通过自定义 fact 增强 MCollective 推送更新元数据的灵活性

目前由于 Facter 并不全面，许多关于主机和环境的信息并没有作为 Facter 的 fact。编写自定义的 fact，可以让节点的 facter 包含更多的元数据 fact，增加 MCollective 选择元数据定位主机的灵活性。

1 自定义节点变量

首选，需要在每个节点自定义一个 facts 文档，文档中包含了每个节点自定义的 fact 信息。为了方便管理，所有变量的值都必须事先定义好，可在 puppet 服务端定义一个 fact 变量列表，里面包含所有节点的自定义 fact 信息。然后，节点根据各自的主机特性选择合适的 fact 信息。

```
[root@puppetserver ~]# vim /etc/mcollective/facts.txt #收集并定义所有节点的 fact 信息，仅仅作为查看用
fact_certname=<自定义>
fact_apply1=apache
fact_apply2=php
fact_apply3=mysql
fact_apply4=java
fact_apply5=tomcat
fact_apply6=oracle
fact_apply7=nginx
fact_apply8=jboss
fact_apply9=haproxy
fact_apply10=db2
…
```

```
[root@agent1 ~]# cat /etc/mcollective/facts.txt #假设 agent1 节点具有以下 fact 变量信息
fact_certname=agent1.kisspuppet.com #puppet 认证用，可写成其他名称
fact_apply3=mysql
fact_apply4=java
fact_apply10=db2
[root@agent2 ~]# cat /etc/mcollective/facts.txt #假设 agent1 节点具有以下 fact 变量信息
fact_certname=agent2.kisspuppet.com #puppet 认证用，可写成其他名称
fact_apply2=php
fact_apply3=mysql
fact_apply7=nginx
```

2 创建 file 资源模块

由于自定义 fact 信息属于每个节点的特性，放在 agents（存放单个节点个性模块的目录）目录中，可将这部分定义成一个 class 包含到每个节点的 class agentN{}中。

```
[root@puppetserver ~]# cat /etc/puppet/agents/modules/agent1/manifests/init.pp
class agent1{
  include agent1::facts
}
class agent1::facts{
  file{ "/etc/mcollective/facts.txt":
    owner   => "root",
    group   => "root",
    mode    => 0400,
    content => template("agent1/facts.txt.erb"),
    backup  => 'main',
  }
}
...
```

```
[root@puppetserver agents]# cat modules/agent1/templates/facts.txt.erb
------------Some custom facts variables-------------
fact_certname=agent1.kisspuppet.com
fact_apply3=mysql
fact_apply4=java
fact_apply10=db2
...
-----------------------------------------------------

[root@puppetserver agents]# cat modules/agent1/manifests/init.pp
class agent1{
  include agent1::facts
}
class agent1::facts{
  file{ "/etc/mcollective/facts.txt":
    owner   => "root",
    group   => "root",
    mode    => 0400,
    content => template("agent1/facts.txt.erb"),
    backup  => 'main',
  }
```

```
}
...
[root@puppetserver agents]# cat modules/agent2/templates/facts.txt.erb
------------Some custom facts variables-------------
fact_certname=agent2.kisspuppet.com
fact_apply2=php
fact_apply3=mysql
fact_apply7=nginx
...
-----------------------------------------------------
```

3 创建 fact 模块

3.1 创建全局模块

新建一个模块可命名为 public，放在 environment（存放基础环境的模块）模块中，自定义 fact fact_apply.rb（过滤各自的自定义 fact 信息）

```
[root@puppetserver puppet]# cat environment/modules/public/lib/facter/fact_apply.rb
# certname is usered for /etc/puppet/puppet.conf
Facter.add("fact_certname") do
   setcode do
       Facter::Util::Resolution.exec("/bin/grep 'fact_certname='
/etc/mcollective/facts.txt |awk -F= '{print $2}'")
     end
end
# fact_apply1~N.rb
#
Facter.add("fact_apply1") do
   setcode do
       Facter::Util::Resolution.exec("/bin/grep 'fact_apply1='
/etc/mcollective/facts.txt |awk -F= '{print $2}'")
     end
end

Facter.add("fact_apply2") do
   setcode do
       Facter::Util::Resolution.exec("/bin/grep 'fact_apply2='
/etc/mcollective/facts.txt |awk -F= '{print $2}'")
     end
end
…
Facter.add("fact_apply10") do
```

```
   setcode do
       Facter::Util::Resolution.exec("/bin/grep 'fact_apply10='
/etc/mcollective/facts.txt |awk -F= '{print $2}'")
     end
end
```

3.2 设置局部模块

如果自定义的 fact 属于某一个模块下具有的特性，只需要将 fact 信息定义到对应的模块中即可，无需创建全局 fact 模块，比如放在 mysql 模块中等。

4 开启模块插件功能

当 pluginsync 选项设置为 true 后，就打开了“模块中的插件”功能。当 agent 连接到 master 时，每一个 agent 都会检查他们的模块中的自定义代码。Puppet 会将这些自定义代码同步到相关的 agent 中。然后他们就能在这些 agent 中使用了。

```
[root@puppetserver ~]# vim /etc/puppet/puppet.conf
[main]
pluginsync = true
…
[root@agent2 ~]# vim /etc/puppet/puppet.conf
[main]
pluginsync = true
…
```

5 节点上测试自定义 fact

5.1 节点运行 puppet 命令更新

```
[root@agent2 ~]# puppet agent --test
info: Retrieving plugin
notice: /File[/var/lib/puppet/lib/facter/fact_apply.rb]/ensure: defined content as
'{md5}03bdfe12d6f40fb8abe0bd407dab6d69'
info: Loading downloaded plugin /var/lib/puppet/lib/facter/fact_apply.rb #自动下载
info: Loading facts in /var/lib/puppet/lib/facter/backup_date.rb
info: Loading facts in /var/lib/puppet/lib/facter/fact_apply.rb #自动载入
info: Caching catalog for agent2.kisspuppet.com
info: Applying configuration version '1381211740'
```

5.2 通过节点查看自定义 fact 是否生效

```
[root@agent2 ~]# facter -p | grep fact_
fact_apply2 => php
fact_apply3 => mysql
```

```
fact_apply7 => nginx
fact_certname => agent1.kisspuppet.com
[root@agent1 facter]# facter -p | grep fact_
fact_certname => agent2.kisspuppet.com
fact_apply10 => db2
fact_apply3 => mysql
fact_apply4 => java
```

6 MCollective 客户端测试自定义 fact

```
[root@puppetserver facter]# mco inventory agent1.kisspuppet.com | grep fact_
      fact_apply10 => db2
      fact_apply3 => mysql
      fact_apply4 => java
fact_certname => agent1.kisspuppet.com
[root@puppetserver facter]# mco inventory agent2.kisspuppet.com | grep fact_
      fact_apply2 => php
      fact_apply3 => mysql
      fact_apply7 => nginx
fact_certname => agent2.kisspuppet.com
```

7 通过自定义 facter 定位主机触发更新

7.1 触发更新 fact_apply4=java 的主机

自定义 fact fact_apply4='java'的主机目前只有 agent1

```
[root@puppetserver facter]# mco puppet -v runonce  mco facts -v --with-fact
fact_apply4='java'
Discovering hosts using the mc method for 2 second(s) .... 1
 * [ ============================================================> ] 1 / 1
agent1.kisspuppet.com                    : OK
    {:summary=>      "Started a background Puppet run using the 'puppet agent --
onetime --daemonize --color=false --splay --splaylimit 30' command"}
---- rpc stats ----
           Nodes: 1 / 1
     Pass / Fail: 1 / 0
      Start Time: Tue Oct 08 14:24:08 +0800 2013
  Discovery Time: 2003.39ms
      Agent Time: 1091.75ms
      Total Time: 3095.14ms
```

7.2 触发更新 fact_apply3=mysql 和系统为 RHEL5.7 的主机

自定义 fact fact_apply3='mysql'的主机有 agent1 和 agent2，系统为 RHEL5.7 的主机只有 agent2(通过系统自带 fact 获取)，取交集，只有 agent2 会被触发更新。

```
[root@puppetserver facter]# mco puppet -v runonce   rpc --np -F
operatingsystemrelease='5.7' -F  fact_apply3='mysql'
Discovering hosts using the mc method for 2 second(s) .... 1
agent2.kisspuppet.com                    : OK
   {:summary=>     "Started a background Puppet run using the 'puppet agent --
onetime --daemonize --color=false --splay --splaylimit 30' command"}
---- rpc stats ----
          Nodes: 1 / 1
    Pass / Fail: 1 / 0
     Start Time: Tue Oct 08 14:22:23 +0800 2013
  Discovery Time: 2004.56ms
     Agent Time: 1092.00ms
     Total Time: 3096.56ms
```

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