**Università degli Studi di Parma**
**Dipartimento di Matematica e Informatica**
**Corso di Laurea in Informatica**

**Giulio Destri**

# Apache from source: Il server Web PHP, Java e Mono, SSH/SFTP

# Introduzione

Manutenere un server Web basato su Linux è una operazione ormai routinaria per molti system administrator e webmaster. Ma rimane sempre un dilemma notevole per quanto riguarda l'applicazione di patch ed aggiornamenti, in special modo quelli di sicurezza. Infatti, se ci si affida completamente alla rete di aggiornamenti della distribuzione Linux utilizzata, il pericolo di errori da dipendenza e simili dovrebbe essere ridotto al minimo. Ma d'altro canto, nonostante l'azione di tantissimi gruppi di lavoro in rete, non sempre gli aggiornamenti "ufficiali" delle distribuzioni coprono con tempestività le necessità di sicurezza, costringendo spesso gli amministratori di sistema ad intervenire più o meno "manualmente". Inoltre i meccanismi di aggiornamento sono fortemente associati alla partcolare distribuzione Linux usata e anzi, spesso, addirittura alla release di tale distribuzione.

Il lavoro necessario per mettere in piedi da zero un server senza usare una distribuzione Linux "codificata" come RedHat o Ubuntu è normalmente proibitivo rispetto alle disponibilità di tempo degli amministratori di sistema. E' questo storicamente il motivo che ha portato alla decadenza di distribuzioni come la Slackware, che, composte dei soli moduli pacchettizzati, se da un lato consentivano il massimo controllo del sistema, dall'altro rendevano molto lunghe le operazioni di installazione e configurazione.

Per cercare di unire i vantaggi di entrambi i modi di procedere, in questa serie di articoli viene presentato un approccio ibrido. Per capire meglio tale approccio, pensiamo di dividere gli strati di software che formano la nostra installazione Linux in strati, in base al loro ruolo e al fatto che tali strati ospitino o no servizi destinati ad un accesso dall'esterno della macchina, via rete TCP/IP. Ricordiamo anzitutto che il nostro è un server Web, destinato ad ospitare solo alcuni servizi accessibili dall'esterno. Supponiamo inoltre che esso sia un server completo, che ospita, accanto a siti statici, anche applicativi Web realizzati con le tre tecnologie maggiormente diffuse: PHP, Java (limitando l'analisi al Web container Tomcat, usato anche dal diffuso application server JBoss) e ASP.NET con Mono. Pertanto gli strati possono essere indicati come segue.

1. Il Kernel: contiene IPTables e possiamo supporre che le vulnerabilità siano rare e comunque coperte efficacemente dagli aggiornamenti propri delle distribuzioni.
2. Le librerie ed il software di base come i compilatori: sono quelli della distribuzione e anche per essi può essere considerato valido il discorso delle vulnerabilità.
3. I database MySQL e Postgres: non sono visibili direttamente dall'esterno in quanto le loro porte TCP devono essere obbligatoriamente schermate da IPTables. Per essi si può usare la versione compresa nella distribuzione oppure, se interessati a funzionalità aggiuntive (si pensi, ad esempio, al MySQL 5.1) si può procedere con una compilazione da zero, che sarà spiegata in uno dei successivi articoli. Possono essere ovviamente presenti vulnerabilità negli applicativi Web che vi accedono (come per esempio PhpMyAdmin) e vanno trattate nella manutenzione di tali applicatvi, che non saranno riportate in questa serie di articoli.
4. Le librerie Zlib e OpenSSL: sono la parte più interna di tanti servizi accessibili direttamente dall'esterno, possono essere soggette a vulnerabilità e può quindi essere desiderabile compilarle da zero.
5. Il sistema di amministrazione remota SSH: è, dopo web e FTP, il servizio più attaccato dai pirati informatici e quindi deve essere compilato da zero a partire dall'ultima versione stabile rilasciata.
6. Il server Web Apache: è l'elemento centrale di tutto il sistema, deve essere compilato da zero a partire dall'ultima versione stabile.

7. Il sistema PHP e le librerie su cui poggia: è il frame work applicativo forse più usato in ambiente Web, per cui è un elemento fondamentale e dovrebbe sempre essere aggiornato all'ultima versione stabile. La sua installazione sarà presentata nel corrente articolo.
8. Il sistema Java e Tomcat: anche esso dovrebbe sempre essere aggiornato all'ultima versione.
9. Il sistema MONO: se l'installazione è facile per quelle distribuzioni Linux che dispongono del pacchetto, compilare da zero è complesso.

Per evitare qualsiasi conflitto con eventuali package già installati entro il server dalla distribuzione installata viene usata la seguente metodologia, simile a quelle usate per la manutenzione di Solaris.

- L'installazione di tutti i package mantenuti "a mano" non viene fatta in **/usr** né in **/usr/local**, ma entro la cartella **/opt**.
- Ogni applicazione viene installata in un proprio albero di cartelle, indipendente dalle altre e da eventuali altre versioni, ottenuto dalla configurazione, dalla compilazione e dal make install, ove presente nel pacchetto dei sorgenti.
- Per ogni applicazione individuata come necessaria per il server Web, viene creata la seguente struttura in **/opt**:
  **/opt/<nome applicazione>/<versione>/<insieme di cartelle dell'applicazione>**
  Ad esempio, per il server web apache, se la versione usata è la 2.2.21, la struttura diventa, per la cartella iniziale o radice di Apache
  **/opt/apache/2.2.21**
  Entro la quale poi stanno le sottocartelle bin, conf ecc…
  Analogamente, per PHP versione 5.3.9, la cartella iniziale diventa
  **/opt/php/5.3.9**
- Qualora sia necessaria la visibilità degli eseguibili così installati a livello globale, si crea un link ad essi entro la cartella **/usr/local/bin** o **/usr/local/sbin**
- Si modificano e/o creano gli script dei servizi inserendo in essi i riferimenti ai link e/o alle cartelle reali dell'applicazione che viene da essi controllata.

In tal modo, quando si deve aggiornare una versione alla successiva, la si compila ed installa in una cartella adiacente (es /opt/apache/2.2.22), si fanno tutti i test del caso e, solo quando il risultato è soddisfacente si cambiano i link e/o i percorsi presenti negli script dei servizi, in modo tale da consentire sempre comunque un rapido ritorno alla versione precedente in caso di anomalie. Periodicamente, se necessario per lo spazio disco, si può procedere ad una rimozione delle versioni vecchie non più utilizzate.
Questa configurazione ha il vantaggio di essere indipendente dalla distribuzione utilizzata, di consentire un rapido backup delle installazioni dei programmi (è sufficiente, infatti, il salvataggio del contenuto della cartella **/opt**, oltre che, ovviamente, delle configurazioni del sistema e dei servizi in **/etc**) e di consentire la coesistenza di diverse versioni degli applicativi e librerie utilizzati.
Uno svantaggio evidente è la necessità di gestire a mano le dipendenze tra gli applicativi di più alto livello come Apache e PHP e le librerie come OpenSSL.
Inoltre, al cambio di ogni release
Nei prossimi paragrafi vedremo passo dopo passo la compilazione ed installazione dei vari elementi sopra citati.

# Compilare ed installare un sistema pacchettizzato tarball

I software i cui sorgenti sono pacchettizzati in formato tarball sono per fortuna la maggioranza.
Si consiglia di creare una cartella con tutti i package sorgenti ed una in cui vengono fatte estrazioni dei sorgenti e loro compilazione.
Il procedimento che viene usato per ottenere da essi gli eseguibili è il seguente, descritto con l'esempio delle librerie zlib versione 1.2.5

Posizionarsi entro la cartella di lavoro scelta.

Scaricare il software con wget dal suo repository ufficiale dei sorgenti
**`wget http://www.gzip.org/zlib/zlib-1.2.5.tar.gz`**

eventualmente provvedere al controllo del checksum MD5

scompattarlo entro la cartella
**`tar –zxvf zlib-1.2.5.tar.gz`**

creare la cartella destinazione sotto ***/*opt**
**`mkdir –p /opt/zlib/1.2.5`**

entrare nella cartella apposite
**`cd zlib-1.2.5`**

configurare il software rispetto alla propria installazione di Linux e specificare la cartella radice per l'installazione finale
**`./configure --shared --prefix=/opt/zlib/1.2.5`**

Compilare
**`make`**

Verificare l'avvenuta compilazione
**`make test`**

Installare nella cartella finale entro ***/*opt**
**`make install`**

Nei package successivi la cui compilazione dipende dalle librerie installate, occorre configurare il software indicando le cartelle delle librerie usate e specificando sempre la cartella radice per l'installazione finale Nell'esempio sotto indicato è la configurazione di OpenSSH affinchè utilizzi le librerie OpenSSL e Zlib installate dai sorgenti.

```
./configure --prefix=/opt/ssh/5.9p1 --with-ssl-
dir=/opt/openssl/1.0.0e  --with-zlib=/opt/zlib/1.2.5
```

Dopodichè, se non vi sono errori si deve compilare ed installare
```
make
make install
```

## Installare pacchetti binari

Nell'insieme di pacchetti necessari, non tutti sono disponibili in formato tarball. Per esempio Java è disponibile in formato shell auto esplodente.
Java è necessario per il funzionamento corretto di tutti gli applicativi basati sul suo runtime. Purtroppo molte distribuzioni contengono versioni di Java non aggiornate, per cui è preferibile scaricare ed installre l'ultima versione direttamente dal sito **java.sun.com**.
Il file indipendente dalla distribuzione è jdk-<versione>-linux-i586.bin, nel caso corrente è **jdk-6u30-linux-i586.bin**, che è un eseguibile autoinstallante.
Una volta copiatolo in una cartella temporanea, ad esempio **/tmp**, occorre seguire i passi di seguito riportati

rendere eseguibile il file
```
chmod 750 /tmp/jdk-6u30-linux-i586.bin
```

creare la cartella per Java in **/opt** (se non esiste ancora)
```
mkdir –p /opt/java
```

posizionarsi in tale cartella
```
cd /opt/java
```

eseguire il file
```
/tmp/jdk-6u30-linux-i586.bin
```

Rispondere alle domande che vengono poste accettando in primo luogo la richiesta (yes)

Al termine della installazione è stata creata una cartella **/opt/java/jdk1.6.0_30**. Tale cartella deve essere impostata come valore per la variabile di ambiente JAVA_HOME nelle fasi successive della installazione e in tutti quegli script che prevedono l'avvio di applicazioni Java come ad esempio Tomcat.

A questo punto si sono visti i diversi tipi di pacchetti da installare.
Di seguito vengono ora presentati gli script per generare l'ambiente completo per Apache con Java e PHP e quello per costruire l'ambiente Mono a partire dai sorgenti.

Gli esempi di script sono disponibili con licenza LGPL e sono anche scaricabili direttamente dagli indirizzi:
http://www.giuliodestri.it/utility/web_platform_creation.txt
http://www.giuliodestri.it/utility/mono_platform_creation.txt
Ovviamente dopo averli modificati secondo le proprie esigenze, devono essere salvati con estensione .sh e l'indicazione della shell usata per poter essere lanciati.
Questi e tutti gli altri script sono poi riuniti nel file
http://www.giuliodestri.it/utility/webplatformfull.tgz

```
#########################################################################
#
# Script template for PHP/Java Web Platform creation from source tarball
# (c) 2005-2012 AREA Professional - Giulio Destri
# http://www.areaprofessional.net/giulio.destri
# Released in 2012 under LGPL License
#
#########################################################################
#
# This script template has been widely used to maintain production servers
# based on RedHat/Fedora/CentOS 4.x and 5.x platform
# It has been partially tested on CentOS 6.x and Debian 4.x
# Feel free to adapt it to your needs
# Please report any bug and any improvement suggestion to
# giulio.destri@areaprofessional.net
#
#########################################################################
#
# SOFTWARE TARBALL and URL
#
# OpenSSL         downloadable from http://www.openssl.org
# ZLib            downloadable from http://www.zlib.net
# OpenSSH         downloadable from
http://mi.mirror.garr.it/1/OpenBSD/OpenSSH/portable/
#
# Postgres  downloadable from http://www.postgresql.org
# MySQL     downloadable from http://www.mysql.com
#
# Java            downloadable from http://java.sun.com
# Apache Httpd    downloadable from http://httpd.apache.org
# Apache Tomcat downloadable from http://tomcat.apache.org
# JK Connector    downloadable from http://tomcat.apache.org/download-
connectors.cgi
#
# LIBMCRYPT       downloadable from http://sourceforge.net/projects/mcrypt/
# LibGD     downloadable from http://www.libgd.org/Main_Page
# LibJpeg   downloadable from http://www.ijg.org/
# FreeType  downloadable from
http://download.savannah.gnu.org/releases/freetype/
# LibPNG    downloadable from http://www.libpng.org/pub/png/libpng.html
# LibIConv  downloadable from http://www.gnu.org/software/libiconv/
# Lib IMAP  downloadable from ftp://ftp.cac.washington.edu/imap/
# PHP             downloadable from http://www.php.net
#
#########################################################################
#
# Starting from now 2 working directory are defined:
# - TAR_DIR       = repository where to download and store tarballs
# - COMPILE_DIR = folder where to compile tarballs
#
# defined value are just for example
#
#########################################################################
#
#########################################################################
#
# VARIABLE DEFINITION
#
#########################################################################

# WORKING FOLDER
export TAR_DIR=/data/condivisa/lavoro/linuxprod/tar
export COMPILE_DIR=/data/condivisa/lavoro/linuxprod/compile
```

```
# INSTALLATION BASE FOLDER
export INSTALL_DIR_PREFIX=/opt
export INSTALL_IMAGELIB_DIR_PREFIX=/opt/imglib

# SOFTWARE PACKAGE VERSIONS
export OPENSSL_VERSION=1.0.0e
export ZLIB_VERSION=1.2.5
export SSH_VERSION=5.9p1
export POSTGRES_VERSION=8.1.23
export MYSQL_VERSION=5.0.77
export JAVA_VERSION=6u30
export JAVA_HOME=$INSTALL_DIR_PREFIX/java/jdk1.6.0_30
export HTTPD_VERSION=2.2.21
export TOMCAT_VERSION=5.5.34
export JKCON_VERSION=1.2.23
export LIBJPEG_VERSION=8c
export FREETYPE_VERSION=2.4.6
export LIBPNG_VERSION=1.5.4
export LIBMCRYPT_VERSION=2.5.8
export LIBGD_VERSION=2.0.34
export LIBICONV_VERSION=1.14
export VSFTPD_VERSION=2.3.4
export LIBIMAP_VERSION=2007f
export CURL_VERSION=7.22.0
export PHP_VERSION=5.3.9

# JAVA ARCHITECTURE FOR DOWNLOADING THE RIGHT PACKAGE, please select one
#export JAVA_CPU_ARCH=i586
export JAVA_CPU_ARCH=x64

##############################################################################
##############################################################################
#
# SOFTWARE TARBALL DOWNLOAD
#
##############################################################################

# TAR_DIR becomes pwd
cd $TAR_DIR

wget http://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz
wget http://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz.md5

wget http://artfiles.org/openbsd/OpenSSH/portable/openssh-$SSH_VERSION.tar.gz

wget http://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-
Site/en_US/-/USD/VerifyItem-Start/jdk-6u23-linux-
i586.bin?BundledLineItemUUID=lwiJ_hCwvHQAAAEt4v0AGViF&OrderID=bPKJ_hCwSNoAAAEt1P
0AGViF&ProductID=QhOJ_hCw.dUAAAEsFIMcKluK&FileName=/jdk-6u23-linux-
i586.binhttp://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-
Site/en_US/-/USD/VerifyItem-Start/jdk-6u23-linux-
i586.bin?BundledLineItemUUID=lwiJ_hCwvHQAAAEt4v0AGViF&OrderID=bPKJ_hCwSNoAAAEt1P
0AGViF&ProductID=QhOJ_hCw.dUAAAEsFIMcKluK&FileName=/jdk-6u23-linux-i586.bin

wget http://download.oracle.com/otn-pub/java/jdk/6u27-b07/jdk-6u27-linux-
i586.bin

wget http://mirror.nohup.it/apache//httpd/httpd-$HTTPD_VERSION.tar.gz
wget http://mirror.nohup.it/apache/tomcat/tomcat-5/v$TOMCAT_VERSION/bin/apache-
tomcat-$TOMCAT_VERSION.tar.gz

wget http://www.ijg.org/files/jpegsrc.v$LIBJPEG_VERSION.tar.gz
```

```
wget http://download.savannah.gnu.org/releases/freetype/freetype-
$FREETYPE_VERSION.tar.gz
wget http://download.savannah.gnu.org/releases/freetype/freetype-
$FREETYPE_VERSION.tar.gz.sig
wget http://prdownloads.sourceforge.net/libpng/libpng-
$LIBPNG_VERSION.tar.gz?download
wget ftp://ftp.cac.washington.edu/imap/imap-$LIBIMAP_VERSION.tar.gz
wget http://ftp.gnu.org/pub/gnu/libiconv/libiconv-$LIBICONV_VERSION.tar.gz

wget http://curl.haxx.se/download/curl-$CURL_VERSION.tar.gz

wget http://www.php.net/get/php-$PHP_VERSION.tar.gz/from/de2.php.net/mirror

#############################################################################
#############################################################################
#
# SOFTWARE TARBALL COMPILATION
#
#############################################################################

#############################################################################
# Compilation and installation of  OpenSSL
#############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION

tar -zxvf $TAR_DIR/openssl-$OPENSSL_VERSION.tar.gz
cd openssl-$OPENSSL_VERSION

./config --prefix=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION/ \
--openssldir=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION/openssl

make
make install

#############################################################################
# Compilation and installation of ZLIB
#############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/zlib/$ZLIB_VERSION

tar -zxvf $TAR_DIR/zlib-$ZLIB_VERSION.tar.gz
cd zlib-$ZLIB_VERSION

./configure --shared --prefix=$INSTALL_DIR_PREFIX/zlib/$ZLIB_VERSION

make
make test
make install

#############################################################################
# Compilation and installation of OpenSSH
#############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/ssh/$SSH_VERSION

tar -zxvf $TAR_DIR/openssh-$SSH_VERSION.tar.gz
cd openssh-$SSH_VERSION

./configure --prefix=$INSTALL_DIR_PREFIX/ssh/$SSH_VERSION \
```

```
      --with-ssl-dir=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION \
      --with-zlib=$INSTALL_DIR_PREFIX/zlib/$ZLIB_VERSION

make
make install

#############################################################################
# Compilation and installation of PostgreSQL
# See or include Postgres Script
#
#############################################################################
# Compilation and installation of MySQL
# See or include MySQL Script
#
#############################################################################
# Installation of Oracle JAVA
#############################################################################

mkdir -p $INSTALL_DIR_PREFIX/java

cd $INSTALL_DIR_PREFIX/java

chmod 755 $TAR_DIR/jdk-$JAVA_VERSION-linux-$JAVA_CPU_ARCH.bin

$TAR_DIR/jdk-$JAVA_VERSION-linux-$JAVA_CPU_ARCH.bin

#############################################################################
# Compilation and installation of Apache HTTPD
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION

tar -zxvf $TAR_DIR/httpd-$HTTPD_VERSION.tar.gz
cd httpd-$HTTPD_VERSION

./configure --prefix=$INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION \
      --enable-ssl \
      --with-ssl=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION \
      --with-mod_jk

make
make install

#############################################################################
# Installation of Apache Tomcat
#############################################################################
mkdir -p $INSTALL_DIR_PREFIX/tomcat
cd $INSTALL_DIR_PREFIX/tomcat
tar -zxvf $TAR_DIR/apache-tomcat-$TOMCAT_VERSION.tar.gz

#############################################################################
# Compilation and installation of Apache Tomcat JK Connector
# Warning: due to strange runtime problems it's strongly suggested
# to use JK version 1.2.23
# JK module (file mod_jk.so) is directly installed in Apache modules folder
#############################################################################

cd $COMPILE_DIR

tar -zxvf $TAR_DIR/tomcat-connectors-$JKCON_VERSION-src.tar.gz
cd tomcat-connectors-$JKCON_VERSION-src/native/

./configure --with-apxs=$INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION/bin/apxs \
```

```
       --with-java-home=$JAVA_HOME/

make

$INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION/build/libtool \
      --finish $INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION/modules
make install

##############################################################################
#
# GRAPHICAL LIBRARY FOR PHP
#
##############################################################################

##############################################################################
# Compilation and installation of JPEG LIB
##############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION

tar -zxvf $TAR_DIR/jpegsrc.v$LIBJPEG_VERSION.tar.gz
cd jpeg-$LIBJPEG_VERSION

./configure --enable-shared \
      --enable-static \
      --prefix=$INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION

make

mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION/include
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION/lib
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION/bin
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION/man
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION/man1
make install

##############################################################################
# Compilation and installation of LibPNG
##############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/libpng/$LIBPNG_VERSION

tar -zxvf $TAR_DIR/libpng-$LIBPNG_VERSION.tar.gz
cd libpng-$LIBPNG_VERSION

./configure --prefix=$INSTALL_IMAGELIB_DIR_PREFIX/libpng/$LIBPNG_VERSION

make
make install

##############################################################################
# Compilation and installation of FreeType
##############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/freetype/$FREETYPE_VERSION

tar -zxvf $TAR_DIR/freetype-$FREETYPE_VERSION.tar.gz
cd freetype-$FREETYPE_VERSION

./configure --prefix=$INSTALL_IMAGELIB_DIR_PREFIX/freetype/$FREETYPE_VERSION
```

```
make
make install

#############################################################################
# Compilation and installation of LIBMCRYPT
#############################################################################

cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/libmcrypt/$LIBMCRYPT_VERSION

tar -zxvf $TAR_DIR/libmcrypt-$LIBMCRYPT_VERSION.tar.gz
cd libmcrypt-$LIBMCRYPT_VERSION

./configure --prefix=$INSTALL_IMAGELIB_DIR_PREFIX/libmcrypt/$LIBMCRYPT_VERSION

make
make install

#############################################################################
# Compilation and installation of LibGD
# Warning: a strange bug has been detected in 2.0.35, use 2.0.34 instead
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/gdlib/$LIBGD_VERSION

tar -zxvf $TAR_DIR/gd-$LIBGD_VERSION.tar.gz
cd gd-$LIBGD_VERSION

./configure --prefix=$INSTALL_IMAGELIB_DIR_PREFIX/gdlib/$LIBGD_VERSION
      --with-zlib=$INSTALL_DIR_PREFIX/zlib/$ZLIB_VERSION \
      --with-freetype=$INSTALL_IMAGELIB_DIR_PREFIX/freetype/$FREETYPE_VERSION \
      --with-jpeg=$INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION

make
make install

#############################################################################
# Compilation and installation of LIBICONV
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_IMAGELIB_DIR_PREFIX/libiconv/$LIBICONV_VERSION

tar -zxvf $TAR_DIR/libiconv-$LIBICONV_VERSION.tar.gz

cd libiconv-$LIBICONV_VERSION
./configure \
--prefix=$INSTALL_IMAGELIB_DIR_PREFIX/libiconv/$LIBICONV_VERSION

make
make install

#############################################################################
# Compilation and installation of LIB IMAP
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/imap/$LIBIMAP_VERSION

tar -zxvf $TAR_DIR/imap-$LIBIMAP_VERSION.tar.gz
cd imap-$LIBIMAP_VERSION
make lr5

cd $COMPILE_DIR
cp -pr imap-$LIBIMAP_VERSION/* /opt/imap/$LIBIMAP_VERSION/
```

```
#############################################################################
# Compilation and installation of LIB CURL
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/curl/$CURL_VERSION

tar -zxvf $TAR_DIR/curl-$CURL_VERSION.tar.gz
cd curl-$CURL_VERSION

./configure --prefix=$INSTALL_DIR_PREFIX/curl/$CURL_VERSION \
      --with-ssl=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION

make
make install

#############################################################################
#
# Compilation and installation of PHP
#
#############################################################################
cd $COMPILE_DIR
mkdir -p $INSTALL_DIR_PREFIX/php/$PHP_VERSION

tar -jxvf $TAR_DIR/php-$PHP_VERSION.tar.bz2
cd php-$PHP_VERSION

# PHP Configuration: be careful in modifying it
./configure --prefix=$INSTALL_DIR_PREFIX/php/$PHP_VERSION \
      --with-mysql=mysqlnd --with-
pgsql=$INSTALL_DIR_PREFIX/postgres/$POSTGRES_VERSION \
      --with-apxs2=$INSTALL_DIR_PREFIX/apache/$HTTPD_VERSION/bin/apxs \
      --with-openssl-dir=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION \
      --with-zlib=$INSTALL_DIR_PREFIX/zlib/$ZLIB_VERSION \
      --enable-gd-native-ttf \
      --with-gd=$INSTALL_IMAGELIB_DIR_PREFIX/gdlib/$LIBGD_VERSION \
      --enable-exif \
      --with-jpeg-dir=$INSTALL_IMAGELIB_DIR_PREFIX/jpeg/$LIBJPEG_VERSION \
      --with-png-dir=$INSTALL_IMAGELIB_DIR_PREFIX/libpng/$LIBPNG_VERSION \
      --with-freetype-
dir=$INSTALL_IMAGELIB_DIR_PREFIX/freetype/$FREETYPE_VERSION \
      --with-ttf \
      --with-mcrypt=$INSTALL_IMAGELIB_DIR_PREFIX/libmcrypt/$LIBMCRYPT_VERSION \
      --with-iconv=$INSTALL_IMAGELIB_DIR_PREFIX/libiconv/$LIBICONV_VERSION \
      --enable-mbstring=all \
      --with-imap=$INSTALL_DIR_PREFIX/imap/$LIBIMAP_VERSION \
      --with-imap-ssl=$INSTALL_DIR_PREFIX/openssl/$OPENSSL_VERSION \
      --with-curl=$INSTALL_DIR_PREFIX/curl/$CURL_VERSION \
      --with-pear=$INSTALL_DIR_PREFIX/php/$PHP_VERSION/pear

make
make test
make install

# copy configuration file in new installation folder
/bin/cp /etc/php.ini $INSTALL_DIR_PREFIX/php/$PHP_VERSION/lib/

#############################################################################

#############################################################################
# END OF FILE
#############################################################################
```

```
#########################################################################
#
# Script template for MONO Web Platform creation from source tarball
# (c) 2009-2012 AREA Professional - Giulio Destri
# http://www.areaprofessional.net/giulio.destri
# Released in 2012 under LGPL License
#
#########################################################################
#
# This script template has been used to maintain production servers
# based on RedHat/Fedora/CentOS 5.x platform
# Feel free to adapt it to your needs
# Please report any bug and any improvement suggestion to
# giulio.destri@areaprofessional.net
#
#########################################################################
#
# See also http://blog.palehorse.net/2008/11/06/my-adventures-installing-mono-
20-on-centos-4-to-work-with-apache-via-mod_mono/
#
#########################################################################
#
# PREREQUISITES
# Install with yum
#   glib*
#   libexif
#   giflib
#
#########################################################################
#
# Starting from now 2 working directory are defined:
# - TAR_DIR       = repository where to download and store tarballs
# - COMPILE_DIR = folder where to compile tarballs
#
# defined value are just for example
#
#########################################################################
#
#########################################################################
#
# VARIABLE DEFINITION
#
#########################################################################

# WORKING FOLDER
export TAR_DIR=/data/condivisa/lavoro/prod/mono/tar
export COMPILE_DIR=/data/condivisa/lavoro/prod/mono/compile

# INSTALLATION BASE FOLDER
export INSTALL_DIR_PREFIX=/opt
export INSTALL_MONO_PREFIX=$INSTALL_DIR_PREFIX/dotnet/mono

# SOFTWARE PACKAGE VERSIONS
export MONO_VERSION=2.10.8
export MONO_BASIC_VERSION=2.10
export MONO_GLUEZILLA_VERSION=2.6
export MONO_LIBGDI_VERSION=2.10
export MONO_ADDINS_VERSION=0.6.2
export MONO_XSP_VERSION=2.10.2
export MONO_MODMONO_VERSION=2.10
export MONO_MONODOC_VERSION=2.0
export MONO_MONOTOOLS_VERSION=2.11
export GTK_VERSION=2.12.11
```

```
export GNOMESHARP_VERSION=2.24.0
export WEBKITSHARP_VERSION=0.3

export HTTPD_VERSION=2.2.21

export PKG_CONFIG_PATH=$INSTALL_MONO_PREFIX/$MONO_VERSION/lib/pkgconfig

# DOWNLOAD MONO WEB SITE
export MONO_WEB_SITE=download.mono-project.com
export DLOAD_PROTOCOL=http://

#####################################################################
# Tarball folder creation
#####################################################################
mkdir -p $TAR_DIR/Mono
mkdir -p $TAR_DIR/Xsp
mkdir -p $TAR_DIR/MonoDevelop
mkdir -p $TAR_DIR/GTK
mkdir -p $TAR_DIR/DevelTool
mkdir -p $TAR_DIR/Other

#####################################################################
#
# MONO TARBALL DOWNLOAD
#
#####################################################################

cd $TAR_DIR/Mono
wget http://$MONO_WEB_SITE/sources/mono/mono-$MONO_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/mono-basic/mono-basic-
$MONO_BASIC_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/libgdiplus/libgdiplus-
$MONO_LIBGDI_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/gluezilla/gluezilla-
$MONO_GLUEZILLA_VERSION.tar.bz2

cd $TAR_DIR/Xsp
wget http://$MONO_WEB_SITE/sources/xsp/xsp-$MONO_XSP_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/mod_mono/mod_mono-
$MONO_MODMONO_VERSION.tar.bz2

cd $TAR_DIR/MonoDevelop
wget http://$MONO_WEB_SITE/sources/monodoc/monodoc-$MONO_MONODOC_VERSION.zip
wget http://$MONO_WEB_SITE/sources/mono-tools/mono-tools-
$MONO_MONOTOOLS_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/gtk-sharp212/gtk-sharp-$GTK_VERSION.tar.bz2
wget http://$MONO_WEB_SITE/sources/gnome-desktop-sharp2/gnome-desktop-sharp-
$GNOMESHARP_VERSION.tar.bz2

cd $TAR_DIR/Other
wget http://$MONO_WEB_SITE/sources/webkit-sharp/webkit-sharp-
$WEBKITSHARP_VERSION.tar.bz2

#####################################################################################
#####################################################################################
#
# SOFTWARE TARBALL COMPILATION
#
#####################################################################################

#####################################################################################
# Destination folder creation
#####################################################################################
```

```
mkdir -p $INSTALL_MONO_PREFIX/$MONO_VERSION

#############################################################################
# Compilation and installation of LibGDI
#############################################################################

cd $COMPILE_DIR

tar -jxvf $TAR_DIR/Mono/libgdiplus-$MONO_LIBGDI_VERSION.tar.bz2
cd libgdiplus-$MONO_LIBGDI_VERSION

./configure --prefix=$INSTALL_MONO_PREFIX/$MONO_VERSION

make
make install

#############################################################################
# Compilation and installation of MONO
# Warning: it's VERY long (about 40 minutes)
#############################################################################

cd $COMPILE_DIR
tar -jxvf $TAR_DIR/Mono/mono-$MONO_VERSION.tar.bz2

cd mono-$MONO_VERSION

./configure --prefix=$INSTALL_MONO_PREFIX/$MONO_VERSION \
      --with-libgdiplus=$INSTALL_MONO_PREFIX/$MONO_VERSION \

make
make install

#############################################################################
# First link to MONO executables in /usr/local/bin
#############################################################################

ln -f -s $INSTALL_MONO_PREFIX/$MONO_VERSION/bin/* /usr/local/bin

#############################################################################
# Compilation and installation of XSP (Mono Tomcat-like web container)
#############################################################################

cd $COMPILE_DIR

tar -jxvf $TAR_DIR/Xsp/xsp-$MONO_XSP_VERSION.tar.bz2
cd xsp-$MONO_XSP_VERSION

./configure --prefix=$INSTALL_MONO_PREFIX/$MONO_VERSION

make
make install

#############################################################################
# Apache should be already installed in /opt/apache/$HTTPD_VERSION
#############################################################################

#############################################################################
# Compilation and installation of MOD_MONO (Mono JK-like connector)
#############################################################################

cd $COMPILE_DIR
```

```
tar -jxvf $TAR_DIR/Xsp/mod_mono-$MONO_MODMONO_VERSION.tar.bz2
cd mod_mono-$MONO_MODMONO_VERSION

./configure --prefix=$INSTALL_MONO_PREFIX/$MONO_VERSION \
      --with-apxs=/opt/apache/$HTTPD_VERSION/bin/apxs \
      --with-mono-prefix=$INSTALL_MONO_PREFIX/$MONO_VERSION \
      --with-mono-default-config-dir=$INSTALL_MONO_PREFIX/$MONO_VERSION/etc/mono
\
      --enable-debug

make
make install

###############################################################################
# Final link to MONO executables in /usr/local/bin
###############################################################################

ln -f -s $INSTALL_MONO_PREFIX/$MONO_VERSION/bin/* /usr/local/bin

###############################################################################
# Link to MONO libraries in /usr/local/lib
###############################################################################

ln -f -s $INSTALL_MONO_PREFIX/$MONO_VERSION/lib/* /usr/local/lib

###############################################################################
# Run-time DLL cache activation
###############################################################################

echo '\n' > /etc/ld.so.conf.d/mono-i386.conf
echo '/opt/dotnet/mono/2.10.8/lib' > /etc/ld.so.conf.d/mono-i386.conf

ldconfig

###############################################################################

###############################################################################
# END OF FILE
###############################################################################
```

# Configurazione del sistema

Una volta ultimata compilazione ed installazione occorre configurare il sistema.

## *Configurazione di OpenSSH*

Occorre decidere se si vogliono usare le nuove chiavi del server che sono state generate durante l'installazione o se devono essere copiate entro la cartella le chiavi preesistenti nel sistema. Le chiavi sono presenti entro la cartella **<radice di ssh>\etc** e quindi, nel nostro caso, **/opt/ssh/5.9p1/etc**, cartella che contiene anche i due file principali di configurazione, **sshd_config** del servizio server e **ssh_config** del client. Nella installazione di default di molte distribuzioni le chiavi ed i file di configurazione di SSH sono invece presenti entro la cartella **/etc/ssh**.
All'inizio del file del servizio sshd, situato normalmente in **/etc/init.d/sshd**, sono presenti le definizioni delle variabili che definiscono il funzionamento di ssh nel sistema. E' necessario quindi impostare i valori di tali variabili alla nuova configurazione, come nell'esempio sotto riportato

```
KEYGEN=/opt/ssh/5.9p1/bin/ssh-keygen
SSHD=/opt/ssh/5.9p1/sbin/sshd
RSA1_KEY=/opt/ssh/5.9p1/etc/ssh_host_key
RSA_KEY=/opt/ssh/5.9p1/etc/ssh_host_rsa_key
DSA_KEY=/opt/ssh/5.9p1/etc/ssh_host_dsa_key
PID_FILE=/var/run/sshd_59.pid
SSHD_CONF=/opt/ssh/5.9p1/etc/sshd_config
```

Il file contenente il PID definito nella penultima riga è stato rinominato in sshd_59.pid per evitare conflitti con le versioni precedentemente installate nel sistema se queste devono continuare a funzionare.
Sempre per evitare conflitti tra le varie versioni, è consigliabile rinominare il file **/var/lock/subsys/sshd** usato entro le funzioni di start, stop e condrestart in **/var/lock/subsys/sshd_59**.
Infine, per consentire il test e ripristino delle varie versioni, anche i file sshd vanno rinominati in sshd_59 o comunque un suffisso dipendente dalla versione mentre sshd sia il link alla versione correntemente attiva.
Se invece il sistema è sotto il controllo di una utility come Webmin è bene non usare file pid ed altri distinti dalle versioni di produzione, ma usare sempre solo una versione attiva.
A questo punto SSH è installato correttamente nel sistema. Nel caso di eventuale installazione di una versione successiva conviene poi conservare le chiavi, rinominando la nuova cartella **<radice di ssh>\etc** e copiando la etc della versione precedente.

## *Configurazione dei moduli di Apache*

Molte distribuzioni con il servizio Apache preinstallato presentano la possibilità di inserire in una apposita cartella i file di configurazione dei singoli moduli. Similmente la compilazione delle ultime versioni di Apache httpd ha un approccio simile, con il file principale ed una serie di file secondari inclusi.
Di seguito sono presentati i file di configurazione di PHP, del JK Connector e di Tomcat, di mod_mono ed XSP (il modulo di Mono simile a Tomcat che esegue le applicazioni ASP.NET).

```
###########################################################################
# PHP 5 MODULE ACTIVATION
###########################################################################
AddType application/x-httpd-php     .php .phtml .php3 .php4 .php5
LoadModule php5_module              modules/libphp5.so
###########################################################################
# END PHP 5 MODULE ACTIVATION
###########################################################################




###########################################################################
# JK CONNECTOR MODULE ACTIVATION
###########################################################################
# Load mod_jk
#
LoadModule jk_module modules/mod_jk.so

# Configure mod_jk
#
# WORKER FILE DEFINITION
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info # (or debug if testing)

JkMount /*.jsp ajp13
###########################################################################
# END JK CONNECTOR MODULE ACTIVATION
###########################################################################
```

```
##############################################################################
# WORKER.PROPERTIES
# Adapted from Tomcat site example
# JDK 1.6.0_30
# Tomcat 5.5.34
# JK Connector 1.2.23
##############################################################################
# CATALINA_HOME of Tomcat installation
workers.tomcat_home=/opt/tomcat/apache-tomcat-5.5.34

# JAVA_HOME of Java installation
workers.java_home=/opt/java/jdk1.6.0_30

# DOS or UNIX path separator
ps=/

##############################################################
# Tomcat worker list
worker.list=ajp12, ajp13

#------ DEFAULT ajp12 WORKER DEFINITION ------------------------------
#---------------------------------------------------------------------
#
# Defining a worker named ajp12 and of type ajp12
# Note that the name and the type do not have to match.
#
worker.ajp12.port=8007
worker.ajp12.host=localhost
worker.ajp12.type=ajp12
#
# Specifies the load balance factor when used with
# a load balancing worker.
# Note:
#  ----> lbfactor must be > 0
#  ----> Low lbfactor means less work done by the worker.
#worker.ajp12.lbfactor=1

#
#------ DEFAULT ajp13 WORKER DEFINITION ------------------------------
#---------------------------------------------------------------------
#
#
# Defining a worker named ajp13 and of type ajp13
# Note that the name and the type do not have to match.
#
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
#
# Specifies the load balance factor when used with
# a load balancing worker.
# Note:
#  ----> lbfactor must be > 0
#  ----> Low lbfactor means less work done by the worker.
# worker.ajp13.lbfactor=1
#
#------ DEFAULT LOAD BALANCER WORKER DEFINITION ----------------------
#---------------------------------------------------------------------
#

#
# The loadbalancer (type lb) workers perform wighted round-robin
# load balancing with sticky sessions.
```

```
# Note:
#  ----> If a worker dies, the load balancer will check its state
#        once in a while. Until then all work is redirected to peer
#        workers.
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=ajp12, ajp13


#
#------ DEFAULT JNI WORKER DEFINITION--------------------------------
#---------------------------------------------------------------------
#

#
# Defining a worker named inprocess and of type jni
# Note that the name and the type do not have to match.
#
worker.inprocess.type=jni

#
#------ CLASSPATH DEFINITION ----------------------------------------
#---------------------------------------------------------------------
#

#
# Additional class path components.
#
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)lib$(ps)tomcat.jar

#
# Setting the command line for tomcat.
# Note: The cmd_line string may not contain spaces.
#
worker.inprocess.cmd_line=start

#
# The JVM that we are about to use
#
# This is for Java2
#
# Windows
#worker.inprocess.jvm_lib=$(workers.java_home)$(ps)jre$(ps)bin$(ps)classic$(ps)j
vm.dll
# Unix - Sun VM or blackdown
worker.inprocess.jvm_lib=$(workers.java_home)$(ps)jre$(ps)lib$(ps)i386$(ps)class
ic$(ps)libjvm.so

#
# Setting the place for the stdout and stderr of tomcat
#
worker.inprocess.stdout=$(workers.tomcat_home)$(ps)logs$(ps)inprocess.stdout
worker.inprocess.stderr=$(workers.tomcat_home)$(ps)logs$(ps)inprocess.stderr

#########################################################################
# END WORKER PROPERTIES
#########################################################################
```

```
#####################################################################
# MOD_MONO MODULE ACTIVATION
#####################################################################
# mod_mono.conf

# Achtung! This file may be overwritten
# Use 'include mod_mono.conf' from other configuration file
# to load mod_mono module.

<IfModule !mod_mono.c>
    LoadModule mono_module modules/mod_mono.so
</IfModule>

<IfModule mod_headers.c>
    Header set X-Powered-By "Mono"
</IfModule>

AddType application/x-asp-net .aspx
AddType application/x-asp-net .asmx
AddType application/x-asp-net .ashx
AddType application/x-asp-net .asax
AddType application/x-asp-net .ascx
AddType application/x-asp-net .soap
AddType application/x-asp-net .rem
AddType application/x-asp-net .axd
AddType application/x-asp-net .cs
AddType application/x-asp-net .vb
AddType application/x-asp-net .master
AddType application/x-asp-net .sitemap
AddType application/x-asp-net .resources
AddType application/x-asp-net .skin
AddType application/x-asp-net .browser
AddType application/x-asp-net .webinfo
AddType application/x-asp-net .resx
AddType application/x-asp-net .licx
AddType application/x-asp-net .csproj
AddType application/x-asp-net .vbproj
AddType application/x-asp-net .config
AddType application/x-asp-net .Config
AddType application/x-asp-net .dll
DirectoryIndex index.aspx
DirectoryIndex Default.aspx
DirectoryIndex default.aspx
#####################################################################
# END MOD_MONO MODULE ACTIVATION
#####################################################################
```

Una volta configurati i moduli, si devono attivare i singoli virtual host di Apache corrispondenti alle singole applicazioni Web. Di seguito sono presentate le configurazioni di virtual host di PHP, Java, ASP.NET, supponendo che i siti virtuali corrispondano a www.examplephp.loc, www.examplejava.loc e www.examplemono.loc e che le loro cartelle radice siano, rispettivamente, /home/web/php, /home/web/java, /home/web/mono.
Per quanto riguarda l'applicativo Java, occorre che nella cartella $CATALINA_BASE/conf/Catalina/localhost, come definita nello script di avvio del servizio Tomcat più sotto riportato, sia presente un file che imposta il path dell'applicativo, chiamato nomeapplicativo.xml, in questo esempio examplejava.xml.

```
#############################################################
# AREAPROFESSIONAL PHP TEST SITE
#############################################################
<VirtualHost *:80>
    ServerAdmin webmaster@examplephp.loc
    ServerName www.examplephp.loc
    <Directory "/home/web/php">
            Options FollowSymLinks
    </Directory>
    DirectoryIndex index.html index.php
    DocumentRoot "/home/web/php"
</VirtualHost>
#############################################################
# END AREAPROFESSIONAL PHP TEST SITE
#############################################################


#############################################################
# AREAPROFESSIONAL JAVA TEST SITE
#############################################################
JkMount /examplejava/* ajp13
<VirtualHost *:80>
    ServerAdmin webmaster@examplejava.loc
    ServerName www.examplejava.loc
    Alias  /examplejava "/home/web/java"
    <Directory "/home/web/java">
            Options FollowSymLinks
    </Directory>
    DirectoryIndex index.html index.jsp
    DocumentRoot "/home/web/java"
</VirtualHost>
#############################################################
# END AREAPROFESSIONAL JAVA TEST SITE
#############################################################


#############################################################
# AREAPROFESSIONAL MONO TEST SITE
#############################################################
<VirtualHost *:80>
    ServerAdmin webmaster@examplemono.loc
    ServerName www.examplemono.loc
    <Directory "/home/web/mono">
            Options FollowSymLinks
    </Directory>
    DocumentRoot "/home/web/mono"
    MonoAutoApplication enabled
    AddMonoApplications area "/:/home/web/mono"
    MonoExecutablePath area /opt/dotnet/mono/2.10.8/bin/mono
    MonoServerPath area /opt/dotnet/mono/2.10.8/bin/mod-mono-server2
    SetHandler mono_module
    DirectoryIndex index.aspx
    DirectoryIndex Default.aspx
    DirectoryIndex default.aspx
</VirtualHost>
#############################################################
# END AREAPROFESSIONAL MONO TEST SITE
#############################################################
```

## *Creazione del servizio Tomcat*

Esistono diversi modi con cui far partire Tomcat come servizio. Di seguito viene riportato uno script di gestione del servizio Tomcat, aggiungibile ai servizi che partono in automatico tramite il chkconfig.

```
#!/bin/bash
#
# chkconfig: 2345 85 15
# description: tomcat
# processname: tomcat
# Source function library.
. /etc/init.d/functions

RETVAL=$?
export JAVA_HOME=/opt/java/jdk1.6.0_30
export JRE_HOME=/opt/java/jdk1.6.0_30/jre
# Tomcat local configuration folder
export CATALINA_BASE="/home/web/tomcat_area"
# Tomcat binary folder
export CATALINA_HOME="/opt/tomcat/apache-tomcat-5.5.34"
export PATH=$PATH:$JAVA_HOME/bin
RETVAL=0

start() {
        if [ -f $CATALINA_HOME/bin/startup.sh ];
          then
          echo $"Starting Tomcat"
            # TOMCAT 5.x startup command with tomcat user
            /bin/su tomcat $CATALINA_HOME/bin/startup.sh
          fi
          RETVAL=$?
          echo
}
stop() {
        if [ -f $CATALINA_HOME/bin/shutdown.sh ];
          then
          echo $"Stopping Tomcat"
            # TOMCAT 5.x shutdown command with tomcat user
            /bin/su tomcat $CATALINA_HOME/bin/shutdown.sh
          fi
          RETVAL=$?
          echo
}

case "$1" in
 start)
      start
      ;;
 stop)
      stop
      ;;
  restart)
      stop
      start
      ;;
 *)
      echo $"Usage: $0 {start|stop|restart}"
      exit 1
      ;;
esac
exit $RETVAL
```

Ed ecco invece il file di configurazione che definisce il sito virtuale examplejava e che deve trovarsi in $CATALINA_BASE/conf/Catalina/localhost, nell'esempio corrente quindi /home/web/tomcat_area//conf/Catalina/localhost/examplejava.xml.

```
<!--
    Context configuration file for the Example Tomcat Java Web App
-->

<Context path="/examplejava" docBase="/home/web/java" debug="0"
Privileged="true" reloadable="true">

</Context>
```

## *Creazione di un certificato SSL e attivazione di HTTPS in Apache*

OpenSSL consente anche di creare un certificato per Apache, così da poter attivare il protocollo HTTPS. A causa dei limiti intrinseci del meccanismo, può essere creato un certificato valido per ogni coppia nome sito-IP.
Il certificato “autoprodotto” comunque, non essendo validato da una Certification Auhority, sarà segnalato come invalido dai browser e richiederà una abilitazione esplicita per accedere al sito stesso.
Di seguito viene riportato uno script per la generazione di un certificato e la configurazione per l’attivazione di un virtual host su HTTPS in Apache, supposto rispondere al sito virtuale www.examplessl.com posto nella cartella /home/web/ssl.
Secondo la convenzione suggerita nella documentazione di Apache, il nome del certificato è lo stesso del sito.

```
#####################################################################
# SSL CERTIFICATE CREATION
#####################################################################
export OPENSSL_VERSION=1.0.0e
export HTTPD_VERSION=2.2.21
#####################################################################
export LONGFILE01=/boot/vmlinuz-2.6.32-220.2.1.el6.i686
export LONGFILE02=/boot/vmlinuz-2.6.32-71.29.1.el6.i686
export LONGFILE03=/boot/vmlinuz-2.6.32-71.el6.i686
#####################################################################
export SERVER_NAME=www.examplessl.loc
export DAYLIFE_CERT=3650
#####################################################################
# SSL CERTIFICATE CREATION IN /etc/apache/conf/keys
mkdir -p /etc/apache/cert
mkdir -p /etc/apache/keys
cd /etc/apache/conf/keys

# Use of long (kernel) file for random initialization
/opt/openssl/$OPENSSL_VERSION/bin/openssl genrsa \
      -rand $LONGFILE01:$LONGFILE02:$LONGFILE03 \
      -out $SERVER_NAME.key 1024

# RSA creation
/opt/openssl/$OPENSSL_VERSION/bin/openssl rsa -in $SERVER_NAME.key \
    -out $SERVER_NAME.key.unsecure

# key creation
/opt/openssl/$OPENSSL_VERSION/bin/openssl rsa -noout -text -in $SERVER_NAME.key

# interactive phase: insert here information required for certificate
/opt/openssl/$OPENSSL_VERSION/bin/openssl req -new -key $SERVER_NAME.key \
 -out $SERVER_NAME.csr

/opt/openssl/$OPENSSL_VERSION/bin/openssl req -noout -text -in $SERVER_NAME.csr

/opt/openssl/$OPENSSL_VERSION/bin/openssl x509 -req -days $DAYLIFE_CERT \
        -in $SERVER_NAME.csr -signkey $SERVER_NAME.key \
        -out $SERVER_NAME.cert

cp -p $SERVER_NAME.cert /etc/apache/cert

#####################################################################
# END SSL CERTIFICATE CREATION
#####################################################################
```

```
#####################################################################
# SSL MODULE ACTIVATION
#####################################################################
<IfDefine SSL>
      LoadModule ssl_module modules/mod_ssl.so
</IfDefine>
#####################################################################
# END SSL MODULE ACTIVATION
#####################################################################

#############################################################
# AREAPROFESSIONAL SSL TEST SITE
#############################################################
# SSL High level
SSLProtocol all
SSLCipherSuite HIGH:MEDIUM

# HTTPS port 443 activation
Listen 443

# Site configuration
NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile \
    /etc/apache/conf/keys/www.examplessl.loc.cert
    SSLCertificateKeyFile \
    /etc/apache/conf/keys/www.examplessl.loc.key
    ServerAdmin giulio.destri@areaprofessional.net
    ServerName www.examplessl.loc
    DocumentRoot "/home/web/ssl"
</VirtualHost>
#############################################################
# END AREAPROFESSIONAL SSL TEST SITE
#############################################################
```

Una volta inserite le applicazioni PHP (il classico file phpinfo.php), Java e Mono, scaricabili dal http://www.giuliodestri.it/utility/webplatformexample.tgz ed impostate le entry dei siti virtuali in un DNS o in un file /etc/hosts, tutto dovrebbe funzionare.

## Conclusioni

In questo articolo è stato presentato un procedimento per creare un server Web completo basato su Apache e PHP, Java e Mono partendo dai sorgenti dei relativi pacchetti.
Il procedimento è stato esaustivamente testato su Linux RedHat/CentOS 5.x a 32 bit, solo parzialmente su altre piattaforme Linux.
Per segnalazioni di bug, suggerimenti, commenti, scrivete a:
giulio.destri@areaprofessional.net