**Lab 4**

Oakland University / CSI3660

Fall 2018

**Due October 29th@ 11:59pm**

**50 points**

This lab is intended to get you setup with a working LAMP stack. What is LAMP?

This is LAMP:

L: **L**INUX

A: **A**PACHE

M: **M**ySQL/**M**ariaDB

P: **P**HP

This is also LAMP, but that’s another discussion entirely:

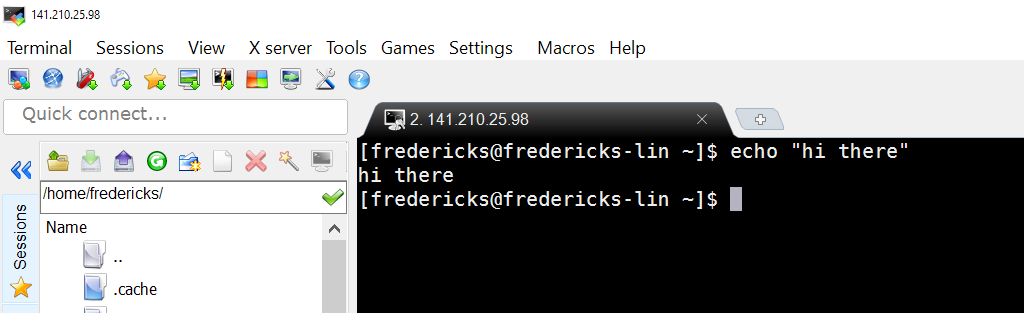


Right, moving on…

Configuration and Setup

Open up a terminal connection. By this point, you should **not** be logging into vcenter through the browser anymore. Use PuTTY, MobaXterm, Cygwin, OSX/Linux terminal, whichever you prefer. But you should be seeing something like this from here on out:

If you still need help setting this up, here are a few overviews:



PuTTY: <https://mediatemple.net/community/products/dv/204404604/using-ssh-in-putty->

MobaXterm: <http://blog.jez.io/2014/09/28/setting-up-mobaxterm-for-ssh-on-windows/>

OSX Terminal: <https://mediatemple.net/community/products/dv/204405144/using-ssh-in-terminal.app-(mac-os-x)>

Remember, to SSH you type in: ssh <yourusername>@<your VM ip address>

Or setup a saved session, depending on your software.

That’s it. Onto the actual work!

**LAB PROCEDURE**

We are going to do threetasks. The first twoyou can do in class. The third, including the lab writeup,is your next homework assignment.

**Step 0) Install Linux**

Already done, nice work!

**Step 1) Install and configure a basic LAMP server**

**Step 1.1 – Apache)**

First thing we’re going to do is install Apache. This is a fairly standard service for serving the HTTP protocol…effectively hosting websites on a server.

First, check to see if the httpd service is running on your machine:

**$ systemctl list-unit-files | grep 'httpd'**

It should say ‘off’ for all runlevels. That means we need to get it up and running.

We are going to install Apache through the yum package manager. Run the following command to run any necessary updates:

**$ sudo yum update**

This will install updates to all installed software on your machine. Security updates too. If it asks you if you want to update at various points, just type ‘y’ and hit enter. We’ll let it update the whole system for now. This might take a little while…your system hasn’t been updated in a while.

Install Apache, in case if it wasn't already pre-installed.

**$ sudo yum install httpd**

To start Apache, all we need to do is the following:

**$ sudo systemctl start httpd**

And let’s make sure that the service starts in all necessary runlevels:

**$ sudo systemctl enable httpd**

Finally, let’s make sure that the service is running:

**$ sudo systemctl status httpd**

That should do it for Apache.

Next we need to ensure that we open the appropriate port (port 80) to allow traffic to/from your web server. Keep in mind that Oakland’s firewall only allows traffic on port 80, so if you wish to run something on your server not on port 80 then you’ll need to either be on campus or on the VPN.

Open the port with the **firewall-cmd** command (we’ll get into firewalls a bit later, but for now just assume that this command opens up your firewall to internet traffic):

**$ sudo firewall-cmd --permanent --add-service=http**

**$ sudo systemctl restart firewalld**

This effectively allows traffic on port 80 to your Ethernet card.

You should now be able to visit your web server through a browser. Load up your favorite browser and type in your IP address into the URL bar. For reference, if you visit my personal VM webpage (<http://141.210.25.14/>) you’ll see the Scientific Linux test page. You should see the same if you visit your IP address in a browser now.**Take a screenshot of your browser page with the default Apache splash page for your report.**

**Step 1.2 – MySQL)**

MySQL and MariaDB are two common database engines that are commonly used. MariaDB tends to be recommended these days, as it was a fork of MySQL after Oracle purchases MySQL.

For MariaDB:

**$ sudo yum install mariadb-server mariadb**

Start the service:

**$ sudo systemctl start mariadb**

And make sure it starts on boot:

**$ sudo systemctl enable mariadb**

And finally set your database administrative passwords, following the prompts:

**$ mysql\_secure\_installation**

It will first ask you about the root password – you haven’t set one for MySQL yet so leave it blank (hit enter). Next, it will ask you to set one..give it a password of your choice. Easiest would be to use your system’s root password, though that somewhat goes against good practice. But, for a VM its fine to use the same password.

Hit enter to skip through the remaining steps (yes will be selected by default).

**Step 1.3 – PHP)**

Lastly, we’ll install the PHP scripting language to allow server-side processing. The following command will setup PHP and the MySQL bindings necessary.

**$ sudo yum install php php-mysql php-gd php-pear**

Restart the Apache web server so that PHP can be properly loaded:

**$ sudo systemctl restart httpd**

**Step 2) Create a basic website for your project**

**Step 2.1 – Permissions)**

Before we get started with creating a website to host project artifacts, we’ll need to set appropriate permissions. By default, the root account owns the web directory (/var/www/html).

Typically we don’t want to open this up wide, however to make our lives easier for this course let’s give the CSI3660 group ownership of that folder.

**$ sudo groupadd CSI3660**

**$ sudo chgrp –R CSI3660 /var/www/html**

**$ sudo chmod -R 775 /var/www/html**

This will give your project group access to create web pages. It should also enable you to create pages without having to sudo each and every time.

*Keep in mind that you need to be a part of the CSI3660 group as well. If you aren’t already:*

***$ sudo usermod –a –G CSI3660<your username>***

Note you'll need to log out and log back in for the new group membership to take effect.

**Step 2.2 – Starting Webpage)**

First, let’s check that PHP is running. Create a file, named index.php, in /var/www/html

(So, there should exist: /var/www/html/test.php)

In this file, put the following code:

**<?php**

**phpinfo();**

**?>**

Now, visit the following browser page: http://<your\_ip\_address>/test.php and you should see a table of PHP attributes that are loaded on your system. **Take a screenshot of what is visible for your lab report (you don’t need to show me the *entire list,* just enough to show that PHP is running).**

*You’re on the right path if this is at the top of the page:*



**Step 2.3 – A Real Webpage)**

Now, let’s create a real webpage. Create a new file in /var/www/html, called “index.php.” This file is what is loaded by default when you visit your IP address (index.html will also work if you don’t plan to use PHP in your webpage, but for this lab we’re using PHP).

A simple structure might look like this:

**<html>**

**<head>**

**<title>CSI3660 Project Webpage</title>**

**</head>**

**<body>**

**<h1>Welcome to my CSI3660 website</h1>**

**<p>Description of my project</p>**

**<?php**

**for ($i = 0; $i < 5; $i++) {**

**echo "<img src='http://i.imgur.com/ITOi4Jn.gif' /><br />";**

**}**

**?>**

**</body>**

**</html>**

**---**

**We'll leave this site as is for now.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**HOMEWORK REQUIREMENTS**

Do the same for your Google Cloud Microinstance. That server should always be online (as it is a microinstance, it should be free and never impact your overall credits). That server will host your project webpage.

The Google server should have the following page:

1. An index page (landing page) that describes your overall project, including:
   1. A list of all the members of your team
   2. A description of your project
      1. Make sure the overview is appropriate for a grade school audience

Eventually, this will be updated with instructions for accessing/running your service.

**REPORT REQUIREMENTS**

**50 points total**

For your submission, you can copy and paste the following into a new document.

* [4 points] Copy and paste a **screenshot** of the default Apache webpage that you took in Step 1.1
* [6 points] Why do you need to **start** and **enable** the **httpdservice?**
* [6 points] Do a Google search (or look in your book) at the **firewall-cmd** command. What is the purpose of this command, and why do you need to run it?
* [3 points] Why do we need to run the **mysql\_secure\_installation** command after installing MySQL?
* [6 points] What are **two** differences between MariaDB and MySQL? Provide your source.
* [5 points] Place your screenshot of the PHP settings page here:
* [20 points] Provide the **external** IP address of your Google website and a screenshot of it.

**Extra Credit:**

(10 points) Setup phpMyAdmin and demonstrate it working.