Introduction

Path Options

- `configure` command: specify directory or file paths for a variety of elements

```
./configure --help

# --SWITCH=DEFAULT # Usage
--prefix=/usr/local/nginx       # Base folder Nginx installed
--sbin-path=/sbin/nginx                    # Nginx Binary file
--conf-path=/conf/nginx.conf    # Main Config File
--error-log-path=logs/error.log               # Error Log
--pid-path=/logs/nginx.pid                     # pid file
--lock-path=/logs/nginx.lock                    # Lock File
--with-perl=             # Perl Binary File, used to run Perl scripts
--http-log-path
--http-proxu-temp-path
--http-fastcgi-temp-path
--builddir
```

Miscellaneous Options

Options available in the configuration script

```
--with-mail # enables mail sv proxy module. Sups POP3, IMAP4, SMTP.
Disabled by default
--with-mail_ssl_module # Enables SSL support for the mail server proxy
--without-mail_pop3_module # Disables POP3 module for the mail server
proxy. Enabled by default when mail server proxy module is enabled
--without-mail_imap_module
--without-mail_smtp_module
--with-rtsig_module # enables rtsig
--with-select_module # enables select module. Enabled by default
--without-select_module
--with-poll_module # poll event notification mecanism
```

```
--user=... # Default user account to start nginx worker processes. Used
only if you do not specify the group directive in the configuration file
--group=... # DEfault user group to start Nginx worker processes. Used only
if you do not specify the group directive in the configuration file
```

```
--with-ipv6 # Enables IPv6
--without-http
--without-http-cache
--add-moudle=PATH # Adds a third-party module to the compile process.
--with-debug # Enables aditinal debugging information to be logged
```

Config Examples

Regular HTTP and HTTPS Servers

- HTTP and HTTPS content enabled
- Mail-related options disabled

```
./configre --user=www-data --group=www-data --with-http_ssl_module --with-
http_realip_module
```

Mail Server Proxy

```
./configure --user=www-data --group=www-data --with-mail --with-
mail_ssl_module
```

Compiling and Installing the Program

Once the configure script is succesfully executed you can proceed with compiling Nginx by the make command in the project source directory

```
make
```

A succesful build shoudl result in the appearance of a final message ````make: leaving directory```

The next step is installing the application

```
make install
```

It performs a fre simple operations copying binaries and config files to the install folder. Also creates directories to store log and HTML files.

Controlling Nginx Service

The default location for the output files is /usr/local/nginx.

User And Group

A very common source of trouble wieh setting up Nginx is invalid file access per,issions. You often end up getting 504 Firbidden HTTP errors.

- Nginx master process: This should be started as root, to open TCP sockets on any ports. IF you do not start as root, std ports such 80 or 554 will not be accessible.
- Nginx Worker processes: Automatically spawned by the master process under the account you specified in the configuration file with the user directive.

Starting and Stopping the Daemon

You can start nginx by running Nginx binary without any switches. You may control the daemon by stopping it, restarting it, or simply reloading its configuration. Controlling is done by sending signals to the process using the nginx -s command

- nginx -s stop
- nginx -s quit
- nginx -s reopen
- nginx -s reload

An alternative way to terminate the process in desperate cases only is to use the `kill` or ````killall``` commands with root privileges

```
killall nginx
```

Testing Config

Testing the validity of your config will become crucial if you constantly tweak yoyr server setup. The following command will be useful to allows you to check the syntax, validity and integrity of your configuration

```
/usr/local/nginx/sbin/nginx -t
```

The -t swtich stands for test configuration. Nginx will parse the configuration anew and let you know whether is valid or not. A Valid coinfiguration file does not necssarily eman Nginx will start, thouhgh as there might be additional probles such as socket issues, invalid paths, or incorrect access permissions.

- Manipulating your configuration files when server is in produciton is a dangerous thing to do and should be avoided when possible.
- The best practice in this case is to pace your new configration into a separate teporary file and run the test on that file.
- Nginx makes it possible by offering the -c switch

```
./nginx -t -c /home/username/test.conf
```

This command will [arse /home/alex/test.conf and make sure it is a valid Nginx configuration file. When its done, after making sure that new file is valid, proceed to replacing your current confguration file and reload the server configuration.

```
cp -i /home/alex/test.conf /usr/local/nginx/conf/nginx.conf
./nginx -s reload
```

Adding Nginx as a System Service

In this section we will create a script that will tansform the Nginx daemon into an actual system service. The daemon will be controllable using standard commands and it will be launched automatically con system startup and stopped on system shutdown

The Linux Based system startup process is managed by a daemon called `init` which functions in a way that is inherited from the old SystemV.

This daemon functions on the principle of runlevels, which represent the state of the computer.

0. System is halted
1. single user mode
2. multiuser mode
3. full multiuser mode
4. not used
5. grahic interface mode
6. system reboot

You can manually initiate a runlevel transition using telinit command. 0 to shutdown, 6 to reboot.

For each runlevel transition, a set of services are executed:

- When computer is stopped, runlevel is 0
- Turn it on: 0 to default startup runlevel, defined by your own system configuration in `/etc/inittab`. Debian and Ubntu use runlevel 2.

For each runlevel, there is a directory containing scripts to be executed (rcX.d) Service startup scripts will indeed be placed in init.x and links will be created by tools placing them in the proper directories.

An init script, also kown as service startup script or sysv script, is a shell script respecting a certain standard. The script controls a daemon app byh responding to commands such as start, stop and others, which are triggered at two levels

1. When computer starts, if the service is scheduled to be started for the system runlevel, the init daemon will run the script with the statr argument.

```
# OS with service command
service httpd start
```

```
# OS without service command
/etc/init.d/httpd start
```

The script must accept at least the start, stop, restart, force-reload and status comands, as the will be used by the system respectivamente. It is often interesting to provide further options, such as a reload argument to reload the service configuration. Or a try-restart argument to stop and start the service again