

深度分析 puppet 自带 filebucket 的备份恢复功能

日常情况下都是通过 puppetmaster 端进行推送变更，推送可以用 MC 实现，至于变更的回滚可以通过 SVN 活该 Git 辅助实现。存在这样一种情况，在一个复杂的环境中，存在多个管理员，而你是负责变更推送的，他们只是做应用的，很多时候是需要登录系统权限的，有一次，一个管理员为了满足他的应用并没有告知给你，登录到系统之后，修改了一个配置文件，完后也没有保存，而这个配置文件恰恰是你这边 puppetmaster 管理的配置文件，在你做 puppet 推送之后，那个管理员的配置文件被还原了。而那个管理员发现应用出了问题，要求你给他还原，你如何做呢？

答案是可以通过 filebucket 功能完成，那么那么多机器，如何集中管理这些单版本呢。

1、首先定义 filebucket 模块

要求所有节点都将被修改的配置文件保存至本地的

`/puppet/version/hosts/${fact_certname}`，其中变量 `fact_certname` 为获取的 `certname` 名自定义的 `fact` 变量，如何自定义 `fact` 可参考 <http://kisskisspuppet.com/2013/11/15/my-fact/>

```
[root@linuxmaster1poc manifests]# cat site.pp
import 'nodes/*.pp'
$puppetserver = 'puppetmaster.kisspuppet.com'
filebucket { 'main':
  path => "/puppet/version/hosts/${fact_certname}",
}
```

备注：有以上模块之后，只要在 `file` 模块中添加 `backup => 'main'`，那么这个文件就会被保存到以上目录中。

2、定义 clientbucketdir 和 bucketdir

参数 `clientbucketdir` 为当 `file` 模块中没有定义 `backup => 'main'` 时被保存的路径，默认在 `/var/lib/puppet/clientbucket`

参数 `bucketdir` 为使用 `puppet filebucket backup [file]` 保存的路径

```
[root@linux57poc ~]# cat /etc/puppet/puppet.conf
[main]
  user = puppet
  group = puppet
  vardir = /var/lib/puppet
  logdir = /var/log/puppet
```

```
rundir = /var/run/puppet
ssldir = $vardir/ssl
pluginsync = true
clientbucketdir = /puppet/version/hosts/puppet_linux57poc.kisspuppet.com
bucketdir = /puppet/version/hosts/puppet_linux57poc.kisspuppet.com
[agent]
  server = puppetmaster.kisspuppet.com
  classfile = $vardir/classes.txt
  localconfig = $vardir/localconfig
  runinterval=86400
  report = true
  authconfig = /etc/puppet/namespaceauth.conf
  usecacheonfailure = false
  certname = puppet_linux57poc.kisspuppet.com
  environment = production
# listen = true
# puppetport = 8139
# bindaddress = 192.168.100.125
```

3、file 模块中添加 `backup => 'main',`

```
class puppet_linux57poc{
  include puppet_linux57poc::motd,puppet_linux57poc::facts
}

class puppet_linux57poc::motd{
  package{ setup:
    ensure => present,
  }
  file{ "/etc/motd":
    owner   => "root",
    group   => "root",
    mode    => 0400,
    content => template("puppet_linux57poc/motd.erb"),
    backup  => 'main',
    require => Package["setup"],
  }
}

class puppet_linux57poc::facts{
  file{ "/etc/mcollective/facts.txt":
    owner   => "root",
    group   => "root",
```

```
mode    => 0400,
content => template("puppet_linux57poc/facts.txt.erb"),
backup  => 'main',
      }
}
```

4、测试 filebucket 的恢复功能

修改节点被管理的配置文件/etc/motd,然后运行 puppet 命令还原

```
[root@linux57poc ~]# echo aa>>/etc/motd
[root@linux57poc ~]# puppet agent -t
info: Retrieving plugin
info: Loading facts in /var/lib/puppet/lib/facter/fact_apply.rb
info: Caching catalog for puppet_linux57poc.kisspuppet.com
info: Applying configuration version '1386061716'
notice: /Stage[main]/Mcollective::Facter/File[/etc/mcollective/facts.yaml]/content:
--- /etc/mcollective/facts.yaml 2013-12-03 17:26:35.000000000 +0800
+++ /tmp/puppet-file20131203-14159-i1qgrq-0 2013-12-03 17:56:59.000000000 +0800
@@ -1,89 +1,89 @@
. . .

info: FileBucket adding {md5}3997ee041b3277fda12dbd849fac47e7
info: /Stage[main]/Mcollective::Facter/File[/etc/mcollective/facts.yaml]:
Filebucketed /etc/mcollective/facts.yaml to main with sum
3997ee041b3277fda12dbd849fac47e7
notice: /Stage[main]/Puppet_linux57poc::Motd/File[/etc/motd]/content:
--- /etc/motd 2013-12-03 17:56:49.000000000 +0800
+++ /tmp/puppet-file20131203-14159-1c90psx-0 2013-12-03 17:56:59.000000000 +0800
@@ -7,4 +7,3 @@
  rubyversion = 1.8.7
  .....
-----
-aa

info: FileBucket adding {md5}6db65f0eb756ee96ef8e615e914fa0ee
info: /Stage[main]/Puppet_linux57poc::Motd/File[/etc/motd]: Filebucketed /etc/motd
to main with sum 6db65f0eb756ee96ef8e615e914fa0ee
notice: /Stage[main]/Puppet_linux57poc::Motd/File[/etc/motd]/content: content
changed '{md5}6db65f0eb756ee96ef8e615e914fa0ee' to
'{md5}7c6613a110541a050c8a8f51fc89dab2'
notice: Finished catalog run in 0.69 seconds
```

可以看到/etc/motd 的 MD5 值从 '6db65f0eb756ee96ef8e615e914fa0ee' 变成 '7c6613a110541a050c8a8f51fc89dab2'

```
[root@linux57poc ~]# ll
/puppet/version/hosts/puppet_linux57poc.kisspuppet.com/6/d/b/6/5/f/0/e/6db65f0eb756
ee96ef8e615e914fa0ee/
total 8
-r--r----- 1 nfsnobody nfsnobody 311 Dec  3 17:56 contents
-rw-r----- 1 nfsnobody nfsnobody  10 Dec  3 17:56 paths
```

通过 puppet filebucket 恢复

```
[root@linux57poc ~]# puppet filebucket --local backup /etc/motd
/etc/motd: 6db65f0eb756ee96ef8e615e914fa0ee

[root@linux57poc ~]# ll
/puppet/version/hosts/puppet_linux57poc.kisspuppet.com/6/d/b/6/5/f/0/e/6db65f0eb756
ee96ef8e615e914fa0ee/
total 8
-r--r----- 1 nfsnobody nfsnobody 311 Dec  3 17:56 contents
-rw-r----- 1 nfsnobody nfsnobody  10 Dec  3 17:56 paths
[root@linux57poc ~]#

[root@linux57poc ~]# >/etc/motd
[root@linux57poc ~]# puppet filebucket --local restore /etc/motd
6db65f0eb756ee96ef8e615e914fa0ee
[root@linux57poc ~]# cat /etc/motd
-----a few of facter values-----
myhostname = linux57poc
eth0_ip = 192.168.100.125
kernel = 2.6.18-274.el5
system release = Red Hat Enterprise Linux Server release 5.7 (Tikanga)
puppetversion = 2.7.23
rubyversion = 1.8.7
.....
-----
aa
[root@linux57poc ~]#
```

如何集中管理呢，可以将所有节点那个目录挂在到一台服务器上，这样就集中了，然后通过扫描命令进行查看

```
[root@linuxreportpoc ~]# ll /puppet/version/hosts
total 12
drwxr-x--- 12 nfsnobody nfsnobody 4096 Dec  3 17:56 puppet_linux57poc.kisspuppet.com
drwxr-x--- 10 nfsnobody nfsnobody 4096 Dec  3 17:08 puppet_linux58poc.kisspuppet.com
```

```
drwxr-x--- 11 nfsnobody nfsnobody 4096 Dec  3 17:09 puppet_linux64poc.kisspuppet.com
```

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，投稿邮箱：admin@kisspuppet.com

微信公众号：“**puppet2014**”，可搜索加入，也可以扫描以下二维码

