

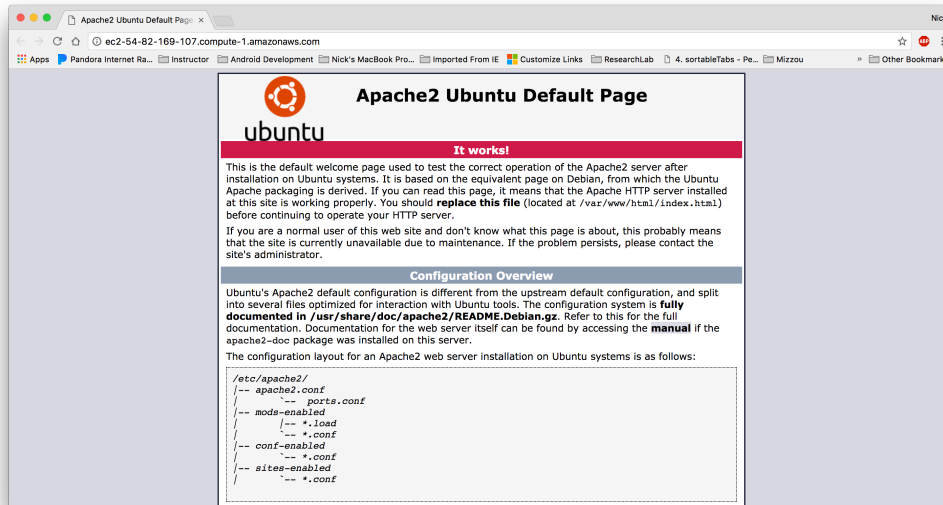
Virtual Hosts in Apache

Created by: Professor Wergeles for the University of Missouri CS4830

After successfully installing Apache, we can navigate to our virtual machine in a web browser.

For example: <http://ec2-54-82-169-107.compute-1.amazonaws.com/>

Doing that will bring up Apache's default webpage:



This page is located within /var/www/html/



`/var/www/html/` is the default web-accessible directory, but we can change that. Let's take a look at Apache's default configuration, located in `/etc/apache2/`

```
nmw6g7@greentree:~$ cd /var/www/html/
nmw6g7@greentree:/var/www/html$ ls
index.html
nmw6g7@greentree:/var/www/html$ cd /etc/apache2/
nmw6g7@greentree:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
nmw6g7@greentree:/etc/apache2$
```

Within Apache's main configuration file, ***apache2.conf***, you'll find the following:

```
# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

This is the default security model for Apache. The first `<Directory>` container states that everything on the file-system is **NOT** web-accessible (`Require all denied`).

That's followed by two exceptions: the second and third <Directory> containers state that "/usr/share" and "/var/www" are both web-accessible (Require all granted).

So, **apache2.conf** allows access to /var/www. To point incoming requests to that location, another configuration file is needed. For that, open, the **000-default.conf** file located in /etc/apache2/sites-available/

```

4830 — nmw6g7@greentree: /etc/apache2/sites-available — ssh -i CS4830.pem nmw6g7@ec2-54-82-169-107.compute-1.ama...
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# 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/html

# 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
</VirtualHost>

vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
~
~
~
~
~
~
~

```

000-default.conf defines a <VirtualHost> container. This container provides contact info (ServerAdmin), tells Apache where to serve content from (DocumentRoot), and specifies a destination for access and error data (ErrorLog/CustomLog).

To customize these options, create your own `.conf` file. I've created one named after my server's domain name, **`ec2-54-82-169-107.compute-1.amazonaws.com`**

```
4830 — nmw6g7@greentree: /etc/apache2/sites-available — ssh -i CS4830.pem nmw6g7@ec2-54-82-1...
VirtualHost *:80>
# Set server name and contact details
ServerName ec2-54-82-169-107.compute-1.amazonaws.com
ServerAdmin wergelesn@missouri.edu

# Point Apache to the following directory...
DocumentRoot /home/nmw6g7/CS4830/public_html/

# Store access and error logs here
ErrorLog /home/nmw6g7/CS4830/log/error.log
CustomLog /home/nmw6g7/CS4830/log/access.log combined

# Grant access to the DocumentRoot
<Directory /home/nmw6g7/CS4830/public_html/>
    Require all granted
</Directory>
</VirtualHost>
~
~
~
```

If you don't want to move your page for Challenge1 then set up like this (Method 2):

```
4830 — nmw6g7@greentree: /etc/apache2/sites-available — ssh -i CS4830.pem ubuntu...
VirtualHost *:80>
# Set server name and contact details
ServerName ec2-54-82-169-107.compute-1.amazonaws.com
ServerAdmin wergelesn@missouri.edu

# Point Apache to the following directory...
#DocumentRoot /home/nmw6g7/CS4830/public_html/
DocumentRoot /var/www/html/

# Store access and error logs here
ErrorLog /var/www/html/log/error.log
CustomLog /var/www/html/log/access.log combined

# Grant access to the DocumentRoot
<Directory /var/www/html/>
    Require all granted
</Directory>
</VirtualHost>
~
```

Things to Note:

- The `ServerName` has been updated to the virtual machine's domain name
- The `DocumentRoot` now points to a folder within my user's home directory
- The `ErrorLog` and `CustomLog` also point to a folder within `/home/"pawprint"/CS4830/`
- The last step is to grant access to the new `DocumentRoot`
 - We saw these rules previously in **`apache2.conf`**
 - The `<Directory>` container here states that all files and folders within `/home/"pawprint"/CS4830/public_html/` will be web-accessible

The next step is to create the directories mentioned by DocumentRoot, ErrorLog, CustomLog, and <Directory>. From /home/"pawprint"/CS4830 do the following:

```
4830 — nmw6g7@greentree: ~ — ssh -i CS4830.pem nmw6g7@ec2-54-82-169-107.compute-1.amazon...
[nmw6g7@greentree:~$ mkdir CS4830
[nmw6g7@greentree:~$ mkdir CS4830/public_html
[nmw6g7@greentree:~$ mkdir CS4830/log
[nmw6g7@greentree:~$ ls
CS4830
[nmw6g7@greentree:~$ ls CS4830/
log public_html
nmw6g7@greentree:~$
```

Or for Method 2:

```
4830 — nmw6g7@greentree: /var/www/html — ssh -i CS4830.pem ubuntu@ec2-54-82-16...
[nmw6g7@greentree:~$ cd /var/www/html/
[nmw6g7@greentree:/var/www/html$ ls
index.html
[nmw6g7@greentree:/var/www/html$ sudo cp index.html oldIndex.html
[nmw6g7@greentree:/var/www/html$ ls
index.html oldIndex.html
[nmw6g7@greentree:/var/www/html$ sudo rm index.html
[nmw6g7@greentree:/var/www/html$ ls
oldIndex.html
[nmw6g7@greentree:/var/www/html$ sudo echo "hello world from /var/www/html" > index.html
-su: index.html: Permission denied
[nmw6g7@greentree:/var/www/html$ ls
oldIndex.html
[nmw6g7@greentree:/var/www/html$ sudo vi index.html
[nmw6g7@greentree:/var/www/html$ ls
index.html oldIndex.html
nmw6g7@greentree:/var/www/html$
```

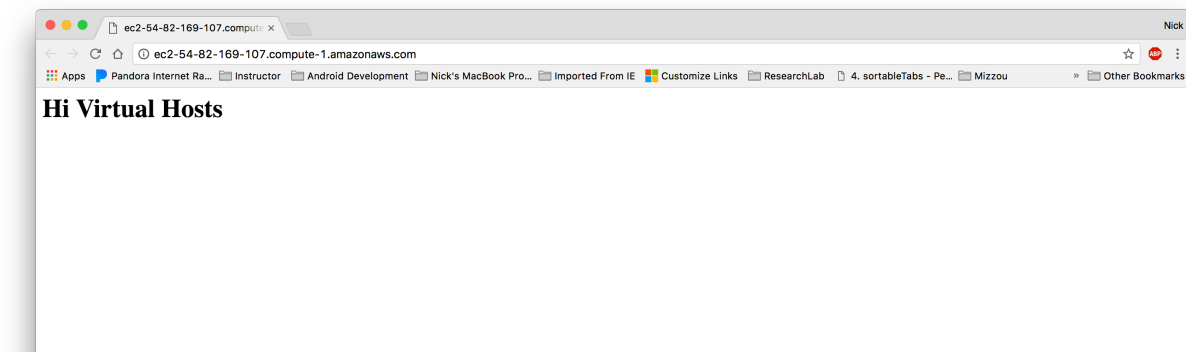
Let's also create a quick index.html page within public_html/ directory:

```
4830 — nmw6g7@greentree: ~ — ssh -i CS4830.pem nmw6g7@ec2-54-82-169-107.compute-1.amazon...
[nmw6g7@greentree:~$ mkdir CS4830
[nmw6g7@greentree:~$ mkdir CS4830/public_html
[nmw6g7@greentree:~$ mkdir CS4830/log
[nmw6g7@greentree:~$ ls
CS4830
[nmw6g7@greentree:~$ ls CS4830/
log public_html
[nmw6g7@greentree:~$ echo "<h1>Hi Virtual Hosts</h1>" > CS4830/public_html/index.html
[nmw6g7@greentree:~$ ls CS4830/public_html/
index.html
nmw6g7@greentree:~$
```

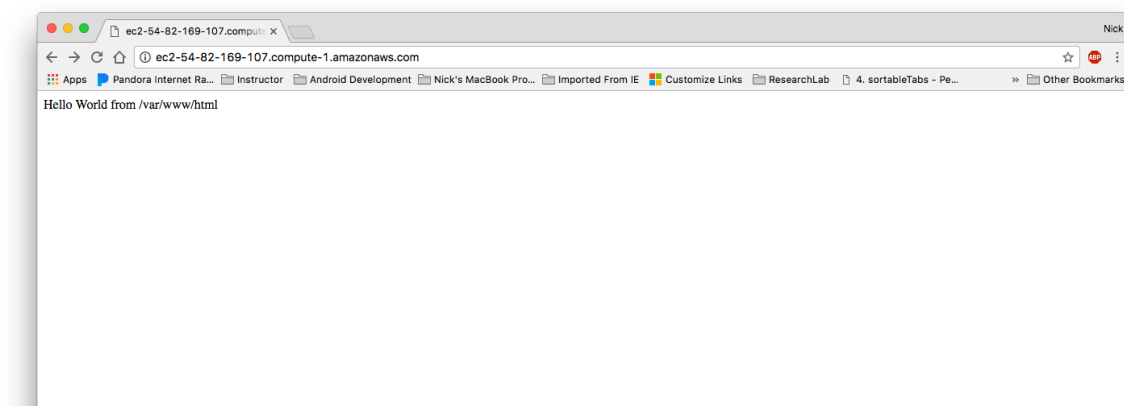
Finally, we need to disable the old virtual host (a2dissite) and enable the new one (a2ensite). After that, Apache should be restarted.

```
nmw6g7@greentree:~$ ls
CS4830
nmw6g7@greentree:~$ cd /etc/apache2/
nmw6g7@greentree:/etc/apache2$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
nmw6g7@greentree:/etc/apache2$ sudo a2ensite ec2-54-82-169-107.compute-1.amazonaws.com.conf
Enabling site ec2-54-82-169-107.compute-1.amazonaws.com.
To activate the new configuration, you need to run:
    service apache2 reload
nmw6g7@greentree:/etc/apache2$ sudo service apache2 reload
nmw6g7@greentree:/etc/apache2$
```

Now we can reach our content in a web browser!



Or Method 2:



What About Multiple Domains?

It is possible to host more than one website from a single server. To do that, we can use name-based virtual hosts. Here is a link to Apache's documentation:

<http://httpd.apache.org/docs/2.4/vhosts/name-based.html>

I have the domain name: <http://professorwergeles.com>

Which I purchased from: <http://namecheap.com>

If I want my domain to point to my Amazon EC2 virtual machine which we made in class, I'll need to start by creating another <VirtualHost> configuration file in /etc/apache2/sites-available/

I'll name this file ***professorwergeles.com.conf***



```
VirtualHost *:80>
# Set server name and contact details
ServerName professorwergeles.com
ServerAdmin wergelesn@missouri.edu

# Point Apache to the following directory...
DocumentRoot /home/nmw6g7/professorwergeles.com/public_html/

# Store access and error logs here
ErrorLog /home/nmw6g7/professorwergeles.com/log/error.log
CustomLog /home/nmw6g7/professorwergeles.com/log/access.log combined

# Grant access to the DocumentRoot
<Directory /home/nmw6g7/professorwergeles.com/public_html/>
    Require all granted
</Directory>
</VirtualHost>
~
~
~
```

1,1 All

This file is similar to ***ec2-54-82-169-107.compute-1.amazonaws.com.conf*** except that I've updated the ServerName and all of my filepaths.

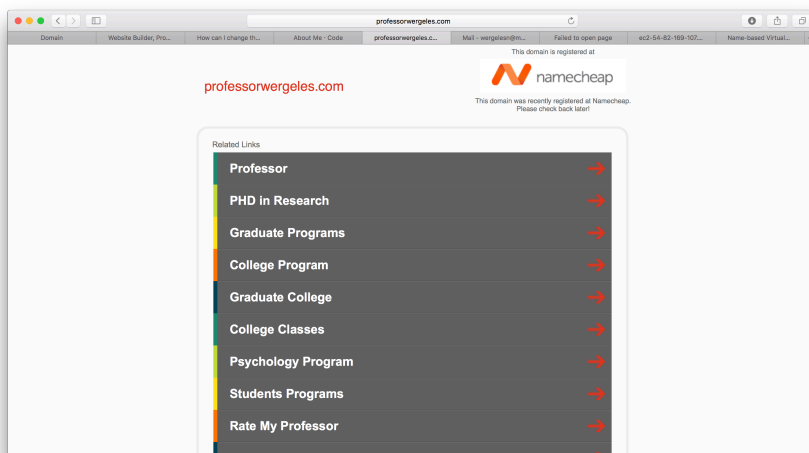
Next we need to create those files and folders:

```
4830 — nmw6g7@greentree: ~/professorwergeles.com/public_html — ssh -i CS4830.pem nmw6g7@ec2-54-82-169-107.compu...
nmw6g7@greentree: /etc/apache2/sites-available$ cd
nmw6g7@greentree: ~$ mkdir professorwergeles.com
nmw6g7@greentree: ~$ mkdir professorwergeles.com/public_html
nmw6g7@greentree: ~$ mkdir professorwergeles.com/log
nmw6g7@greentree: ~$ ls
CS4830 professorwergeles.com
nmw6g7@greentree: ~$ cd professorwergeles.com/
nmw6g7@greentree: ~/professorwergeles.com$ ls
log public_html
nmw6g7@greentree: ~/professorwergeles.com$ cd public_html/
nmw6g7@greentree: ~/professorwergeles.com/public_html$ ls
nmw6g7@greentree: ~/professorwergeles.com/public_html$ echo "Hello from professorwergeles.com" > index.html
nmw6g7@greentree: ~/professorwergeles.com/public_html$ ls
index.html
nmw6g7@greentree: ~/professorwergeles.com/public_html$
```

We can then enable and reload:

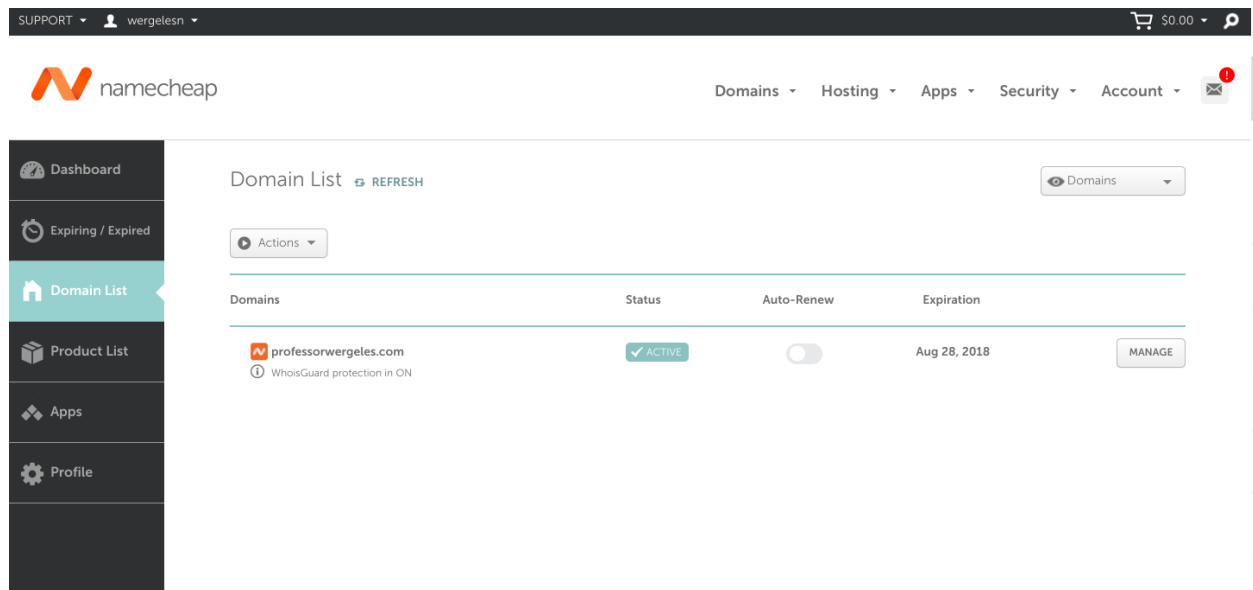
```
4830 — nmw6g7@greentree: /etc/apache2/sites-available — ssh -i CS4830.pem nmw6g7@ec2-54-82-169-107.compute-1.ama...
nmw6g7@greentree: ~/professorwergeles.com/public_html$ cd /etc/apache2/
nmw6g7@greentree: /etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
nmw6g7@greentree: /etc/apache2$ cd sites-available/
nmw6g7@greentree: /etc/apache2/sites-available$ ls
000-default.conf  ec2-54-82-169-107.compute-1.amazonaws.com.conf
default-ssl.conf  professorwergeles.com.conf
nmw6g7@greentree: /etc/apache2/sites-available$ sudo a2ensite professorwergeles.com.conf
Enabling site professorwergeles.com.
To activate the new configuration, you need to run:
  service apache2 reload
nmw6g7@greentree: /etc/apache2/sites-available$ sudo service apache2 reload
nmw6g7@greentree: /etc/apache2/sites-available$ sudo service apache2 reload
nmw6g7@greentree: /etc/apache2/sites-available$
```

If you go to professorwergeles.com then you will see this

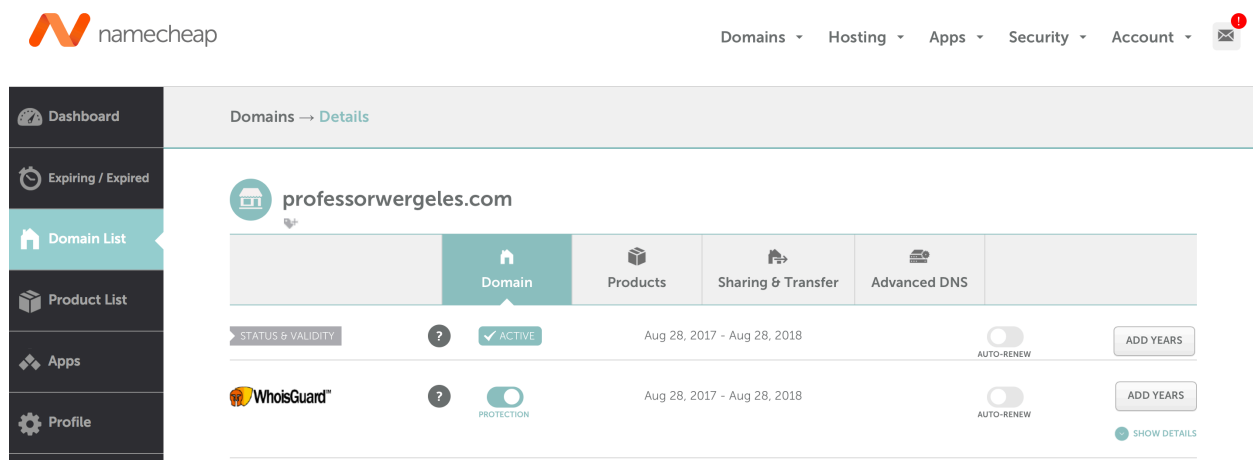


That is because we have not completed the final step which is to map our domain name to our Amazon virtual machine. This process will vary depending on where you've purchased a domain name from.

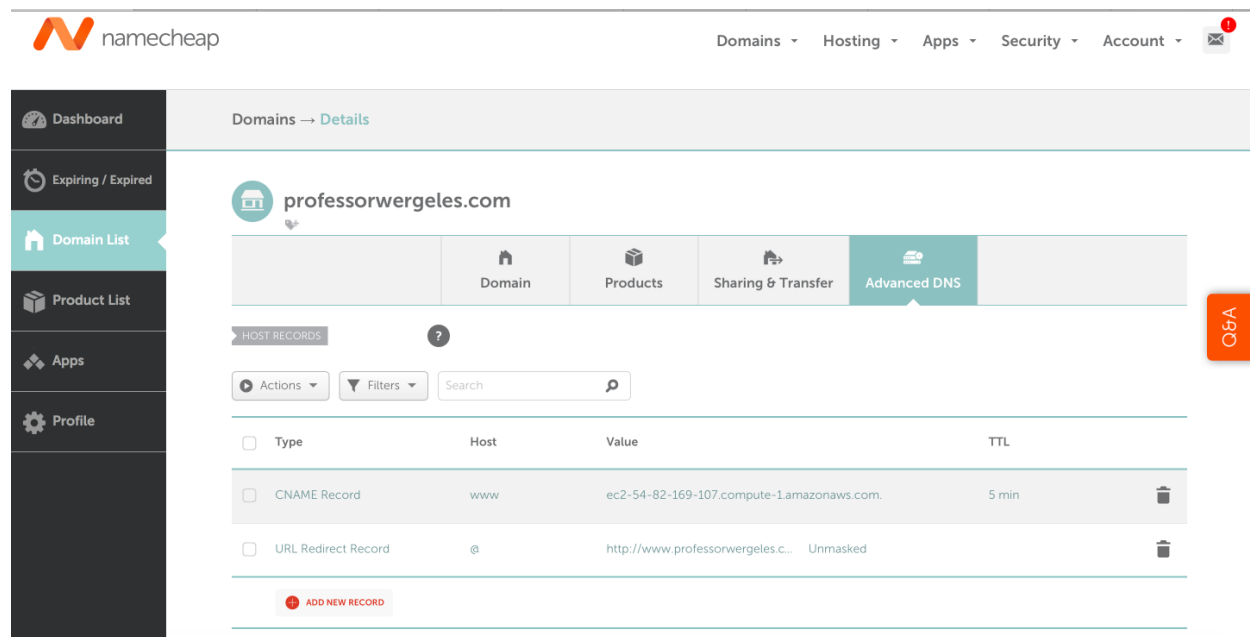
On <http://namecheap.com>, after you login, you go to your dashboard. From the dashboard, you will click on the "Domain List" tab on the left-hand side of the page.



Then click on "Manage" button next to the domain you would like to setup.



Now you can select the “Advanced DNS” tab near the top.



Under the advanced settings, there is a control panel where you can setup a CNAME:

<input type="checkbox"/>	Type	Host	Value	TTL	
<input type="checkbox"/>	CNAME Record	www	ec2-54-82-169-107.compute-1.amazonaws.com.	5 min	
<input type="checkbox"/>	URL Redirect Record	@	http://www.professorwergeles.c...	Unmasked	

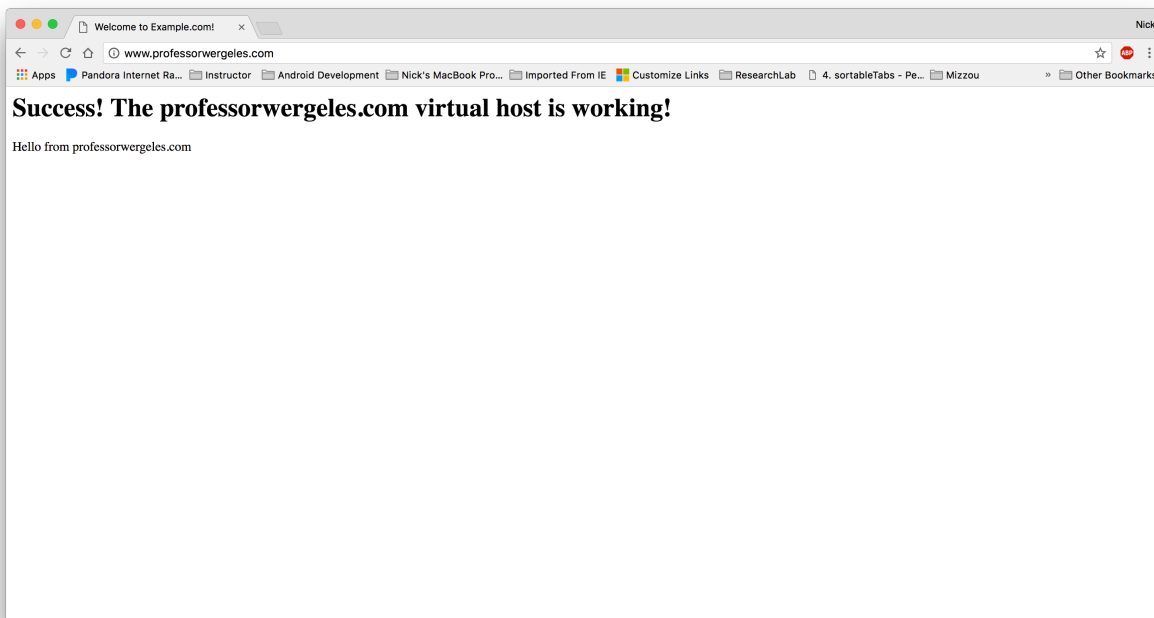
CNAME stands for Canonical Name. It matches a domain to another domain. In our case, it's mapping <http://professorwergeles.com> to <http://ec2-54-82-169-107.compute-1.amazonaws.com>

For more information on CNAME and DNS, take a look at the following guide:
<https://www.linode.com/docs/networking/dns/introduction-to-dns-records>

Note:

- It takes time for these changes to go into effect
- Just how long depends on a value called Time-to-Live (TTL).

Now, we should be able to successfully access professorwergeles.com:



Note:

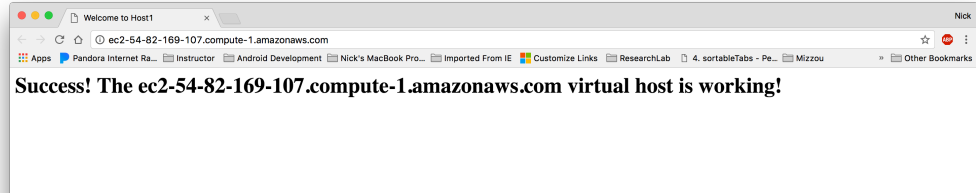
- If your page is not showing up and going to the original index.html page we first set up for **ec2-54-82-169-107.compute-1.amazonaws.com** then your configuration file might not be pointing to the right address
- For example
 - If we type “professorwergeles.com” in the URL of a browser, the browser will change the address to www.professorwergeles.com
 - Since we did not handle this in the configuration file
- Therefore, you will need to add an alias in the professorwergeles.com.conf file



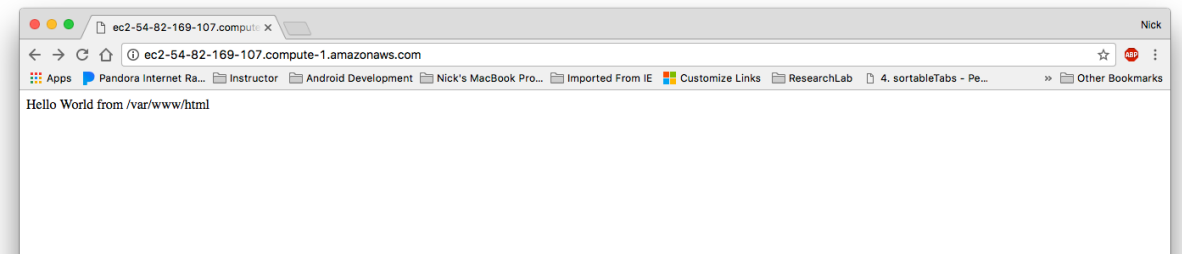
-
- Make sure to restart apache after changing the configuration file
 - “sudo service apache2 restart”

Test your result:

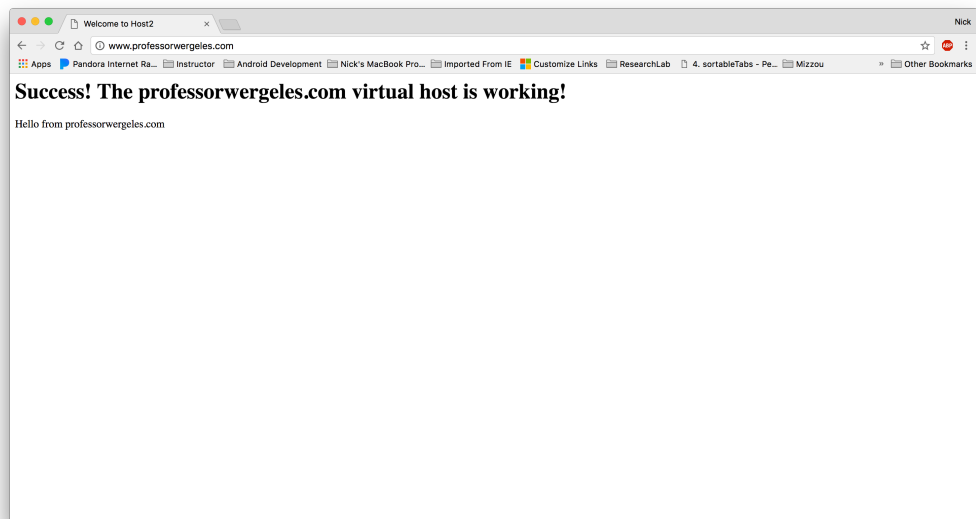
- Go to:
 - <http://ec2-54-82-169-107.compute-1.amazonaws.com/>
 - You should see



-
- Or method 2



-
- Go to:
 - <http://www.professorvergeles.com/>
 - You should see



-