

Puppet Resources – Cheatsheet

Los **resources** son el corazón de Puppet. Cada uno representa algo que Puppet puede gestionar: archivos, paquetes, servicios, usuarios, permisos, etc.

Formato general:

```
tipo { 'título':
  atributo => valor,
  atributo => valor,
}
```

Tipos de Recursos Más Usados

file

Gestiona archivos y directorios.

```
file { '/ruta/al/archivo':
  ensure  => file,      # present / absent / directory / link
  content => 'texto...', 
  source   => 'puppet:///modules/modulo/archivo',
  mode     => '0644',
  owner    => 'root',
  group   => 'root',
}
```

package

Instala, elimina o actualiza paquetes.

```
package { 'nginx':
  ensure => installed,  # present / absent / latest
}
```

service

Controla servicios del sistema.

```
service { 'nginx':
  ensure => running,      # running / stopped
  enable  => true,        # inicia con el sistema
  provider => 'systemd',
}
```

user

Crea, modifica o elimina usuarios.

```
user { 'axel':
  ensure     => present,
  uid       => '1001',
  home     => '/home/axel',
  shell     => '/bin/bash',
  managehome => true,
}
```

group

Gestiona grupos.

```
group { 'devs':
  ensure => present,
}
```

exec

Permite ejecutar comandos (cuando no existe otro recurso apropiado).

```
exec { 'update_system':
  command => '/usr/bin/apt update',
  path    => ['/usr/bin', '/usr/sbin'],
  unless  => 'test -f /tmp/ya_actualizado',
}
```

⚠️ Regla de oro: Usar `exec` solo si un recurso nativo no existe.

cron

Gestiona jobs de cron.

```
cron { 'backup':
  ensure  => present,
  user    => 'root',
  hour    => 2,
  minute   => 0,
  command  => '/usr/local/bin/backup.sh',
}
```

mount

Gestiona puntos de montaje.

```
mount { '/mnt/data':
  ensure  => mounted,
  device  => '/dev/sdb1',
  fstype   => 'ext4',
  atboot   => true,
}
```

ssh_authorized_key

Agrega claves SSH.

```
ssh_authorized_key { 'axel@laptop':
  ensure => present,
  user   => 'axel',
  type   => 'ssh-rsa',
  key    => 'AAAAB3NzaC1yc2EAAAQABAAQ...',
}
```

notify

Imprime mensajes en el output de Puppet (útil para debug).

```
notify { 'Instalación completa':
  message => 'Todo OK!',
}
```

Atributos comunes en varios recursos

Atributo	Significa
ensure	Estado deseado (present, absent, running, installed, etc.)
require	Dependencia: espera a otro recurso
before	Debe ejecutarse antes de otro recurso
notify	Notifica a otro recurso si cambia
subscribe	Se reactiva si otro recurso cambia
provider	Selecciona backend (systemd, apt, yum, etc.)
alias	Nombre alternativo del recurso

Ver recursos del sistema (formato Puppet)

Muy útil para aprender:

```
puppet resource user  
puppet resource package nginx  
puppet resource service ssh
```

Ejemplo combinando recursos

```
package { 'nginx':  
  ensure => installed,  
}  
  
file { '/var/www/html/index.html':  
  ensure  => file,  
  content => 'Hola Puppet!',  
  notify  => Service['nginx'],  
}  
  
service { 'nginx':  
  ensure  => running,  
  enable   => true,  
}
```