Lab 3: LAMP & VirtualHosts

michael.ferrie@edinburghcollege.ac.uk

Introduction

The LAMP stack is a commonly used combination of technologies that can be found on web servers. Create a logbook of this with your student number and name, you can upload this to the Moodle when complete, add a screenshot with a description to the logbook where specified.

Part 1: Apache Configuration

Ensure you have apache2 installed and running, and you can see a web page on the servers ip address, or on the localhost address – add this as the first screenshot to the logbook. Commands to remember:

```
apt install apache2 -y
systemctl start apache2
systemctl stop apache2
systemctl status apache2
systemctl restart apache2
ip address show (ip a)
```

Part 2: MariaDB Installation

Become root, install MariaDB and run the sql secure installation:

```
su -
apt install mariadb-server -y
mysql_secure_installation
```

Because you just installed MariaDB and haven't made any configuration changes yet, this password will be blank, so just press ENTER at the prompt. The next prompt asks you whether you'd like to set up a database root password. Type N and then press ENTER. In Debian, the root account for MariaDB is tied closely to automated system maintenance, so we should not change the configured authentication methods for that account. Doing so would make it possible for a package update to break the database system by removing access to the administrative account. From there, you can press Y and then ENTER to accept the defaults for all the subsequent questions.

Part 3: MariaDB - Creating a Simple Database

Enter mariadb as the root user, just press enter after this command:

```
mariadb -u root -p
```

Now enter these commands to create a database:

```
create database shop;

use shop;

CREATE TABLE products_tbl( product_id INT NOT NULL AUTO_INCREMENT,
product_name VARCHAR(100) NOT NULL, PRIMARY KEY ( product_id ) );
show tables;

INSERT INTO products_tbl SET product_name = ‘My First Product’;
select * from products_tbl;
```

Careful when you enter this get all the characters exactly right, if you succeed each time you change a row in the database should say ‘query OK, 1 row affected’ – add a screenshot showing your table and your entry into the new table with it’s auto-generated primary key - Exit MariaDB.

```
exit;
```

Now, any time you want to access the database server:

```
mariadb -u root -p
```

Part 4: Install & Test PHP

Debian will provide the correct php version, install it from apt:

```
apt install php libapache2-mod-php php-mysql -y
```

Edit this config file:

```
vi /etc/apache2/mods-enabled/dir.conf
```

It will look like this:

```
<IfModule mod_dir.c>
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
index.htm </IfModule>
```

Move the PHP index file (highlighted) to the first position after the DirectoryIndex specification, like this, remember to cut in vi? Would be good to use this here.

```
<IfModule mod_dir.c>
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml
index.htm </IfModule>
```

Write quit or SHIFT+ZZ the file then restart Apache, make sure Apache starts again, if it doesn't start you've broken it, once the server restarts make the info file.

```
apachectl configtest
systemctl restart apache2
systemctl status apache2
```

Now verify php has worked by creating the php info file:

```
vi /var/www/html/info.php
```

```
<?php
phpinfo();
?>
```

Open a browser and go to http://{ your_servers _ip }/info.php

Add a screenshot of the server info page displaying. Once you see the php info page, you have successfully installed the LAMP stack, well done.

Part 5: Virtual Hosts

We are going to run a second website on a different port number. We will create a config file for the website. First make a new directory for the site as root, calling this site2, then change ownership from root to your ec username, then reset the permissions recursively on all files in the wwwroot. Then make a new site with a heading that says Site 2 in the new site2 directory:

```
mkdir -p /var/www/html/site2
chown -R ec:ec /var/www/html/site2/
chmod -R 755 /var/www/
cd /var/www/html/site2/
```

```
vi index.html
```

```
<!DOCTYPE html>
<head>
<title>Site 2</title>
</head>
<body bgcolor="yellow">
<h1>Welcome to Site 2</h1>
```

```
</body>
</html>
```

Now we need a configuration file so that Apache can host the site. Change directory to:

```
/etc/apache2/sites-available
```

Make a copy of the default config file so we can use that for our new site:

```
cp 000-default.conf site2.conf
```

Now edit the new config file, you will notice there are a lot of comments in the file, use dd in vim's normal mode to remove them and then edit the file so it looks exactly like this:

```
<VirtualHost *:8000>
ServerAdmin bob@site2.com
DocumentRoot
/var/www/html/site2
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Add a screenshot of your VirtualHost file when you have completed this.

We need to tell the server to listen on port 8000 as well as 80, then enable the new site:

`vi /etc/apache2/ports.conf` **← add Listen 8000 under Listen 80**

```
a2ensite site2.conf
systemctl restart apache2
```

Now if you go to your ip you should have a website and if you go to your ip:8000 you should have a second website. Add a screenshot of both sites to the logbook.