Apache+Passenger 代替 puppet 自带低性能 web 服务 WEBRick

描述：puppet 使用 SSL(https)协议来进行通讯，默认情况下，puppet server 端使用基于 Ruby 的 WEBRick HTTP 服务器。由于 WEBRick HTTP 服务器在处理 agent 端的性能方面并不是很强劲，因此需要扩展 puppet，搭建 Apache 或者其他强劲的 web 服务器来处理客户的 https 请求。

需要解决的问题：

- 扩展传输方式：提高性能并增加 Master 和 agent 之间的并发连接数量。
- 扩展 SSL：采用良好的 SSL 证书管理方法来加密 Master 和 agent 之间的通讯。

参考：http://projects.puppetlabs.com/projects/1/wiki/Using_Passenger

1 使用 Ruby Gem 安装 Passenger

```
[root@puppetserver etc]# yum install ruby-devel ruby-libs rubygems libcurl-devel
[root@puppetserver etc]# yum install httpd httpd-devel apr-util-devel apr-devel
mod_ssl
[root@puppetserver repos]# gem install --local passenger-4.0.19.gem #自动解决依赖关系，进入 gem 包目录进行安装
Building native extensions.  This could take a while...
Successfully installed rake-10.0.1
Successfully installed daemon_controller-1.1.5
Successfully installed rack-1.5.2
Successfully installed passenger-4.0.19
```

2 整合 Apache 和 Passenger

```
[root@puppetserver rpms]# yum install gcc-c++  gcc openssl-devel #源码包编译安装（安装需要 apache gcc gcc-c++ openssl-devel 开发包的支持）
[root@puppetserver etc]# passenger-install-apache2-module #按照相关提示解决依赖关系，安装完成之后会显示
…
The Apache 2 module was successfully installed.
Please edit your Apache configuration file, and add these lines:
   LoadModule passenger_module /usr/lib/ruby/gems/1.8/gems/passenger-
4.0.19/buildout/apache2/mod_passenger.so
   PassengerRoot /usr/lib/ruby/gems/1.8/gems/passenger-4.0.19
   PassengerDefaultRuby /usr/bin/ruby

After you restart Apache, you are ready to deploy any number of Ruby on Rails
```

```
applications on Apache, without any further Ruby on Rails-specific
configuration!
…
```

3 配置 Apache 和 Passenger

创建虚拟主机并加载 passenger 相关模块，注意证书路径要和 puppet 实际证书路径对应。虚拟主机配置 Apache 以监听在 8140 端口，并且使用 SSL 和 Puppet Master 生成的证书对所有通讯进行加密。同时还将配置 Passenger 来使系统的 Ruby 解释器并且提供 Rack 配置文件 `config.ru` 的路径

```
[root@puppetserver conf.d]# vim passenger.conf
LoadModule passenger_module /usr/lib/ruby/gems/1.8/gems/passenger-
4.0.19/buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
   PassengerRoot /usr/lib/ruby/gems/1.8/gems/passenger-4.0.19
   PassengerRuby /usr/bin/ruby
   PassengerHighPerformance on
   PassengerMaxPoolSize 12
   PassengerPoolIdleTime 1500
   PassengerStatThrottleRate 120
 # RailsAutoDetect On
</IfModule>
Listen 8140  #监听 TCP 8140 端口，这是 PuppetMaster 服务器的标准端口
<VirtualHost *:8140>
        SSLEngine on  #开始 ssl 加密
        SSLProtocol -ALL +SSLv3 +TLSv1
        SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:-LOW:-SSLv2:-EXP #开启 ssl 加密

        SSLCertificateFile
/var/lib/puppet/ssl/certs/puppetserver.kisspuppet.com.pem
        SSLCertificateKeyFile
/var/lib/puppet/ssl/private_keys/puppetserver.kisspuppet.com.pem
        SSLCertificateChainFile /var/lib/puppet/ssl/ca/ca_crt.pem
        SSLCACertificateFile    /var/lib/puppet/ssl/ca/ca_crt.pem
        SSLCARevocationFile     /var/lib/puppet/ssl/ca/ca_crt.pem #打开证书撤销功能，
当我们颁发或撤销 Puppet agent 的证书时，Puppet cert 命令会自动更关心 ca_crl.pem 文件
        SSLVerifyClient optional
        SSLVerifyDepth  1
        SSLOptions +StdEnvVars #配置 Apache 来验证 Puppet agent 证书的真实性。验证的结果会
被保存在这个环境变量中，运行在 Passenger 中的 Puppet master 进程会使用这个变量来认证 Puppet
agent。

#Puppet agent 证书验证的结果会以客户端请求头的形式存放在标准环境中。
```

```
        RequestHeader unset X-Forwarded-For
        RequestHeader set X-SSL-Subject %{SSL_CLIENT_S_DN}e
        RequestHeader set X-Client-DN %{SSL_CLIENT_S_DN}e
        RequestHeader set X-Client-Verify %{SSL_CLIENT_VERIFY}e

        DocumentRoot /etc/puppet/rack/puppetmaster/public/
        RackBaseURI /
#Rack 为 Web 服务器提供了用来和 Puppet 这样的 Ruby HTTP 服务交换请求和响应的一些常用 API。
Rack 经常被用于在多台 Web 服务器上部署如 Puppet Dashboad 这样的 web 程序。
        <Directory /etc/puppet/rack/puppetmaster/>  #虚拟主机部分
                Options None
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
</VirtualHost>
```

```
[root@c1.inanu.net]# service httpd configtest  #检查 apache 配置语法是否正确
Warning: DocumentRoot [/etc/puppet/rack/puppetmaster/public/] does not exist
Syntax OK
```

备注：有关 puppet 虚拟主机配置可参考默认配置

```
/usr/share/puppet/ext/rack/files/apache2.conf
```

4 准备 config.ru 配置文件

```
[root@puppetserver rack]# mkdir -p /etc/puppet/rack/puppetmaster/{public,tmp} #为
Rack 和 Puppet master 的 rack 程序实例创建框架目录。
[root@puppetserver rack]# cp /usr/share/puppet/ext/rack/files/config.ru
/etc/puppet/rack/puppetmaster/
[root@puppetserver rack]# vim /etc/puppet/rack/puppetmaster/config.ru  #默认即可

# a config.ru, for use with every rack-compatible webserver.
# SSL needs to be handled outside this, though.

# if puppet is not in your RUBYLIB:
# $:.unshift('/opt/puppet/lib')

$0 = "master"

# if you want debugging:
```

```
# ARGV << "--debug"

ARGV << "--rack"
require 'puppet/application/master'
# we're usually running inside a Rack::Builder.new {} block,
# therefore we need to call run *here*.
run Puppet::Application[:master].run
```

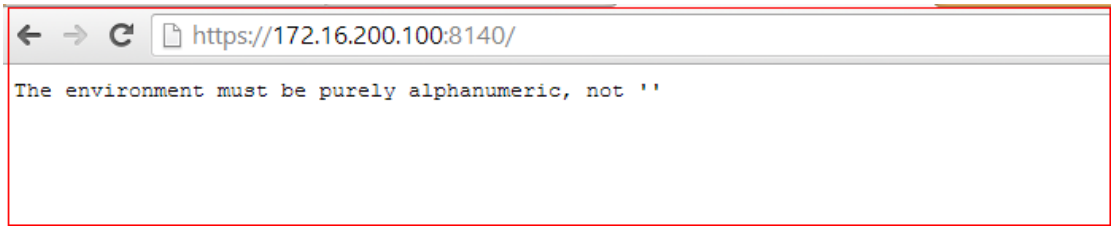
备注： 如果需要最新的 Rack 配置文件，可以在 Puppet 最新发行版的 ext 目录找到。也可以在 https://github.com/puppetlabs/puppet/tree/master/ext/rack/files 找到。

```
[root@puppetserver rack]# chown puppet. /etc/puppet/rack/puppetmaster/config.ru
#Rack 配置文件 config.ru 的用户和组应该是 puppet。当 Apache 启动时，Passenger 会检查这个文件的所有者，并将其使用的账号从 root 切换到权限较低的 puppet 账户。
```

5 在 Apache 中测试 PuppetMaster

```
[root@puppetserver ~]# /etc/rc.d/init.d/puppetmaster stop #停止 puppetmaster 进程
[root@puppetserver ~]# chkconfig puppetmaster off #防止开机自动启动
[root@puppetserver ~]# /etc/rc.d/init.d/httpd start #启动 apache 服务
[root@puppetserver ~]# chkconfig httpd off  #设置开机自动启动
[root@puppetserver ~]# netstat -nlp | grep 8140  #监听 8140 端口
tcp        0      0 :::8140                 :::*                    LISTEN      4162/httpd
```

测试一：通过浏览器（IE 版本<9）访问 https://172.16.200.100:8140/，出现以下信息，说明配置正确



测试二：在节点上运行 puppet 程序，在服务器端通过 apache 访问日志查看是否有 puppet 的请求，如果返回状态吗“200”表明这次请求时成功的。

```
[root@puppetserver conf.d]# tailf  /var/log/httpd/access_log
172.16.200.101 - - [22/Jul/2013:10:30:34 +0800] "GET
/production/file_metadata/modules/mysql/etc/my.cnf? HTTP/1.1" 200 298 "-" "-"
172.16.200.101 - - [22/Jul/2013:10:30:34 +0800] "GET
/production/file_metadata/modules/motd/etc/motd? HTTP/1.1" 200 295 "-" "-"
172.16.200.101 - - [22/Jul/2013:10:30:35 +0800] "PUT
/production/report/agent1.kisspuppet.com HTTP/1.1" 200 14 "-" "-"
172.16.200.101 - - [22/Jul/2013:10:30:40 +0800] "POST
/production/catalog/agent1.kisspuppet.com HTTP/1.1" 200 8346 "-" "-"
```

```
172.16.200.101 - - [22/Jul/2013:10:30:41 +0800] "GET
/production/file_metadata/modules/ssh/etc/ssh/sshd_config? HTTP/1.1"
```

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