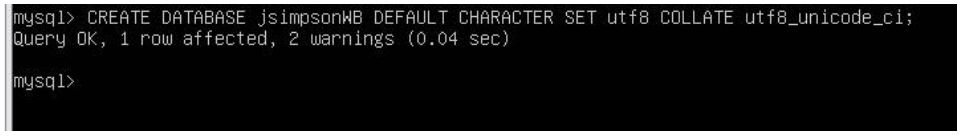
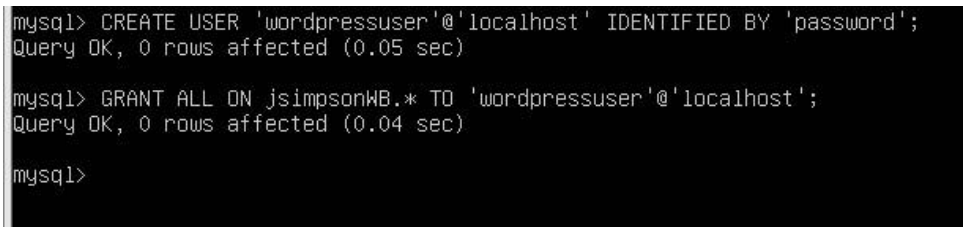
OVERVIEW:

The purpose of this lab was to get hands-on experience installing and securing web servers using Apache and WordPress. I started by setting up a LAMP stack (Linux, Apache, MySQL, PHP) on an Ubuntu server, followed by installing WordPress. I used MySQL to create a database for the website. After setting up WordPress, I focused on securing the site to ensure it communicates securely over HTTPS. This process helped me develop valuable skills for deploying and securing web servers in real-world environments.

ANALYSIS:

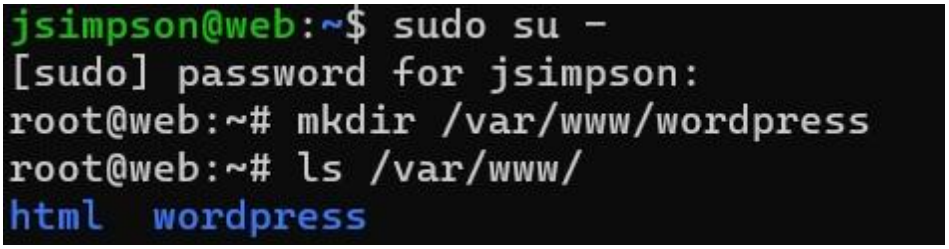
The first step was to install the LAMP stack. This was done by following the lab part 1 video on canvas. During the installation a database was created for the Wordpress site as well.





After the LAMP Stack was installed I installed Wordpress.

After I SSH into the VM I decieded to change to the root user and make a directory for Wordpress



I then needed to enable .htaccess because Wordpress and many of its plugins use these files

```
root@web:~# vim /etc/apache2/sites-available/000-default.conf
```

```
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/wordpress

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
        <Directory /var/www/wordpress/>
                AllowOverride All
        </Directory>
/irtualHost>
```

The next step was enabling the reqrite module so that we can utilize the Wordpress permalink feature.

```
root@web:~# a2enmod rewrite
Enabling module rewrite.
```

The changes that were made then needed to be enabled

```
root@web:~# apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'Serve
rName' directive globally to suppress this message
Syntax OK
```

We can not see that the website has changed

Now that the server software is configured Wordpress can be installed

```
root@web:~# cd /tmp
curl -O https://wordpress.org/latest.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 22.3M  100 22.3M    0     0  1281k      0  0:00:17  0:00:17 --:--:-- 4819k
root@web:/tmp# ls
latest.tar.gz
snap-private-tmp
systemd-private-0c284ffdcee84202b6b5fa210b8176c6-apache2.service-nqUu3a
systemd-private-0c284ffdcee84202b6b5fa210b8176c6-ModemManager.service-WlLjT6
systemd-private-0c284ffdcee84202b6b5fa210b8176c6-systemd-logind.service-xZHidv
systemd-private-0c284ffdcee84202b6b5fa210b8176c6-systemd-resolved.service-VNADoX
systemd-private-0c284ffdcee84202b6b5fa210b8176c6-systemd-timesyncd.service-HAKTFa
vmware-root_675-3980232795
root@web:/tmp#
```

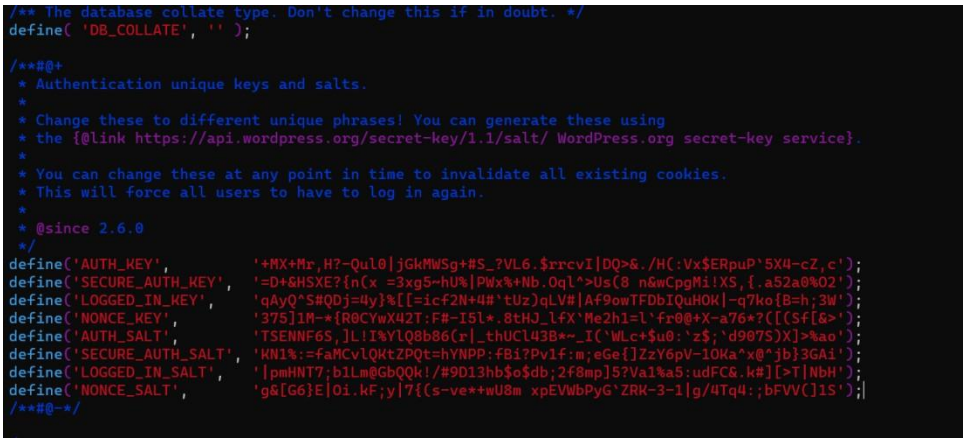
```
root@web:/tmp# tar xzvf latest.tar.gz
wordpress/wp-admin/js/media.js
wordpress/wp-admin/js/editor-expand.min.js
wordpress/wp-admin/js/media-gallery.min.js
wordpress/wp-admin/js/common.min.js
wordpress/wp-admin/js/tags-box.min.js
wordpress/wp-admin/js/svg-painter.min.js
wordpress/wp-admin/js/custom-background.js
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
root@web:/tmp#
```

```
root@web:/tmp# touch /tmp/wordpress/.htaccess
root@web:/tmp# cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
root@web:/tmp# mkdir /tmp/wordpress/wp-content/upgrade
root@web:/tmp# sudo cp -a /tmp/wordpress/. /var/www/wordpress
root@web:/tmp#
```

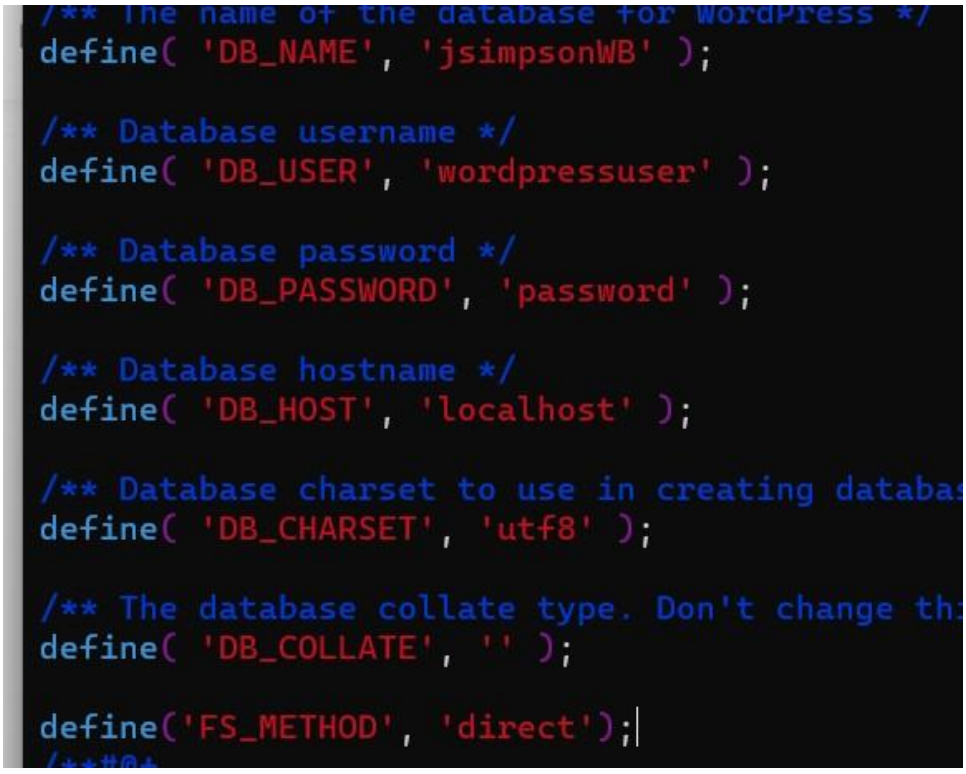
Ownership and permissions then needed to be set up

```
root@web:/tmp# sudo chown -R www-data:www-data /var/www/wordpress
root@web:/tmp# sudo find /var/www/wordpress/ -type d -exec chmod 750 {} \;
sudo find /var/www/wordpress/ -type f -exec chmod 640 {} \;
root@web:/tmp#
```

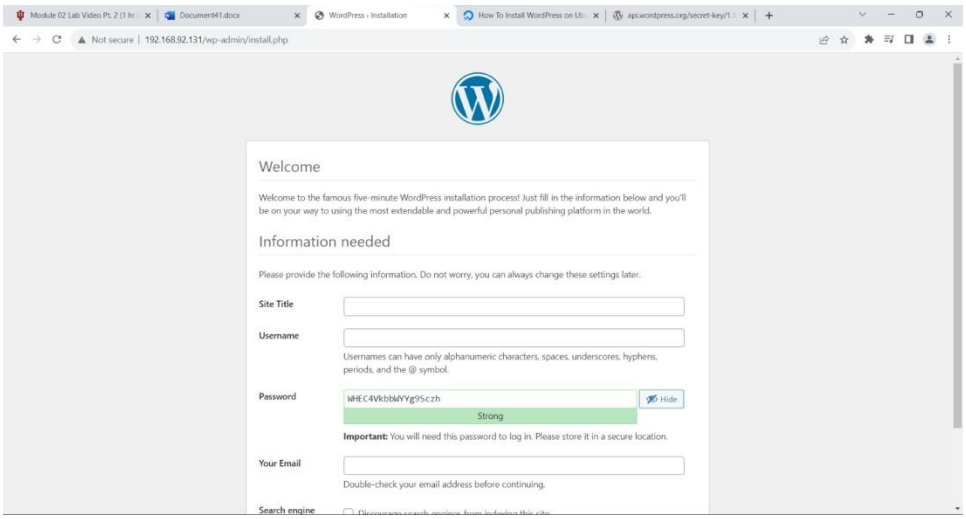
After that we needed to make changes to the main Wordpress config file. The secret keys were needed to provide a level of security. They were obtained from the https://api.wordpress.org/secret-key/1.1/salt/ website.



The next step was to modify some of the database connectoin settings as well



Now that those settings were changed we could complete the install through the web interface. Going to the IP we are greeted with a new window where we can enter info and install wordpress



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title CIT415 Webserver

Username jsimpson

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password password Hide

Very weak

Important: You will need this password to log in. Please store it in a secure location.

Confirm Password ☑ Confirm use of weak password

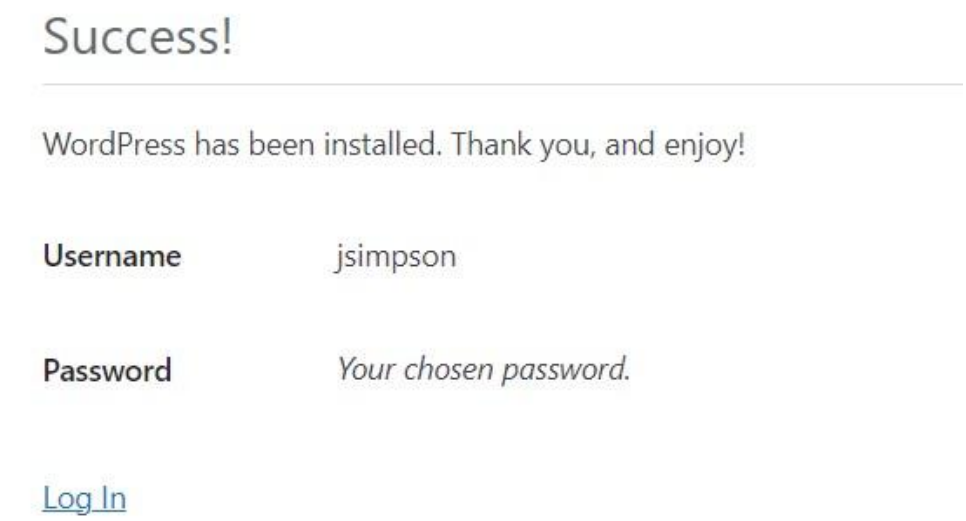
Your Email jsimpson@jsimpson.fail

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress



Now that wordpress was successfully installed we needed to secure the web server with a self-signed SSL certificate.

Using the command line we can create the certificate using this “sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apacheselfsigned.crt”

```
root@web:/var/www/wordpress# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

You then are asked to fill out some information about the certificate

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Indiana
Locality Name (eg, city) []:IUPUI
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CIT
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:web.jsimpson.fail
Email Address []:jsimpson@jsimpson.fail
root@web:/var/www/wordpress#
```

Now that the certificate is available we need to update the apache config to use it.

```
<VirtualHost *:443>
   ServerName web.jsimpson.fail
   DocumentRoot /var/www/web.jsimpson.fail

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.cr
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsign
</VirtualHost>

<VirtualHost *:80>
        ServerName web.jsimpson.fail
        Redirect / https://web.jsimpson.fail/
</VirtualHost>
```

To test that it is working I created an html file with the following text “it worked!”

```
root@web:/var/www/wordpress# sudo mkdir /var/www/web.jsimpson.fail
root@web:/var/www/wordpress# sudo nano /var/www/web.jsimpson.fail/index.html
root@web:/var/www/wordpress#
```

```
<h1>it worked!</h1>
```

Now if we go to the website it is using the certificate and we are greeted with