NGINX

from zero to hero

Installation scenarios

Installation methods

- In Nginx, you have two ways of installation: either to download and install the precompiled binaries, or to compile it from source.
- The advantages of using precompiled binaries is that it is very easy to setup. Just one or two lines on the command line and Nginx is installed.
- However, sometimes you are obliged to compile Nginx from source like:
  - Although Nginx is widely supported on well known Linux distributions, you may be working on a system where there are no precompiled binaries available. Accordingly you'll have to compile it from source files.
  - As mentioned in the previous section, Nginx is modular by design. This means that enabling/disabling modules is done by specifying the appropriate command line options in the compile command.
- Both of those methods will be discussed in this section.

Common prerequisites for precompiled binaries installation

- Whether you will be working on Ubuntu or Centos, the following packages need to be installed to have a successful Nginx installation:
  - Lynx: this is a text-based web browser. It will aid in situation where you want to check whether or not the web server is running through the CLI.
  - Wget: this is a tool used to download files from the Internet using HTTP, HTTPS or FTP.
  - Vim or Nano: the editor that will be used to work with various configuration files throughout your work with Nginx.
- All the above tools can be installed using the native package manager of each system. That is, yum for Centos and Red Hat based systems, and apt-get for Ubuntu and Debian based ones.

LAB 01: Installing Nginx on Centos 7

- In this lab we are going to use Centos 7 installed on VirtualBox. You can VMWare player or whatever virtualization platform you favor. You can also use a physical standalone server for this lab. You can download VirtualBox from www.virtualbox.org and Centos 7 from www.centos.org

- All what you have to do is create a new file /etc/yum.repos.d/nginx.repo and add the following lines:

```
[Nginx]
name="Nginx Repository"
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1
```

- Then issue `yum -y install nginx`

- That's it! You can verify that you have a successful Nginx installation by issuing `nginx -v` or `nginx -V` for a more verbose output.

- You may want to enable port 80 on your firewall (if it is turned on). This can be done by issuing

```
firewall-cmd --permanent --add-port=80/tcp && firewall-cmd --reload
```

LAB 02: Installing Nginx on Ubuntu 16.10 server

- Similar steps should be followed on Ubuntu, with the due differences.

- To add the appropriate repository to the Ubuntu package sources, edit `/etc/apt/sources` and add the following lines:

  ```
  deb http://nginx.org/packages/mainline/ubuntu/ yakkety nginx
  deb-src http://nginx.org/packages/mainline/ubuntu/ yakkety nginx
  ```

- Additionally, you'll need to download and install the GPG key to be able to download software from the newly added repository. Execute the following commands on the CLI:

  ```
  wget http://nginx.org/keys/nginx_signing.key
  sudo apt-key add nginx_signing.key
  ```

- Start the installation process by updating the repo sources and then installing the software:

  ```
  sudo apt-get update
  sudo apt-get install nginx
  ```

- Verify that you have a successful installation by issuing the following command:

  ```
  nginx -v
  ```

  It should give you some information about your Nginx installation including the version. Type `nginx -V` for a more verbose output. Please take a note of the output of the later command as we are going to use it when compiling the program from source.

- To enable Nginx traffic through the firewall issue the following command: `iptables -I INPUT -p tcp --dport 80 -j ACCEPT`

Post installation

- After you've installed Nginx, let's have a look at the files and directories that got created as a result of this installation:
  - The configuration files are located in `/etc/nginx`
  - The binary file that starts the daemon is located under `/usr/sbin`. You will need root privileges to start or stop the service.
  - The web directory from which Nginx serves web pages by default is located under `/usr/share/nginx`
  - Log files can be found under `/var/log/nginx`. You will find `access.log` file, which logs all requests that is received by the server, and `error.log` which logs all error messages encountered by the server.
  - Using your browser (or Lynx), head on to the IP address of the server, you should see the welcome screen of Nginx.

LAB 03: Installing Nginx from source

- Compiling any program from source is not as easy as installing the precompiled binaries. Nginx is no exception but sometimes it is worth the effort.

- Before we can start compiling the source files we need to prepare the OS environment by installing some prerequisites:

  - Download the source file of Nginx using wget (or whatever tool of your choice) from http://nginx.org/en/download.html

  - The build tools: These are various commands in addition to the gcc compiler that will be used in the compilation process. On Centos issue `yum -y group install "Development Tools"` on Ubuntu issue `apt-get install build-essential`

  - Perl Compatible Regular Expressions: they are needed by Nginx for various scenarios including URL rewriting. On Ubuntu issue `apt-get install libpcre3 libpcre3-dev` on Centos issue `yum -y install pcre pcre-devel`

  - OpenSSL: this would be used by Nginx when serving HTTPS content. On Ubuntu issue `apt-get install openssl libssl-dev`. On Centos issue `yum -y install openssl openssl-devel`

- If you built any software before you are already aware that the operation basically involves three commands: `./configure`, `make`, and `make install`

- In Nginx compilation, the `./configure` command is the one with the most concern. Combined with several command-line options you enable/disable modules and you can also change a lot of the default behavior of the server. For example, Nginx by default uses a user called nginx with a group of the same name to be the owner of the daemon. You can change this to another user using `--user=`*`user`* and `--group=`*`group`*. Beware, however, that root should <u>never</u> be used as the process owner.

- For a list of the different command line options that can be used with Nginx you can have a look at the official page: http://nginx.org/en/docs/configure.html or issue `./configure --help` from inside the Nginx uncompressed directory.

- You can also add this-party modules (modules not developed by Nginx team) by downloading the module file, uncompressing it, and pointing using `./configure --add-module=/path/to/module`.

- Last thing before compiling Nginx is to create the necessary user and group that will be used by the process. You can create any user/group but let's stick with the defaults: user `groupadd -r nginx && useradd -r nginx -g nginx`. Optionally you can set a password for the user although no one should normally use this account for logging in to the system.

- Finally let's issue the ./configure command, it is the same on Centos and Ubuntu (make sure to use sudo on Ubuntu):

  ```
  ./configure --prefix=/etc/nginx --user=nginx --group=nginx --sbin-path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --pid-path=/var/run/nginx.pid --lock-path=/var/run/nginx.lock --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --with-http_gzip_static_module --with-http_stub_status_module --with-http_ssl_module --with-pcre --with-file-aio --with-http_realip_module --without-http_proxy_module
  ```

- Following it, issue `make && make install` (on Ubuntu make sure you use sudo).

- Now Nginx has been installed successfully on your system. You can verify that by issuing `nginx -V`. Notice the different message that was printed indicating different compile options that were used.

- Start the service using `service nginx start`

- Navigate to the IP address of your machine using Lynx or your favorite browser. You should see the welcome message of Nginx.