

# My nginx version update script

---

Ever wanted a quick update on your SPDY patched nginx? This is what I use.

---

nginx version are rolling out quite quick; after making several mistakes when trying to upgrade by hand I finally put together an update script. Use it as you wish.

bash

```
#!/bin/bash

usage() {
cat << EOF
Usage: nginx-update.sh VERSION [upgrade]

Script to download, configure, build and optionally install
nginx [VERSION].

Examples:

upgrade.nginx-version 1.3.9
Gets nginx version 1.3.9 source code, modules code and
compile the binary.

upgrade.nginx-version 1.3.9 upgrade
Gets nginx version 1.3.9 source code, modules code and
compile the binary and if every test runs without error
( nginx -t ) nginx running nginx version is replaced with the
new binary ( and restarted ).

EOF
}

if [ -z "$1" ]; then
    usage
    exit 1
```

```
fi
```

```
#apt-get -y build-dep nginx
#dpkg -l | grep -w " build-essential " > /dev/null || apt-get
install -y build-essential
#dpkg -l | grep -w " nginx-common " > /dev/null || apt-get
install -y nginx-common
#dpkg -l | grep -w " git " > /dev/null || apt-get install -y
git
#dpkg -l | grep -w " subversion " > /dev/null || apt-get
install -y subversion

# get the basedir by reading the absolute path of the script
# and the dir if this absolute path
SCRIPT=`readlink -f $0`
SCRIPTBASE=`dirname $SCRIPT`
# the root for the whole process
BASE="/usr/src"
VERSION="nginx-$1"
BASEDIR="$BASE/$VERSION"

# where the modules are stored
MODULESDIR="$BASE/nginx-3rdparty"

# list of the modules
# unused: ngx_pagespeed -> 'https://github.com/pagespeed/
ngx_pagespeed.git'
#           ngx_syslog_patch -> https://github.com/
yaoweibin/nginx_syslog_patch.git
#
MODULES=( 'echo-nginx-module' 'headers-more-nginx-module'
'ngx_upstream_status' 'ngx-fancyindex' 'ngx_devel_kit' 'set-
misc-nginx-module' )
MODULESREPO=( 'https://github.com/agentzh/echo-nginx-
module.git' 'https://github.com/agentzh/headers-more-nginx-
module.git' 'https://github.com/petermolnar/
ngx_upstream_status.git' 'https://github.com/aperezdc/ngx-
fancyindex.git' 'https://github.com/simpl/nginx_devel_kit.git'
'https://github.com/openresty/set-misc-nginx-module' )
```

```

# nginx normal binary
NGINXBIN="/usr/sbin/nginx"

# backup nginx binaries
NGINXBACKUPDIR="/root/nginx-backups/"
DATE=`date +%Y-%m-%d`
NGINXBACKUP=$NGINXBACKUPDIR/nginx-$DATE

# the freshly compiled nginx binary
NGINXOBJ="objs/nginx"

OPENSSLVERSION="openssl-1.0.2g"
OPENSSLTAR="$OPENSSLVERSION.tar.gz"
OPENSSL="$BASE/$OPENSSLVERSION"
OPENSSLSOURCE="http://www.openssl.org/source/$OPENSSLTAR"

# create dirs if needed
if [ ! -d "$NGINXBACKUPDIR" ]; then
    mkdir -p $NGINXBACKUPDIR
fi

# create module directories
if [ ! -d "$MODULESDIR" ]; then
    mkdir -p $MODULESDIR
fi

# check if nginx tar exist
if [ -f "$BASE/$VERSION.tar.gz" ]; then
    rm "$BASE/$VERSION.tar.gz"
fi

# check if tar extractions exist
if [ -d "$BASEDIR" ]; then
    rm -rf "$BASEDIR"
fi

cd $BASE

# get nginx
echo "Getting nginx source from nginx.org"
wget http://nginx.org/download/$VERSION.tar.gz

```

```
echo "Extracting nginx source"
tar xzf $VERSION.tar.gz

# get openssl if not exist
if [ ! -d "$OPENSSL" ]; then
    wget --output-document=$OPENSSLTAR $OPENSSLSOURCE
    tar xzf $OPENSSLTAR
fi

#echo "Changing dir to nginx base"
#cd $BASEDIR

# basic configuration parameters
CONFIG="./configure \
    --prefix=/etc/nginx \
    --conf-path=/etc/nginx/nginx.conf \
    --error-log-path=/var/log/nginx/error.log \
    --http-client-body-temp-path=/var/lib/nginx/body \
    --http-fastcgi-temp-path=/var/lib/nginx/fastcgi \
    --http-log-path=/var/log/nginx/access.log \
    --http-proxy-temp-path=/var/lib/nginx/proxy \
    --lock-path=/var/lock/nginx.lock \
    --pid-path=/var/run/nginx.pid \
    --without-http_browser_module \
    --without-http_empty_gif_module \
    --without-http_geo_module \
    --without-http_referer_module \
    --without-http_scgi_module \
    --without-http_split_clients_module \
    --without-http_ssi_module \
    --without-http_userid_module \
    --without-http_uwsgi_module \
    --with-http_ssl_module \
    --with-openssl=$OPENSSL \
    --with-http_stub_status_module \
    --with-http_mp4_module \
    --with-http_flv_module \
    --with-ipv6 \
    --with-http_realip_module \
    --with-http_v2_module"
```

```

# run through all modules and pull the repositories for
updates
echo "Updating modules, if possible"
for ((i=0; i<${#MODULES[@]}; i++)); do
    MOD=${MODULESDIR}/${MODULES[$i]}

    if [ ! -d "$MOD" ]; then
        cd $MODULESDIR
        echo "Trying to check out module ${MODULES[$i]}"
        if [[ "${MODULESREPO[$i]}" == *git* ]]; then
            git clone ${MODULESREPO[$i]} ${MODULES[$i]}
        elif [[ "${MODULESREPO[$i]}" == *svn* ]]; then
            svn checkout ${MODULESREPO[$i]} ${MODULES[$i]}
        fi
    else
        cd $MOD
        echo "Trying to update module ${MODULES[$i]}"
        if [ -d ".git" ]; then
            git pull -a -f -v
        elif [ -d ".svn" ]; then
            svn update
        fi
    fi

    if [ "${MODULES[$i]}" == "nginx_syslog_patch" ]; then
        cd $BASEDIR
        patch -p1 < $MOD/syslog_1.4.0.patch
    fi

    CONFIG=$CONFIG " --add-module=$MOD"
done

# return to the basedir, run configure and make
cd $BASEDIR

echo "Applying config"
eval $CONFIG

echo "Starting to compile"
make -j4

```

```
# if anything was added at the end of the call, replace the
current running nginx
# if the test is successful
if [ "$2" == "upgrade" ]; then
    TEST=`./$NGINXOBJ -t 2>&1 | grep "successful" | wc -l`
    if [ $TEST -ne 0 ]; then
        cp $NGINXBIN $NGINXBACKUP
        cp -f $NGINXOBJ $NGINXBIN
        OLDPID=`cat /var/run/nginx.pid`
        kill -USR2 $OLDPID
        sleep 3
        kill -QUIT $OLDPID
    else
        eval "./$NGINXOBJ -t"
    fi
fi
```

Created by Peter Molnar <[mail@petermolnar.net](mailto:mail@petermolnar.net)>, published at 2012-09-28 08:00 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/nginx-version-update-script/> , licensed under CC-BY-4.0 .