GcT9iZ65W0otuQZ14zlinux8omm50qgYU8tINlvKHllfe  
kxbGFstUUsqSto2ckGkUE0cz450DmARbE80gAQgA5Is2RwxW4Ig0s7QV7fG2EvZy  
0uJh3dXzRFeDoUl79zZ6tRGXrrw9y2axJGgya+6gm3k940g172QwQdxs1mL1HLeu  
RXPf9Li2VmRAuxb0eyPDcHLQr95/hLWGxucMd4jKQ4RJLuebNlVdUPNyL/ITlqgC  
28JAW7W/kXeSGkC5YQ5ZNUZzMzDtmEaw/RlH/KdPZHvQjJyf/E+vVz0JV87+xwf2  
MP6gnTxS5bGpcnm6dZs38cc8WysPuRwxNofFbzhWpaUV7xXb1lAniAxLFE0nvw5+  
VyC68heZ+Kl0uuZpjhrH0rhTCnmQpCgdZ6ZiLosCe5QbNCRLkQ1ZHel7JwQyxQAR  
AQABAAf9GdoSc3J2oFLbo0G5yk7Zauc0QV0It3x0VLJ06gljK6les30wNHyIG+o7  
lvPvYiydv8B06qwxr6jcWZLav9x1QACaBrKvDmJXg8m21cLouEYpHR7rsr0fHQ6g  
irslep7/TChaU+MweDKrSdUj4T8HjrbmDeA5Afa62vRU00Hnhk4Pq+fuSne/GxTB  
3RnfxXmgWn0d9+GKizm2BzrLxTGUKxoQ4Sho/fy8VDAx60TnpcoU0yxMBGpo4yF5  
Z++xZdwko9zpSQtMltGyXpJjYq1a1Avz8sujwy0EZjLA5gGP8Znr9Y0D6VudHEhv  
1sXZrHS7zPkuKnEg9j47mS4pMymp0QQA51QuoQU1ESP7PqYtbdCYNNnWRx4BaRqD  
EkfUKZRu+MgADfMFsnfX/00WJWe6DFIVdiEFq/n5aQnEb2gkr9bCugTfAomKNr1e  
+60hhF0mBlhymx56HhNtbBGE0lNs8fVPez00qh0iYqH0AFXJ1/Q011x+nCp+u2Ci  
NQy9afFFDbUEAPzq/eQaih3HaUJKxlqjodkSdwo7DqSKTT1Cc9xSvjmu+QHHq5+d  
DHk7ULkbbQQXvSk01i0RdvKqC/t2NEfqpV7iG6JbAF60+iSwXl0RKpRJPNy6rBn7

```
ItF1u+XwpgrFFBMqjTb9msdYy2ThzXffU13GGjJPb5eDtqTD2lu0BTrrA/40lTUS
9vkzVnyhXzBDlONHopt0I3EgmqwF6DTHiIAncBDWuXqt8DVjFTyCEdVayGRPsRyM
IWVa1zyYKfM7142Jedxk1JGyNFE+C4PcTbDSj01SRoGS1421fzvpXJKaYpE6g/uQ
ZtqIx3d9C+idn9Kwhm1jh5oR19nQqSLYe6n6TT7BiQEfBBgBAGAJBQJB80gAhsM
AAoJEBKA0aIFIFr0JwQH/2QYUqpGUSaTP1l+iCoi8mtansAXirB+xqVywAfywHzS
1a8R5q7pbG3aTzh578ptpRCFF9UgBv0gaUXe5eE+prms2xeQ08Cevk+7aBe/xm+D
dfRP51JcKvHYWtZMgoHRK2yTqYkmVuajXIMA11LU9qs0h91HSSEeEX+X2XTHC9gS
D90M06Nbs/bgUPZnX0ExKSUvenNhNo0FI3hMnqNx8xdg64X/6rxidootzTfgc+1H
pXiofzczqDErBmcMst0b0JAadq+BdPo0J0T/o0nfyjVx1uBokkMUn8uJtsKQWzl+
rbiN19KARb60/utd1lrR0dhAuFCXlyryjm94HHVRGy0=
=QW+0
-----END PGP PRIVATE KEY BLOCK-----
```

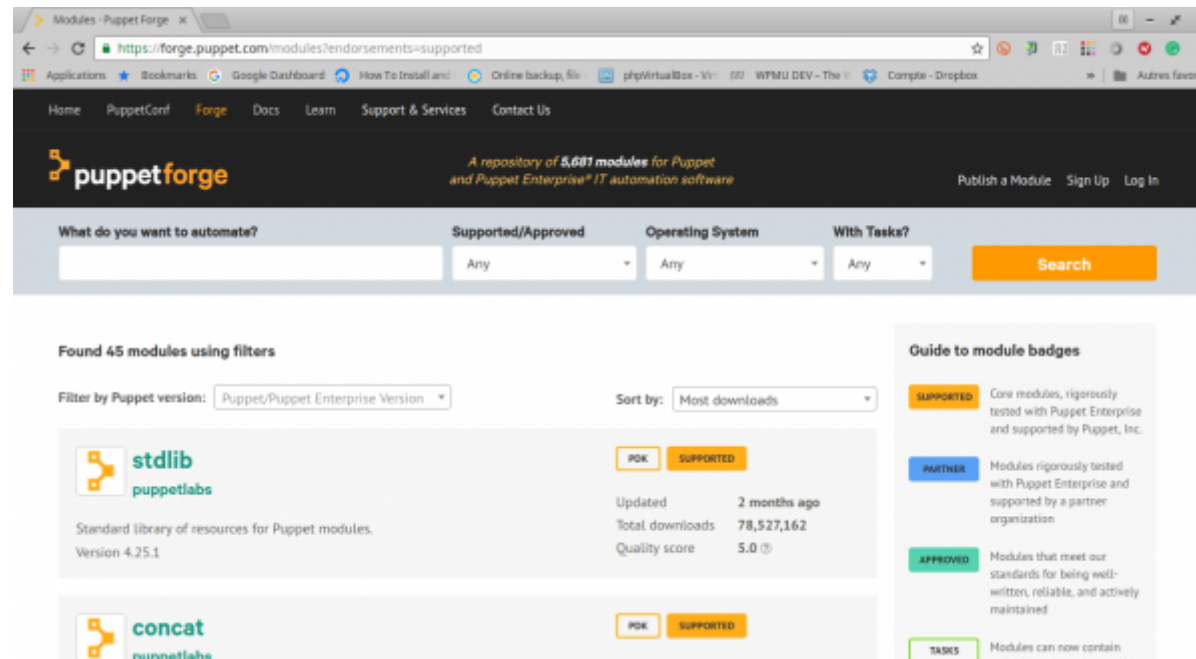
Sur les autres neuds, importez la clef :

```
# sudo gpg --import key.txt
```

## Modules

Les Modules de Puppet sont des morceaux de code réutilisables et qui gèrent un service ou un serveur de production et qui sont partagés sur **Puppet Forge**.

Puppet Forge est un repository public de modules disponible à l'adresse suivante <https://forge.puppet.com/modules?endorsements=supported> :



Il existe dans Puppet Forge des modules pour la plupart des serveurs, par exemple :

- MySQL/MariaDB,
- PostgreSQL,
- SQL Server,
- Apache,
- Nginx,
- Java,
- Tomcat,
- PHP,
- Ruby,
- Rails,
- Amazon AWS,
- Docker,
- Elasticsearch,
- Redis,
- Cassandra,

- Git,
- Iptables,
- etc ...

Les modules sont regroupés en deux groupes :

- **Supported** - les modules bénéficient du support de Puppet,
  - URL - <https://forge.puppet.com/modules?endorsements=supported>,
- **Approved** - les modules ne bénéficient pas du support de Puppet mais ils sont approuvés par ce dernier pour l'utilisation dont ils ont été conçus,
  - URL - <https://forge.puppet.com/modules?endorsements=approved>.

## LAB #25 - Installer des Modules

Le gestionnaire de modules de Puppet s'appelle **r10k**. Le gestionnaire utilise un fichier qui s'appelle **Puppetfile** qui se trouve dans le répertoire de l'environnement :

```
vagrant@ubuntu-xenial:~$ cd /etc/puppetlabs/code/environments/pbg
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ ls
data  hiera.yaml  Puppetfile
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ cat Puppetfile
forge 'http://forge.puppetlabs.com'

mod 'garethr/docker', '5.3.0'
mod 'puppet/archive', '1.3.0'
mod 'puppet/staging', '2.2.0'
mod 'puppetlabs/apache', '2.0.0'
mod 'puppetlabs/apt', '3.0.0'
mod 'puppetlabs/aws', '2.0.0'
mod 'puppetlabs/concat', '4.0.1'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql', '3.11.0'
mod 'puppetlabs/stdlib', '4.17.1'
mod 'stahnma/epel', '1.2.2'
```

```
mod 'pbg_ntp',  
  :git => 'https://github.com/bitfield/pbg_ntp.git',  
  :tag => '0.1.4'
```

Dans ce fichier :

- la variable **forge** spécifie le repository à utiliser,
- la variable **mod** spécifie le nom et la version du module à installer.

Exécutez la commande suivante pour que le gestionnaire traite ce fichier :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo r10k puppetfile install --verbose  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/docker  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/archive  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/staging  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/apache  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/apt  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/aws  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/concat  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/docker_platform  
WARN      -> Puppet Forge module 'puppetlabs-docker_platform' has been deprecated, visit  
https://forge.puppet.com/puppetlabs/docker_platform for more information.  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/mysql  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/stdlib  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/epel  
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/pbg_ntp
```

Pour tester si le module **stdlib** a été correctement installé, exécutez la commande suivante :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo puppet apply --environment pbg -e  
"notice(upcase('hello'))"  
Notice: Scope(Class[main]): HELLO  
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.14 seconds  
Notice: Applied catalog in 0.04 seconds
```



**Important** - La fonction **upcase** fait partie du module **stdlib**.

Le module **apache** nécessite que les modules **concat** et **stdlib** soient installés. Par contre la commande **r10k** ne gère pas de dépendances. La gestion des dépendances est donc manuelle. Afin d'aider l'administrateur à identifier les dépendances de tel ou tel module, il existe un Gem appelé **generate\_puppetfile**. Installez donc ce Gem :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo gem install generate-puppetfile
Fetching: colorize-0.8.1.gem (100%)
Successfully installed colorize-0.8.1
Fetching: generate-puppetfile-1.0.0.gem (100%)
Successfully installed generate-puppetfile-1.0.0
Parsing documentation for colorize-0.8.1
Installing ri documentation for colorize-0.8.1
Parsing documentation for generate-puppetfile-1.0.0
Installing ri documentation for generate-puppetfile-1.0.0
Done installing documentation for colorize, generate-puppetfile after 1 seconds
2 gems installed
```

Utilisez maintenant ce Gem pour générer le puppetfile pour le module **docker\_platform** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo
/opt/puppetlabs/puppet/lib/ruby/gems/2.4.0/gems/generate-puppetfile-1.0.0/bin/generate-puppetfile
puppetlabs/docker_platform
```

Installing modules. This may take a few minutes.

Your Puppetfile has been generated. Copy and paste between the markers:

```
=====
forge 'https://forge.puppet.com'
```

```
# Modules discovered by generate-puppetfile
mod 'garethr/docker',      '5.3.0'
mod 'puppetlabs/apt',      '3.0.0'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/stdlib',   '4.25.1'
mod 'stahnma/epel',        '1.3.0'
```

Pour générer une liste de dépendances à jour, y compris leurs versions, pour un fichier puppetfile existant il convient d'utiliser la commande suivante :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo
/opt/puppetlabs/puppet/lib/ruby/gems/2.4.0/gems/generate-puppetfile-1.0.0/bin/generate-puppetfile -p
/etc/puppetlabs/code/environments/pbg/Puppetfile
```

Installing modules. This may take a few minutes.

Your Puppetfile has been generated. Copy and paste between the markers:

```
=====
forge 'https://forge.puppet.com'
```

```
# Modules discovered by generate-puppetfile
mod 'garethr/docker',      '5.3.0'
mod 'puppet/archive',      '3.0.0'
mod 'puppet/staging',      '3.2.0'
mod 'puppetlabs/apache',   '3.1.0'
mod 'puppetlabs/apt',      '3.0.0'
mod 'puppetlabs/aws',      '2.1.0'
mod 'puppetlabs/concat',   '4.2.1'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql',    '5.4.0'
mod 'puppetlabs/stdlib',   '4.25.1'
mod 'puppetlabs/translate', '1.1.0'
```

```
mod 'stahnma/epel',          '1.3.0'
# Discovered elements from existing Puppetfile
mod 'pbg_ntp',
  :git => 'https://github.com/bitfield/pbg_ntp.git',
  :tag => '0.1.4'
=====
```

Notez que le Puppetfile existant n'est pas mis à jour :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ cat Puppetfile
forge 'http://forge.puppetlabs.com'

mod 'garethr/docker', '5.3.0'
mod 'puppet/archive', '1.3.0'
mod 'puppet/staging', '2.2.0'
mod 'puppetlabs/apache', '2.0.0'
mod 'puppetlabs/apt', '3.0.0'
mod 'puppetlabs/aws', '2.0.0'
mod 'puppetlabs/concat', '4.0.1'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql', '3.11.0'
mod 'puppetlabs/stdlib', '4.17.1'
mod 'stahnma/epel', '1.2.2'

mod 'pbg_ntp',
  :git => 'https://github.com/bitfield/pbg_ntp.git',
  :tag => '0.1.4'
```

Mettez à jour donc le Puppetfile :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ vi Puppetfile
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ cat Puppetfile
forge 'https://forge.puppet.com'
```



```
# Modules discovered by generate-puppetfile
mod 'garethr/docker',          '5.3.0'
mod 'puppet/archive',          '3.0.0'
mod 'puppet/staging',          '3.2.0'
mod 'puppetlabs/apache',       '3.1.0'
mod 'puppetlabs/apt',          '3.0.0'
mod 'puppetlabs/aws',          '2.1.0'
mod 'puppetlabs/concat',       '4.2.1'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql',        '5.4.0'
mod 'puppetlabs/stdlib',       '4.25.1'
mod 'puppetlabs/translate',    '1.1.0'
mod 'stahnma/epel',            '1.3.0'
# Discovered elements from existing Puppetfile
mod 'pbg_ntp',
  :git => 'https://github.com/bitfield/pbg_ntp.git',
  :tag => '0.1.4'
```

Dernièrement, utilisez le gestionnaire des modules **r10k** pour installer les modules :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo r10k puppetfile install --verbose
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/docker
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/archive
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/staging
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/apache
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/apt
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/aws
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/concat
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/docker_platform
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/mysql
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/stdlib
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/translate
INFO    -> Updating module /etc/puppetlabs/code/environments/pbg/modules/epel
```

```
INFO      -> Updating module /etc/puppetlabs/code/environments/pbg/modules/pbg_ntp
```

## LAB #26 - Utilisation des Modules

### puppetlabs/mysql

Commencez par créer le fichier **module\_mysql.pp** :

```
vagrant@ubuntu-xenial:~$ vi module_mysql.pp
vagrant@ubuntu-xenial:~$ cat module_mysql.pp
# Install MySQL and set up an example database
include mysql::server

mysql::db { 'cat_pictures':
  user      => 'greebo',
  password => 'tabby',
  host      => 'localhost',
  grant     => ['SELECT', 'UPDATE'],
}
```

La première ligne de ce fichier install le serveur MySQL en incluant la **classe** mysql avec le **paramètre** server. Le format de l'include est donc **classe::paramètre**. Lorsque Puppet rencontre cette ligne il recherche automatiquement Hiera pour toute clef qui correspond au nom du paramètre et utilise les valeurs. Dans ce cas, les valeurs se trouvent dans le fichier **/etc/puppetlabs/code/environments/pbg/data/common.yaml** et sont au nombre de deux :

```
...
mysql::server::root_password: 'hairline-quotient-inside-tableful'
mysql::server::remove_default_accounts: true
...
```

```
vagrant@ubuntu-xenial:~$ tail /etc/puppetlabs/code/environments/pbg/data/common.yaml
ensure: present
```

```
uid: 1904
shell: '/bin/bash'
mysql::server::root_password: 'hairline-quotient-inside-tableful'
mysql::server::remove_default_accounts: true
apache::default_vhost: false
pbg_ntp_params::version: 'latest'
pbg_ntp_params2::start_at_boot: true
pbg_ntp_params2::version: 'latest'
pbg_ntp_params2::service_state: 'running'
```



**Important** - Le mot de passe de root pour MySQL **hairline-quotient-inside-tableful** est ici en clair. En production, ce mot de passe serait crypté comme nous avons déjà vu.

Revenons au fichier **module\_mysql.pp**. A la suite de la première ligne est une ressource - **mysql::db** :

```
mysql::db { 'cat_pictures':
  user      => 'greebo',
  password  => 'tabby',
  host      => 'localhost',
  grant     => ['SELECT', 'UPDATE'],
}
```



**Important** - Le nom de la ressource **cat\_pictures** est le nom de la base de données. Les attributs **user**, **password**, **host** et **grant** indiquent que l'utilisateur **greebo** peut se connecter à MySQL à partir du **localhost** en utilisant le mot de passe **tabby** et qu'il aura les privilèges **SELECT** et **UPDATE** sur la base de données **cat-pictures**.

Appliquez le manifest **module\_mysql.pp** :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment=pbq module_mysql.pp
Notice: Compiled catalog for ubuntu-xenial.ief2i.fr in environment pbq in 4.49 seconds
Notice: /Stage[main]/Mysql::Server::Install/Package[mysql-server]/ensure: created
Notice: /Stage[main]/Mysql::Server::Config/File[mysql-config-file]/ensure: defined content as
'{md5}44e7aa974ab98260d7d013a2087f1c77'
Notice: /Stage[main]/Mysql::Server::Root_password/Mysql_user[root@localhost]/password_hash: changed password
Notice: /Stage[main]/Mysql::Server::Root_password/File[/root/.my.cnf]/ensure: defined content as
'{md5}4bb1978026fab523a39a7fd27e4e39c2'
Notice: /Stage[main]/Mysql::Client::Install/Package[mysql_client]/ensure: created
Notice: /Stage[main]/Main/Mysql::Db[cat_pictures]/Mysql_database[cat_pictures]/ensure: created
Notice: /Stage[main]/Main/Mysql::Db[cat_pictures]/Mysql_user[greebo@localhost]/ensure: created
Notice: /Stage[main]/Main/Mysql::Db[cat_pictures]/Mysql_grant[greebo@localhost/cat_pictures.*]/ensure: created
Notice: Applied catalog in 270.54 seconds
```

Vérifiez ensuite que vous pouvez vous connecter avec l'utilisateur **greebo** et le mot de passe **tabby** :

```
vagrant@ubuntu-xenial:~$ mysql -ugreebo -p
Enter password: tabby
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW databases;
+-----+
| Database          |
+-----+
```

```
| information_schema |  
| cat_pictures      |  
+-----+  
2 rows in set (0.01 sec)  
  
mysql> USE cat_pictures;  
Database changed  
mysql> exit  
Bye  
vagrant@ubuntu-xenial:~$
```

## puppetlabs/apache

Créez le fichier **module\_apache.pp** :

```
vagrant@ubuntu-xenial:~$ vi module_apache.pp  
vagrant@ubuntu-xenial:~$ cat module_apache.pp  
include apache  
  
apache::vhost { 'cat-pictures.com':  
  port      => '80',  
  docroot    => '/var/www/cat-pictures',  
  docroot_owner => 'www-data',  
  docroot_group => 'www-data',  
}  
  
file { '/var/www/cat-pictures/index.html':  
  content => "<img src='http://bitfieldconsulting.com/files/happycat.jpg'>",  
  owner   => 'www-data',  
  group   => 'www-data',  
}
```

La première ligne installe apache. Lorsque Puppet rencontre cette ligne il recherche automatiquement Hiera pour toute clef qui correspond au nom du

paramètre et utilise les valeurs. Dans ce cas, la valeur se trouve dans le fichier **/etc/puppetlabs/code/environments/pbg/data/common.yaml** :

```
...
  apache::default_vhost: false
...
```

```
vagrant@ubuntu-xenial:~$ tail /etc/puppetlabs/code/environments/pbg/data/common.yaml
  ensure: present
  uid: 1904
  shell: '/bin/bash'
mysql::server::root_password: 'hairline-quotient-inside-tableful'
mysql::server::remove_default_accounts: true
apache::default_vhost: false
pbg_ntp_params::version: 'latest'
pbg_ntp_params2::start_at_boot: true
pbg_ntp_params2::version: 'latest'
pbg_ntp_params2::service_state: 'running'
```



**Important - apache::default\_vhost: false** désactive l'hôte virtuel de la page de test d'Apache.

Revenons au fichier **module\_apache.pp**. A la suite de la première ligne est une ressource - **apache::vhost** :

```
apache::vhost { 'cat-pictures.com':
  port          => '80',
  docroot        => '/var/www/cat-pictures',
  docroot_owner  => 'www-data',
  docroot_group  => 'www-data',
}
```



**Important** - Le nom de la ressource **cat-pictures.com** est le nom de domaine de l'hôte virtuel. Les attributs **port**, **docroot**, **docroot\_owner** et **docroot\_group** indiquent que l'hôte virtuel écoute sur le port **80**, que les pages à servir par apache se trouvent dans le répertoire **/var/www/cat-pictures** et que ce répertoire appartient à l'utilisateur **www-data** et est associé avec le groupe **www-data** .

A la fin du fichier se trouve une ressource de type **file** :

```
file { '/var/www/cat-pictures/index.html':  
  content => "<img src='http://bitfieldconsulting.com/files/happycat.jpg'>",  
  owner   => 'www-data',  
  group   => 'www-data',  
}
```



**Important** - Cette ressource crée le fichier **/var/www/cat-pictures/index.html** appartenant à l'utilisateur **www-data**, étant associé au groupe **www-data** et ayant un contenu de **<img src='http://bitfieldconsulting.com/files/happycat.jpg'>** .

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment=pbp module_apache.pp  
Notice: Compiled catalog for ubuntu-xenial.ief2i.fr in environment pbp in 7.55 seconds  
Notice: /Stage[main]/Apache/Package[httpd]/ensure: created  
Notice: /Stage[main]/Apache/Exec[mkdir /etc/apache2/conf.d]/returns: executed successfully  
Notice: /Stage[main]/Apache/File[/etc/apache2/sites-available/000-default.conf]/ensure: removed  
Notice: /Stage[main]/Apache/File[/etc/apache2/sites-available/default-ssl.conf]/ensure: removed
```

```
Notice: /Stage[main]/Apache/File[/etc/apache2/sites-enabled/000-default.conf]/ensure: removed
Notice: /Stage[main]/Apache::Mod::Reqtimeout/File[reqtimeout.conf]/content: content changed
'{md5}40b45155afb3d14263d12e6fc4a98513' to '{md5}81c51851ab7ee7942bef389dc7c0e985'
Notice: /Stage[main]/Apache::Mod::Alias/File[alias.conf]/content: content changed
'{md5}c6e9f26152898c38e58211c8b362d5c3' to '{md5}cb528041df274fb077800a8e2e64f94e'
Notice: /Stage[main]/Apache::Mod::Autoindex/File[autoindex.conf]/content: content changed
'{md5}bfba7d77669e02b869b92e98215d58fc' to '{md5}2421a3c6df32c7e38c2a7a22afdf5728'
Notice: /Stage[main]/Apache::Mod::Deflate/File[deflate.conf]/content: content changed
'{md5}6649a32153b9afdc53f6898b5a10a2d6' to '{md5}a045d750d819b1e9dae3fbfb3f20edd5'
Notice: /Stage[main]/Apache::Mod::Dir/File[dir.conf]/content: content changed
'{md5}fe4bc5fa3b3cc7a241fe57f8fabc55a1' to '{md5}c741d8ea840e6eb999d739eed47c69d7'
Notice: /Stage[main]/Apache::Mod::Mime/File[mime.conf]/content: content changed
'{md5}7369c2fd5edf2192edbd6d865b632ae5' to '{md5}9da85e58f3bd6c780ce76db603b7f028'
Notice: /Stage[main]/Apache::Mod::Negotiation/File[negotiation.conf]/content: content changed
'{md5}443398efdd41085bcla70047f6e61c95' to '{md5}47284b5580b986a6ba32580b6ffb9fd7'
Notice: /Stage[main]/Apache::Mod::Setenvif/File[setenvif.conf]/content: content changed
'{md5}533f5f92761c2c24d6820f1d7d1c45ad' to '{md5}c7ede4173da1915b7ec088201f030c28'
Notice: /Stage[main]/Apache::Mod::Worker/File[/etc/apache2/mods-available/worker.conf]/ensure: defined content as
'{md5}1e3caf54ba0d71f3502b6cce6cda38d8'
Notice: /Stage[main]/Apache/Concat[/etc/apache2/ports.conf]/File[/etc/apache2/ports.conf]/content: content
changed '{md5}a961f23471d985c2b819b652b7f64321' to '{md5}334fa5cddb9a408ea1ca7a1666b1fc4'
Notice: /Stage[main]/Apache/File[/etc/apache2/apache2.conf]/content: content changed
'{md5}da32fcc6a783acaebf0d74c17c726bf7' to '{md5}f0bb70225191ce965d8c3e8ae992ba4c'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[authz_host]/File[authz_host.load]/content: content changed
'{md5}f529587409471e3feb62256bccaf538a' to '{md5}4b8162e3e3a58d1dc2222e2ad352bd93'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[authz_host]/File[authz_host.load symlink]/target: target
changed '../mods-available/authz_host.load' to '/etc/apache2/mods-available/authz_host.load'
Notice: /Stage[main]/Apache::Mod::Authn_core/Apache::Mod[authn_core]/File[authn_core.load symlink]/target: target
changed '../mods-available/authn_core.load' to '/etc/apache2/mods-available/authn_core.load'
Notice: /Stage[main]/Apache::Mod::Reqtimeout/Apache::Mod[reqtimeout]/File[reqtimeout.load symlink]/ensure:
created
Notice: /Stage[main]/Apache::Mod::Reqtimeout/Apache::Mod[reqtimeout]/File[reqtimeout.conf symlink]/ensure:
created
Notice: /Stage[main]/Apache::Mod::Alias/Apache::Mod[alias]/File[alias.load symlink]/target: target changed
```

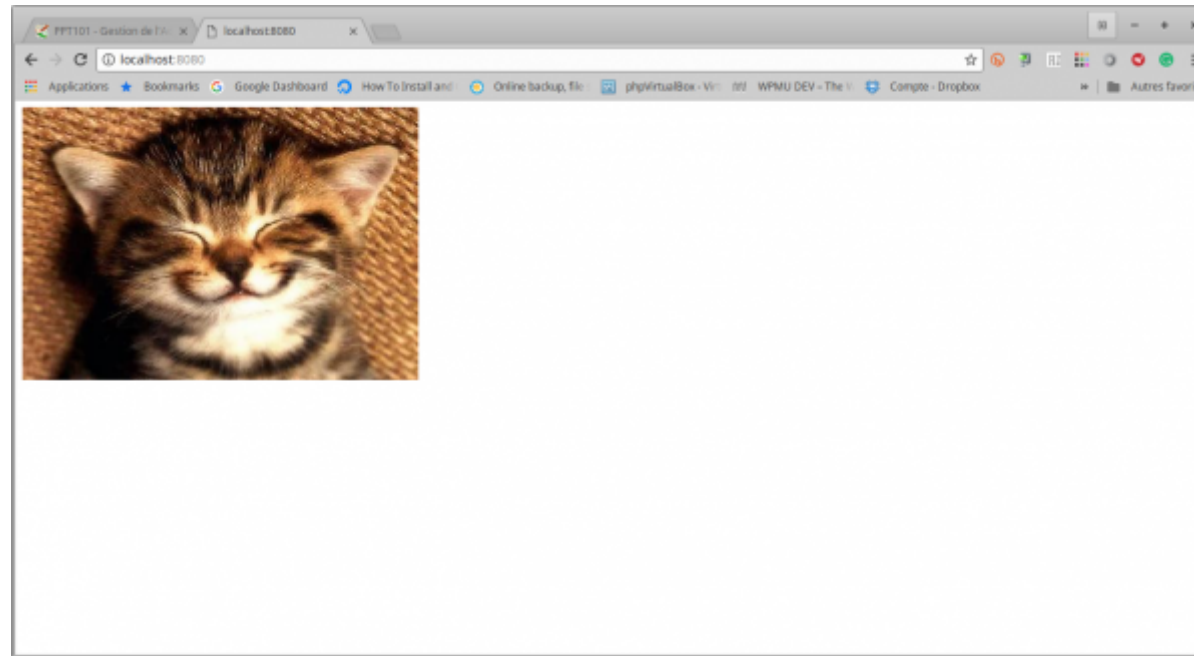


```
'../mods-available/alias.load' to '/etc/apache2/mods-available/alias.load'  
Notice: /Stage[main]/Apache::Mod::Alias/Apache::Mod[alias]/File[alias.conf symlink]/target: target changed  
'../mods-available/alias.conf' to '/etc/apache2/mods-available/alias.conf'  
Notice: /Stage[main]/Apache::Mod::Authn_file/Apache::Mod[authn_file]/File[authn_file.load symlink]/target: target  
changed '../mods-available/authn_file.load' to '/etc/apache2/mods-available/authn_file.load'  
Notice: /Stage[main]/Apache::Mod::Autoindex/Apache::Mod[autoindex]/File[autoindex.load symlink]/target: target  
changed '../mods-available/autoindex.load' to '/etc/apache2/mods-available/autoindex.load'  
Notice: /Stage[main]/Apache::Mod::Autoindex/Apache::Mod[autoindex]/File[autoindex.conf symlink]/target: target  
changed '../mods-available/autoindex.conf' to '/etc/apache2/mods-available/autoindex.conf'  
Notice: /Stage[main]/Apache::Mod::Dav/Apache::Mod[dav]/File[dav.load symlink]/ensure: created  
Notice: /Stage[main]/Apache::Mod::Dav_fs/File[dav_fs.conf]/content: content changed  
'{md5}162db96239f13b5416f369b6306d9db6' to '{md5}e36e2951cff0d4df331652ca6fccdb77'  
Notice: /Stage[main]/Apache::Mod::Dav_fs/Apache::Mod[dav_fs]/File[dav_fs.load]/content: content changed  
'{md5}17f662fd023dbaaab9d89f9a11ae58c1' to '{md5}3fa14d6e9f9d8eb25d09ed4823508f6d'  
Notice: /Stage[main]/Apache::Mod::Dav_fs/Apache::Mod[dav_fs]/File[dav_fs.load symlink]/ensure: created  
Notice: /Stage[main]/Apache::Mod::Dav_fs/Apache::Mod[dav_fs]/File[dav_fs.conf symlink]/ensure: created  
Notice: /Stage[main]/Apache::Mod::Deflate/Apache::Mod[deflate]/File[deflate.load]/content: content changed  
'{md5}6a7a23153796456ee919abdf728fd972' to '{md5}ac4540dd672556b07f900425751f745c'  
Notice: /Stage[main]/Apache::Mod::Deflate/Apache::Mod[deflate]/File[deflate.load symlink]/target: target changed  
'../mods-available/deflate.load' to '/etc/apache2/mods-available/deflate.load'  
Notice: /Stage[main]/Apache::Mod::Deflate/Apache::Mod[deflate]/File[deflate.conf symlink]/target: target changed  
'../mods-available/deflate.conf' to '/etc/apache2/mods-available/deflate.conf'  
Notice: /Stage[main]/Apache::Mod::Dir/Apache::Mod[dir]/File[dir.load symlink]/target: target changed '../mods-  
available/dir.load' to '/etc/apache2/mods-available/dir.load'  
Notice: /Stage[main]/Apache::Mod::Dir/Apache::Mod[dir]/File[dir.conf symlink]/target: target changed '../mods-  
available/dir.conf' to '/etc/apache2/mods-available/dir.conf'  
Notice: /Stage[main]/Apache::Mod::Mime/Apache::Mod[mime]/File[mime.load symlink]/target: target changed '../mods-  
available/mime.load' to '/etc/apache2/mods-available/mime.load'  
Notice: /Stage[main]/Apache::Mod::Mime/Apache::Mod[mime]/File[mime.conf symlink]/target: target changed '../mods-  
available/mime.conf' to '/etc/apache2/mods-available/mime.conf'  
Notice: /Stage[main]/Apache::Mod::Negotiation/Apache::Mod[negotiation]/File[negotiation.load symlink]/target:  
target changed '../mods-available/negotiation.load' to '/etc/apache2/mods-available/negotiation.load'  
Notice: /Stage[main]/Apache::Mod::Negotiation/Apache::Mod[negotiation]/File[negotiation.conf symlink]/target:  
target changed '../mods-available/negotiation.conf' to '/etc/apache2/mods-available/negotiation.conf'
```

```
Notice: /Stage[main]/Apache::Mod::Setenvif/Apache::Mod[setenvif]/File[setenvif.load symlink]/target: target
changed '../mods-available/setenvif.load' to '/etc/apache2/mods-available/setenvif.load'
Notice: /Stage[main]/Apache::Mod::Setenvif/Apache::Mod[setenvif]/File[setenvif.conf symlink]/target: target
changed '../mods-available/setenvif.conf' to '/etc/apache2/mods-available/setenvif.conf'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[auth_basic]/File[auth_basic.load]/content: content changed
'{md5}a92c34a3ce5e67e2ecbd272155ebe072' to '{md5}5eec35bb52e1ae9ff5eec602ed1d2374'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[auth_basic]/File[auth_basic.load symlink]/target: target
changed '../mods-available/auth_basic.load' to '/etc/apache2/mods-available/auth_basic.load'
Notice: /Stage[main]/Apache::Mod::Filter/Apache::Mod[filter]/File[filter.load symlink]/target: target changed
 '../mods-available/filter.load' to '/etc/apache2/mods-available/filter.load'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[authz_core]/File[authz_core.load symlink]/target: target
changed '../mods-available/authz_core.load' to '/etc/apache2/mods-available/authz_core.load'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[access_compat]/File[access_compat.load]/content: content
changed '{md5}f7f77b3eb4005749b583ff14b49f8bf1' to '{md5}a3e92887d86700b30a3869d0e8c159d7'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[access_compat]/File[access_compat.load symlink]/target:
target changed '../mods-available/access_compat.load' to '/etc/apache2/mods-available/access_compat.load'
Notice: /Stage[main]/Apache::Mod::Authz_user/Apache::Mod[authz_user]/File[authz_user.load]/content: content
changed '{md5}bcf988f105564ec1e14b78df56a01cd8' to '{md5}69150f8246499e0135580d8cc16eaeab'
Notice: /Stage[main]/Apache::Mod::Authz_user/Apache::Mod[authz_user]/File[authz_user.load symlink]/target: target
changed '../mods-available/authz_user.load' to '/etc/apache2/mods-available/authz_user.load'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[authz_groupfile]/File[authz_groupfile.load]/content:
content changed '{md5}97a3e14926ce7bf5f2d2a7b93ece648d' to '{md5}bf972e82336c72b20d67d871ef77d4ce'
Notice: /Stage[main]/Apache::Default_mods/Apache::Mod[authz_groupfile]/File[authz_groupfile.load symlink]/ensure:
created
Notice: /Stage[main]/Apache::Mod::Env/Apache::Mod[env]/File[env.load symlink]/target: target changed '../mods-
available/env.load' to '/etc/apache2/mods-available/env.load'
Notice: /Stage[main]/Apache::Mod::Worker/Apache::Mpm[worker]/File[/etc/apache2/mods-
available/worker.load]/ensure: defined content as '{md5}3064ef75f030fbf76986f6f073beb113'
Notice: /Stage[main]/Apache::Mod::Worker/Apache::Mpm[worker]/File[/etc/apache2/mods-enabled/worker.conf]/ensure:
created
Notice: /Stage[main]/Apache::Mod::Worker/Apache::Mpm[worker]/File[/etc/apache2/mods-enabled/worker.load]/ensure:
created
Notice: /Stage[main]/Apache::Mod::Cgid/File[cgid.conf]/content: content changed
'{md5}e8a2836392051bde889cf9c137058273' to '{md5}7dffbb5823bcbb9ab4d3e67ab14d38a0'
```

```
Notice: /Stage[main]/Apache::Mod::Cgid/Apache::Mod[cgid]/File[cgid.load symlink]/ensure: created
Notice: /Stage[main]/Apache::Mod::Cgid/Apache::Mod[cgid]/File[cgid.conf symlink]/ensure: created
Notice: /Stage[main]/Apache/File[/etc/apache2/mods-enabled/mpm_event.conf]/ensure: removed
Notice: /Stage[main]/Apache/File[/etc/apache2/mods-enabled/mpm_event.load]/ensure: removed
Notice: /Stage[main]/Apache/File[/etc/apache2/mods-enabled/status.conf]/ensure: removed
Notice: /Stage[main]/Apache/File[/etc/apache2/mods-enabled/status.load]/ensure: removed
Notice: /Stage[main]/Main/Apache::Vhost[cat-pictures.com]/File[/var/www/cat-pictures]/ensure: created
Notice: /Stage[main]/Main/File[/var/www/cat-pictures/index.html]/ensure: defined content as
'{md5}e274bce610d32200bb7256542f720833'
Notice: /Stage[main]/Main/Apache::Vhost[cat-pictures.com]/Concat[25-cat-
pictures.com.conf]/File[/etc/apache2/sites-available/25-cat-pictures.com.conf]/ensure: defined content as
'{md5}3096e79ddd479b5289234df595f43817'
Notice: /Stage[main]/Main/Apache::Vhost[cat-pictures.com]/File[25-cat-pictures.com.conf symlink]/ensure: created
Notice: /Stage[main]/Apache::Service/Service[httpd]: Triggered 'refresh' from 1 event
Notice: Applied catalog in 66.04 seconds
```

Pour vérifier l'installation et la configuration de l'hôte virtuel, connectez-vous à l'adresse <http://localhost:8080> avec votre navigateur :



Il semblerait que ce petit chaton soit bien content de vous voir !

## **puppet/archive**

Puppet sait installer des logiciels à partir d'un **tarball** (\*.tar.gz) ou à partir d'un fichier **zip**. Créez les fichier **module\_archive.pp** :

```
vagrant@ubuntu-xenial:~$ vi module_archive.pp
vagrant@ubuntu-xenial:~$ cat module_archive.pp
file { ['/var/www':
  ensure => directory,
}

archive { ['/tmp/wordpress.tar.gz':
  ensure      => present,
  extract     => true,
```

```
extract_path => '/var/www',  
source       => 'https://wordpress.org/latest.tar.gz',  
creates      => '/var/www/wordpress',  
cleanup      => true,  
}
```



**Important** - La première ressource est de type **file**. Cette ressource crée si nécessaire le répertoire **/var/www/**.

La deuxième ressource est une **archive** :

```
archive { '/tmp/wordpress.tar.gz':  
  ensure      => present,  
  extract     => true,  
  extract_path => '/var/www',  
  source      => 'https://wordpress.org/latest.tar.gz',  
  creates     => '/var/www/wordpress',  
  cleanup     => true,  
}
```



**Important** - Le nom de la ressource **/tmp/wordpress.tar.gz** indique le nom et l'emplacement de l'archive téléchargé - **latest.tar.gz**. L'attribut **extract** indique à Puppet d'extraire l'archive dans l'attribut **extract\_path**. L'attribut **source** indique à Puppet d'où il faut télécharger l'archive. L'attribut **creates** indique le nom d'un répertoire qui existera une fois l'archive désarchivée. De cette façon si Puppet détecte la présence de ce répertoire il ne procédera pas à l'extraction de l'archive considérant que l'extraction a déjà eu lieu. Dernièrement l'attribut **cleanup** indique à Puppet de supprimer l'archive à la fin du processus.

Appliquez maintenant le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment=pbq module_archive.pp
Notice: Compiled catalog for ubuntu-xenial.ief2i.fr in environment pbq in 0.53 seconds
Notice: /Stage[main]/Main/Archive[/tmp/wordpress.tar.gz]/ensure: download archive from
https://wordpress.org/latest.tar.gz to /tmp/wordpress.tar.gz and extracted in /var/www with cleanup
Notice: Applied catalog in 99.85 seconds
```

Vérifiez que le processus à abouti :

```
vagrant@ubuntu-xenial:~$ ls /var/www
cat-pictures  html  wordpress
```

Copyright © 2018 I2TCH LIMITED