Lab 7 – les ressources

La ressource file

Gère les fichiers, y compris leur contenu, leur propriété et leurs autorisations. Le type file peut gérer des fichiers, des répertoires et des liens symboliques normaux ; le type doit être spécifié dans l'attribut ensure.

```
file { 'resource title':
  path                    => # (namevar) The path to the file to manage.
  backup                  => # Whether (and how) file content should be backed...
  checksum                => # The checksum type to use when determining...
  checksum_value          => # The checksum of the source contents. Only md5...
  content                 => # The desired contents of a file, as a string...
  ctime                   => # A read-only state to check the file ctime. On...
  force                   => # Perform the file operation even if it will...
  group                   => # Which group should own the file.  Argument can...
  ignore                  => # A parameter which omits action on files matching
  links                   => # How to handle links during file actions.  During
  max_files               => # In case the resource is a directory and the...
  mode                    => # The desired permissions mode for the file, in...
  mtime                   => # A read-only state to check the file mtime. On...
  owner                   => # The user to whom the file should belong....
  provider                => # The specific backend to use for this `file...
  purge                   => # Whether unmanaged files should be purged. This...
  recurse                 => # Whether to recursively manage the _contents_ of...
  recurselimit            => # How far Puppet should descend into...
  replace                 => # Whether to replace a file or symlink that...
  selinux_ignore_defaults => # If this is set then Puppet will not ask SELinux...
  selrange                => # What the SELinux range component of the context...
  selrole                 => # What the SELinux role component of the context...
  seltype                 => # What the SELinux type component of the context...
  seluser                 => # What the SELinux user component of the context...
  show_diff               => # Whether to display differences when the file...
  source                  => # A source file, which will be copied into place...
  source_permissions      => # Whether (and how) Puppet should copy owner...
  sourceselect            => # Whether to copy all valid sources, or just the...
  staging_location        => # When rendering a file first render it to this...
  target                  => # The target for creating a link.  Currently...
  type                    => # A read-only state to check the file...
  validate_cmd            => # A command for validating the file's syntax...
  validate_replacement    => # The replacement string in a `validate_cmd` that...
  # ...plus any applicable metaparameters.
}
```

- Copiez le contenu depuis un fichier

```
file{ '/etc/motd':
  source => '/tmp/motd.txt',
}
```

- Copiez le contenu depuis une URL

```
file{ '/tmp/README.md':
  source => 'https://github.com/medsalahmeddeb/devops-
labs/blob/main/README.md',
}
```

- Modifier les permissions d'un fichier

```
file{
  '/tmp/motd.txt':
  ensure => present,
  owner => 'vagrant',
  group => 'vagrant',
  mode => '0600',
}
```

- Créer un lien symbolique

```
file {
  '/tmp/lien':
  ensure => link,
  target ='/etc/motd'
}
```

La ressource cron

Ce module gère les tâches cron en plaçant des fichiers au format /etc/cron.d

cron::job crée des jobs génériques en /etc/cron.d. Il permet de spécifier les paramètres suivants :

```
ensure - facultatif - par défaut "présent"
command - obligatoire - la commande à exécuter
minute - facultatif - par défaut "*"
hour - facultatif - par défaut "*"
date - facultatif - par défaut "*"
month - facultatif - par défaut "*"
weekday - facultatif - par défaut "*"
special - facultatif - par défaut undef
user - facultatif - par défaut "root"
environment - facultatif - par défaut ""
mode - facultatif - la valeur par défaut est "0600"
description - facultatif - par défaut undef
```

- Exemple : sauvegarder la date chaque quart d'heure

```
cron{'run-puppet':
   command =>'date >> /tmp/date ',
   hour    => '*' ,
   minute  => '*/15' ,
}
```

La ressource package

Gérer les packages automatiquement par puppet. Puppet devinera automatiquement le format de paquetage que vous utilisez en fonction de la plate-forme sur laquelle vous vous trouvez, mais vous pouvez le remplacer à l'aide du paramètre provider.

```
package { 'resource title':
  name                 => # The package name.  This is the name that the...
  command              => # The targeted command to use when managing a...
  ensure               => # What state the package should be in. On...
  adminfile            => # A file containing package defaults for...
  allow_virtual        => # Specifies if virtual package names are allowed...
  allowcdrom           => # Tells apt to allow cdrom sources in the...
  category             => # A read-only parameter set by the...
  configfiles          => # Whether to keep or replace modified config files
  description          => # A read-only parameter set by the...
  enable_only          => # Tells `dnf module` to only enable a specific...
  flavor               => # OpenBSD and DNF modules support 'flavors', which
  install_only         => # It should be set for packages that should only...
  install_options      => # An array of additional options to pass when...
  instance             => # A read-only parameter set by the...
  mark                 => # Set to hold to tell Debian apt/Solaris pkg to...
  package_settings     => # Settings that can change the contents or...
  platform             => # A read-only parameter set by the...
  provider             => # The specific backend to use for this `package...
  reinstall_on_refresh => # Whether this resource should respond to refresh...
  responsefile         => # A file containing any necessary answers to...
  root                 => # A read-only parameter set by the...
  source               => # Where to find the package file. This is mostly...
  status               => # A read-only parameter set by the...
  uninstall_options    => # An array of additional options to pass when...
  vendor               => # A read-only parameter set by the...
  # ...plus any applicable metaparameters.
}
```

- Exemple : Installer les packages nécessaires pour le LAMP server

```
Package{ ensure => 'installed' }

package { 'httpd':  }
package { 'mariadb-server':  }
package { 'mariadb':  }
package { 'php':  }
package { 'php-mysql':  }
package { 'php-fpm':  }
```

- Exemple : Désinstaller le package 'php-fpm'

```
package {'php-fpm':
        ensure => 'absent'
}
```

La ressource service

Gérer les services en cours d'exécution. La prise en charge des services varie malheureusement considérablement selon la plate-forme

Les attributs :

```
service { 'resource title':
  name          => # (namevar) The name of the service to run.  This name is...
  ensure        => # Whether a service should be running. Default...
  binary        => # The path to the daemon.  This is only used for...
  control       => # The control variable used to manage services...
  enable        => # Whether a service should be enabled to start at...
  flags         => # Specify a string of flags to pass to the startup
  hasrestart    => # Specify that an init script has a `restart...
  hasstatus     => # Declare whether the service's init script has a...
  logonaccount  => # Specify an account for service...
  logonpassword => # Specify a password for service logon. Default...
  manifest      => # Specify a command to config a service, or a path
  path          => # The search path for finding init scripts....
  pattern       => # The pattern to search for in the process table...
  provider      => # The specific backend to use for this `service...
  restart       => # Specify a *restart* command manually.  If left...
  start         => # Specify a *start* command manually.  Most...
  status        => # Specify a *status* command manually.  This...
  stop          => # Specify a *stop* command...
  timeout       => # Specify an optional minimum timeout (in seconds)
  # ...plus any applicable metaparameters.
}
```

- Démarrer le service httpd

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

- Si certains de vos services manquent de scripts d'initialisation, Puppet peut compenser, comme dans l'exemple suivant :

```
service { "apache2":
  ensure  => running,
  start   => "/usr/sbin/apachectl start",
  stop    => "/usr/sbin/apachectl stop",
  pattern => "/usr/sbin/httpd",
}
```

- Démarrer les services du serveur LAMP

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

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

La ressource user

Gérer les utilisateurs. Ce type est principalement conçu pour gérer les utilisateurs du système, il lui manque donc certaines fonctionnalités utiles pour gérer les utilisateurs normaux.

```
Les attributs
user { 'resource title':
  name                 => # (namevar) The user name. While naming limitations vary
by...
  ensure               => # The basic state that the object should be in....
  allowdupe            => # Whether to allow duplicate UIDs.  Default...
  attribute_membership => # Whether specified attribute value pairs should...
  attributes           => # Specify AIX attributes for the user in an array...
  auth_membership      => # Whether specified auths should be considered the
  auths                => # The auths the user has.  Multiple auths should...
  comment              => # A description of the user.  Generally the user's
  expiry               => # The expiry date for this user. Provide as either
  forcelocal           => # Forces the management of local accounts when...
  gid                  => # The user's primary group.  Can be specified...
  groups               => # The groups to which the user belongs.  The...
  home                 => # The home directory of the user.  The directory...
  ia_load_module       => # The name of the I&A module to use to manage this
  iterations           => # This is the number of iterations of a chained...
  key_membership       => # Whether specified key/value pairs should be...
  keys                 => # Specify user attributes in an array of key ...
  loginclass           => # The name of login class to which the user...
  managehome           => # Whether to manage the home directory when Puppet
  membership           => # If `minimum` is specified, Puppet will ensure...
  password             => # The user's password, in whatever encrypted...
  password_max_age     => # The maximum number of days a password may be...
  password_min_age     => # The minimum number of days a password must be...
  password_warn_days   => # The number of days before a password is going to
  profile_membership   => # Whether specified roles should be treated as the
  profiles             => # The profiles the user has.  Multiple profiles...
  project              => # The name of the project associated with a...
  provider             => # The specific backend to use for this `user...
  purge_ssh_keys       => # Whether to purge authorized SSH keys for this...
  role_membership      => # Whether specified roles should be considered the
  roles                => # The roles the user has.  Multiple roles should...
  salt                 => # This is the 32-byte salt used to generate the...
  shell                => # The user's login shell.  The shell must exist...
  system               => # Whether the user is a system user, according to...
  uid                  => # The user ID; must be specified numerically. If...
  # ...plus any applicable metaparameters.
}
```

- Créer l'utilisateur user1 appartenant au group dev

```
group{'dev':
  ensure   => present,
  gid => '2000',


}
user { 'user21':
  ensure   => present,
  home => '/home/user1',
  shell => '/bin/bash',
  groups => ['dev'],
  uid => '10000',
}
```

- Supprimer l'utilisateur user1

```
user { 'user2':
  ensure   => absent,
}
```