

# Experimenting with pyControl on LTM VE



Jason Rahm, 2010-16-02

Since the [BIG-IP LTM VE](#) is currently a trial version, you won't be doing any production implementations. That's good in this case, because we're going step outside the box a little and install pyControl on your shiny new toy. As with any "hacked" solutions in the BIG-IP shell, it's best to deploy these in development and test scenarios, not production.

## Prerequisites

Beginning in v10.1, python version 2.4 comes installed on the BIG-IP. This makes the task a little simpler, as we'll only need to install `setuptools`, `suds`, and `pyControl`. Before we do, we need to download some packages:

1. [Setup Tools](#)
2. [Suds](#)
3. [pyControl](#)

To transfer these packages to your VE image, you can use [WinSCP](#), [PSCP](#) (from Putty), SCP, etc. From my Win7 desktop, the copy is simple

```
pscp setuptools-0.6c11-py2.4.egg root@10.10.20.254:/var/tmp/  
pscp python-suds-0.3.8.tar.gz root@10.10.20.254:/var/tmp/  
pscp pycontrol.tar.gz root@10.10.20.254:/var/tmp/
```

## Installation

We'll need to install the `setuptools` prior to installing `suds` and `pyControl`. If you give it a go straight away, you'll get an error:

```
[Erno 30] Read-only file system: '/usr/lib/python2.4/site-packages/test-easy-install-3910.write-test'  
The installation directory you specified (via --install-dir, --prefix, or the distutils default setting) was:  
/usr/lib/python2.4/site-packages/
```

This error is triggered due to the `/usr` partition being read-only as shown in the `/etc/mstab` file:

```
[root@localhost:Active] config # cat /etc/mstab | grep /usr  
/dev/mapper/vg--db--hda-set.1._usr /usr ext3 ro,noatime 0 0
```

So before we install the `setuptools`, we need to remount the `/usr` partition as read-write since it is read-only by default in v10.1

```
[root@localhost:Active] config # mount -o remount,rw /dev/vg-db-hda /usr  
[root@localhost:Active] config # cat /etc/mstab | grep /usr  
/dev/mapper/vg--db--hda-set.1._usr /usr ext3 rw 0 0
```

Now that we have a read-write partition, we can proceed to install the `setuptools`.

```
[root@localhost:Active] tmp # sh setuptools-0.6c11-py2.4.egg
Processing setuptools-0.6c11-py2.4.egg
creating /usr/lib/python2.4/site-packages/setuptools-0.6c11-py2.4.egg
Extracting setuptools-0.6c11-py2.4.egg to /usr/lib/python2.4/site-packages
Adding setuptools 0.6c11 to easy-install.pth file
Installing easy_install script to /usr/bin
Installing easy_install-2.4 script to /usr/bin

Installed /usr/lib/python2.4/site-packages/setuptools-0.6c11-py2.4.egg
Processing dependencies for setuptools==0.6c11
Finished processing dependencies for setuptools==0.6c11
```

With setuptools installed, we can proceed to install suds and pycontrol

```
#Suds
cd /var/tmp/
tar xvfz python-suds-0.3.8.tar.gz
cd python-suds-0.3.8
python setup.py install

#pyControl
cd /var/tmp/
tar xvfz pycontrol.tar.gz
cd trunk
python setup.py install
```

## Verify the Installation

Once the packages are installed, we enter the shell and make sure the version and build information is accessible. Suds should be 0.3.8 GA, pyControl should be 2.0 with at least build r76.

```
[root@localhost:Active] trunk # python
Python 2.4.3 (#1, Feb 6 2010, 00:58:54)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-46)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import suds
>>> suds.__version__
'0.3.8'
>>> suds.__build__
'GA R627-20091217'
>>> import pycontrol.pycontrol as pc
>>> pc.__version__
'2.0'
>>> pc.__build__
'r76'
```

## Test the Installation

Now that we finally have an installation in place, let's test it out! First we specify the assigned management interface, then we try the localhost.

```
[root@localhost:Active] trunk # python
Python 2.4.3 (#1, Feb 6 2010, 00:58:54)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-46)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pycontrol.pycontrol as pc
>>> b = pc.BIGIP(
...     hostname = '10.10.20.254',
...     username = 'admin',
...     password = 'admin',
...     fromurl = True,
...     wsdl = ['Networking.VLAN'])
>>> b.Networking.VLAN.get_list()
[server1, server2, clientaccess]
>>> b = pc.BIGIP(
...     hostname = 'localhost',
...     username = 'admin',
...     password = 'admin',
...     fromurl = True,
...     wsdl = ['LocalLB.VirtualServer'])
>>> b.LocalLB.VirtualServer.get_list()
[geo-vip]
```

Note that I had to define a name server in the system configuration or I received a “temporary failure in name resolution,” which is odd since I used an IP address and a name in the /etc/hosts file.

## Conclusion

Hopefully this was a fun exercise in the possible. If you haven’t had the resources to mess with the iControl API due to system constraints, BIG-IP access, etc, now there’s no excuse. So go download the BIG-IP LTM VE and start playing!

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](http://f5.com)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)