

Instalación del modulo SAPRFC en Red Hat Linux 9

Autores:

- Jaime Díaz (jdiaz@esabel.net)
- Vincent Vaurette (php@vaurette.net)

Preliminares:

La instalación fue realizada en un sistema con Red Hat Linux 9 (RHL9) al cual se le actualizó el GNU C Compiler (gcc) a la version 3.3.4 (RHL9 incluye la version 3.2.2). Aunque esto no es necesario, es altamente recomendable debido a problemas de seguridad descubiertos en las últimas semanas, respecto a la generacion de codigo con perdidas (leaks) que pueden bloquear el sistema con un simple programa.

Se compiló, además, las versiones 2.0.50 de Apache httpd y PHP 4.3.8. Si bien la recompilación del httpd no es necesaria, también es recomendable debido a las pocas opciones especiales que se requieren para el modulo SAPRFC. Adicionalmente existen, al menos, 6 vulnerabilidades descubiertas desde la version 2.0.47 (incluida con RHL9) solucionadas en la version 2.0.50. Poseer el código fuente de PHP es un requerimiento para poder compilar el modulo SAPRFC.

Por último, es necesario poseer el SAP RFC SDK (version NON-UNICODE) para la plataforma utilizada (en nuestro caso, Linux_32). La misma se puede descargar desde <http://service.sap.com/swdc>.

Para instalarla se necesita, también, el programa SAPCAR que se utiliza para descomprimir archivos SAR. Esta utilidad puede conseguirse en el FTP de SAP (<ftp://ftp.sap.com>) dentro del directorio /pub/linuxlab/contrib .

Para poder compilar los paquetes es necesario instalar los paquetes de desarrollo que se encuentran en los Cds de RHL9.

Instalacion de Apache httpd:

El código fuente del httpd puede ser descargado desde <http://httpd.apache.org>.

Una vez obtenido el mismo, debe ser descomprimido con el siguiente comando:

```
tar xvfz httpd-2.0.50.tar.gz
```

Esto generará una carpeta llamada httpd-2.0.50 la cual contendrá todos los

archivos fuentes del demonio httpd.
Allí dentro ejecutamos el comando:

```
./configure --enable-so --otras-opciones
```

Solamente es necesaria la opción `--enable-so` para poder utilizar el módulo de PHP. El resto de las opciones disponibles pueden consultarse ejecutando:

```
./configure --help
```

Este comando realizará las comprobaciones de dependencias necesarias para la exitosa compilación del paquete.
Una vez finalizado, procedemos a la compilación propiamente dicha, ejecutando el comando:

```
make
```

Por último, si el comando `make` finalizó sin errores, ejecutamos, como usuario `root` el comando:

```
make install
```

Este último comando instalará todos los archivos dentro de `/usr/local/apache2` a no ser que se haya modificado el parámetro de compilación `--prefix`. En tal caso, se encontrarán donde se especificó allí.

Para verificar la correcta instalación, debe modificarse levemente el `httpd.conf` ubicado en `/usr/local/apache2/conf`. Con modificar el parámetro `ServerName` debería ser suficiente.

Luego, de esto ejecutamos el demonio con el siguiente comando:

```
/usr/local/apache2/bin/apachectl start
```

Instalación del SAP RFC SDK:

Una vez obtenidos los dos archivos (el SDK y la utilidad `SAPCAR`) procederemos a la instalación del mismo. Para esto, debemos descomprimir los archivos que se encuentran dentro del archivo `SAR` y, luego, copiarlos a la ubicación correspondiente.

Para descomprimirlos, y suponiendo que la utilidad `SAPCAR` y el SDK se encuentran en el mismo directorio en el cual estamos posicionados, ejecutamos:

```
./SAPCAR -xf ARCHIVO.SAR
```

Esta operación creará una carpeta llamada **rfcsdk** en el directorio actual. Antes

de copiarla debemos (como root) crear la carpeta `/usr/sap` y, luego, `/usr/sap/rfcsdk`, debido a que deberemos linkear luego desde PHP funciones del SDK es recomendable que el este directorio creado tenga permisos 755 (podemos lograrlo ejecutando `chmod -R 755 /usr/sap` (sin comillas) para modificar los permisos recursivamente.

Esto concluye la instalación del SAP RFC SDK.

Instalación de PHP:

La compilación de PHP suele ser la parte más complicada de todo el proceso, debido a que se necesita saber de antemano que soporte necesitaremos y se deben satisfacer las dependencias necesarias para que se pueda compilar dicho soporte. Esto implica la instalación de paquetes RPM adicionales que incluyen los archivos de desarrollo para un soporte específico (por ejemplo, PostgreSQL, el cual será nuestro ejemplo utilizado).

Para poder compilar PHP con soporte para PostgreSQL deberemos tener instalado el paquete `postgresql-devel` el cual contiene los archivos necesarios (se instala con el comando `rpm -ivh postgresql-devel`). La determinación de los componentes necesarios es fundamental, debido a que si no se lo compila, luego habrá que recompilar todo PHP para agregar el soporte faltante.

Habiendo bajado el archivo conteniendo el código fuente de PHP (de <http://www.php.net>) procedemos a descompactarlo con el siguiente comando:

```
tar xvfz php-x.y.z.tar.gz
```

Esto creará una carpeta llamada `php-x.y.z` a la cual debemos ingresar. Una vez allí ejecutamos el comando **configure**, indicando las opciones de compilación que deseamos. En nuestro caso, queremos soporte para PostgreSQL por lo que lo haremos de esta manera:

```
./configure --with-pgsql \  
    --with-apx2=/usr/local/apache2/bin/apxs \  
    --otras-opciones
```

La opción **apx2** se utiliza para indicar que, además, queremos compilar el módulo PHP para Apache httpd (también conocido como `mod_php`). El script comprobará que se encuentren todas las dependencias necesarias para la compilación y, si termina sin errores, ejecutamos el siguiente comando:

```
make
```

Esta operación puede tomar un tiempo considerable según la cantidad de memoria y el procesador que tengamos.

Si concluye exitosamente (debe verse el mensaje “**Build complete**”) ejecutamos, como usuario **root**, este último comando:

```
make install
```

Ahora debemos copiar la librería del `mod_php` adonde Apache httpd espera encontrarla. Para esto ejecutamos este comando:

```
cp -p .libs/libphp4.so /usr/local/apache2/modules
```

Luego deberemos copiar el archivo de configuración **php.ini** desde los ejemplos que trae el paquete PHP con el siguiente comando:

```
cp -p php.ini-recommended /usr/local/lib/php.ini
```

La ubicación correcta del archivo `php.ini` depende de si se modificó (o no) el parámetro `--prefix` durante la ejecución del comando `./configure` anteriormente visto. El ejemplo dado solo sirve si **no** se modificó dicho parámetro.

Por último, deberemos verificar que las siguientes directivas se encuentran dentro del archivo `/usr/local/apache2/conf/httpd.conf` para que los archivos PHP se reconozcan como tales y sean pasados al módulo PHP.

```
#Solo debe haber una línea con esta directiva
LoadModule php4_module          modules/libphp4.so
```

```
# Agregar index.php al DirectoryIndex
DirectoryIndex index.html index.php
```

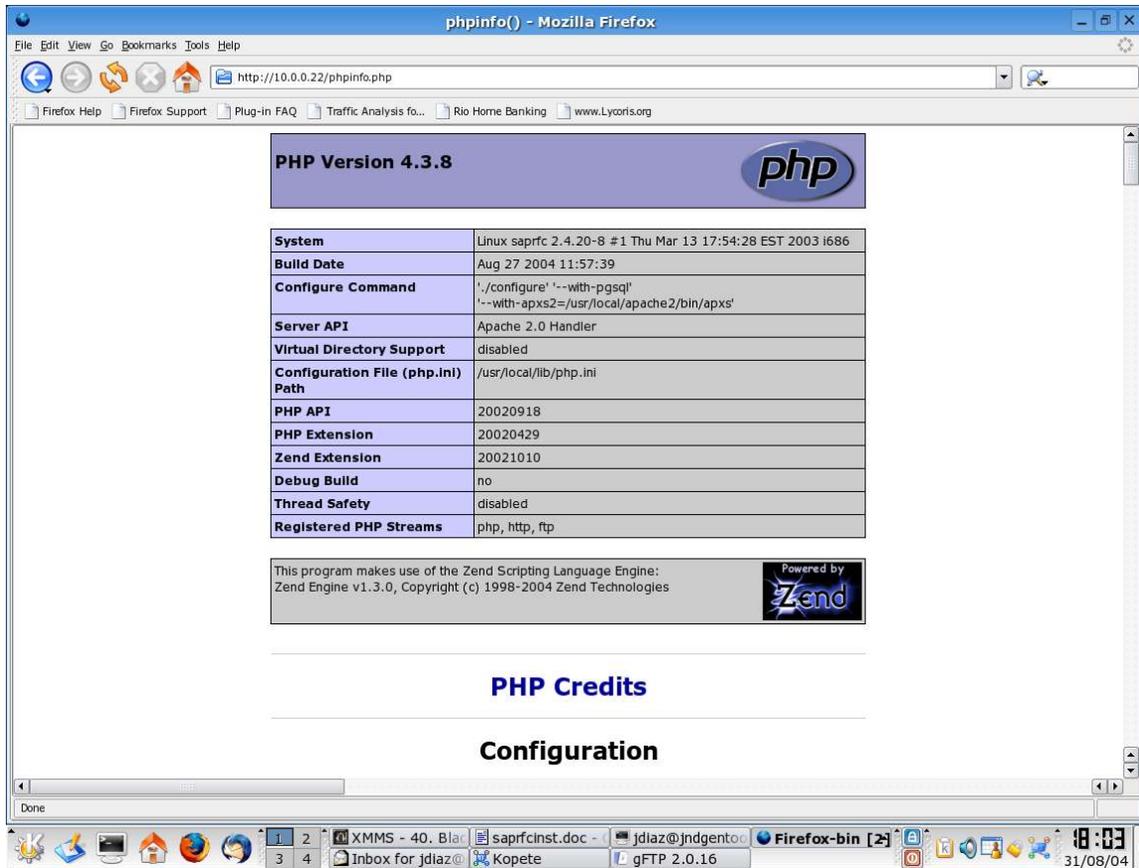
```
AddType application/x-httpd-php php
```

Si estas directivas no existieran deben ser agregadas.

Para probar el correcto funcionamiento del módulo, creamos un archivo con extensión `php` que contenga el siguiente código:

```
<?php
phpinfo()
?>
```

Y lo ubicamos dentro del `DocumentRoot` de Apache httpd. Si al solicitarlo vemos una salida similar a la de la figura, entonces, el módulo funciona correctamente.



Si esto funciona correctamente, hemos concluido la instalación de PHP.

Instalación SAP RFC:

Puesto que la forma de compilación indicada en la documentación del producto no funciona bajo RHL9, deberemos compilar los componentes necesarios en forma manual, y luego agregar la extensión a PHP que comenzará a utilizarlo.

Para realizar esto debemos tener instalado el SAP RFC SDK y tener disponible los fuentes de PHP ya que varias funciones se linkean contra el.

Primero de todo, descompactamos los fuentes del saprfc a un directorio con el siguiente comando:

```
tar xvfz saprfc-x.y.z.tar.gz
```

Esto creará un directorio llama **saprfc-x.y.z**, debemos ingresar a el (`cd saprfc-x.y.z`) y ejecutar los siguientes comando, exactamente en el mismo orden que aquí se describe.

Primer Comando:

```
gcc -fpic -fomit-frame-pointer -fexpensive-optimizations
-D_COMPILE_DL_SAPRFC \
-I/usr/local/include -I. -I/home/php-4.3.8 \
-I/home/php-4.3.8/Zend -I/home/php-4.3.8/TSRM \
-I/home/php-4.3.8/main -I/usr/sap/rfcsdk/include -c saprfc.c -o saprfc.o
```

Segundo Comando:

```
gcc -fpic -fomit-frame-pointer -fexpensive-optimizations
-D_COMPILE_DL_SAPRFC \
-I/usr/local/include -I. -I/home/php-4.3.8 \
-I/home/php-4.3.8/Zend -I/home/php-4.3.8/TSRM \
-I/home/php-4.3.8/main -I/usr/sap/rfcsdk/include -c rfccal.c -o rfccal.o
```

Tercer Comando:

```
ld -shared -L/usr/local/lib -L/usr/sap/rfcsdk/lib -lrfccm saprfc.o
rfccal.o -o php_saprfc.so
```

Cuarto Comando:

```
strip -R .stab -R .stabstr -R .comment -R .note php_saprfc.so
```

Quinto Comando:

```
strip -s php_saprfc.so
```

Por supuesto, para que estos comandos funcionen correctamente, debemos reemplazar `/home/php-4.3.8` por el directorio donde hayamos desempaquetado el código fuente de PHP. Lo mismo vale para el SAP RFC SDK si optamos por instalarlo en un directorio diferente a `/usr/sap/rfcsdk`. Estos cinco comandos sucesivos crearán un archivo en el directorio actual llamado **php_saprfc.so** el cual debe ser copiado al directorio de extensiones de PHP. Este mismo puede consultarse (y modificarse) buscando el parametro “**extension_dir**” dentro del archivo **php.ini**. En el caso de una instalación por default, bastará ejecutar el siguiente comando para realizar esta operación:

```
cp php_saprfc.so /usr/local/lib/php/
```

Por último, debemos agregar la extensión de **saprfc** al archivo **php.ini** agregando la siguiente sentencia a continuación de “**extension_dir**”

```
extension = php_saprfc.so
```

Solo nos queda reiniciar el servidor web, mediante un **start** y **stop** y no un **restart**. Podemos usar el archivo **saprfc_test.php** para la comprobación del correcto funcionamiento.