1. **Virtual Host**

Virtual host refers to different websites running in a single web server. To create a virtual host, go to the root directory of Apache. This is where we will store all the resources for a website. The root directory of apache is as follows:

/var/www

We configured the server to contain the resources (group4a, group4b, group4c) to be used in creating a

virtual host. The folder “html” contains the default welcome page of apache web server



Next, we created a configuration file for the virtual host. Configuration files contains the settings as to how the virtual host would behave when accessed by a client. The configuration files are stored in the directory:

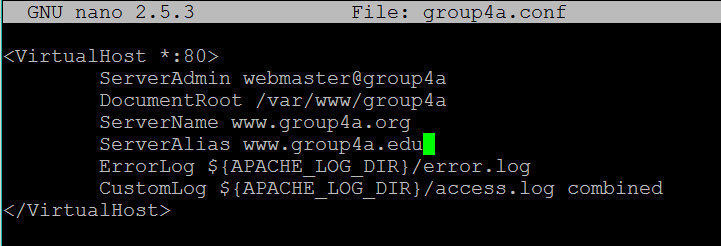
/etc/apache2/sites-available

Now inside the sites-available folder, we created the files named group4a.conf, group4b.conf, and group4c.conf as the configuration files for the three virtual host.



Using a text editor, we edited the configuration files. Remember to always use the command “sudo” so that the server would know that the command is from the administrator.

sudo nano group4a.conf



ServerAdmin – the email of the site administrator.

DocumentRoot – the location for the resources of the website.

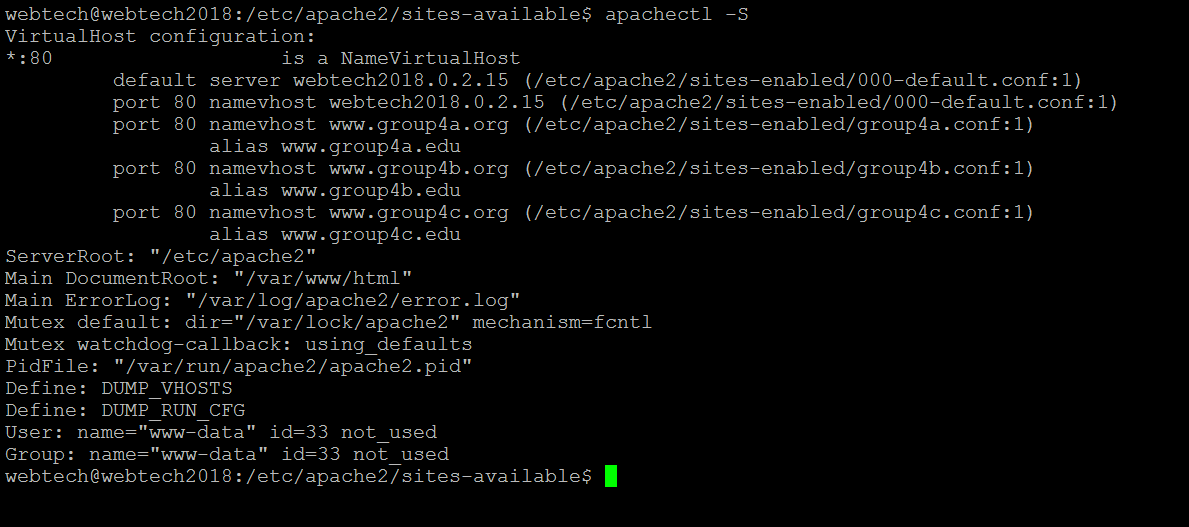
ServerName – the main domain name of the website.

ServerAlias – other domain names that could refer to the website.

After typing the configurations, save the file.

For debugging purposes, the following commands are used:

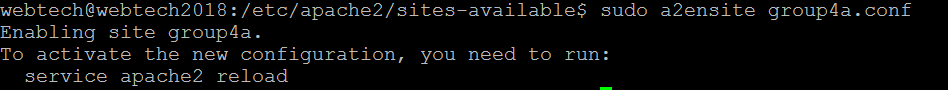
apachectl –S



This command will show a description of the configuration file that may be used to uncover errors when debugging.

Next, we need to enable the virtual host configurations by typing the following command:

sudo a2ensite group4a.conf

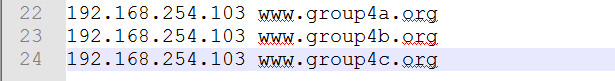


To commit the changes, restart apache by typing the following command

sudo service apache2 restart

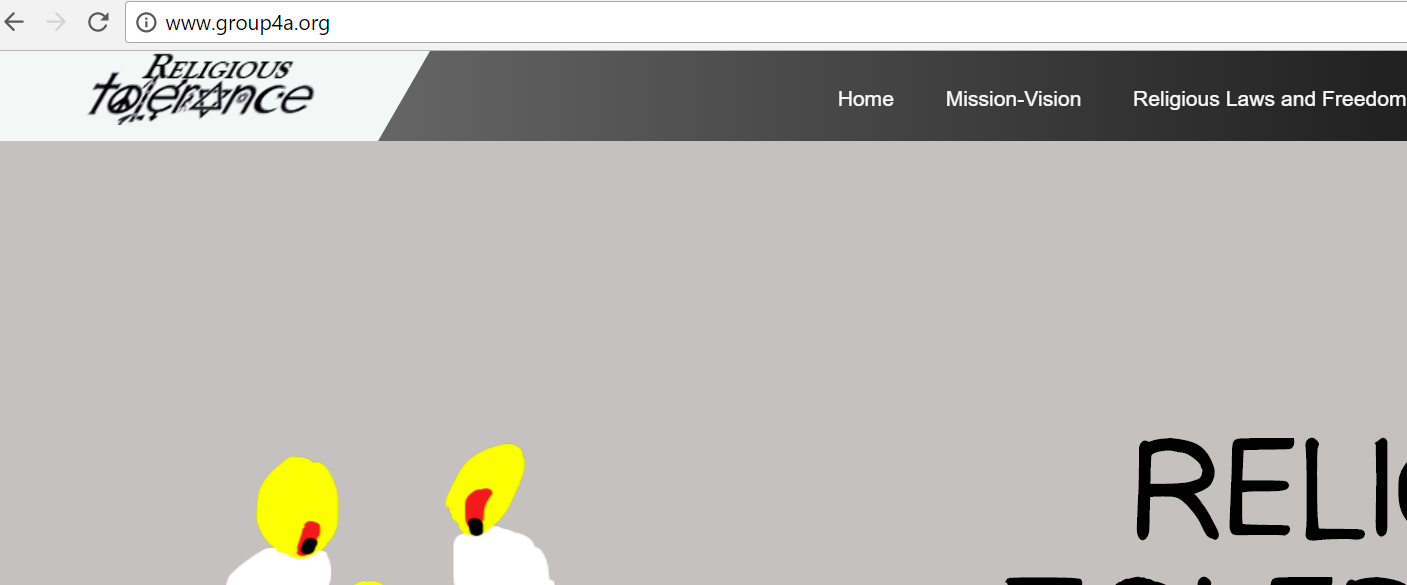
In order for the client to access the websites, we need to modify the hosts file of the client to act as a DNS system. You can find the hosts file in the following directory:

Windows\System32\drivers\etc\host



Using a text editor, type in the ipaddress of the server and the ServerName in the host file. You can also use the ServerAlias as a domain name. By doing this, every request for [www.group4a.org](http://www.group4a.org) would direct the computer to the ip address of the server which would connect it to the virtual host.

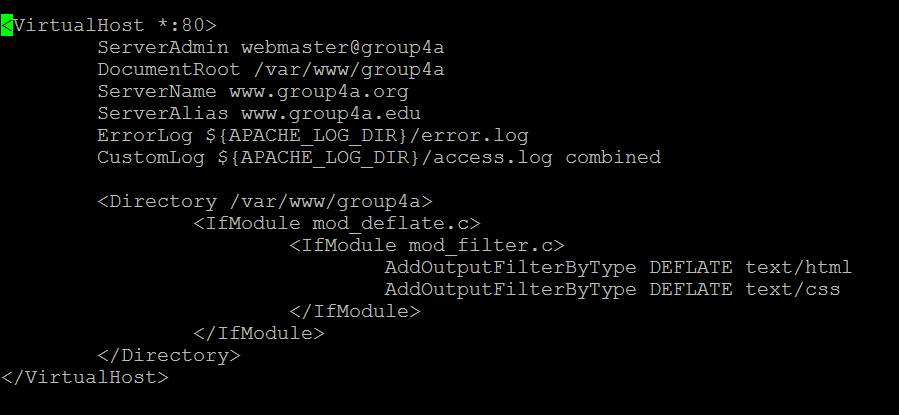
Finally, in order to check if the virtual host is working, type the ServerName in any browser.



1. **Content Compression**

Compression is a function in apache that allows the server to compress files that will by access by the clients. This minimize the size of the resource, thus making clients access it easier and faster.

Using a text editor, open the configuration file of a virtual host and type the following commands inside the <VirtualHost \*:80> </VirtualHost>

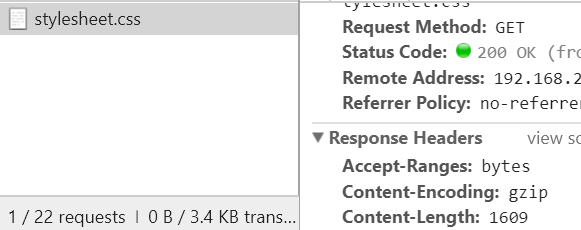


mod\_filter.c is the configuration that is accessed for compression. You can add resource types by using “AddOutputFilterByType DEFLATE type”. The command in the screenshot means that all html and css files will be compressed whenever the server serve it to a client.

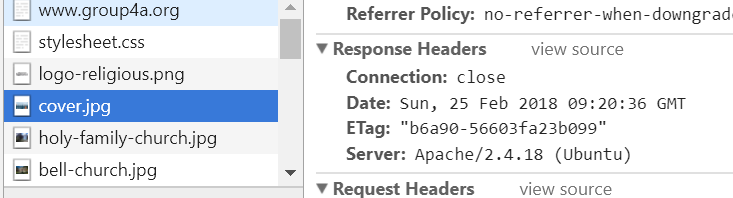
Restart apache to commit all configurations made

sudo service apache2 restart

To check whether the content compression is working, open your browser and open the developer tools. Click Network, reload the page and open the html file and the css file. As you can see from the screenshot below, the css file has a content-encoding that is equivalent to gzip. This means that the compression configuration is working.



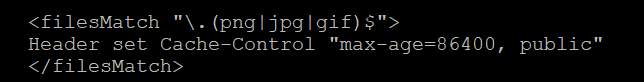
Cover.jpg doesn’t have a gzip because it was not included in the compression configuration.



1. **Content Caching**

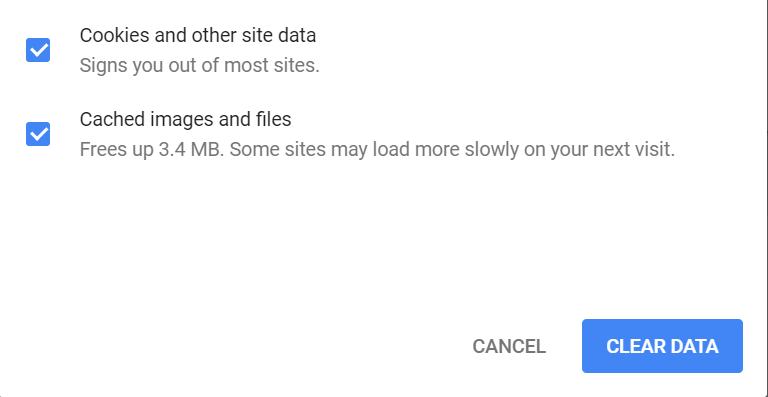
Apache’s caching features allow the contents retrieved by the clients to be stored locally in the browser. This will speed up retrieval the next time the client access the resources.

In order to enable the caching, add the following settings to the configuration files.



This means that all png, jpg, and gif will be cached and stored for 86400 seconds (24 hours). After 24 hours, the cached content will be deleted.

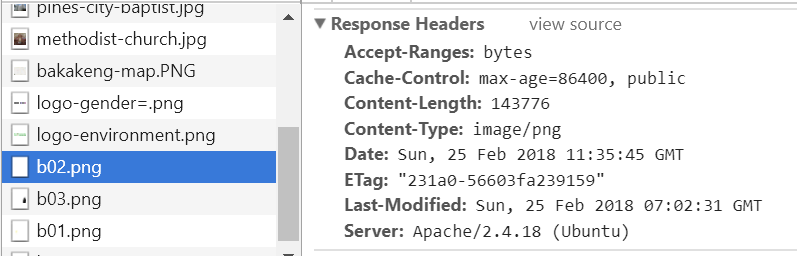
To check whether the resources are cached, go to your browser and clear all cached images and files.



Next load one of your website then close it afterwards. Then turn off your internet connection.

Try to load the website again, notice that the site can still load without internet connection.

Another way to check it is by using the developer tool of the browser. If the caching feature is enabled, the response header will contain the max-age.



1. **Content Negotiation**

Content Negotiation is a feature of apache that can choose the best resource to be served base on the client’s preferences.

1. **Accept-Header**

Let us first create a virtual host named webtech1.negotiate.org. A folder named negotiate is created to contain the resources for the virtual host. These resources are two files named content.html and content.txt



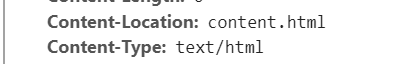
Create a configuration file for the virtual host. Inside, put the basic settings like the ServerName, ServerAlias, etc. Next, add the configuration to enable Multiviews. Save and enable the configuration, then restart apache.



Multiviews is one variant of content negotiation. For example, the virtual host webtech1.negotiate.org contains two file with the same name but different type. One is an html, and the other is a plain txt file. If I accessed

<http://webtech.negotiate.org/content>

without telling the file extension, it’s up to the server to negotiate base on the client’s prefence.



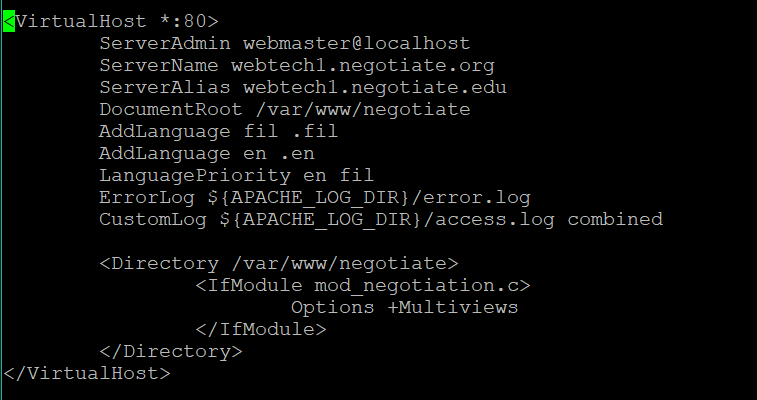
According to the screenshot above, the server choose content.html because it best matches the client’s preference.

1. **Accept-Language**

Inside the resource folder for webtech.negotiate.org, add the files language.html.en and language.html.fil

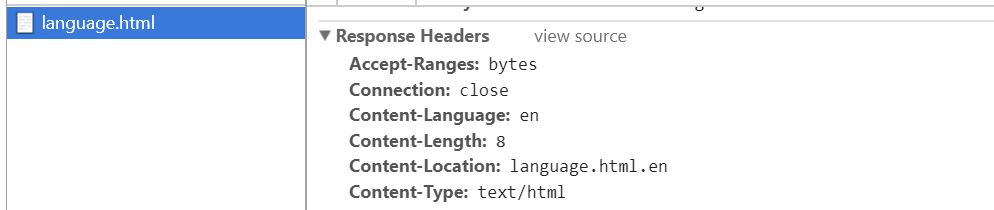


Add the following settings to the configuration file of webtech.negotiate.org virtual host



AddLanguage – add a language and its extension

LanguagePriority – this tells the order of priority when serving the resources. In this case, en (English) has more priority than fil (Filipino) so English will be served first.



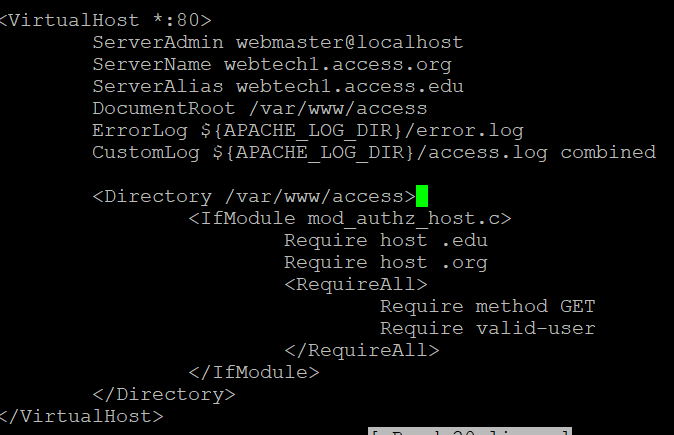
1. **Access Control**

Access Control is a feature of apache wherein the admin can configure and control who and what can access resources from a website.

First, let us create a virtual host named webtech1.access.org with folder named access that will contain the resource index.html



Next open the configuration file for webtech1.access.org to edit the setting for the virtual host. Add the following settings to activate the access control feature.



These commands specify that only domain names who ends with edu and org can access the resources.

Require host .edu

Require host .org

These commands specify that only GET and HEAD method can access the resources, the other method needs to be accessed by valid-user and needs authentication.

<RequireAll>

Require method GET

Require valid-user

</RequireAll>

If the client trying to access the resources doesn’t match to any access control configuration, the server would return a forbidden message.



To configure the virtual host such that only clients that has user credentials can access the resources, let us create a file that will contain the users and passwords.

We created the folder passwd to contain the password configurations for the webtech1.access.org virtual host.

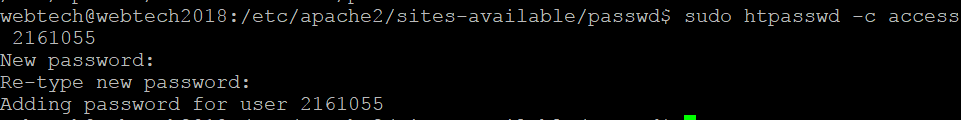


Go inside the folder passwd and type the following command:

sudo htpasswd –c access 2161055

This command means that it will create a new password file using htpasswd

It will prompt the administrator to type in the password for user xxx.



To add another user simply remove –c.

sudo htpasswd access newuser

Then add the following settings to a virtual host configuration file that needs authorization to access the resources.

Try accessing the virtual host by typing the domain name of the virtual host. It required authentication before it allowed an access to the resources.

