**Configuration**

Virtual Host: **group4a.org**

Pre-req: (This is assuming that *group4a.org* has already been created, enabled, and tested, as the following involves merely configuring.)

**A. Enable serving up compressed content (HTML/CSS)**

When a browser requests a file or website, the web server sends the requested information back. However, with websites nowadays featuring bulky texts and multimedia attachments, it can often appear as a large file. By compressing it, usually gzip, before it is sent to the browser, it reduces bandwidth and download time, allowing the website to be loaded faster. On Apache, the module *mod\_deflate* provides us a way to do that.

1. Enable the Apache module.

|  |
| --- |
| sudo a2enmod deflate |

1. Reload or restart the Apache service, as prompted.

|  |
| --- |
| sudo service apache2 restart |

1. To configure the virtual host, proceed to where its *conf* file is located in the server:

|  |
| --- |
| cd /etc/apache2/sites-enabled |

1. Edit the conf file that was previously created.

|  |
| --- |
| sudo vi group4a.org.conf |

1. Add the following to the file, so that the virtual host will compress only HTML and CSS files:

|  |
| --- |
| AddOutputFilterByType DEFLATE text/html text/css |

End result:

|  |
| --- |
| # group4a.org.conf  <VirtualHost \*:80>  # already existing code  <Directory /var/www/group4a.org/public\_html>  AddOutputFilterByType DEFLATE text/html text/css  </Directory>  </VirtualHost> |

**B. Enable cache of files (PNG, JPG, GIF)**

Whenever a browser sends a request to the server, it downloads the file or information it requested, saving merely a copy. Sending requests multiple times, however, would also mean re-downloading the same files again and again. This can lead to slow loading time, so to remedy this, we can cache the files by storing them within the browser. When the same request is sent again, it first checks if the files are already in the cache first, and displays it if it is. To ensure that the browser is always retrieving the latest version of the files, the cache can be configured.

On Apache, the module *mod\_cache* provides a way for server-side caching, but since we’re only focusing on the cache’s time limits, we will be using the module *mod\_expires* instead, which determines the max-age field.

1. Enable the Apache module.

|  |
| --- |
| sudo a2enmod expires |

1. Reload or restart the Apache service, as prompted.

|  |
| --- |
| sudo service apache2 restart |

1. To configure the virtual host, proceed to where its *conf* file is located in the server.

|  |
| --- |
| cd /etc/apache2/sites-enabled |

1. Edit the conf file that was previously created.

|  |
| --- |
| sudo vi group4a.org.conf |

1. Add the following to the file, so that the virtual host will cache PNG, JPG, and GIF files for up to 24 hours from the time they were accessed.

|  |
| --- |
| ExpiresActive on  ExpiresByType image/png "access plus 24 hours"  ExpiresByType image/jpg "access plus 24 hours"  ExpiresByType image/gif "access plus 24 hours" |

Note: if the configuration requires that the max-age be based on the file’s last modification time, then “M” or “modification” may be used.

End result:

|  |
| --- |
| # group4a.org.conf  <VirtualHost \*:80>  # already existing code  <Location />  ExpiresActive on  ExpiresByType image/png "access plus 24 hours"  ExpiresByType image/jpg "access plus 24 hours"  ExpiresByType image/gif "access plus 24 hours"  </Location>  </VirtualHost> |