Version : **2022.01**

Dernière mise-à-jour : 2023/09/15 09:56

LDF804 - Hiera et Modules

Contenu du Module

Préparation

```
vagrant@ubuntu-xenial:~$ sudo apt-get install git-core zlib1g-dev build-essential libssl-dev libreadline-dev
libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libcurl4-openssl-dev software-properties-common
```

```
libffi-dev
...
vagrant@ubuntu-xenial:~$ cd
vagrant@ubuntu-xenial:~$ git clone https://github.com/excid3/asdf.git ~/.asdf
vagrant@ubuntu-xenial:~$ echo '. "$HOME/.asdf/asdf.sh"' >> ~/.bashrc
vagrant@ubuntu-xenial:~$ echo '. "$HOME/.asdf/completions/asdf.bash"' >> ~/.bashrc
vagrant@ubuntu-xenial:~$ echo 'legacy_version_file = yes' >> ~/.asdfrc
vagrant@ubuntu-xenial:~$ echo 'export EDITOR="code --wait"' >> ~/.bashrc
vagrant@ubuntu-xenial:~$ exec $SHELL
vagrant@ubuntu-xenial:~$ asdf plugin add ruby
vagrant@ubuntu-xenial:~$ asdf install ruby 3.2.2
...
vagrant@ubuntu-xenial:~$ gem update --system
vagrant@ubuntu-xenial:~$ vi .tool-versions
vagrant@ubuntu-xenial:~$ cat .tool-versions
ruby 3.2.2
vagrant@ubuntu-xenial:~$ ruby -v
ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux]
```

Hiera

Présentation

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

```
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 nœuds. La multiplication de cette tâche pour 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 minimaliste :

```
---
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.sample
data  environment.conf  hiera.yaml  manifests  modules
```

Copiez **data**, **environment.conf** et **hiera.yaml** du répertoire **/etc/puppetlabs/code/environments/production.sample** vers le répertoire **/etc/puppetlabs/code/environments/production** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production.sample$ sudo cp -r data  environment.conf
hiera.yaml /etc/puppetlabs/code/environments/production
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/production.sample$ cd ../production
```

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

LAB #1 - 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:~$ sudo 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 0.03 seconds
Notice: Applied catalog in 0.01 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
vagrant@ubuntu-xenial:~$
```

LAB #2 -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}"
  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'
```

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

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

Tandis quun 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:~$ sudo 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 pbg lookup_hash.pp
Notice: Scope(Class[main]): pxe_just_once is enabled
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.02 seconds
Notice: Applied catalog in 0.01 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:~$ sudo 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 pbg lookup_hash_dot.pp
Notice: Scope(Class[main]): web_root is public_html
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.03 seconds
Notice: Applied catalog in 0.01 seconds
```

La donnée obtenue se trouve dans le fichier **/etc/puppetlabs/code/environments/pbg/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 #3 - 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:~$ sudo 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 considérer 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 nœud 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 #4 -Créer des Ressources avec les Données d'Hiera

Commencez par créer le fichier **hiera_users.pp** :

```
vagrant@ubuntu-xenial:~$ sudo 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 pbg hiera_users.pp
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.04 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 0.14 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:~$ sudo 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/pbg/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 0.03 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 0.17 seconds
```

Consultez maintenant la fin du fichier **/etc/passwd** :

```
vagrant@ubuntu-xenial:~$ tail /etc/passwd
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
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 #5 - 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 gpgme
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (1.4.20-1ubuntu3.3).
```

```
The following NEW packages will be installed:
  rng-tools
0 upgraded, 1 newly installed, 0 to remove and 7 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 (76.0 kB/s)
Selecting previously unselected package rng-tools.
(Reading database ... 74569 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 ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.23) ...
Setting up rng-tools (5-0ubuntu3) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.23) ...
```

Installez ensuite le support gpg pour Hiera :

```
vagrant@ubuntu-xenial:~$ gem install hiera-eyaml-gpg
```

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 (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

NE METTEZ PAS DE PASSPHRASE. APPUYEZ SIMPLEMENT SUR [ENTREE] <----------------------------------ATTENTION
```

```
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 D327661B 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/D327661B 2020-02-12
      Key fingerprint = 38E1 5B5E EBDF F39F F49B  ED2A 9646 8A98 D327 661B
uid                  puppet <puppet@i2tch.co.uk>
sub   2048R/73C27290 2020-02-12
```

La clef est visible :

```
pub   2048R/D327661B 2020-02-12
      Key fingerprint = 38E1 5B5E EBDF F39F F49B  ED2A 9646 8A98 D327 661B
uid                  puppet <puppet@i2tch.co.uk>
```

```
sub   2048R/73C27290 2020-02-12
```

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

```
vagrant@ubuntu-xenial:~$ vi /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/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 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[]!
```

Éditez 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,hQEMA5MfXRRzwnKQAQf/fRBVVlz42D4cIHMtZmyUhNE8Kxdzn+lAsTfu7a8NbbN0NfNmYcCZxWr2h0rVoEdQKUiMdawr5kwsMMU1aGbWb
0GH5LDwde8XXxDnr4dnACevG5G/UJkhhsc8cpfJSk9yCyb0rGu9dM8/PJuh768p5xwVslu5lQkNE5gVx14Vh2xMqyqio2DUhWbKphzw234GBK9g9E
```

```
ys2ucsi/aRJtAgxz4N0eD/+E2xsaBBTXr4rOZ1SXUlLGflS6opaOkFb5WTdUoGReP0YxFOd6hhysL9fltHYAgBzvP06BUvlmvnC4QqAy+q3z7Xj65
NyRpvGYoLTrkoaSav650DaoqklsQ5D9JSAbt1nI/TZCjJiPqT6vKPPcDs2Y4kvOA6KKWMR1G32cBicWc9+sWiDtO2NxJrf4noI6rujmmZUnRkkd9e
CWnUtIvSgb3G55Y965e6Pl110jlPpw==]
---
```

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 pbg test_secret
--- This is a test secret
```

Créez maintenant le script **eyaml_edit.sh** :

```
vagrant@ubuntu-xenial:~$ sudo vi eyaml_edit.sh
vagrant@ubuntu-xenial:~$ cat eyaml_edit.sh
#!/bin/bash
eyaml edit --gpg-always-trust --gpg-recipients=puppet@i2tch.co.uk
/etc/puppetlabs/code/environments/pbg/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 134 Feb 12 09:37 /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 134 Feb 12 09:37 /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 automatique 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 deuxième 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 pbg new_secret
--- Somebody wakes up
```

Dernièrement il faut copier la clef GPG a chaque nœud. 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

lQOYBF5DvOEBCADjpHyyFInVw5v+2kWYnlWxYny5LAz4jdJIU4IhJtpBpOivU5GE
P6X7u7izFfOclByUtPQuJlFKDzuXgCPXpH31Ifx0HCMdhM3SR1mNc+kWI65eQ23q
CCPd3d3t9dYjWOBnfczKwxKcztBYzrlDhUK7u1Wkdm+Z2n6d8y3PhP+IDvzW3H0K
1gbCsFKDF7cTlkQJQuyAFl7TggRfAbmD5z65LVlNBwVe4YBKpngF2+6SJ9T5j7MT
24VR+l9UfBmTm1F+HGn7RUqQDutCxEvRLqxf1fcSr5HKK9Cs50jWJ+whjlcdEV40
BzWe/0Kjz90SR8Un9RQS2iTzFx99GurmYcG7ABEBAAEAB/wI+btQ6i5lF0lXSxPj
aJrIW7G/O1Vnl2/rGh/PXtDMNcxW3VyncjsBKe2mYf8w9UYLNvdJuokJ+n/HAW5o
Wjv9JlHl4hYgW9Qhrl6BjUhrbwy4+r+isp93aAAQGdlPm3l9myAv/nEFbcIZPk1f
tCllbYV67pCH6G2FnMj5nazVmG6mPDEsE5duEJ8Zx6L8Y4ETPqCnhil8+X9cgL4B
bSODCK2jbjoXIzKWXtU0EeHJB6Nv0TTd/zbie9XfU65qGOrvSVlZgvTYem8q9OoL
+VCqZghTa6O/jzZWNvKyuFqrx2J8TIaNAhZuHx9gJ/HitXd95g/ncQzOCe/qep3q
7glBBADwMtPac43LvOo44KvJzuuFBaY940LIWP38cZEZtXs4yaeVQ2A5KiYTuHVz
0kCfFZ6roZ/jARKyVNikyqL2hTMYiCjUY4V4b8fKLjcTYhazz+kpANJ0DX+MgDqx
zuWXzHFVT1/ZgyR+nvi15c9MPSXtv8Ta9Qe8cTu/0jx8nCZYnwQA8p40SQEDuOCE
mQAR5PGYxK+hDRyMKAAqCoDY/rMpwCwioMHv9a5Dk0BqhRpVxuBCLichR9SPRokl
p/7+nIodOPDFjWI4dEM45eVit/MepPl3PBfPuY5pGzc/H/pGjPzvNb+Gb+6kjcQI
YgPQzy7BuPS3LXmqW+qZywT4i9usVWUD/355kUCeLKfidvsTp8oKq+6+pz4Sn29P
GvCh1os7UJrPuCQ4ixKFwjlMyDccjQdPvfF8Jivv6Kmxt/PccYXQfBltJ7ubILF0
q6auSw53h8gvwmHeNEV6UnygS0diO21ARgWWQhmWFmvrTHm2T6rlJRCEZXlYf3Jj
x0+yW5eLcU21QDW0G3B1cHBldCA8cHVwcGV0QGkydGNoLmNvLnVrPokBOAQTAQIA
IgUCXkO84QIbAwYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQlkaKmNMnZhtF
awgAkRtzy6MJvbRQ9o6Qs9bAiRPnf/78nTOVzfq6wvANp01rQSuf62yOTrM27nZ9
w2MGgUNCOWIL7hJ1H/kvlSJ6hsSshmxDbidppWbmSZFW8guLztpVvc/8JQd2Ums1
Y/nGAIxeAVpnZf5s+ZY5LDSHRUmol36/8FdnII9mtsm+wypiauxzzmpAJwVCA715
dybg7JH39AkYpr9pqoJt0XRLjOcZM1nzDSPu3GkFf4HYH0ENnON6QMm0ut2oAfba
TGx8ZJYhCpELqg0e+0hOtUXknY8bBDTE0sTBGU+s7BAnrWbvgtiA3RD+UDA1x0Sm
9N086OcDEg7KIQf9R0WfyeVRvJ0DmAReQ7zhAQgA9M/4MxXNWw0mlk+dpuERf3bu
k+zDr69mOa95DKLS6B/gOXOKQ9f4Ly5UoIpNaHCc3TafYH5MlQt7aTk+8KvQ4Z4n
OEbSFtBlTbHc62f/PDGWT3YtbUcPMW1/zHqUMEwn0YLVS5Xwt8kHB/bkrBFxAgG8
qCoCV0Lw8ff7kg1lPkkH7+iaUdqzdhLbFrlbs0CQTDoc5+z+ijKXiIuio8XBmJNJ
KSqvtLgmERIIMMgxkNbf+9Y6hoSNFa9GrFZMoSx9OnaeI0ttwdYPHktrYhiJZgtN
```

```
ehxPYnTvByctK6gwzWkfiQS0/H9Jj8wK1WWsGbyaQ63933jNNhnQSsfIW2HoSwAR
AQABAAf7BQYzoNDEuyak8RbHNIKP+l61fMGyLmvaHDr2sAScWIeFu/BGa1YAZfg
0r6j4g3VmlChqrhM2A45RSqQmR3dlpPJWEVT0HYMbjDFSOuEeHbID+9r9T8tF6WY
ptipK+iUNJ+LwxZbr1nn1AkEHmfpOtJQTd821xWfnkepmRC/UCzjPytUmcshFRSf
n4E3Ok1aikTzo4THSJrOxCjRHLq/o2mXdIIkgCr/Na7q0lPSHZqSpAab92Q/ESTk
fXrVPmKcV34qgcK6MeId0/7MwQ/tl3Eel87uLEJEAfXkvVp+p6j5NTkVenFHFPEV
+0Tudm/py5SBkIWsNojoF/vGjcsXIQQA+GgPpxcQdW/5XCIKn9lsRYwGY3sxu1KF
GLogSujl7/uNGUOx42jFs/XkdJHGTI/OWPIT2OMiZ759DeHzV3tUJEcPIAZxK//E
T7zCmywJD6RNsOvbz8XluqWe5eTjXS1c0YCmdR4YwduuZpgT7T1yCNrX/JQLjVl0
EL5XDfS44VsEAPxLyD6UG7qnLbpQ1c2kzjs3EUqPe7p7/Y46zI1M4yzAw2elNyGB
6Ek+vxYk4XQOCQ1M5lsqq7fS694e1Wa8bW5wW4BpIdTFPulsdE4McAqH7xeGifig
2obAt+tMPnLn3/tj3RG1JPSIf0w8w86WjqneZ8YYXzcRKDK40Cx+VFfRBACYkvtv
AAXNUlhTxGiS7xQkNy+EARbN37/J0HmqJ1VpfWRCGRzjx6pasfhQgNPhxDvPvQza
5zAm4nOt9LOim6igKN+aSIk99KdtRYKxPR1h/vF8iPRLcvbLWnZ2Sc02mbzEyRLy
O4yVUFCLQHXRI8iQhM6cdn8RlIaZ2AWoPUwjnUYiiQEfBBgBAgAJBQJeQ7zhAhsM
AAoJEJZGipjTJ2YbvVIH/Rea3HIQJN09gDtqiYJTZYM9KTVQlqyG0frTBqVHMEXD
1U07bt4j6oaML9fkywJbNJu16z4Tj5RST+w0k7qCwUqgsAdpwUK/a+oWFxVmDbin
Lk/TRM/XytvpSJ1FgeMdF6n8ivP+W9yh2UFsxR68ZezryOAl9tz4k33E1e/GPZUA
TlKtl9HAAPWMyNznCwA4yZ/c+NxuwXFp+/oFVcDJjkjcCX7Sdaihlkhtnw4PBB/u
PR2k5sYm7AV88AuUzIf2vxVuNbmMHKJT1qQJt4p4TdPeCtofhI4r1/GTTi4xyPpg
jnxZNEX8j4nW37ulyW6sHElUZFdhPDMdb8vZIyy3JYI=
=UwyQ
-----END PGP PRIVATE KEY BLOCK-----
```

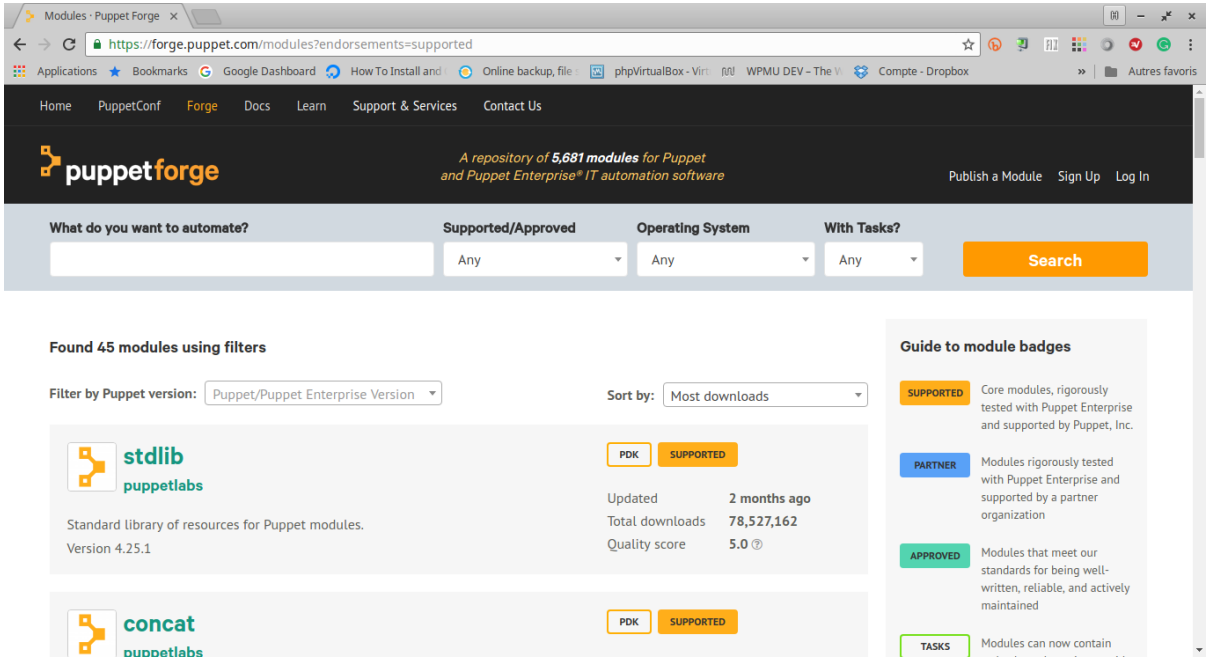
Sur les autres nœuds, importez la clef :

```
# 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 il sont approuvés par ce dernier pour l'utilisation dont ils ont été conçus,
  - URL - **https://forge.puppet.com/modules?endorsements=approved**.

LAB #6 - 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.

Supprimez les trois dernières lignes du fichier **Puppetfile** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ vi 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'
```

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     -> Using Puppetfile '/etc/puppetlabs/code/environments/pbg/Puppetfile'
INFO     -> Updating module /etc/puppetlabs/code/environments/pbg/modules/docker
WARN     -> Puppet Forge module 'garethr-docker' has been deprecated, visit
https://forge.puppet.com/garethr/docker for more information.
INFO     -> Updating module /etc/puppetlabs/code/environments/pbg/modules/archive
INFO     -> Updating module /etc/puppetlabs/code/environments/pbg/modules/staging
WARN     -> Puppet Forge module 'puppet-staging' has been deprecated, visit
https://forge.puppet.com/puppet/staging for more information.
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
```

Pour tester si le module **stdlib** est 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 0.01 seconds
Notice: Applied catalog in 0.01 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-1.1.0.gem (100%)
Successfully installed generate-puppetfile-1.1.0
Parsing documentation for colorize-0.8.1
Installing ri documentation for colorize-0.8.1
Parsing documentation for generate-puppetfile-1.1.0
Installing ri documentation for generate-puppetfile-1.1.0
Done installing documentation for colorize, generate-puppetfile after 0 seconds
2 gems installed
```

Installez le paquet **ruby-dev** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo apt install ruby-dev
```

Utilisez maintenant le Gem **generate-puppetfile** pour générer le puppetfile pour le module **docker_platform** :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ 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',               '1.3.1'
================================================================================
```

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 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',             '4.4.0'
mod 'puppet/staging',             '3.2.0'
mod 'puppetlabs/apache',          '5.4.0'
mod 'puppetlabs/apt',             '3.0.0'
mod 'puppetlabs/aws',             '2.1.0'
mod 'puppetlabs/concat',          '6.2.0'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql',           '10.3.0'
mod 'puppetlabs/stdlib',          '4.25.1'
mod 'puppetlabs/translate',       '2.1.0'
mod 'stahnma/epel',               '1.3.1'
================================================================================
```

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'
```

Mettez à jour donc le Puppetfile :

```
vagrant@ubuntu-xenial:/etc/puppetlabs/code/environments/pbg$ sudo 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',             '4.4.0'
mod 'puppet/staging',             '3.2.0'
mod 'puppetlabs/apache',          '5.4.0'
mod 'puppetlabs/apt',             '3.0.0'
mod 'puppetlabs/aws',             '2.1.0'
mod 'puppetlabs/concat',          '6.2.0'
mod 'puppetlabs/docker_platform', '2.2.1'
mod 'puppetlabs/mysql',           '10.3.0'
mod 'puppetlabs/stdlib',          '4.25.1'
mod 'puppetlabs/translate',       '2.1.0'
```

```
mod 'stahnma/epel',               '1.3.1'
```

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     -> Using Puppetfile '/etc/puppetlabs/code/environments/pbg/Puppetfile'
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
WARN     -> Puppet Forge module 'puppet-staging' has been deprecated, visit
https://forge.puppet.com/puppet/staging for more information.
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
```

LAB #7 - Utilisation des Modules

puppetlabs/mysql

Commencez par créer le fichier **module_mysql.pp** :

```
vagrant@ubuntu-xenial:~$ sudo 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=pbg module_mysql.pp
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.38 seconds
Notice: /Stage[main]/Mysql::Server::Config/File[/etc/mysql]/ensure: created
Notice: /Stage[main]/Mysql::Server::Config/File[/etc/mysql/conf.d]/ensure: created
Notice: /Stage[main]/Mysql::Server::Config/File[mysql-config-file]/ensure: defined content as
'{md5}3cb51becc6dc4533c36b0212cba6091b'
Notice: /Stage[main]/Mysql::Server::Install/Package[mysql-server]/ensure: created
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 32.59 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: 5.7.29-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2020, 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 (0.00 sec)

mysql> USE cat_pictures;
Database changed
mysql> exit
Bye
```

puppetlabs/apache

Créez le fichier **module_apache.pp** :

```
vagrant@ubuntu-xenial:~$ sudo 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='https://3.bp.blogspot.com/-w1McqyQflFM/VUEsrgAphKI/AAAAAAAAAdY/lbMsugNdGto/s1600/why-do-cats-purr-happycat.j
pg'>",
  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='https://3.bp.blogspot.com/-w1McqyQflFM/VUEsrgAphKI/AAAAAAAAAdY/lbMsugNdGto/s1600/why-do-cats-purr-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='https://3.bp.blogspot.com/-w1McqyQflFM/VUEsrgAphKI/AAAAAAAAAdY/lbMsugNdGto/s1600/why-do-cats-purr-happycat.jpg'>** .

Appliquez le manifest :

```
vagrant@ubuntu-xenial:~$ sudo puppet apply --environment=pbg module_apache.pp
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.98 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::Service/Service[httpd]: Triggered 'refresh' from 1 event
Notice: Applied catalog in 8.94 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:~$ sudo 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=pbg module_archive.pp
Notice: Compiled catalog for ubuntu-xenial in environment pbg in 0.07 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 4.44 seconds
```

Vérifiez que le processus à abouti :

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