STEP BY STEP

SETTING UP WORDPRESS ON A VPS

UBUNTU 18.04

APACHE UPDATE

BEFORE COMPLETING THIS SECTION OF THE COURSE

Please ensure you have secured your server as follows:

- □ Logged in as the root user and performed the initial steps as the root user
- □ Logged in as a non-root user and completed the initial steps as the non-root user
- □ Setup and are logging in to your server using SSH Key authentication
- □ Configured the Firewall
- □ Configured Fail2Ban

Your server must be secured before you continue with this section of the course.

IMPORTANT NOTICE

Copying and then pasting commands from a pdf can produce errors when pasting the commands into your terminal emulator.

Confirm the accuracy & correctness of the command before pressing enter.

The information contained in this pdf file is designed to be used in conjunction with the video lectures. It is not to be used without referring to the video lectures as certain sections may have been omitted from the pdf file.

INSTALL APACHE:

We need to install apache2 and the apache2-utils packages.

```
sudo apt-get install apache2 apache2-utils
```

INSTALL MARIADB

We need to install the MariaDB package

```
sudo apt install mariadb-server
```

INSTALL PHP7.2

We need to install various php modules that are needed by apache and WordPress.

```
sudo apt-get install php7.2-fpm php7.2-opcache php7.2-gd php7.2-mysql
php7.2-json php7.2-mbstring php7.2-curl php7.2-cli php7.2-xml php7.2-zip
php7.2-soap php7.2-bcmath php7.2 php-imagick php-ssh2 php7.2-common
```

We need to enable and disable a few apache modules:

```
sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php7.2-fpm
sudo a2dismod mpm_prefork
sudo a2enmod mpm_event
sudo service apache2 restart
```

SECURE APACHE

We need to prevent apache from displaying a directory index in the event that no index file is present and we need to enable the headers module - this module provides directives to control and modify HTTP request and response headers. Headers can be merged, replaced or removed.

```
sudo a2dismod -f autoindex
sudo a2enmod headers
```

Now, we need to to open the security,conf file, that is located in the /etc/apache2/conf-available directory:

```
sudo nano /etc/apache2/conf-available/security.conf
```

Make the following changes:

ORIGINAL VALUE	MODIFIED VALUE
ServerTokens OS	ServerTokens Prod
ServerSignature On	ServerSignature Off
#Header set X-Content-Type-Options: "nosniff"	Header set X-Content-Type-Options: "nosniff"
#Header set X-Frame-Options: "sameorigin"	Header set X-Frame-Options: "sameorigin"
ADD TO FILE	Header set X-XSS-Protection: "1; mode=block"
ADD TO FILE	FileETag None
ADD TO FILE	Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure

Close, save and restart apache:

```
sudo systemctl restart apache2
```

We need to edit the dir.conf file, that is located in the /etc/apache2/mods-available directory:

```
sudo nano /etc/apache2/mods-available/dir.conf
```

Remove all values in the DirectoryIndex directive and add only index.php

```
ORIGINAL: DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
MODIFIED: DirectoryIndex index.php
```

We need to restrict the HTTP Methods to get, post and head requests only, we are also going to allow .htaccess files to be used to configure our sites: open the apache2.conf file, located in the /etc/apache2 directory:

```
sudo nano /etc/apache2/apache2.conf
```

Changes:

```
<Directory /usr/share>
        AllowOverride None
        Require all granted
<LimitExcept GET POST HEAD>
       deny from all
</LimitExcept>
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
<LimitExcept GET POST HEAD>
          deny from all
</LimitExcept>
</Directory>
```

We need to enable the apache rewrite module to allow the use of .htaccess files

```
sudo a2enmod rewrite
```

To enable the changes we have made to various apache configuration files, apache needs to be restarted:

```
sudo service apache2 restart
```

SECURE MARIADB

Run the built in secure mysql script to remove the dangerous default values

```
sudo mysql_secure_installation
```

SECURE PHP7.2

Open the php.ini file, located in the /etc/php/7.2/fpm directory:

```
sudo nano /etc/php/7.2/fpm/php.ini
```

Change the following values:

```
allow_url_fopen=Off
cgi.fix_pathinfo=0
```

You need to restart the fpm process after making any changes to the php.ini file.

```
sudo systemctl restart php7.2-fpm
```

The information contained in this pdf file is designed to be used in conjunction with the video lectures. It is not to be used without referring to the video lectures as certain sections may have been omitted from the pdf file.

OPTIMIZE APACHE

We have enabled the events Multi-Processing Module.

You need to open the apache2.conf file, located in the /etc/apache2 directory:

```
sudo nano /etc/apache2/apache2.conf
```

Add the following to the end of the file:

```
# HTTP 2 Enable
Protocols h2 http/1.1
```

```
# mpm events

<IfModule mpm_event_module>
      StartServers                 6
      ServerLimit                  16
      ThreadsPerChild              25
      MaxRequestWorkers            400
      MinSpareThreads              200
      MaxSpareThreads              400
      MaxConnectionsPerChild       10000
</IfModule>
```

Close the apache2.conf file and the enable the http2 module

```
sudo a2enmod http2
```

To prevent the error message: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName, we need to create a config file called servername.conf and add the name localhost to the file.

```
echo "ServerName localhost" | sudo tee /etc/apache2/conf-available/servername.conf
```

Now we need to use the a2enconf command to enable the configuration file

```
sudo a2enconf servername
```

After this restart apache to enable the configuration

```
sudo systemctl restart apache2
```

We are going to optimize apache further after creating our first virtual host

OPTIMIZE MARIADB

Database optimizing is an expansive and in-depth topic. This course covers only the basics of editing your MariaDB configuration. We will look at basic MariaDB tuning later in the course.

OPTIMIZE PHP7.2

Open the php.ini file, located in the /etc/php/7.2/fpm directory:

```
max_input_vars = 3000
memory_limit = 256M
upload_max_filesize = 100M
post_max_size = 100M
```

OPCACHE

```
[opcache]
opcache.enable=1
opcache.memory_consumption=192
opcache.interned_strings_buffer=16
opcache.max_accelerated_files=7963
opcache.validate_timestamps=0
opcache.revalidate_freq=0
```

Restart fpm and apache

```
sudo systemctl restart php7.2-fpm && sudo systemctl restart apache2
```

Calculating opcache.max_accelerated_files value

Controls how many PHP files, at most, can be held in memory at once. It's important that your project has LESS FILES than whatever you set this at. RUN THE COMMAND LISTED HEREUNDER and set value higher than number returned.

```
cd /var/www
find . -type f -print | grep php | wc -l
```

CREATE A VIRTUAL HOST

Change to the directory: /etc/apache2/sites-available/

A listing will display the file 000-default.conf. Make a copy and name the file your_domain_name.conf

```
sudo cp 000-default.conf your_domain_name.com.conf
```

Modify the file to reflect your domain name:

```
<VirtualHost *:80>
     ServerName example.com
     ServerAlias www.example.com
     ServerAdmin webmaster@example.com
     DocumentRoot /var/www/example.com/public_html
     ErrorLog /var/log/apache2/example.com_error.log
     CustomLog /var/log/apache2/example.com_access.log combined
</VirtualHost>
```

Enable your site:

```
sudo a2ensite your_domain_name.com.conf
sudo systemctl restart apache2
```

INSTALLING WPCLI

```
cd
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
chmod +x wp-cli.phar
sudo mv wp-cli.phar /usr/local/bin/wp
wp --info
```

PWGEN - Password Generator

```
sudo apt-get install pwgen
```

CREATING A DATABASE

```
CREATE DATABASE db_name;
CREATE USER 'db_user'@'localhost' identified by 'password';
GRANT ALL PRIVILEGES ON db_name.* to 'db_user'@'localhost';
FLUSH PRIVILEGES;
exit
```

INSTALL WORDPRESS SITE

```
wp core download

wp core config --dbname= --dbuser= --dbpass= --dbprefix=

wp core install --url= --title='' --admin_user= --admin_password= --admin_email=
```

MODIFY WP-CONFIG.PHP

```
/** Allow Direct Updating Without FTP */
define('FS_METHOD', 'direct');
/** Disable Editing of Themes and Plugins Using the Built In Editor */
define('DISALLOW_FILE_EDIT', true);
```

CREATE .HTACCESS FILE

```
cd /var/www/example.com/public_html
touch .htaccess
sudo chown $USER:www-data .htaccess
sudo chmod 664 .htaccess wp-config.php
```

SECURE WP SITE

OWNERSHIP:

cd /var/www/example.com/

```
sudo chown -R $USER:www-data example.com
```

cd /var/www/example.com/public_html/

```
sudo chown -R www-data:www-data wp-content/
```

PERMISSIONS:

```
sudo find /var/www/example.com/public_html/ -type d -exec chmod 755 {} \;

sudo find /var/www/example.com/public_html/ -type f -exec chmod 644 {} \;
```

WP-CONFIG AND HTACCESS

```
cd /var/www/example.com/public_html

sudo chmod 664 wp-config.php .htaccess
```

Change .htaccess and wp-config.php permissions back to 644 after configuring and setting up site.

ENABLE HTTPS USING FREE LET'S ENCRYPT CERTIFICATES

Install Certbot Repository

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
```

Install Certbot

```
sudo apt install python-certbot-apache
```

Install SSL Certificate

```
sudo certbot --apache -d example.com -d www.example.com
```

Current settings result in an A rating with ssllabs.com

We need to edit the ssl configuration to obtain an A+ rating:

```
cd /etc/apache2
sudo mkdir ssl/
```

Generate the DH param file

```
sudo openssl dhparam -out dhparam.pem 2048
```

Configure SSL:

Modify: /etc/apache2/mods-available/ssl.conf

```
sudo nano /etc/apache2/mods-available/ssl.conf
```

DO NOT modify the SSLProtocol in the ssl.conf file, the procedure has changed.

```
SSLProtocol do not modify in the ssl.conf file
```

The SSLProtocol directives follow on the next page.

ADD to the bottom of the ssl.conf file:

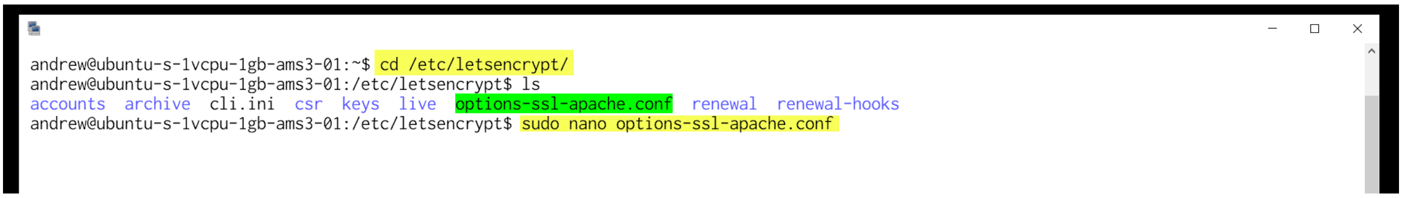
```
#SSL Stapling, only in httpd 2.3.3 and later
SSLUseStapling          on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
SSLStaplingCache        shmcb:/var/run/ocsp(128000)
# DHE (Diffie-Hellman key exchange)
SSLOpenSSLConfCmd Curves secp384r1
SSLOpenSSLConfCmd DHParameters "/etc/apache2/ssl/dhparam.pem"
```

Change to the /etc/letsencrypt/ directory and open the file: options-ssl-apache.conf file

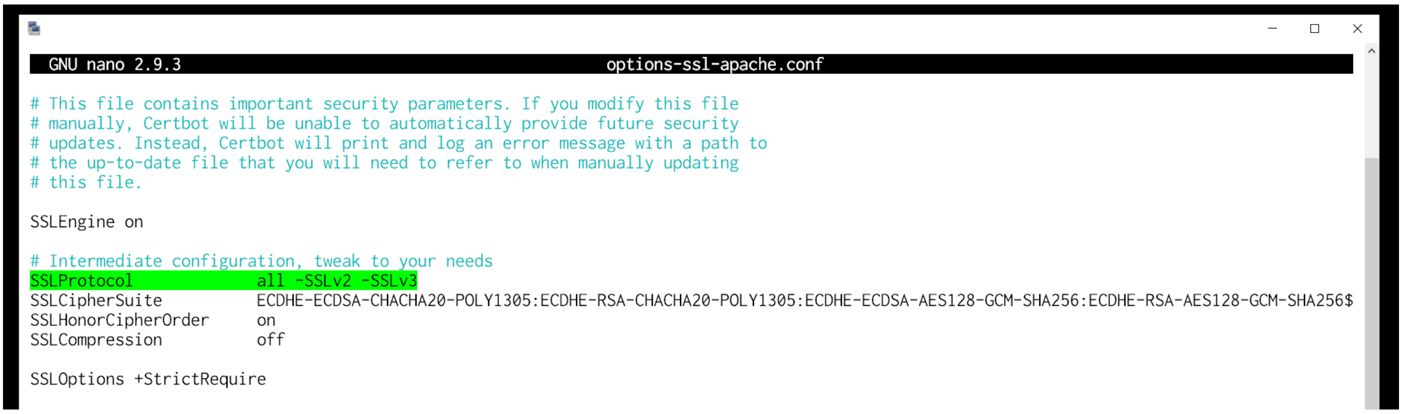
```
cd /etc/letsencrypt/

sudo nano options-ssl-apache.conf
```

Change SSLProtocol to:

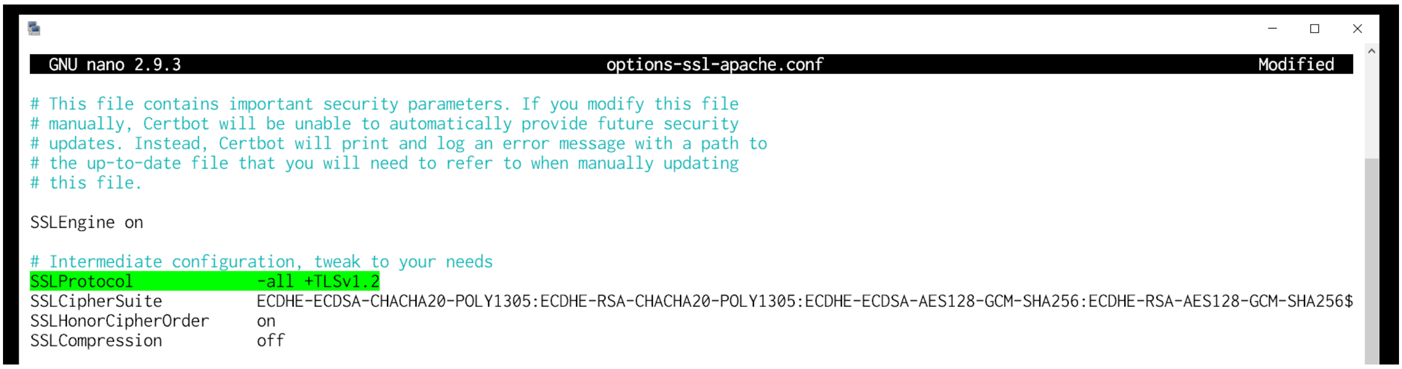
```
SSLProtocol -all +TLSv1.2
```



```
andrew@ubuntu-s-1vcpu-1gb-ams3-01:~$ cd /etc/letsencrypt/
andrew@ubuntu-s-1vcpu-1gb-ams3-01:/etc/letsencrypt$ ls
accounts  archive  cli.ini  csr  keys  live  options-ssl-apache.conf  renewal  renewal-hooks
andrew@ubuntu-s-1vcpu-1gb-ams3-01:/etc/letsencrypt$ sudo nano options-ssl-apache.conf
```



```
  GNU nano 2.9.3                                        options-ssl-apache.conf

# This file contains important security parameters. If you modify this file
# manually, Certbot will be unable to automatically provide future security
# updates. Instead, Certbot will print and log an error message with a path to
# the up-to-date file that you will need to refer to when manually updating
# this file.

SSLEngine on

# Intermediate configuration, tweak to your needs
SSLProtocol             all -SSLv2 -SSLv3
SSLCipherSuite          ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256$
SSLHonorCipherOrder     on
SSLCompression          off

SSLOptions +StrictRequire
```



```
  GNU nano 2.9.3                                        options-ssl-apache.conf                                        Modified

# This file contains important security parameters. If you modify this file
# manually, Certbot will be unable to automatically provide future security
# updates. Instead, Certbot will print and log an error message with a path to
# the up-to-date file that you will need to refer to when manually updating
# this file.

SSLEngine on

# Intermediate configuration, tweak to your needs
SSLProtocol             -all +TLSv1.2
SSLCipherSuite          ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256$
SSLHonorCipherOrder     on
SSLCompression          off
```

Modify your sites LE generated ssl virtual host file

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

Add underneath SSL Certificate paths:

```
Header always set Strict-Transport-Security "max-age=15552000; includeSubDomains;"
```

OPTIMIZE WORDPRESS - W3TC

Before installing w3tc, ensure that your .htaccess and wp-config.php in the public_html directory are writeable by apache. Ownership should be $USER:www-data and the permissions should be 664.