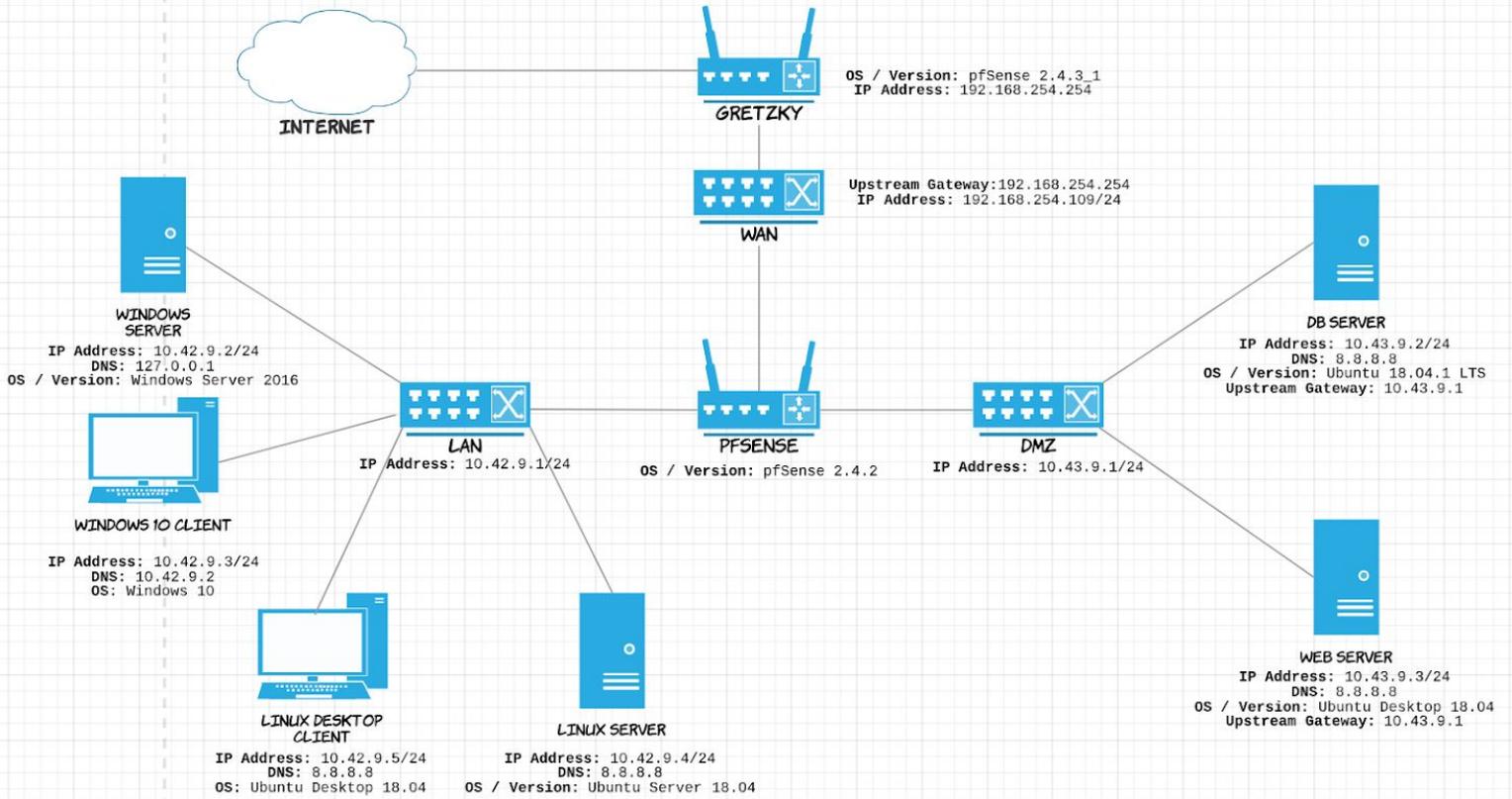


Daniel Goings
dccoings@buffalo.edu
UBNetDef - LAMP Stack
October 7 2018

UBNETDEF NETWORK TOPOLOGY

dcgoings | October 8, 2018



Introduction

The purpose of this report is to get familiar with the **LAMP Stack**, in which we will be using a **Linux** operating system (Ubuntu), the **Apache** server, **MariaDB** database system, and the **PHP** programming language. Our end goal is to set up and manage a WordPress using this stack. Finally, we will have a better understanding of the current network topology, which is reflected on the previous page.

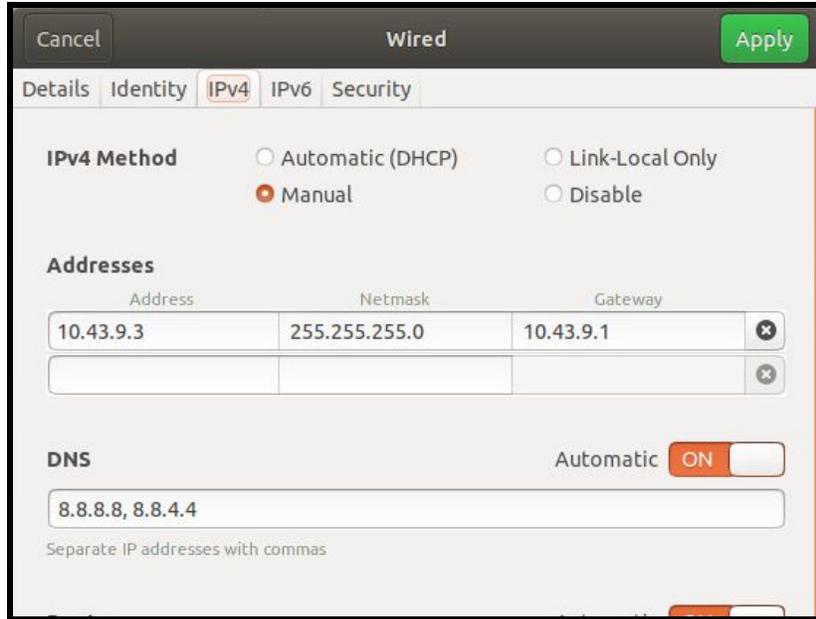
Prerequisites

In order to follow along with this report, it is expected that you already have the a VM Client installed and are logged in. Additionally, you should already have your machines available, one dedicated as a database server and another as a web server. For the purposes of this report, we will be using Ubuntu Desktop 18.04 for the web server, and it is assumed that your database server will already be configured and assigned a static IP.

Step 1: Assigning Static IP to Web Server

For the purposes of this walkthrough, I installed Ubuntu Desktop 18.04 as the operating system of choice for the web server. Keep in mind, Ubuntu Server likely would have been a better choice considering its purpose as a web server.

As previously shown in my Linux report, we must initially connect our server with a static IP (**Network Settings > Settings Cog > IP4 tab**). Since we are connecting this on the DMZ (**10.43.9.1/24**) and **10.43.9.2/24** is being used for the database server, we will assign **10.43.9.3/24** to our web server. Additionally, we will use the DMZ as our upstream gateway and **8.8.8.8** and **8.8.4.4** for DNS.



After applying these settings, restart the network (turn-off / turn-on from wired settings window), and open the terminal. To check for connectivity, we're going to ping the DMZ, ping Gretzky, and make sure DNS is working by running nslookup, all shown below with expected results.

```
dan@dan-web-server: ~
File Edit View Search Terminal Help
dan@dan-web-server:~$ ping 10.43.9.1
PING 10.43.9.1 (10.43.9.1) 56(84) bytes of data.
64 bytes from 10.43.9.1: icmp_seq=1 ttl=64 time=0.231 ms
64 bytes from 10.43.9.1: icmp_seq=2 ttl=64 time=0.233 ms
64 bytes from 10.43.9.1: icmp_seq=3 ttl=64 time=0.249 ms
^C
--- 10.43.9.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.231/0.237/0.249/0.019 ms
dan@dan-web-server:~$ ping 192.168.254.254
PING 192.168.254.254 (192.168.254.254) 56(84) bytes of data.
64 bytes from 192.168.254.254: icmp_seq=1 ttl=63 time=0.346 ms
64 bytes from 192.168.254.254: icmp_seq=2 ttl=63 time=0.353 ms
64 bytes from 192.168.254.254: icmp_seq=3 ttl=63 time=0.298 ms
^C
--- 192.168.254.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.298/0.332/0.353/0.028 ms
dan@dan-web-server:~$ nslookup ubnetdef.org
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   ubnetdef.org
Address: 128.205.44.157

dan@dan-web-server:~$
```

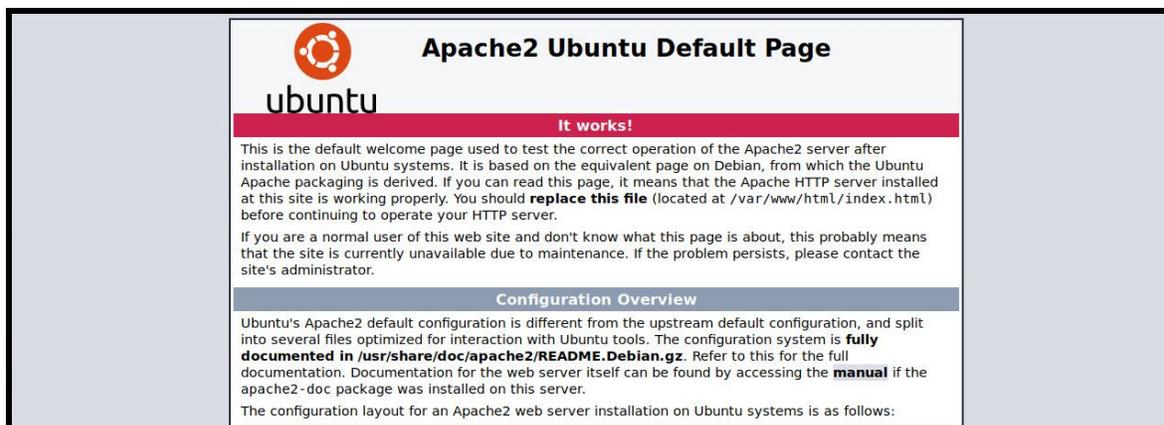
Step 2: Installing Apache

To install Apache, we're going to open up the terminal. The first commands to run are:

```
$ sudo apt update
$ sudo apt install apache2
```

Continue through the prompts and progress bar. If you run into issues running these commands, ensure VMWare Tools are installed, you're connected to the internet and you are running sudo with the correct credentials. The default settings for UFW are also the recommended settings, so it is not necessary to edit anything within `$ sudo ufw app list`

We can confirm that Apache was installed by opening the browser in another client and entering the IP address we assigned for the web server earlier. If installation of Apache was successful, this is what you should see when if you enter `10.43.9.3`



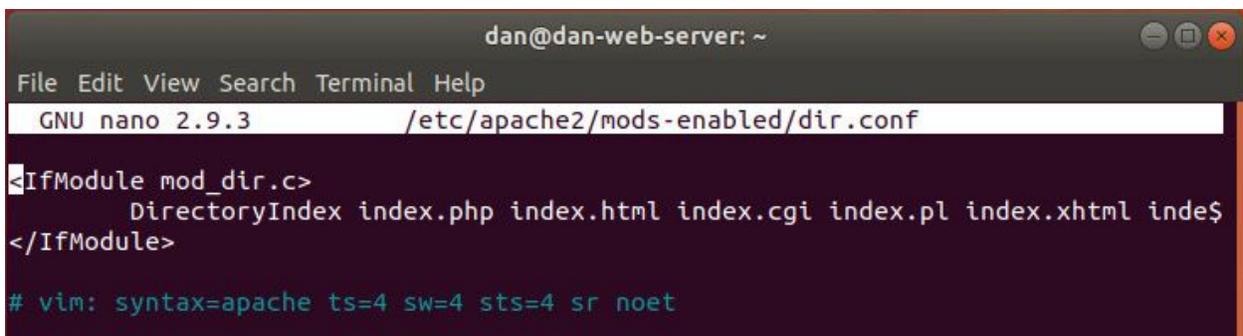
Step 3: Installing PHP

Since we already set up our database server with MariaDB as mentioned in the prerequisites, we can jump right into installing PHP. Now since we are using Ubuntu Desktop 18.04 for our web server, we did not run into any issues with the given commands. However, if you chose a different operating system for your web server, this section may require some outside assistance to work.

However, in our case, we already ran `$ sudo apt update` in a previous step so all that is left is the install command for PHP:

```
$ sudo apt install php libapache2-mod-php php-mysql
```

After following through the prompts and watching the progress bar, the next step is telling the web server to prefer PHP files over other file types. This is simply done by editing the `dir.conf` file inside `/etc/apache2/mods-enabled/`. Ensure that `index.php` is immediately after `DirectoryIndex`, as shown below.

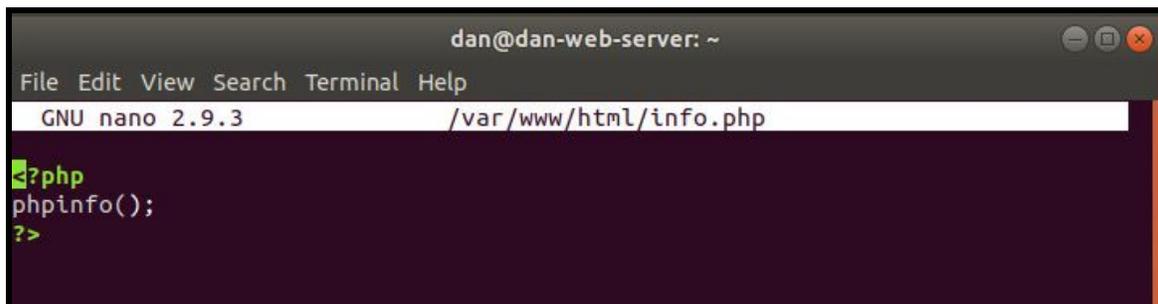


```
dan@dan-web-server: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
  DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index/
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Last thing to do before confirming that we are set up correctly is to restart `apache2`. We simply do this with:

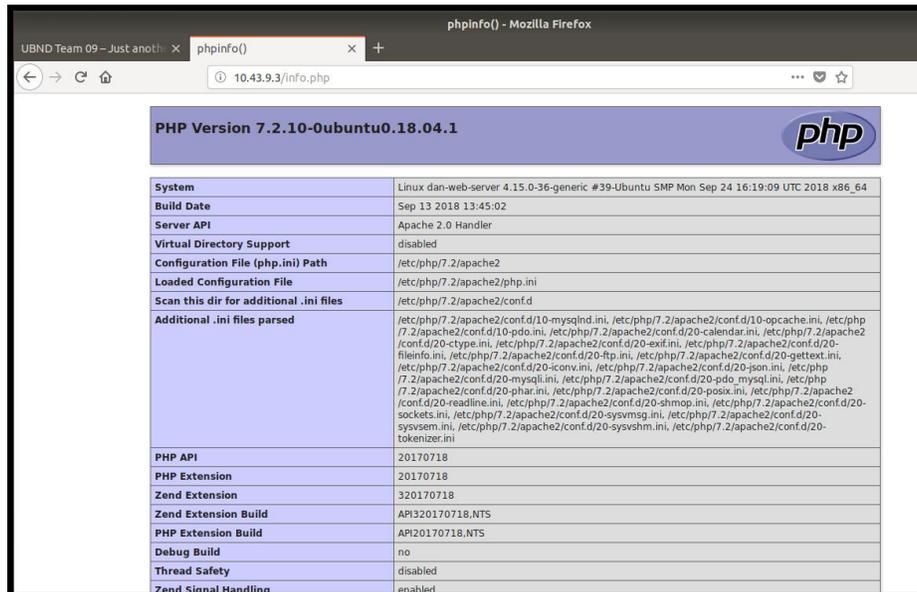
```
$ sudo systemctl restart apache2
```

Now we should be good to go. Just like how we checked if Apache was installed, we can do something similar for PHP. Let's create and edit a file `info.php` inside `/var/www/html` and add the following text inside:



```
dan@dan-web-server: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /var/www/html/info.php
<?php
phpinfo();
?>
```

Our hope is that if everything is set up correctly, we should be able to see the info.php page when we open up this file in our browser from 10.43.9.3, e.g.:



PHP Version 7.2.10-0ubuntu0.18.04.1	
System	Linux dan-web-server 4.15.0-36-generic #39-Ubuntu SMP Mon Sep 24 16:19:09 UTC 2018 x86_64
Build Date	Sep 13 2018 13:45:02
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-ldap.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Success!

Step 4: Installing WordPress

Finally moving on to installing WordPress. Let's first change into the right directory and download the latest WP tar, and untar:

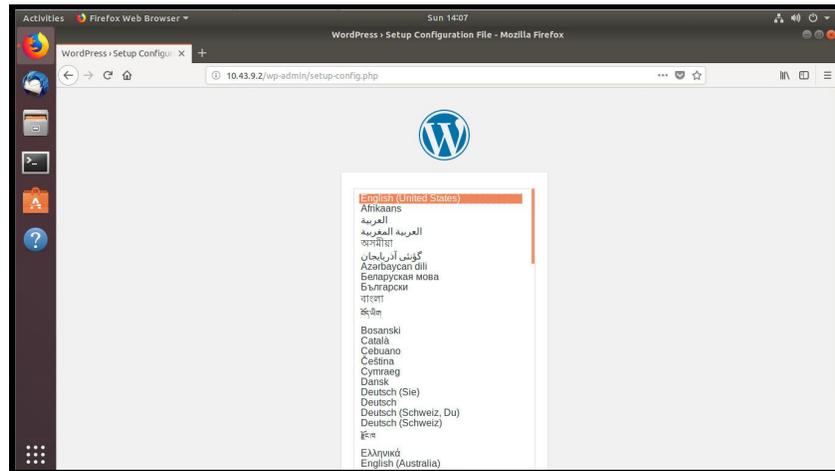
```
$ cd /var/www/html
$ sudo wget http://wordpress.org/latest.tar.gz
$ tar -zxvf latest.tar.gz
```

Now we need to move the files that were just extracted over to the html folder, remove the **index.html** file, and change the ownership:

```
$ mv wordpress/* wordpress/. * .
$ rm index.html
$ sudo chown -R www-data:www-data /var/www/html
```

As we did previously, don't forget to **restart Apache**.

Lastly, we can confirm we are set up correctly by going to **10.43.9.3/wp-admin**. You should be met with the WordPress configuration window.



Select your language.

Click *Let's Go*.

Database Name: wordpress

Username: wordpress

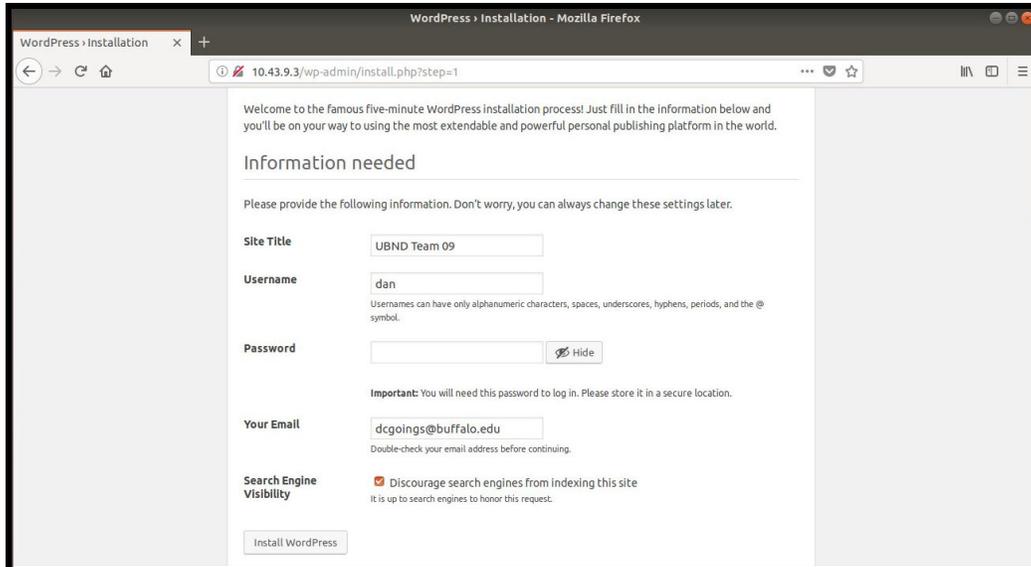
Password: changeme123 (selected earlier in database server setup)

Database Host: 10.43.9.2 (the database server static IP)

Table Prefix: wp_

A screenshot of the WordPress database configuration form. At the top center is the WordPress logo. Below it is a heading: 'Below you should enter your database connection details. If you're not sure about these, contact your host.' The form contains five rows of input fields, each with a label on the left and a description on the right. The fields are: 'Database Name' with 'wordpress', 'Username' with 'wordpress', 'Password' with 'changeme123', 'Database Host' with '10.43.9.2', and 'Table Prefix' with 'wp_'. A 'Submit' button is located at the bottom left of the form area.

After submitting the credentials, we can set up the actual WordPress!



Upon completion, the next time you enter 10.43.9.3 into the browser, we will be greeted with the home page for our WordPress!

