

# Compilar módulo de Apache mod\_python para Centos 7 / Redhat 7

## Introducción.

Check\_mk hasta ahora (dic 2014) continua usando el viejo módulo de python para Apache mod\_python que está ya discontinuado. Es un módulo que empezaron a desechar ya hace tiempo las distribuciones en favor del módulo mod\_wsgi que se ha impuesto por diversas razones. Las nuevas versiones de los distros más populares no incluyen este módulo por lo que si lo necesitas tendrás que compilarlo tu mismo. Hasta que CMK no cambie de módulo python habrá que hacerlo...

Las pruebas realizadas son con SELINUX desactivado. Cuidado también con configurar FirewallD para que puedas acceder a tu puerto 80.

## Compilar e instalar

Para instalar mod\_python en nuestro Apache deberemos compilarlo. Necesitamos primero instalar los fuentes de apache ya que necesitamos un binario que se incluye en estos (apxs). También necesitamos los fuentes de python

```
yum install httpd httpd-devel python-devel
yum groupinstall "Development Tools"
```

Bajamos el módulo de su página <http://modpython.org/>. OJO. Necesario versión 3.5.x.

Lo descomprimos y comenzamos con el proceso de compilación.

```
tar zxfv mod_python-3.5.0.tg
cd mod_python-3.5.0
./configure -with-apxs=/usr/bin/apxs
```

Antes de hacer el make debemos modificar un par de ficheros. Tenemos más información al respecto al final del artículo en “Referencias”.

**src/Makefile.** Añadiremos las líneas marcadas

```
version.c:
    @MP_GIT_SHA=$(git describe --always); \
    echo > version.c ; \
    echo "/* THIS FILE IS AUTO-GENERATED BY Makefile */" >> version.c ; \
    echo "#include \"mp_version.h\"" >> version.c ; \
    include/mod_python.h include/psp_flex.h include/psp_parser.h \
    include/requestobject.h include/tableobject.h include/connobject.h \
    include/finfoobject.h include/hlistobject.h include/mp_version.h \
    include/_pspmodule.h include/psp_string.h include/serverobject.h \
    include/util.h \
    echo "const char * const mp_git_sha = \"\${MP_GIT_SHA}\";" >> version.c ; \
```

**dist/version.sh.** Comentamos alguna linea y añadimos otra modificada

```
#!/bin/sh
MPV_PATH=""`dirname $0`../src/include/mp_version.h"
MAJ=`awk '/MP_VERSION_MAJOR/ {print $3}' $MPV_PATH`
MIN=`awk '/MP_VERSION_MINOR/ {print $3}' $MPV_PATH`
PCH=`awk '/MP_VERSION_PATCH/ {print $3}' $MPV_PATH`
#GIT=`git describe --always`
#echo $MAJ.$MIN.$PCH-$GIT
echo $MAJ.$MIN.$PCH-
```

Ahora si, podemos acabar de compilar.

```
make
make install
```

## ***Probando el módulo.***

Para probarlo de forma sencilla seguimos las instrucciones de la [siguiente página](#) para realizar un test usando nuestra estructura por defecto de Apache. Añadimos en nuestro apache un directorio:

```
mkdir /var/www/html/test
chown apache.apache /var/www/test
```

Creamos un nuevo fichero de configuración p.e. `/etc/httpd/conf.d/test_mod_python.conf` para manejar los ficheros .py

```
# FILE /etc/httpd/conf.d/test_mod_python.conf
<Directory /var/www/html/test>
    AddHandler mod_python .py
    PythonHandler mptest
    PythonDebug On
</Directory>
```

Creamos un fichero en dicho directorio `/var/www/html/test/mptest.py` para realizar el test.

```
# FILE /var/www/html/test/mptest.py
from mod_python import apache
def handler(req):
    req.content_type = 'text/plain'
    req.write("Hello World!")
    return apache.OK
```

Activamos en Apache en módulo de python. Creamos p.e.. un archivo `/etc/httpd/conf.modules.d/01-mod_python.conf`

```
# FILE /etc/httpd/conf.modules.d/01-mod_python.conf
LoadModule python_module modules/mod_python.so
```

Reiniciamos Apache.

```
systemctl restart httpd.service
```

Probamos acceder a la URI para obtener un “Hello Word!”

<http://server/test/mpptest.py>

## Referencias.

Si no se modifican los ficheros fuente según se indica no funcionará `mod_python` y nos encontraremos problemas en el `error_log` de Apache del tipo siguiente:

```
[Sun Dec 07 10:31:13.539980 2014] [[:error] [pid 25070] make_obcallback: could
not import mod_python.apache.\n
Traceback (most recent call last):
  File "/usr/lib64/python2.7/site-packages/mod_python/__init__.py", line 25, in
<module>
    from . import version
  File "/usr/lib64/python2.7/site-packages/mod_python/version.py", line 3
    version = "fatal: Not a git repository (or any of the parent
directories): .git
^
SyntaxError: EOL while scanning string literal
[Sun Dec 07 10:31:13.540476 2014] [[:error] [pid 25070] make_obcallback: Python
path being used "['/usr/lib64/python27.zip', '/usr/lib64/python2.7',
'/usr/lib64/python2.7/plat-linux2', '/usr/lib64/python2.7/lib-tk',
'/usr/lib64/python2.7/lib-old', '/usr/lib64/python2.7/lib-dynload',
'/usr/lib64/python2.7/site-packages', '/usr/lib/python2.7/site-packages']".
[Sun Dec 07 10:31:13.540593 2014] [[:error] [pid 25070] get_interpreter: no
interpreter callback found.
[Sun Dec 07 10:31:13.540752 2014] [[:error] [pid 25070] [client
192.168.164.1:55805] python_handler: Can't get/create interpreter.
```

La solución es la comentada que está “aplicada” del siguiente enlace.

<http://www.yeboyzq.com/linux/ruanjiananzhuangwei/495.html>