puppet 资源 exec 详细介绍（附案例）

一、资源介绍

Description

Executes external commands. It is critical that all commands executed using this mechanism can be run multiple times without harm, i.e., they are idempotent. One useful way to create idempotent commands is to use the checks like creates to avoid running the command unless some condition is met.

Note that you can restrict an exec to only run when it receives events by using the refreshonly parameter; this is a useful way to have your configuration respond to events with arbitrary commands.

Note also that if an exec receives an event from another resource, it will get executed again (or execute the command specified in refresh, if there is one).

There is a strong tendency to use exec to do whatever work Puppet can’t already do; while this is obviously acceptable (and unavoidable) in the short term, it is highly recommended to migrate work from exec to native Puppet types as quickly as possible. If you find that you are doing a lot of work with exec, please at least notify us at Puppet Labs what you are doing, and hopefully we can work with you to get a native resource type for the work you are doing.

Autorequires: If Puppet is managing an exec’s cwd or the executable file used in an exec’s command, the exec resource will autorequire those files. If Puppet is managing the user that an exec should run as, the exec resource will autorequire that user.

Attributes

```
exec { 'resource title':
  command     => # (namevar) The actual command to execute.  Must either be...
  creates     => # A file to look for before running the command...
  cwd         => # The directory from which to run the command.  If
  environment => # Any additional environment variables you want to
  group       => # The group to run the command as.  This seems to...
  logoutput   => # Whether to log command output in addition to...
  onlyif      => # If this parameter is set, then this `exec` will...
  path        => # The search path used for command execution...
  provider    => # The specific backend to use for this `exec...
  refresh     => # How to refresh this command.  By default, the...
  refreshonly => # The command should only be run as a refresh...
  returns     => # The expected return code(s).  An error will be...
  timeout     => # The maximum time the command should take.  If...
  tries       => # The number of times execution of the command...
  try_sleep   => # The time to sleep in seconds between...
  umask       => # Sets the umask to be used while executing this...
  unless      => # If this parameter is set, then this `exec` will...
  user        => # The user to run the command as.  Note that if...
  # ...plus any applicable metaparameters.
}
```

1、实现功能

1.1 远程执行系统命令，其实就是 **shell** 的调用

1.2 由于 **exec** 是一次性执行资源，在不同类里面 **exec** 名字可相同。

二、系统环境

1、puppet 服务端

```
Release: RHEL6.4
HOSTNAME: puppetserver.kisspuppet.com
TCP/IP: 172.16.200.100/24
Packages:
puppet-server-2.7.21-1.el6.noarch
mcollective-client-2.2.4
activemq-5.5.0
```

2、puppet 节点

```
Release: RHEL5.8
HOSTNAME: agent1.kisspuppet.com
TCP/IP: 172.16.200.101/24
Packages:
puppet-2.7.21-1.el5
mcollective-2.2.4-1.el5
```

3、puppet 节点

```
Release: RHEL6.4
```

```
HOSTNAME: agent3.kisspuppet.com
TCP/IP: 172.16.200.103/24
Packages:
puppet-2.7.21-1.el6
mcollective-2.2.4-1.el6
```

三、支持参数

1.1 command => "mkdir /tmp/rhel5/nginx ", 被执行的命令，必须为被执行命令的绝对路径。

1.2 cwd =>"/tmp/rhel5/nginx", 指定命令执行的目录。如果目录不存在，则命令执行失败。

1.3 environment => "PATH=/home/puppetfans", 为命令设定额外的环境变量。要注意的是如果你用这个来设定 PATH，那么 PATH 的属性会被覆盖。多个环境变量应该以数组的形式来设定。

1.4 group => 定义运行命令的用户组。在不同的平台下的运行的结果无法确定，由于不同用户运行命令的时候，变量是不变的，所以这是平台的问题，而不是 Ruby 或 Puppet 的问题。

1.5 logoutput => on_failure|true|false 是否记录输出。默认会根据 exec 资源的日志等级（loglevel） 来记录输出。若定义为 on_failure，则仅在命令返回错误的时候记录输出。可取的值为：true，false 和其他合法的日志等级。

1.6 onlyif =>"/bin/ls /usr/local/nginx/conf", 如果这个参数被设定了，则 exec 只会在 onlyif 设定的命令返回 0 时才执行。

1.7 path => ["/bin/", "/sbin/" , "/usr/bin/", "/usr/sbin/"] 命令执行的搜索路径。如果 path 没有被定义，命令需要使用绝对路径。路径可以以数组或以冒号分隔的形式来定义。

1.8 creates => "/tmp/rhel5/nginx", 指定命令所生成的文件。如果提供了这个参数，那么命令只会在所指定的文件不存在的情况的被执行：

1.9 refresh =>true|false 定义如何更新命令。当 exec 收到一个来自其他资源的事件时，默认只会重新执行一次命令。不过这个参数允许你定义更新时执行不同的命令。

1.10 refreshonly =>true|false 该属性可以使命令变成仅刷新触发的，也就是说只有在一个依赖的对象被改变时，命令才会被执行。只有同时使用 subscribe 或 notify 才有意义

1.11 returns 指定返回的代码。如果被执行的命令返回了其他的代码，一个错误（error）会被返回。默认值是 0，可以定义为一个由可以接受的返回代码组成的数组或单值。

1.12 timeout => 0 命令运行的最长时间。如果命令运行的时间超过了 timeout 定义的时间，那么这个命令就会被终止，并作为运行失败处理。当定义为负值时就会取消运行时间的限制。timeout 的值是以秒为单位的。

1.13 unless => "/bin/ls /usr/local/nginx/conf",如果这个变量被指定了，那么 exec 会执行，除非 unless 所设定的命令返回 0

1.14 user => "nginx", 定义运行命令的用户。 注意如果你使用了这个参数，那么任何的错误输出不会在当下被捕捉，这是 Ruby 的一个 bug。

三、资源示例

1、示例一

1.1 实现功能

*要求实现使用 chkconfig 命令将节点的 iptables 和 ip6tables 服务关闭，并记录错误日志

```
1.2 配置说明
class motd::exec {
  include motd::exec1
}
class motd::exec1 {
  exec {  [  "chkconfig iptables off",
         "chkconfig ip6tables off",]:
    path      => ["/usr/bin","/usr/sbin","/bin","/sbin"],
    logoutput => on_failure,
  }
}
```

```
1.3 客户端 agent3 测试
[root@agent3 ~]# puppet agent --test
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/backup_date.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply1.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply3.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply2.rb
info: Caching catalog for agent3.kisspuppet.com
info: Applying configuration version '1378284783'
notice: /Stage[main]/Motd::Exec1/Exec[chkconfig iptables off]/returns: executed
successfully
notice: /Stage[main]/Motd::Exec1/Exec[chkconfig ip6tables off]/returns: executed
successfully
notice: Finished catalog run in 0.17 seconds
```

2、示例二

2.1 实现功能

*要求节点上创建用户和组 nginx，UID 和 GID 都为 1000

*要求从服务器下载 nginx-0.8.42.tar.gz 源码包到节点/tmp/rhel5/nginx 目录下

*要求解压源码包，并编译安装到指定目录下

```
2.2 配置说明
class source {
  include source::file1,source::exec1,source::exec2,source::exec3,source::user
    notify { "nstallation nginx package through the source code nginx-
0.8.42.tar.gz":
      withpath => true,
    }
}
class source::user{
  group { "nginx":   #建立组 nginx
    ensure => present,
    gid    => 1000
  }
  user { "nginx":   #建立用户 nginx
    ensure     => present,
    uid        => 1000,
    gid        => 1000,
    groups     => ["nginx"],
#   membership => minimum,
    shell      => "/sbin/nologin",
    require    => Group["nginx"]
  }

}
class source::file1{  #远程下载 nginx 源码包
  file{ "nginx":
    name    => "/tmp/rhel5/nginx/nginx-0.8.42.tar.gz",
    owner   => "root",
    group   => "root",
    mode    => 0700,
    source  => "puppet://$puppetserver/modules/source/nginx-0.8.42.tar.gz",
    backup  => 'main',
    require => Class["source::exec1"],
   }
}
class source::exec1{
  exec {"create nginx_pag":
    command => "mkdir /tmp/rhel5/nginx ",
    path    => ["/usr/bin","/usr/sbin","/bin","/sbin"],
    creates => "/tmp/rhel5/nginx", #目录或文件不存在的情况下执行 command
  }
}
```

```
class source::exec2{
  exec { "install nginx":
    cwd       =>"/tmp/rhel5/nginx",  #目录存在的情况下执行 command
    command   =>"tar -zxvf nginx-0.8.42.tar.gz && cd nginx-0.8.42 &&./configure --
user=nginx --group=nginx --prefix=/usr/local/nginx --without-http-cache &&
make&&make install",
    path      => ["/usr/bin","/usr/sbin","/bin","/sbin"],
    logoutput => on_failure,
    unless    => "/bin/ls /usr/local/nginx/conf",  #命令返回值不为 0 的情况下执行
commond
    require   => Class[source::file1,source::user]
    notify    => Class["source::exec3"],
  }
}
class source::exec3{
  exec { "updatedb":
    command     => "updatedb",
    path        => ["/usr/bin","/usr/sbin","/bin","/sbin"],
    refreshonly => true,  #触发更新的时候执行 command
    subscribe   => Class["source::exec2"],
  }
}
[root@puppetserver manifests]#
```

```
2.3 客户端 agent3 测试
[root@agent3 rhel5]# cat /etc/passwd | grep nginx
[root@agent3 rhel5]# cat /etc/group | grep nginx
[root@agent3 rhel5]# ll /tmp/rhel5/
total 0
[root@agent3 rhel5]# ll /usr/local/ |  grep nginx
[root@agent3 ~]# puppet agent --test
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/backup_date.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply1.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply3.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply2.rb
info: Caching catalog for agent3.kisspuppet.com
info: Applying configuration version '1378366520'
notice: /Stage[main]/Source::Exec1/Exec[create nginx_pag]/returns: executed
successfully
```

```
notice: /Stage[main]/Source/Notify[nstallation nginx package through the source
code nginx-0.8.42.tar.gz]/message: nstallation nginx package through the source
code nginx-0.8.42.tar.gz
notice: /Stage[main]/Source/Notify[nstallation nginx package through the source
code nginx-0.8.42.tar.gz]/message: defined 'message' as 'nstallation nginx package
through the source code nginx-0.8.42.tar.gz'
notice: /Stage[main]/Source::File1/File[nginx]/ensure: defined content as
'{md5}2818e8b03512b239f1238d702703bcf3'
notice: /Stage[main]/Source::User/Group[nginx]/ensure: created
notice: /Stage[main]/Source::User/User[nginx]/ensure: created
notice: /Stage[main]/Source::Exec2/Exec[install nginx]/returns: executed
successfully
info: /Stage[main]/Source::Exec2/Exec[install nginx]: Scheduling refresh of
Class[Source::Exec3]
info: Class[Source::Exec2]: Scheduling refresh of Exec[updatedb]
info: Class[Source::Exec3]: Scheduling refresh of Exec[updatedb]
notice: /Stage[main]/Source::Exec3/Exec[updatedb]: Triggered 'refresh' from 2
events
notice: Finished catalog run in 18.83 seconds
[root@agent3 rhel5]# cat /etc/passwd | grep nginx
nginx:x:1000:1000::/home/nginx:/sbin/nologin
[root@agent3 rhel5]# cat /etc/group | grep nginx
nginx:x:1000:nginx
[root@agent3 rhel5]# ll /tmp/rhel5/nginx/
total 632
drwxr-xr-x. 8 nginx nginx   4096 Sep  5 14:29 nginx-0.8.42
-rwx------. 1 root  root  642593 Sep  5 14:29 nginx-0.8.42.tar.gz
[root@agent3 rhel5]# ll /usr/local/nginx/
total 16
drwxr-xr-x. 2 root root 4096 Sep  5 14:30 conf
drwxr-xr-x. 2 root root 4096 Sep  5 14:30 html
drwxr-xr-x. 2 root root 4096 Sep  5 14:30 logs
drwxr-xr-x. 2 root root 4096 Sep  5 14:30 sbin
```

第二次执行，由于设置了 `unless => "/bin/ls /usr/local/nginx/conf"`，当命令返回结果为 0 的时候，exec 是不会执行的。其次设置了 `refreshonly => true`,配合 `notify` 和 `subscrive` 只有在更改的情况下才会触发更新

```
[root@agent3 ~]# puppet agent --test
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/backup_date.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply1.rb
info: Loading facts in /var/lib/puppet/lib/facter/my_apply3.rb
```

```
info: Loading facts in /var/lib/puppet/lib/facter/my_apply2.rb
info: Caching catalog for agent3.kisspuppet.com
info: Applying configuration version '1378366520'
notice: /Stage[main]/Source/Notify[nstallation nginx package through the source
code nginx-0.8.42.tar.gz]/message: nstallation nginx package through the source
code nginx-0.8.42.tar.gz
notice: /Stage[main]/Source/Notify[nstallation nginx package through the source
code nginx-0.8.42.tar.gz]/message: defined 'message' as 'nstallation nginx package
through the source code nginx-0.8.42.tar.gz'
notice: Finished catalog run in 0.32 seconds
```

——————————————————————————————————————

为了能够和大家更好的交流和学习 Puppet，本人 2014 年又新开辟了微信公众号进行交流学习，目前已经有 300 多人同时收听，喜欢 Puppet 的大神们可自行加入哦。

如果你有好的有关 Puppet 的咨询也可以给我投稿，投稿地址：admin@kisspuppet.com

微信公众号：“puppet2014”，可搜索加入，也可以扫描以下二维码

——————————————————————————————————————