# Review:

What command is best used to start/stop our Apache web server?

* sudo systemctl start/stop/restart/reload apache

What command give us detailed status of our Apache web server?

* apachectl status

Where is the /www directory created upon Apache installation (in Ubuntu)?

* /var

Where are the Apache config files located?

* /etc/apache2

Where are the apache log files located?

* /var/log/apache2

What are the two important log files?

* error.log & access.log

# Part 5 – Setting Up Virtual Host

When using Apache web server, we can use *virtual hosts* (similar to server blocks, or virtual directories in IIS) to encapsulate configuration details and host more than one domain (or site) from a single server.

Apache on Ubuntu has one server block enabled by default that is configured to server documents from the **/var/www/html** directory. While this works for a single site, it can become unwieldy if you are hosting multiple sites.

Let’s create a custom virtual site:

1. Create a directory structure to host our web page(s):
   1. Run: **sudo mkdir /var/www/coos291**

Text

Description automatically generated

1. Next we need to make sure the ownership and permissions correct:
   1. **sudo chown -R $USER:$USER /var/www/coos291** <- recursive is useful if not empty
   2. **sudo chmod -R 755 /var/www/coos291** <- It’s already 755, more useful if files have been created already within the coos291 directory.

Text

Description automatically generated

1. Let’s create an HTML page in this directory called **index.html** within the /coos291 directory.

Text

Description automatically generated

Enter some HTML code, and then try to load the web page by ip address.

* 1. Oh snap! The default Apache2 web page is loaded:Graphical user interface, text, application, email

     Description automatically generated

We need to tell the server that when a certain URL is accessed, that the server should server different pages. To do that, we have to create a **Virtual Host**!

* 1. We need to create a new configuration file for the new Virtual Host (coos291). The Virtual Hosts configuration files are located in the **/etc/apache2/sites-available** directory. Let’s look at what exists there already:

We have 2 files here by default, “000-default.conf” and “default-ssl.conf”

Text

Description automatically generated

Let’s view the 000-default.conf: **vi 000-default.conf**

Text

Description automatically generated

* + 1. Server Admin – This is an email address that is displayed to users if any errors occure on the server that are displayed on the browser.
    2. DocumentRoot – Directory that forms that main document tree visible from the web.
    3. ErrorLog – Path and filename to the error.log file for this Virtual Host
    4. CustomLog – Path and filename of the access.log file for this Vitual Host.

1. Let’s create a new configuration file for a new Virtual Host (coos291)
   1. Type the following: **sudo vi /etc/apache2/sites-available/coos291.conf**
   2. OR **sudo cp 000-default.conf coos291.conf** and then change the directives as follows:

Text

Description automatically generated

1. We need to enable the configuration file using the **a2ensite** too (short for Apache2 enable site).
   1. Type the following: **sudo a2ensite coos291.conf**

Text

Description automatically generated

* 1. Before we activate the new config we should always run **apachectl configtest** to ensure there is no errors in the .conf file. If errors are present Apache will stop.
  2. Finally we run **sudo systemctrl reload apache2**
  3. Modify the hosts file: **vi /etc/hosts**, add in the line **127.0.1.1 coos291.com**

Text

Description automatically generated

* 1. We should now be able to load the coos291 website by name:

Graphical user interface, text, application

Description automatically generated

Let’s create another custom Virtual Host, COOS294:

1. Let’s make a **/coos294** subdiretory in **/var/www**:
   1. Then copy the index.html from /var/www/coos291 to /var/www/coos294
   2. Modify the ownership and permissions, the directory isn’t empty this time, so -R (recursive) is useful!

Text

Description automatically generated

1. Ultimately, what we want to set up is when coos291.com is accessed, we will serve pages from /var/www/coos291, and when coos294 is accessed we will serve pages from /var/www/coos294.
   1. To do this we need to create an additional Virtual Host for the coos294 site. Create the **coos294.conf** file in the **/etc/apache2/sites-available** directory.
      1. We can copy the coos291.conf file and make the appropriate changes.

Text

Description automatically generated

1. Next we enable the virtual host via the a2ensite command, run configtest, and then reload apache:
   1. Type: **sudo a2ensite coos294.conf** then **sudo apache2ctl configtest**, then **sudo systemctl reload apache2**.

Text

Description automatically generated

* 1. Next change the /var/www/coos294/index.html file contents and then load coos294.com in Firefox:

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

# Virtual Host Matching

Apache has an order to how it matches Virtual Hosts (MEMORIZE THIS):

1. Look for all exact IP matches – (for when the web server has multiple IP addresses.)
2. Only if there is no exact match, go by wildcards. (The wildcard is the \* in the Virtual Host .conf file).
3. If there are multiple matches, and the request contains a Host: header, look for the first matching ServerName or ServerAlias. (This is the most common situation.)
4. Otherwise, **take the first match**.

# .HTACCESS

An **.htaccess** file is a way to configure the details of your website without needing to alter the server config files. (The . indicates it is hidden.)

You can created the .htaccess file in a text editor (make sure to only name it .htaccess without any other extensions or names.)

Additionally, the placement of the .htaccess file is important. The configurations in that file will affect everything in its directory and the directories under it.

The benefit of the .htaccess file is that root permissions are not needed to reload the Apache2 service.

The disadvantage of .htaccess is that it slows down your web server as the .htaccess file needs to be looked for and then processed if it exists every time a page is accessed.

## Adding Custom Error Pages:

One of the things Apache autoconfigures is the error page. Let’s add and configure an .htaccess file to our coos291 site which will override the default error page.

1. Use **vi .htaccess** and add the line: **ErrorDocument 404 /new404.html**

Text

Description automatically generated

This directive tell the Apache server to serve **new404.html** as it’s response to a 404 error. We can add additional ErrorDocument directives for our other HTTP status codes (401 Unauthorized, etc).

1. The .htaccess file is ignored until we tell Apache to allow it. To enable the **.htaccess**. to work, we need to enable the additional directive in our **coos291.conf** within the /etc/apache2 directory. (Keep in mind the location of the enabling matters! – Which .conf file you add this too, if you do the apache2.conf file it will apply to ALL Virtual Hosts!)

Graphical user interface, text, application

Description automatically generated

Add the following Directory to coos291.conf before the </VirtualHost>

|  |
| --- |
| <Directory>  Options Indexes FollowSymLinks MultiViews  AllowOverride All  Order allow,deny  allow from all  </Directory> |

Text

Description automatically generated

This allows the coos291 directory to have it’s own custom .htaccess file to override any configuration directives in the global file.

1. Next we want to configtest, reload and check the status:
   1. **sudo apachectl configtest**
   2. **sudo systemctl reload apache2**
   3. **sudo systemctl status apache2**

Text

Description automatically generated

1. Next return to your /var/www/coos291 directory and create a new **new404.html** file. \*\*You do not need to sudo this!
   1. **cd /var/www/coos291**
   2. **vi new404.html**  and add some text or html code.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated