Installing the NGINX 1 Web Server

This document describes how to install the version of the NGINX 1 web server that is located on the class website, version 1.0.15. You may obtain this version either by clicking here or by going to the class web page and clicking on the appropriate link under the Software tab. You should use this version of NGINX, and not later versions available at nginx.org, as these versions will not properly compile on Sun SPARC architecture, such as cs-server.usc.edu. If you need assistance you should read the online documentation at https://nginx.org/en/docs/.

The following procedure describes how to build and install the NGINX 1.0.15 source distribution. Once again, this procedure works ONLY FOR the version of NGINX located on the course web site, namely 1.0.15.

Please run your server on cs-server.usc.edu; DO NOT run your server on nunki.usc.edu or aludra.usc.edu, otherwise, your account will be suspended.

Download and Unpack NGINX 1 Package

Step 1: Download the 1.0.15 source version of the NGINX web server from the class website by clicking here or go to the course home page and click on the appropriate link under the Software tab. Upload the file to your class user account home directory.

Assume your user account home directory is **/home/scf-22/myname/**, and you want to unpack the nginx files there.

Step 2: unpack the tar package by running "tar xvf nginx-1.0.15.tar"

Step 3: after several minutes, a directory tree **/home/scf-22/myname/nginx-1.0.15** will be created.

Step 4: you should delete nginx-1.0.15.tar which is no longer needed by running "rm nginx-1.0.15.tar"

Download and Unpack PCRE 8 Library Package

Step 1: Download the 8.40 source version of the PCRE (Perl Compatible Regular expressions) library, required by NGINX, from the class website by clicking here or go to the course home page and click on the appropriate link under the Software tab. Upload the file to your class user account home directory.

Assume your user account home directory is **/home/scf-22/myname/**, and you want to unpack the PCRE files there.

Step 2: unpack the tar package by running "tar xvf pcre-8.40.tar"

Step 3: after several minutes, a directory tree **/home/scf-22/myname/pcre-8.40** will be created.

Step 4: you should delete pcre-8.40.tar which is no longer needed by running "rm pcre-8.40.tar"

Install NGINX 1.0.15 Binary

Connect to the NGINX source distribution directory just created and do a "pwd" command to get the full path to the apache directory, as in:

```
cs-server.usc.edu(2): cd nginx-1.0.15
cs-server.usc.edu(3): pwd
/home/scf-22/csci571b/nginx-1.0.15
cs-server.usc.edu(4):
```

Part 1: Source creation and configuration.

The NGINX installation needs a version of 'sed' that does not truncate long lines. To get that version on cs-server.usc.edu, please run this:

```
setenv PATH "/usr/bin:$PATH"
```

Then type the following command to create the source files and configure them to your "hardware" environment:

```
./configure --prefix=/absolute/path/to/nginx/directory --with-pcre=/absolute/path/to/pcre/directory
```

Note: The directory **/absolute/path/to/nginx/directory/** SHOULD BE a new directory, e.g. /home/scf-22/myname/nginx, so that you can erase the distribution directory /home/scf-22/myname/nginx-1.0.15 at the end of this installation process. Therefore, we strongly recommend that you use a **new directory** for this installation. The directory **/absolute/path/to/pcre/directory/** is the directory where you extracted the PCRE source library, such as /home/scf-22/myname/pcre-8.40.

During installation, which could last as long as 10 minutes, you will see hundreds of output lines. Many will look like **checking for _name_ found/not found**. At the beginning and end of the source code creation, you should see output similar to the one below (most of the "checking" and "creating" lines have been omitted):

```
cs-server.usc.edu(5): ./configure --prefix=/home/scf-22/csci571b/nginx --with-
pcre=/home/scf-22/csci571b/pcre-8.40
checking for OS
 + SunOS 5.10 sun4v
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 4.2.1
checking for gcc -pipe switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
checking for unistd.h ... found
checking for inttypes.h ... found

. . . . . lots of other messages . . . . . .

checking for zlib library ... found
creating objs/Makefile

Configuration summary
  + using PCRE library: /home/scf-22/csci571b/pcre-8.40
  + OpenSSL library is not used
  + md5: using system md library
  + sha1: using system crypto library
  + using system zlib library

  nginx path prefix: "/home/scf-22/csci571b/nginx"
  nginx binary file: "/home/scf-22/csci571b/nginx/sbin/nginx"
  nginx configuration prefix: "/home/scf-22/csci571b/nginx/conf"
  nginx configuration file: "/home/scf-22/csci571b/nginx/conf/nginx.conf"
  nginx pid file: "/home/scf-22/csci571b/nginx/logs/nginx.pid"
  nginx error log file: "/home/scf-22/csci571b/nginx/logs/error.log"
  nginx http access log file: "/home/scf-22/csci571b/nginx/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http uwsgi temporary files: "uwsgi_temp"
  nginx http scgi temporary files: "scgi_temp"
cs-server.usc.edu(9):
```

Part 2: Source compilation and linking.

Do **not** change directory. Then type the following command to compile all source files and create (i.e. "make" in UNIX lingo) the **nginx** binary file:

```
make
```

The compilation will **take a long time** (as long a 20 minutes).

Part 3: Library Linking and installation.

Once compilation is finished, you can install the **nginx** binary using:

```
make install
```

This phase will last about 3 minutes. You will see several lines of the type "make..." and "mkdir...". With many of those lines removed, you will see output similar to the one

below:

```
cs-server.usc.edu(13): make install
make -f objs/Makefile install
test -d '/home/scf-22/csci571b/nginx' ||
mkdir -p '/home/scf-22/csci571b/nginx'
test -d '/home/scf-22/csci571b/nginx/sbin'              ||
mkdir -p '/home/scf-22/csci571b/nginx/sbin'
test ! -f '/home/scf-22/csci571b/nginx/sbin/nginx'              ||
mv '/home/scf-22/csci571b/nginx/sbin/nginx'                  '/home/scf-22/
csci571b/nginx/sbin/nginx.old'
cp objs/nginx '/home/scf-22/csci571b/nginx/sbin/nginx'
test -d '/home/scf-22/csci571b/nginx/conf'              ||
mkdir -p '/home/scf-22/csci571b/nginx/conf'
cp conf/koi-win '/home/scf-22/csci571b/nginx/conf'
cp conf/koi-utf '/home/scf-22/csci571b/nginx/conf'
cp conf/win-utf '/home/scf-22/csci571b/nginx/conf'
test -f '/home/scf-22/csci571b/nginx/conf/mime.types'           ||
cp conf/mime.types '/home/scf-22/csci571b/nginx/conf'
cp conf/mime.types '/home/scf-22/csci571b/nginx/conf/mime.types.default'
test -f '/home/scf-22/csci571b/nginx/conf/fastcgi_params'               ||
cp conf/fastcgi_params '/home/scf-22/csci571b/nginx/conf'
cp conf/fastcgi_params          '/home/scf-22/csci571b/nginx/conf/
fastcgi_params.default'
test -f '/home/scf-22/csci571b/nginx/conf/fastcgi.conf'                 ||
cp conf/fastcgi.conf '/home/scf-22/csci571b/nginx/conf'
cp conf/fastcgi.conf '/home/scf-22/csci571b/nginx/conf/fastcgi.conf.default'
test -f '/home/scf-22/csci571b/nginx/conf/uwsgi_params'                 ||
cp conf/uwsgi_params '/home/scf-22/csci571b/nginx/conf'
cp conf/uwsgi_params            '/home/scf-22/csci571b/nginx/conf/
uwsgi_params.default'
test -f '/home/scf-22/csci571b/nginx/conf/scgi_params'          ||
cp conf/scgi_params '/home/scf-22/csci571b/nginx/conf'
cp conf/scgi_params             '/home/scf-22/csci571b/nginx/conf/
scgi_params.default'
test -f '/home/scf-22/csci571b/nginx/conf/nginx.conf'           ||
cp conf/nginx.conf '/home/scf-22/csci571b/nginx/conf/nginx.conf'
cp conf/nginx.conf '/home/scf-22/csci571b/nginx/conf/nginx.conf.default'
test -d '/home/scf-22/csci571b/nginx/logs'              ||
mkdir -p '/home/scf-22/csci571b/nginx/logs'
test -d '/home/scf-22/csci571b/nginx/logs' ||
mkdir -p '/home/scf-22/csci571b/nginx/logs'
test -d '/home/scf-22/csci571b/nginx/html'              ||
cp -R html '/home/scf-22/csci571b/nginx'
test -d '/home/scf-22/csci571b/nginx/logs' ||
mkdir -p '/home/scf-22/csci571b/nginx/logs'
cs-server.usc.edu(14):
```

If you have reached this point successfully, you now have a ready-to-customize binary. Congratulations!

Configure/Customize NGINX

NGINX 1 uses one configuration file, nginx.conf. We will need to change this configuration file before starting the NGINX 1 web server.

Assuming you have followed the instructions above, the configuration file nginx.conf is contained in the folder **/home/scf-22/myname/nginx/conf**. Change to that directory:

```
cs-server.usc.edu(9): cd /home/scf-22/csci571b/nginx/conf
```

Edit main configuration file, nginx.conf

If not already present, copy of the default configuration file, with:

```
cs-server.usc.edu(10): cp nginx.conf.default nginx.conf
```

Open the file **nginx.conf** using a text editor. **PLEASE** examine the directives shown below and make sure they are properly set.

The following set of directives must be changed:

o listen **port_selected** (for port_selected replace port 80 using the first 5 digits of your class ID, **PLUS 1**, as the base for this value), as in:

```
    server {
        listen       45679;
        server_name  localhost;
```

o add index.php to the root files, as in:

```
        location / {
            root   html;
            index index.html index.php index.htm;
        }
```

o Uncomment the lines the make NGINX a reverse-proxy for Apache PHP scripts, as in:

```
        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        location ~ \.php$ {
            proxy_pass   http://127.0.0.1;
        }
```

o Change the proxy_pass to use the port of your Apache web server by adding :port# after the localhost IP address 127.0.0.1, as in:

```
        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        location ~ \.php$ {
            proxy_pass   http://127.0.0.1:45678;
        }
```

o Uncomment the block that denies access to .htaccess files, if Apache's document root concurs with nginx's one, as in:

```
        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        location ~ /\.ht {
            deny  all;
        }
```

o To make Apache accessible only from the 'localhost' (127.0.0.1), change the Listen argument in the Apache httpd.conf configuration file, and restart Apache, as in:

```
        # Change this to Listen on specific IP addresses as shown below to
        # prevent Apache from glomming onto all bound IP addresses.
        #
        Listen 127.0.0.1:45678
```

You have now completed all the steps to configure, create, install and customize NGINX 1. **Congratulations!**

Starting and Stopping NGINX

Students MUST run their server on cs-server.usc.edu. To start the server, change to the NGINX s*bin* directory (e.g., /home/scf-22/csci571b/nginx/sbin, in the example above) and type **nginx**. The following output should be displayed:

```
cs-server.usc.edu(10): nginx
cs-server.usc.edu(11):
```

It is a good practice to check the error log file, e.g. /home/scf-22/csci571b/nginx/logs/error.log, to ensure your server is properly running. You may see something like this:

```
2017/06/29 14:47:38 [error] 11338#0: *2 open() "/home/scf-22/csci571b/nginx/html/
favicon.ico" failed (2: No such file or directory), client: 209.36.208.226, s
erver: localhost, request: "GET /favicon.ico HTTP/1.1", host: "cs-server.usc.edu:
45679", referrer: "http://cs-server.usc.edu:45679/"
```

This is fine as favicon.ico is normally not present by default.

At this point you should log out and immediately log back in. The reason is that the "temporary" change to the PATH made in Part1, so that the right version of 'sed' program could run, breaks access to the 'ps' program.

You can also check that the NGINX "master" and "worker" processes are running, using the 'ps' UNIX command:

```
cs-server.usc.edu(6): ps -gx
   PID TT       S  TIME COMMAND
  1202 ?        S  0:00 /usr/lsd/openssh/default/sbin/sshd -R
  1665 ?        S  0:00 nginx: master process nginx
  1666 ?        S  0:00 nginx: worker process
  1822 ?        S  0:00 /home/scf-22/csci571b/apache2/bin/httpd -k start
```

```
  1823 ?        S  0:00 /home/scf-22/csci571b/apache2/bin/httpd -k start
  1824 ?        S  0:00 /home/scf-22/csci571b/apache2/bin/httpd -k start
  6437 ?        S  0:00 /usr/lsd/openssh/default/sbin/sshd -R
  6439 pts/3    S  0:00 -tcsh
  7065 pts/3    O  0:00 ps -gx
cs-server.usc.edu(7):
```

If NGINX started successfully, you will see running the 2 highlighted processes above.

Similarly, to stop the server, in directory s*bin*, type **nginx -s stop**. The following output should be displayed:

```
cs-server.usc.edu(12): nginx -s stop
```

Testing Your Server

By issuing the URL

`http://cs-server.usc.edu:port_selected/`

you should see the NGINX default home page below:

Welcome to nginx!

Finishing Up Your Web Server Setup

If you installed NGINX 1 in a directory other than the source distribution, as we recommended, you can now safely delete the distribution directory, its sub-directories and all its files with:

```
cs-server.usc.edu(12): /usr/bin/rm -r ~/nginx-1.0.15
```

Useful Links for Compiling, Installing and Configuring the NGINX Web Server

NGINX Beginner Guide - https://nginx.org/en/docs/beginners_guide.html

NGINX Support - https://nginx.org/en/support.html

How To Configure Nginx as a Reverse Proxy for Apache, by Etel Sverdlov - https://www.digitalocean.com/community/tutorials/how-to-configure-nginx-as-a-reverse-

proxy-for-apache