Deep-Dive 1: Wiki & Dynamic Web Server Selection

**Web server table**

The following table shows the four optional web server software that are available. This table contains the License that each web server uses, its dependencies and scripting language as well as basic features of the software.

| Webserver | License | Dependencies | Scripting Language | Basic Features |
|---|---|---|---|---|
| **Apache HTTP Server** | Apache License 2.0 | Apr-Util-1.6.1<br>ANSI-C Compiler<br>PCRE-8.43<br>Accurate time keeping<br><br>Optional-Dependencies<br>OpenLDAP-2.4.47<br>"(Apr-Util-1.6.1 needs to be installed with ldap support)"(Unknown, 2019-04-01)<br><br>(Linuxfromscratch.org, 2019) | **C,XML** | - Loadable Dynamic Modules<br>- Virtual hosting<br>- IPv6-compatible<br>- HTTP/2 support<br>- Request processing rate limiting<br>- Bandwidth throttling<br>- IP address-based geolocation<br>- User and Session tracking<br>- Real-time status views<br>- Generic expression parser<br>- Reverse proxy with caching.<br>- Highly scalable<br>- Handling of static files, index files<br>- Load Balancing<br>- Multiprocessing Modules(MPMs)<br>- XML Support<br>- Many features implemented as compiled modules which extend core functionality<br>(En.wikipedia.org, 2019) |
| **Apache Tomcat** | Apache License 2.0 | commons-cli<br>tomcat-api<br>tomcat-annotations-api<br>tomcat-catalina<br>tomcat-catalina-ha<br>tomcat-coyote<br>tomcat-dbcp | **Java** | - JSP/Servlet applications.<br>- High availability to facilitate scheduled system upgrades.<br>- Cluster to manage large applications for load balancing |

| | | | | |
|---|---|---|---|---|
| | | tomcat-el-api<br>tomcat-jasper<br>tomcat-jasper-el<br>tomcat-jdbc<br>tomcat-jsp-api<br>tomcat-servlet-api<br>tomcat-tribes<br>tomcat-util<br>tomcat-embed-core<br>ecj | | - Catalina (Apache Tomcat servlet container)<br>- Jasper(Apache Tomcat JSP Engine) 'Parses JSP files to compile them into Java code as servlets (that can be handled by Catalina)' (En.wikipedia.org, 2019) |
| **Lighttpd** | 3-Clause BSD | lighttpd<br>lighttpd-doc<br>ighttpd-filesystem<br>lighttpd-modules-ldap<br>lighttpd-modules-mysql<br><br>(reposcope.com, 2019) | **C** | - Low memory usage "Light-weight (less than 1 MB)" (En.wikipedia.org, 2019)<br>- Small CPU load<br>- speed optimizations<br>- Load Balancing<br>- FastCGI<br>- SCGI<br>- HTTP proxy support<br>- Modules support<br>- Servlet support |
| **Nginx** | 2-Clause BSD | PCRE version 8.42<br><br>zlib version 1.2.11<br><br>OpenSSL version 1.1.0h | **C** | - HTTP proxy/Web server/ Mail proxy<br>- Handling of static files, index files and auto-indexing<br>- Load balancing<br>TLS/SSL with SNI and OC SP stapling support<br>- FastCGI, SCGI, uWSGI su pport with caching<br>- URL rewriting and redirection<br>- IPv6-compatible<br>- Name- and IP address-based virtual servers<br>- TLS/SSL support<br>- STARTTLS support<br>- Media Streaming<br>- reverse proxying for non-Http protocols |

**License Table**

| **Apache License 2.0** | **2-Clause BSD** | **3-Clause BSD** |
|---|---|---|
| Open source license<br>Required to provide copyright notice and disclaimer | Open source license<br>"Only difference between 2-Clause BSD and 3-Clause BSD is that this license omits the non-endorsement clause and adds further disclaimer about views and opinions expressed in the software." (Wikipedia, 22 March 2019) | Open source license<br>Allows for unlimited distribution for any purpose but requires copyright notice and disclaimers and warranties. |

**Scripting Language**

The following table provides the three scripting languages used by the selected web servers and contains a brief list of the attributes of each scripting language.

| **C** | **Java** |
|---|---|
| - Procedural Programming Language<br>- Mid-level Language<br>- function oriented<br>- procedure-oriented<br>- Developed between 1969 and 1973. | - Object oriented programming language<br>- Inheritance<br>- Abstraction<br>- Polymorphism<br>- Encapsulation<br>- data-oriented<br>- Developed in 1995 |

**Selection Criteria Web Server**

The following table shows the selection criteria amongst the researched web servers. The selection criteria is simple but effective as each category is easy to research and compare amongst the different contending web servers.

The ease of installation is a self explaining category as a difficult to install service is time wasted and unnecessary.

Scripting language is more dependent on personal proficiency when it comes to the different languages but Java is generally considered more accessible than C.

Expandability/Upgradability criteria is decided by the amount of modular expandability or plain upgradability of the web service beyond its initial installation.

Popularity is decided by the amount of the 2018 Web server Survey showing the most popular Web servers.

Support is decided by how much support is provided to users of the software.

The security criteria is decided by the security each web server when compared with each other.

Pass =

Fail =

| Name | Ease of Installation | Programmability/Scripting Language | Modularity/Upgradability | Popularity | Support | Security |
|---|---|---|---|---|---|---|
| **Apache HTTP Server** | | | | | | |
| **Apache Tomcat** | | | | | | |
| **Nginx** | | | | | | |
| **Lighttpd** | | | | | | |

**Reason for selection**

- Ease of installation shows that Apache HTTP server is the easiest to install
- Apache is also more flexible with loadable dynamic modules than most other webservers in the list except for Nginx which offers the same type of modular design.
- Another criteria that Apache passes is the popularity criteria which it shares with Nginx again as both are extremely but as of March 2019 Apache is at 44% of the usage of web servers while Nginx is slightly behind at 41.2% (W3techs.com, 2019)

- The final two criteria that both Nginx and Apache share are the support and security with both providing a large amount of support from the user community and documentation that allows users to make constant adjustments to the security of the web servers.
- The final choice was between Apache and Nginx with Apache being decided for its more flexible modularity and ease of installation compared to that of Nginx, although Nginx does stand out for having more advanced features such as media streaming and reverse proxying for non-Http protocols.

**Selection Criteria Scripting Language**

- The following is the selection criteria to compare the two different commonly used scripting languages used by the web server software Accessibility
- Pass =
- Fail =

| Name | Security | Accessibility | Popularity |
|---|---|---|---|
| **C** | | | |
| **Java** | | | |

**Reason for selection**

- The security criteria is a hard choice as both scripting languages are considered mid range in terms of security and are the main scripting languages of most web servers.
- Java is the more accessible and newer scripting language and is considered more accessible and easier to learn as well as being cross platform support 'Java code can be written once and executed from anywhere' (WhiteSource, 2019)
- Both Java and C are incredibly popular, C more for the reason of being developed earlier and Java for the accessibility that it provides.

**Selected Webserver**

**Apache HTTP Server**

**Apache GitHub Repository**

https://github.com/apache/httpd

**Official site**

https://httpd.apache.org/

**License**

Apache License 2.0

**Scripting language**

C,XML

**Brief description**

Apache HTTP Server is a free open source web server software released under the Apache license 2.0. Majority of instances of Apache HTTP server run on Linux,  Apache also runs on windows and a wide variety of Unix systems.

**Installation instructions on Ubuntu**

**Update local package index**

- sudo apt-get update

**Install the apache2 package**

- sudo apt-get install apache2

**Firewall for the webserver**

**Enable**

- sudo ufw enable

**List application profiles**

- sudo ufw app list

**Apache Full: This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic)**

- sudo ufw allow 'Apache Full'

**Check Status of firewall**

- sudo ufw status

**Check Status of Service**

- sudo systemctl status apache2

**Check Server IP address**

- hostname -I

**Open Browser then insert given Ip address to view running webpage**

**Webserver management commands**

**Stop Webserver**

- sudo systemctl stop apache2

**Start Webserver after being stopped**

- sudo systemctl start apache2

**Restart Webserver**

- sudo systemctl restart apache2

**Restart without stopping for configuration changes**

- sudo systemctl reload apache2

**Disable automatic start on boot (Apache is configured to start on system boot)**

- sudo systemctl disable apache2

**Re-enable Apache start on boot**

- sudo systemctl enable apache2

**Enable/Disable Modules**

a2enmod – enables modules (lists all the modules available)

- a2enmod ***

a2dismod – disables modules

- a2dismod ***

**Uninstall Apache**

- sudo apt-get purge -y apache2

**Setting Virtual Host**

“Virtual Hosts encapsulate configuration details and host more than one domain from a single server” (Digitalocean.com, 2019)

**Directory for Domain name using domain.com as the example.**

- sudo mkdir -p /var/www/ domain.com/html

**Assign ownership of the directory**

- sudo chown -R $USER:$USER /var/www/ domain.com/html

**Permissions for web roots**

- sudo chmod -R 755 /var/www/ domain.com

**Creating sample html file using Nano**

- nano /var/www/ domain.com/html/index.html

**Add following to index.html file**

```
<html>
    <head>
        <title>Welcome to domain.com</title>
    </head>
    <body>
        <h1>The domain.com server block is working</h1>
    </body>
</html>
```

**Create new virtual host at directory**

/etc/apache2/sites-available/example.com.conf

- sudo nano /etc/apache2/sites-available/example.com.conf

**Add following to conf file**

```
<VirtualHost *:80>
    ServerAdmin admin@domain.com
    ServerName domain.com
    ServerAlias www.domain.com
    DocumentRoot /var/www/domain.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

**Enable the file**

- sudo a2ensite domain.com.conf

**Disable the default site at 000-default.conf**

- sudo a2dissite 000-default.conf

**Test Configuration**

- sudo apache2ctl configtest

**Restart Apache for changes to take place.**

**Apache Files and Directories**

**Web content.**

/var/www/html

**The Apache configuration directory.**

/etc/apache2

**The Apache configuration file.**

/etc/apache2/apache2.conf

**The directory where per-site V-Hosts can be stored.**

/etc/apache2/sites-available/

**The directory where enabled per-site V-hosts are stored.**

/etc/apache2/sites-enabled/

**File that specifies the ports that Apache will listen on.**

/etc/apache2/ports.conf

**File relationships sites-available and sites-enabled used to store configuration fragments that do not belong in Virtual Host.**

/etc/apache2/conf-available/

/etc/apache2/conf-enabled/

**Directories that contain the available and enabled modules, respectively.**
.load Files will load specific modules.
.conf Files contain the configuration of those modules.

/etc/apache2/mods-available/

/etc/apache2/mods-enabled/

**Server Logs**

**Every request to the web server is stored in this log file unless Apache is configured to do otherwise.**

/var/log/apache2/access.log

**Errors are recorded in this file**

/var/log/apache2/error.log

**Screen Shots**

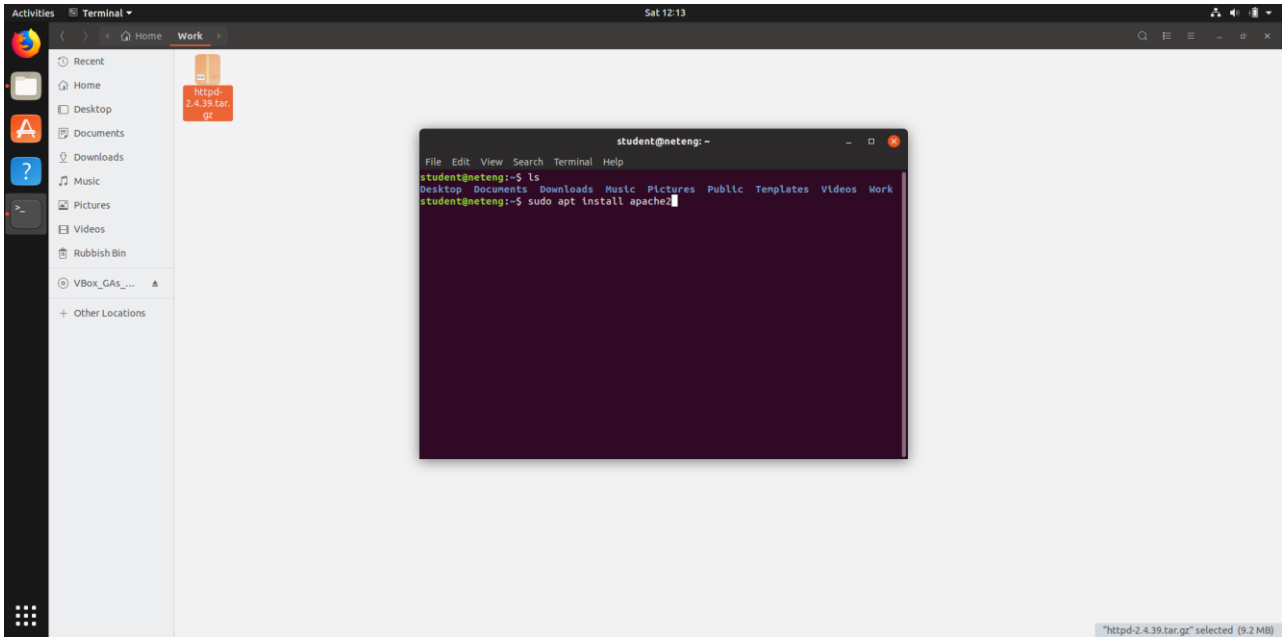


**Figure 1: Installing Apache2 on command terminal**

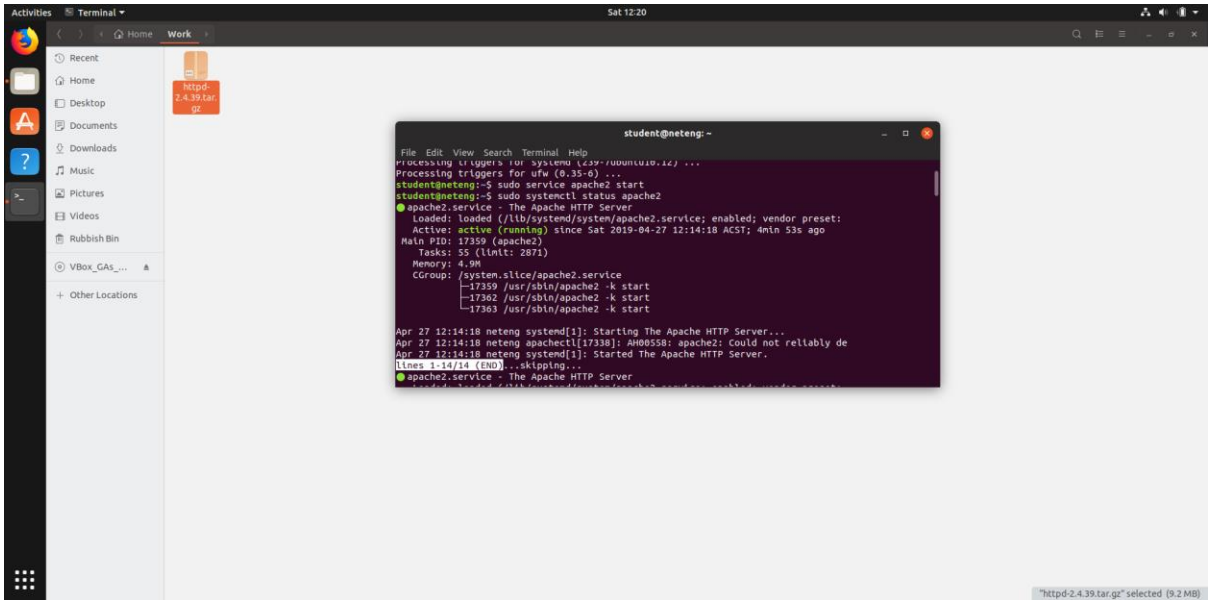


**Figure 2: Check status after starting web server**

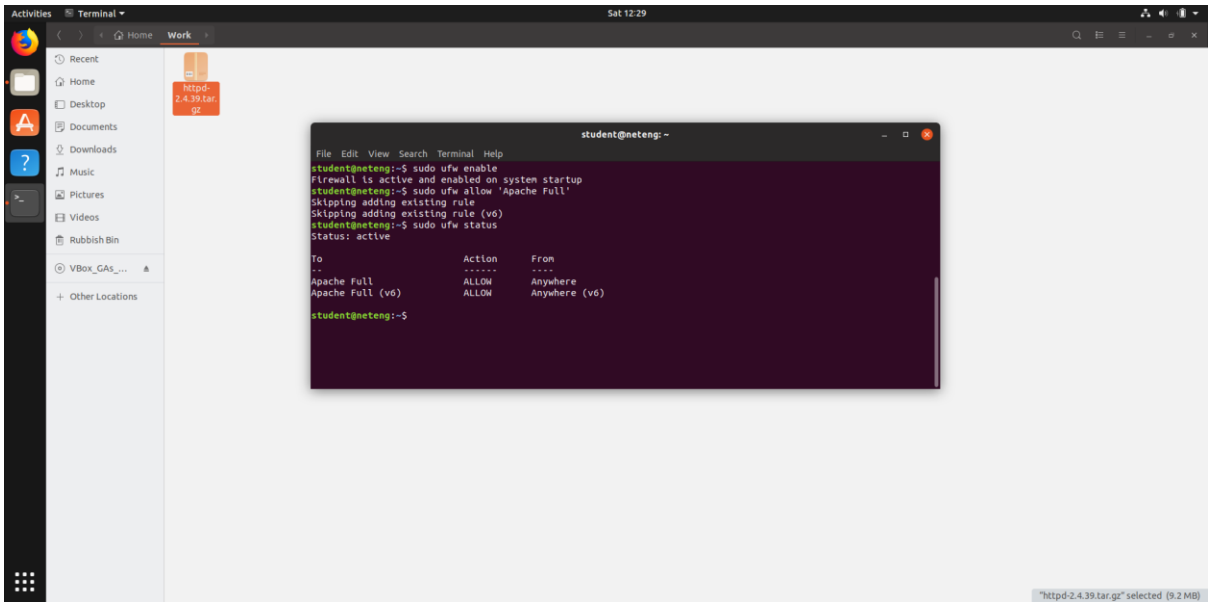


**Figure 3: configure Firewall**

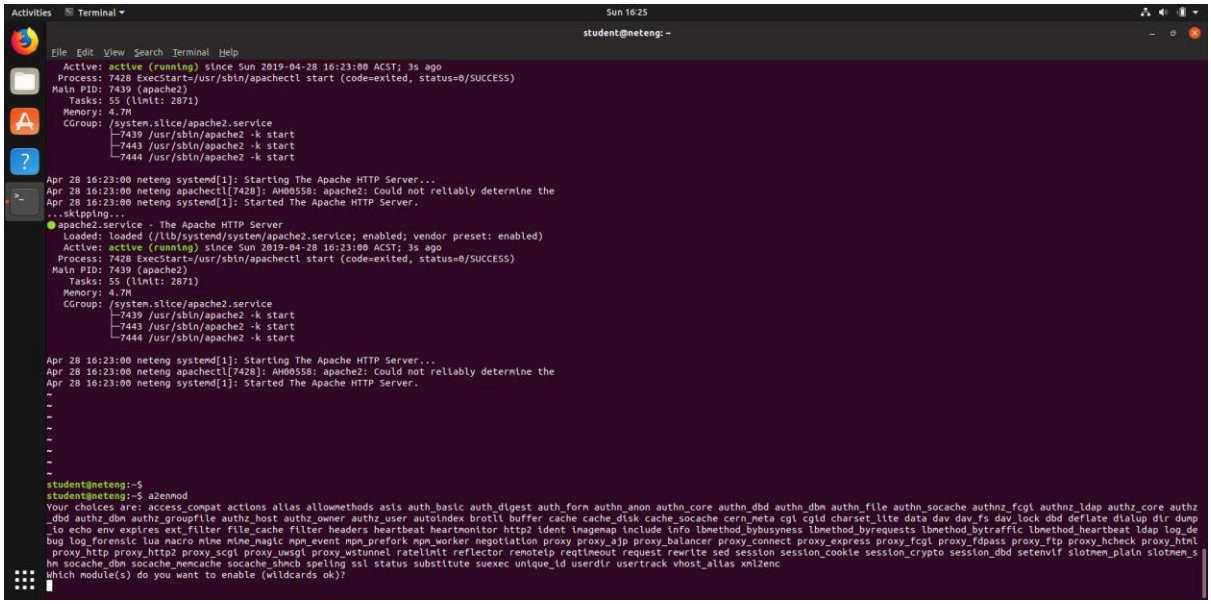


**Figure 4: Listed Modules**

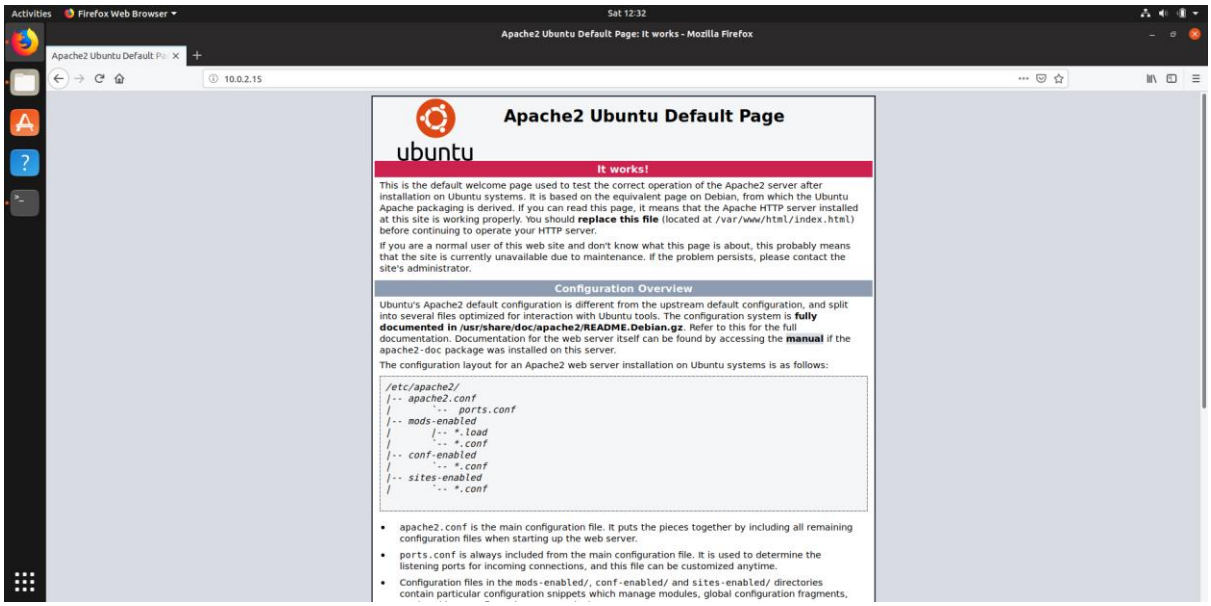


**Figure 6: Running Webserver**

## Bibliography

https://en.wikipedia.org/wiki/Apache_HTTP_Server

https://en.wikipedia.org/wiki/Apache_Tomcat

https://en.wikipedia.org/wiki/Lighttpd

http://www.linuxfromscratch.org/blfs/view/systemd/server/apache.html

https://reposcope.com/package/lighttpd/dependencies

https://httpd.apache.org/

https://w3techs.com/technologies/history_overview/web_server

https://www.whitesourcesoftware.com/most-secure-programming-languages/

## References

En.wikipedia.org. (2019). *Apache HTTP Server*. [online] Available at: https://en.wikipedia.org/wiki/Apache_HTTP_Server [Accessed 20 Apr. 2019].

En.wikipedia.org. (2019). *Apache Tomcat*. [online] Available at: https://en.wikipedia.org/wiki/Apache_Tomcat [Accessed 20 Apr. 2019].

En.wikipedia.org. (2019). *Lighttpd*. [online] Available at: https://en.wikipedia.org/wiki/Lighttpd [Accessed 20 Apr. 2019].

Linuxfromscratch.org. (2019). *Apache-2.4.39*. [online] Available at: http://www.linuxfromscratch.org/blfs/view/systemd/server/apache.html [Accessed 20 Apr. 2019].

reposcope.com. (2019). *lighttpd - Dependencies*. [online] Available at: https://reposcope.com/package/lighttpd/dependencies [Accessed 23 Apr. 2019].

Group, D. (2019). *Welcome! - The Apache HTTP Server Project*. [online] Httpd.apache.org. Available at: https://httpd.apache.org/ [Accessed 23 Apr. 2019].

W3techs.com. (2019). *Historical trends in the usage of web servers, April 2019*. [online] Available at: https://w3techs.com/technologies/history_overview/web_server [Accessed 25 Apr. 2019].

WhiteSource. (2019). *Most Secure Programming Languages*. [online] Available at: https://www.whitesourcesoftware.com/most-secure-programming-languages/ [Accessed 28 Apr. 2019].

Digitalocean.com. (2019). *How To Install the Apache Web Server on Ubuntu 18.04 [Quickstart]* | *DigitalOcean*. [online] Available at: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04-quickstart [Accessed 22 Apr. 2019].