EXEMPLO RESOURCE TYPE

tipo { 'título':  
 param1 => 'valor',  
 param2 => 'valor',  
 param3 => 'valor',  
 metaparam1 => 'valor',  
 metaparam2 => 'valor',  
 function1 => 'valor',  
 function2 => 'valor',  
}

EXEMPLO RESOURCE TYPE USER

user { 'gutocarvalho':  
 ensure => 'present',  
 gid => '500',  
 home => '/home/gutocarvalho',  
 password => '$6$BE6a/5SJ$7mSGRgRZBPVRv5p7OkClaTUwx.hwTLzrpVNEbqvyjFy5yFUdGZeL41b76VC6DXV/eFUG.zFS.wkjTpsO.KfCy0',  
 password\_max\_age => '99999',  
 password\_min\_age => '0',  
 shell => '/bin/bash',  
 uid => '500',  
 }

EXEMPLO RESOURCE TYPE FILE

file { '/etc/resolv.conf':  
 ensure => 'file',  
 content => '{md5}24ebe9c85187eb484686b5e835eb24ae',  
  [ctime](http://www.opengroup.org/onlinepubs/009695399/functions/ctime.html) => 'Tue May 22 13:32:17 -0300 2012',  
 group => '0',  
 mode => '644',  
 mtime => 'Tue May 22 13:32:17 -0300 2012',  
 owner => '0',  
 selrange => 's0',  
 selrole => 'object\_r',  
 seltype => 'net\_conf\_t',  
 seluser => 'system\_u',  
 type => 'file',  
}

VERIFICANDO ESTADOS

puppet resource user**(tipo)** gutocarvalho**(título)**

user { 'gutocarvalho':  
 ensure => 'present',  
 comment => 'Jose Augusto da Costa Carvalho,,,',  
 gid => '50000',  
 groups => ['infra'],  
 home => '/home/gutocarvalho',  
 password => '$6$UE/yPfbi$eq3kd1WPZxv8assXjxc6ZSiNc/YF40dmVknbPxT8LUFn5gKqhCGMbtWV9cobF8xWai1Kk7xSu.aHjAubzQuHA0',  
 password\_max\_age => '99999',  
 password\_min\_age => '0',  
 shell => '/bin/bash',  
 uid => '50000',  
}

[root@galaxy ~]# puppet resource user lucas  
user { 'lucas':  
 ensure => 'absent', //absent significa que o usuário não existe  
}

CRIANDO O USUÁRIO

puppet resource user lucas ensure=present

notice: /User[lucas]/ensure: created  
user { 'lucas':  
 ensure => 'present',  
}

EDITANDO ATRIBUTOS

ralsh user lucas --edit //alterando atributos do usuário lucas

ralsh package htop --edit //alterando atributos do pacote htop

PRINCIPAIS TIPOS DE RECURSOS DO PUPPET

* User
* Exec
* Cron
* File
* Package
* Service
* Notify
* Host
* Mailalias

O QUE É FACTER

Facter é usado para coletar informações do sistema com o puppet.

NOTA: No **resource type** **file** podemos usar os seguintes valores para o parâmetro ensure:

* present = esteja presente
* absent = não esteja presente
* file =
* directory = crie um diretório
* link = crie um link simpbolico
* recurse =

Os parâmetros mais utilizados são:

* owner
* group
* mode
* source
* recurse

### **EXEMPLO CONCEITUAL DE TRIO**

Na configuração em trio vamos atender as seguintes necessidades:

* Um pacote deve ser declarado como presente/instalado
* O serviço relacionado ao pacote deve estar habilitado
  + O serviço relacionado ao pacote de estar rodando
  + O serviço relacionado ao pacote aceita comando restart em seu init script
  + O serviço relacionado ao pacote aceita comando status em seu init script
  + O serviço depende do objeto pacote
* Um arquivo de configuração do serviço deve ser declarado como presente
  + O arquivo de configuração depende do objeto pacote
  + O arquivo deve pertencer ao usuário root
  + O arquivo deve pertencer ao grupo root
  + O arquivo deve ter permissão 644
  + O arquivo de configuração deve notificar o serviço caso ocorra mudança na origem

A tradução e construção destes requisitos acima em código puppet pode ser solucionada desta forma:

package { 'pacote':  
 ensure => present,  
 }  
   
 service { 'servico':  
 ensure => running,  
 enable => true,  
 hasrestart => true,  
 hasstatus => true,  
 require => Package['pacote'],  
 }  
   
 file { 'main.cf':  
 path => "/etc/pacote/pacote.conf",  
 source => "/root/puppet/pacote.conf",  
 owner => 'root',  
 group => 'root',  
 mode => 644,  
 require => Package['pacote'],  
 notify => Service['servico'],  
 }

Veja que a tradução é visualmente mais objetiva que a declaração das necessidades/requisitos, o código é limpo e entendível.