# Create directory structure

* sudo mkdir -p /var/www/example.com/public\_html
* sudo mkdir -p /var/www/test.com/public\_html

# Grant permission

* sudo chown -R $USER:$USER /var/www/example.com/public\_html
* sudo chown -R $USER:$USER /var/www/test.com/public\_html

The $USER variable will take the value of the user you are currently logged in as when you press **Enter**. By doing this, our regular user now owns the public\_html subdirectories where we will be storing our content.

We should also modify our permissions a little bit to ensure that read access is permitted to the general web directory and all of the files and folders it contains so that pages can be served correctly:

* sudo chmod -R 755 /var/www

# Create Demo Page

* nano /var/www/example.com/public\_html/index.html

In this file, create a simple HTML document that indicates the site it is connected to. My file looks like this:

/var/www/example.com/public\_html/index.html

<html> <head>

<title>Welcome to Example.com!</title> </head>

<body> <h1>Success! The example.com virtual host is working!</h1> </body></html>

# Create new virtual host file

Start by copying the file for the first domain:

* sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/example.com.conf

Open the new file in your editor with root privileges:

* sudo nano /etc/apache2/sites-available/example.com.conf

The file will look something like this (I’ve removed the comments here to make the file more approachable):

/etc/apache2/sites-available/example.com.conf

*<VirtualHost \*:80>*

*ServerAdmin admin@example.com*

*ServerName example.com*

*ServerAlias www.example.com*

*DocumentRoot /var/www/example.com/public\_html*

*ErrorLog ${APACHE\_LOG\_DIR}/error.log*

*CustomLog ${APACHE\_LOG\_DIR}/access.log combined*

*</VirtualHost>*

As you can see, there’s not much here. We will customize the items here for our first domain and add some additional directives. This virtual host section matches *any* requests that are made on port 80, the default HTTP port.

# Enable new host file

* sudo a2ensite example.com.conf
* sudo a2ensite test.com.conf

Next, disable the default site defined in 000-default.conf:

* sudo a2dissite 000-default.conf

When you are finished, you need to restart Apache to make these changes take effect:

* sudo systemctl restart apache2

# Setup local host file

Edit file C:\**Windows**\System32\Drivers\etc\hosts

<ip address> test.com

# Redirect request to port 8080

Sudo vi /etc/apache2/sites-available/test.com.conf

Add these lines -

*ProxyPreserveHost On*

*ProxyRequests Off*

*ProxyPass /* [*http://localhost:8080/*](http://localhost:8080/)

*ProxyPassReverse /* [*http://localhost:8080/*](http://localhost:8080/)

*#changes for websocket*

*RewriteEngine On*

*RewriteCond %{HTTP:Upgrade} ^Websocket$ [NC]*

*RewriteCond %{HTTP:Connection} ^Upgrade$ [NC]*

*RewriteRule .\* “ws://localhost:8080%{REQUEST\_URI}” [P]*

After saving enable proxy modules

Sudo a2enmod proxy

Sudo a2enmod proxy\_http

Sudo a2enmod proxy\_wstunnel

Sudo a2enmod rewrite

Sudo systemctl restart apache2