

puppet 实战-puppet agent 命令的中文帮助信息详解

`puppet agent` 在每个节点以守护进程方式运行，默认为每隔 30 分钟向 Master 请求一次，以确认 新的信息并询问是否有变更，然后负责运行编译好的 Catalog 代码。如果在节点配置文件（`puppet.conf`）中配置了"`pluginsync = true`",Agent 同样复制啦取自定义 facts 并处理。除此之外，如果更多考虑安全性，可以关闭 `puppet agent` 的守护进程，通过 `mcollective+MQ` 框架调用 `puppet agent` 命令实现一次性更新。有关 `mcollective+MQ` 框架信息可参考 http://www.rsyslog.org/categories/MCollective_Technology/

命令: `puppet agent` #客户端进程，负责从 Master 获取数据

```
--certname <name> #指定客户端 certname（唯一 ID），通常以 FQDN 命名
--daemonize #发送到后台守护进程，默认选项
--no-daemonize #不发送到后台守护进程
-d|--debug #启用完整的调试模式
--detailed-exitcodes #提供详细的退出代码
--digest <digest> #指定证书指纹算法，默认为 MD5 算法
--disable #禁用，禁止 puppet agent 在此节点运行
--enable #启用，重新运行执行 puppet agent
--fingerprint #显示当前证书的指纹
-h|--help #查看帮助
-l|--logdest syslog|<file>|console #日志发送方式，默认采用 syslog 配置
--no-client #不创建客户端配置文件，当 listen=true 时才有意义
--noop #puppet 运行 Catalog，但不执行配置
-o|--onetime #运行一次，配合--no-daemonize 使用
--serve <handler> #启动另一类的服务
-t|--test #测试用，包含了（'onetime','verbose','ignorecache','no-daemonize','no-usecacheonfailure','detailed-exit-codes','no-splay', and 'show_diff'）
-v|--verbose #显示详细信息
-V|--version #打印 puppet 版本
-w|--waitforcert <seconds> #当 Master 未签署此节点证书时，puppet agent 将等待签署，并默认每 2 分钟重新连接 Master 以确定是否完成签署
```

```
[root@linux57poc ~]# puppet help agent
```

```
puppet-agent(8) -- The puppet agent daemon
```

```
=====
```

```
SYNOPSIS
```

```
-----
```

```
Retrieves the client configuration from the puppet master and applies it to the local host.
```

This service may be run as a daemon, run periodically using cron (or something similar), or run interactively for testing purposes.

USAGE

```
puppet agent [--certname <name>] [-D|--daemonize|--no-daemonize]
  [-d|--debug] [--detailed-exitcodes] [--digest <digest>] [--disable] [--enable]
  [--fingerprint] [-h|--help] [-l|--logdest syslog|<file>|console]
  [--no-client] [--noop] [-o|--onetime] [--serve <handler>] [-t|--test]
  [-v|--verbose] [-V|--version] [-w|--waitforcert <seconds>]
```

DESCRIPTION

This is the main puppet client. Its job is to retrieve the local machine's configuration from a remote server and apply it. In order to successfully communicate with the remote server, the client must have a certificate signed by a certificate authority that the server trusts; the recommended method for this, at the moment, is to run a certificate authority as part of the puppet server (which is the default). The client will connect and request a signed certificate, and will continue connecting until it receives one.

Once the client has a signed certificate, it will retrieve its configuration and apply it.

USAGE NOTES

'puppet agent' does its best to find a compromise between interactive use and daemon use. Run with no arguments and no configuration, it will go into the background, attempt to get a signed certificate, and retrieve and apply its configuration every 30 minutes.

Some flags are meant specifically for interactive use -- in particular, 'test', 'tags' or 'fingerprint' are useful. 'test' enables verbose logging, causes the daemon to stay in the foreground, exits if the server's configuration is invalid (this happens if, for instance, you've left a syntax error on the server), and exits after running the configuration once (rather than hanging around as a long-running process).

'tags' allows you to specify what portions of a configuration you want to apply. Puppet elements are tagged with all of the class or definition names that contain them, and you can use the 'tags' flag to specify one of these names, causing only configuration elements contained within that class or definition to be applied. This is very useful when you are testing new configurations -- for instance, if you are just starting to manage 'ntpd', you would put all of the new elements into an 'ntpd' class, and call puppet with '--tags ntpd', which would only apply that small portion of the configuration during your testing, rather than applying the whole thing.

'fingerprint' is a one-time flag. In this mode 'puppet agent' will run once and display on the console (and in the log) the current certificate (or certificate request) fingerprint. Providing the '--digest' option allows to use a different digest algorithm to generate the fingerprint. The main use is to verify that before signing a certificate request on the master, the certificate request the master received is the same as the one the client sent (to prevent against man-in-the-middle attacks when signing certificates).

OPTIONS

Note that any configuration parameter that's valid in the configuration file is also a valid long argument. For example, 'server' is a valid configuration parameter, so you can specify '--server <servername>' as an argument.

See the configuration file documentation at <http://docs.puppetlabs.com/references/stable/configuration.html> for the full list of acceptable parameters. A commented list of all configuration options can also be generated by running puppet agent with '--genconfig'.

* --certname:

Set the certname (unique ID) of the client. The master reads this unique identifying string, which is usually set to the node's fully-qualified domain name, to determine which configurations the node will receive. Use this option to debug setup problems or implement unusual node identification schemes.

* --daemonize:

Send the process into the background. This is the default.

```
* --no-daemonize:
    Do not send the process into the background.

* --debug:
    Enable full debugging.

* --detailed-exitcodes:
    Provide transaction information via exit codes. If this is enabled, an exit
    code of '2' means there were changes, an exit code of '4' means there were
    failures during the transaction, and an exit code of '6' means there were both
    changes and failures.

* --digest:
    Change the certificate fingerprinting digest algorithm. The default is
    MD5. Valid values depends on the version of OpenSSL installed, but
    should always at least contain MD5, MD2, SHA1 and SHA256.

* --disable:
    Disable working on the local system. This puts a lock file in place,
    causing 'puppet agent' not to work on the system until the lock file
    is removed. This is useful if you are testing a configuration and do
    not want the central configuration to override the local state until
    everything is tested and committed.

    'puppet agent' uses the same lock file while it is running, so no more
    than one 'puppet agent' process is working at a time.

    'puppet agent' exits after executing this.

* --enable:
    Enable working on the local system. This removes any lock file,
    causing 'puppet agent' to start managing the local system again
    (although it will continue to use its normal scheduling, so it might
    not start for another half hour).

    'puppet agent' exits after executing this.

* --fingerprint:
    Display the current certificate or certificate signing request
    fingerprint and then exit. Use the '--digest' option to change the
    digest algorithm used.

* --help:
    Print this help message
```

* --logdest:

Where to send messages. Choose between syslog, the console, and a log file. Defaults to sending messages to syslog, or the console if debugging or verbosity is enabled.

* --no-client:

Do not create a config client. This will cause the daemon to run without ever checking for its configuration automatically, and only makes sense when puppet agent is being run with listen = true in puppet.conf or was started with the '--listen' option.

* --noop:

Use 'noop' mode where the daemon runs in a no-op or dry-run mode. This is useful for seeing what changes Puppet will make without actually executing the changes.

* --onetime:

Run the configuration once. Runs a single (normally daemonized) Puppet run. Useful for interactively running puppet agent when used in conjunction with the --no-daemonize option.

* --serve:

Start another type of server. By default, 'puppet agent' will start a service handler that allows authenticated and authorized remote nodes to trigger the configuration to be pulled down and applied. You can specify any handler here that does not require configuration, e.g., filebucket, ca, or resource. The handlers are in 'lib/puppet/network/handler', and the names must match exactly, both in the call to 'serve' and in 'namespaceauth.conf'.

* --test:

Enable the most common options used for testing. These are 'onetime', 'verbose', 'ignorecache', 'no-daemonize', 'no-usecacheonfailure', 'detailed-exit-codes', 'no-splay', and 'show_diff'.

* --verbose:

Turn on verbose reporting.

* --version:

Print the puppet version number and exit.

* --waitforcert:

This option only matters for daemons that do not yet have certificates

and it is enabled by default, with a value of 120 (seconds). This causes 'puppet agent' to connect to the server every 2 minutes and ask it to sign a certificate request. This is useful for the initial setup of a puppet client. You can turn off waiting for certificates by specifying a time of 0.

EXAMPLE

```
$ puppet agent --server puppet.domain.com
```

DIAGNOSTICS

Puppet agent accepts the following signals:

- * SIGHUP:
Restart the puppet agent daemon.
- * SIGINT and SIGTERM:
Shut down the puppet agent daemon.
- * SIGUSR1:
Immediately retrieve and apply configurations from the puppet master.
- * SIGUSR2:
Close file descriptors for log files and reopen them. Used with logrotate.

AUTHOR

Luke Kanies

COPYRIGHT

Copyright (c) 2011 Puppet Labs, LLC Licensed under the Apache 2.0 License

eg. 节点以 noop 方式去同步 puppetmaster 的测试环境 testing

```
[root@linux57poc ~]# puppet agent --server=puppetmaster.kisspuppet.com --  
environment=testing --verbose --no-daemonize --onetime --noop  
info: Retrieving plugin  
info: Loading facts in /var/lib/puppet/lib/facter/fact_apply.rb  
info: Caching catalog for puppet_linux57poc.dev.shanghaigm.com  
info: Applying configuration version '1389088855'  
notice: /Stage[main]/Allservice::lm_sensors_service/Service[lm_sensors]/ensure:  
current_value running, should be stopped (noop)
```

```
notice: Class[Allservice::Lm_sensors_service]: Would have triggered 'refresh' from 1 events
notice: /Stage[main]/Allservice::Sendmail_service/Service[sendmail]/ensure:
current_value running, should be stopped (noop)
notice: Class[Allservice::Sendmail_service]: Would have triggered 'refresh' from 1 events
notice: /Stage[main]/Allservice::Restorecond_service/Service[restorecond]/ensure:
current_value running, should be stopped (noop)
notice: Class[Allservice::Restorecond_service]: Would have triggered 'refresh' from 1 events
notice: /Stage[main]/Allservice::Bluetooth_service/Service[hidd]/ensure:
current_value running, should be stopped (noop)
notice: Class[Mcollective::Facter]: Would have triggered 'refresh' from 1 events
notice: /Stage[main]/Sysctl::Exec/Exec[sysctl -p]/returns: current_value notrun,
should be 0 (noop)
notice: Class[Sysctl::Exec]: Would have triggered 'refresh' from 1 events
notice: /Stage[main]/Allservice::Bluetooth_service/Service[bluetooth]/ensure:
current_value running, should be stopped (noop)
notice: Class[Allservice::Bluetooth_service]: Would have triggered 'refresh' from 2 events
notice: Stage[main]: Would have triggered 'refresh' from 6 events
notice: Finished catalog run in 3.24 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

