基于 mcollective 的多种 plugins 组合使用功能展示

前一篇文章介绍了 mcollective 的 shell 插件，确实很强大
http://kisspuppet.com/2013/12/13/mcollective*plugins*shell/

接下来介绍 mcollective 的其它 plugins

首先去官网下载各个插件 http://yum.puppetlabs.com

1、在 mcollective client 端和 server 端安装各种 plugins

mcollective-client 端

```
[root@linuxmaster1poc ~]# rpm -qa | grep mco
mcollective-service-common-3.1.2-1.noarch
mcollective-client-2.2.4-1.el6.noarch
mcollective-service-client-3.1.2-1.noarch
mcollective-common-2.2.4-1.el6.noarch
mcollective-iptables-common-3.0.1-1.noarch
mcollective-filemgr-client-1.0.1-1.noarch
mcollective-nrpe-client-3.0.2-1.noarch
mcollective-puppet-client-1.6.0-1.noarch
mcollective-nrpe-common-3.0.2-1.noarch
mcollective-filemgr-common-1.0.1-1.noarch
mcollective-iptables-client-3.0.1-1.noarch
mcollective-puppet-common-1.6.0-1.noarch
mcollective-facter-facts-1.0.0-1.noarch
mcollective-package-client-4.2.0-1.noarch
mcollective-package-common-4.2.0-1.noarch
```

mcollecitve-server 端

```
[root@linux57poc ~]# rpm -qa | grep mco
mcollective-nrpe-common-3.0.2-1
mcollective-puppet-common-1.6.0-1
mcollective-iptables-common-3.0.1-1
mcollective-iptables-agent-3.0.1-1
mcollective-2.2.4-1.el5
mcollective-package-common-4.2.0-1
mcollective-service-common-3.1.2-1
mcollective-service-agent-3.1.2-1
mcollective-puppet-agent-1.6.0-1
mcollective-package-agent-4.2.0-1
```

```
mcollective-filemgr-common-1.0.1-1
mcollective-common-2.2.4-1.el5
mcollective-facter-facts-1.0.0-1
mcollective-filemgr-agent-1.0.1-1
mcollective-nrpe-agent-3.0.2-1
```

以上安装可写个 package 模块执行，以下只针对 mcollective server 端，安装完成之后记得重启服务，如果写了 service 模块可以自动刷新

plugins.pp

```
class mcollective::plugins{
  include mcollective::plugins_puppet,
    mcollective::plugins_facter,
    mcollective::plugins_filemgr,
    mcollective::plugins_iptables,
#   mcollective::plugins_nettest, #这个安装需要依赖包 ruby-net-ping，没找到
    mcollective::plugins_nrpe,
    mcollective::plugins_package,
    mcollective::plugins_service
}

#mco-client need install mcollective-puppet-client and mcollective-puppet-common
class mcollective::plugins_puppet{
  package { ['mcollective-puppet-agent','mcollective-puppet-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}



#mco-client need install mcollective-facter-facts
class mcollective::plugins_facter{
  package { 'mcollective-facter-facts':
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}

#mco-client need install mcollective-filemgr-client and mcollective-filemgr-common
class mcollective::plugins_filemgr{
  package { ['mcollective-filemgr-agent','mcollective-filemgr-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}
```

```
#mco-client need install mcollective-iptables-client and mcollective-iptables-common
class mcollective::plugins_iptables{
  package { ['mcollective-iptables-agent','mcollective-iptables-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}

#mco-client need install mcollective-nettest-client and mcollective-nettest-common
class mcollective::plugins_nettest{
  package { ['mcollective-nettest-agent','mcollective-nettest-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}

#mco-client need install mcollective-nrpe-client and mcollective-nrpe-common
class mcollective::plugins_nrpe{
  package { ['mcollective-nrpe-agent','mcollective-nrpe-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}

#mco-client need install mcollective-package-client and mcollective-package-common
class mcollective::plugins_package{
  package { ['mcollective-package-agent','mcollective-package-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}

#mco-client need install mcollective-service-client and mcollective-service-common
class mcollective::plugins_service{
  package { ['mcollective-service-agent','mcollective-service-common']:
    ensure  => installed,
    require => Class["mcollective::install"]
  }
}
```

conf.pp

```
class mcollective::service{
  service { 'mcollective':
    ensure     => running,
    hasstatus  => true,
    hasrestart => true,
    enable     => true,
    subscribe  => Class['mcollective::config'],
  }
}
```

mcollective-client 端安装好之后，可通过 mco 命令查看

```
[root@linuxmaster1poc ~]# mco
The Marionette Collective version 2.2.4

usage: /usr/bin/mco command <options>

Known commands:

   completion           facts               filemgr
   find                 help                inventory
   iptables             nrpe                package
   ping                 plugin              puppet
   rpc                  service             shell

Type '/usr/bin/mco help' for a detailed list of commands and '/usr/bin/mco help
command'
to get detailed help for a command
```

mcollective-server 端安装好之后，可在 mco-client 端查看

```
[root@linuxmaster1poc ~]# mco inventory linux57poc
Inventory for linux57poc:

   Server Statistics:
                    Version: 2.2.4
                 Start Time: Fri Dec 13 08:15:46 +0800 2013
                Config File: /etc/mcollective/server.cfg
                Collectives: mcollective
            Main Collective: mcollective
                 Process ID: 23268
             Total Messages: 16
    Messages Passed Filters: 16
          Messages Filtered: 0
           Expired Messages: 0
               Replies Sent: 15
        Total Processor Time: 0.71 seconds
                System Time: 0.15 seconds

   Agents: #都加载上了
    discovery       filemgr         nrpe
    package         puppet          rpcutil
    service         shell

   Data Plugins:
    agent           fstat           nrpe
    puppet          resource        service

   Configuration Management Classes:
    No classes applied

   Facts:
    architecture => x86_64
    augeasversion => 0.10.0
    bios_release_date => 06/22/2012
    bios_vendor => Phoenix Technologies LTD
    bios_version => 6.00
    blockdevice_fd0_size => 4096
    blockdevice_hdc_size => 3834736640
。。。
```

注意： 接下来测试各种命令的操作组合，这里只举一些例子，更多信息可参考--help 或者参考官网

2、组合 mcollective 各种 plugins 完成各种任务组合

2.1、停止操作系统为 RHEL5.x 服务器的 crond 任务

先查看 5.x 系统 crond 的状态，使用插件 service、facts

```
[root@linuxmaster1poc ~]# mco service crond status -F operatingsystemmajrelease=5

 * [ ============================================================> ] 2 / 2

   linux57poc: running
   linux58poc: running

Summary of Service Status:
   running = 2


Finished processing 2 / 2 hosts in 184.79 ms
```

然后通过 service 插件停止服务，使用插件 service、facts

```
[root@linuxmaster1poc ~]# mco service crond stop -F operatingsystemmajrelease=5

 * [ ============================================================> ] 2 / 2



Summary of Service Status:

   stopped = 2



Finished processing 2 / 2 hosts in 914.76 ms
```

再次查看过滤的主机 crond 服务是否被停掉，使用插件 service、facts

```
[root@linuxmaster1poc ~]# mco service crond status -F operatingsystemmajrelease=5

 * [ ============================================================> ] 2 / 2

   linux57poc: stopped
   linux58poc: stopped

Summary of Service Status:

   stopped = 2



Finished processing 2 / 2 hosts in 125.87 ms
```

也可以通过 shell 插件实现，使用到插件为 shell、service、facts

```
[root@linuxmaster1poc ~]# mco shell "service crond status" -F
operatingsystemmajrelease=5
Discovering hosts using the mc method for 2 second(s) .... 2
Host: linux57poc
Statuscode: 3
```

```
Output:
crond is stopped
Host: linux58poc
Statuscode: 3
Output:
crond is stopped
```

2.2、使用 mco 对自定义 fact_apply4=app 的主机做一次变更，要求环境为 testing，模式为 noop

首先查看下那些主机具备有这个自定义 fact，使用的插件为 find、inventory

```
[root@linuxmaster1poc ~]# for i in `mco find` ; do echo $i; mco inventory $i | grep
fact_apply4; done
linux58poc
  fact_apply4 => app
linux57poc
linux64poc
  fact_apply4 => app
```

其次按要求做变更即可，使用到的插件为 puppet，facts

```
[root@linuxmaster1poc ~]# mco puppet -v runonce --environment=testing --noop -F
fact_apply4=app
Discovering hosts using the mc method for 2 second(s) .... 2

 * [ =============================================================> ] 2 / 2



linux64poc                            : OK
    {:summary=>      "Started a background Puppet run using the 'puppet agent --
onetime --daemonize --color=false --splay --splaylimit 30 --noop --environment
testing' command"}

linux58poc                            : OK
    {:summary=>      "Started a background Puppet run using the 'puppet agent --
onetime --daemonize --color=false --splay --splaylimit 30 --noop --environment
testing' command"}




---- rpc stats ----
          Nodes: 2 / 2
     Pass / Fail: 2 / 0
```

```
      Start Time: Fri Dec 13 09:10:50 +0800 2013
  Discovery Time: 2003.32ms
      Agent Time: 884.34ms
      Total Time: 2887.67ms
```

变更完成后，迅速查看节点运行情况，使用到的插件为 puppet

```
[root@linuxmaster1poc ~]# mco puppet status

 * [ ============================================================> ] 3 / 3

   linux64poc: Currently idling; last completed run 54 seconds ago
   linux58poc: Currently applying a catalog; last completed run 1 minutes 12 seconds
ago
   linux57poc: Currently stopped; last completed run 22 minutes 57 seconds ago

Summary of Applying:

   false = 2
    true = 1

Summary of Daemon Running:

   running = 2
   stopped = 1

Summary of Enabled:

   enabled = 3

Summary of Idling:

   false = 2
    true = 1

Summary of Status:

               idling = 1
              stopped = 1
   applying a catalog = 1


Finished processing 3 / 3 hosts in 263.72 ms
```

3、远程改所有系统为 RHEL6.4 主机 root 的密码，使用到的插件为 shell，facts

```
[root@linuxmaster1poc ~]# mco shell "echo redhat | passwd root --stdin" -F
operatingsystemrelease=6.4
Discovering hosts using the mc method for 2 second(s) .... 1
Host: linux64poc
Statuscode: 0
Output:
Changing password for user root.
passwd: all authentication tokens updated successfully.
```

4、查看所有节点 puppet 和 facter 安装包的版本信息，使用到的插件为 package

```
[root@linuxmaster1poc ~]# mco package status puppet

 * [ ============================================================> ] 3 / 3

   linux64poc: puppet-2.7.23-1.el6.noarch
   linux57poc: puppet-2.7.23-1.el5.noarch
   linux58poc: puppet-2.7.23-1.el5.noarch

Summary of Arch:

   noarch = 3

Summary of Ensure:

   2.7.23-1.el5 = 2
   2.7.23-1.el6 = 1


Finished processing 3 / 3 hosts in 635.21 ms

[root@linuxmaster1poc ~]# mco package status facter

 * [ ============================================================> ] 3 / 3

   linux58poc: facter-1.7.3-1.el5.x86_64
   linux64poc: facter-1.7.3-1.el6.x86_64
   linux57poc: facter-1.7.3-1.el5.x86_64

Summary of Arch:

   x86_64 = 3
```

```

Summary of Ensure:

   1.7.3-1.el5 = 2
   1.7.3-1.el6 = 1


Finished processing 3 / 3 hosts in 124.99 ms
```

更多的功能可通过以下方式查看：

```
[root@linuxmaster1poc ~]# mco puppet -h

Schedule runs, enable, disable and interrogate the Puppet Agent

Usage: mco puppet [OPTIONS] [FILTERS] <ACTION> [CONCURRENCY|MESSAGE]
Usage: mco puppet <count|enable|status|summary>
Usage: mco puppet disable [message]
Usage: mco puppet runonce [PUPPET OPTIONS]
Usage: mco puppet resource type name property1=value property2=value
Usage: mco puppet runall [--rerun SECONDS] [PUPPET OPTIONS]

The ACTION can be one of the following:

    count    - return a total count of running, enabled, and disabled nodes
    enable   - enable the Puppet Agent if it was previously disabled
    disable  - disable the Puppet Agent preventing catalog from being applied
    resource - manage individual resources using the Puppet Type (RAL) system
    runall   - invoke a puppet run on matching nodes, making sure to only run
               CONCURRENCY nodes at a time
    runonce  - invoke a Puppet run on matching nodes
    status   - shows a short summary about each Puppet Agent status
    summary  - shows resource and run time summaries

        --force                      Bypass splay options when running
        --server SERVER               Connect to a specific server or port
        --tags, --tag TAG             Restrict the run to specific tags
        --noop                       Do a noop run
        --no-noop                    Do a run with noop disabled
        --environment ENVIRONMENT     Place the node in a specific environment for
this run
        --splay                      Splay the run by up to splaylimit seconds
        --no-splay                   Do a run with splay disabled
        --splaylimit SECONDS          Maximum splay time for this run if splay is set
        --ignoreschedules            Disable schedule processing
        --rerun SECONDS              When performing runall do so repeatedly with a
minimum run time of SECONDS

        --np, --no-progress          Do not show the progress bar
    -1, --one                        Send request to only one discovered nodes
        --batch SIZE                 Do requests in batches
        --batch-sleep SECONDS        Sleep time between batches
        --limit-seed NUMBER          Seed value for deterministic random batching
        --limit-nodes, --ln, --limit COUNT
                                     Send request to only a subset of nodes, can be a
percentage
    -j, --json                       Produce JSON output
        --display MODE               Influence how results are displayed. One of ok,
all or failed
    -c, --config FILE                Load configuratuion from file rather than
default
    -v, --verbose                    Be verbose
    -h, --help                       Display this screen

Common Options
    -T, --target COLLECTIVE          Target messages to a specific sub collective
        --dt, --discovery-timeout SECONDS
                                     Timeout for doing discovery
    -t, --timeout SECONDS            Timeout for calling remote agents
    -q, --quiet                      Do not be verbose
        --ttl TTL                    Set the message validity period
        --reply-to TARGET            Set a custom target for replies
        --dm, --disc-method METHOD   Which discovery method to use
        --do, --disc-option OPTION   Options to pass to the discovery method
        --nodes FILE                 List of nodes to address

Host Filters
    -W, --with FILTER                Combined classes and facts filter
    -S, --select FILTER              Compound filter combining facts and classes
    -F, --wf, --with-fact fact=val   Match hosts with a certain fact
    -C, --wc, --with-class CLASS     Match hosts with a certain config management
class
    -A, --wa, --with-agent AGENT     Match hosts with a certain agent
    -I, --wi, --with-identity IDENT  Match hosts with a certain configured identity

The Marionette Collective 2.2.4
```

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
