# Week #4

**Week 4 - Persistent, Non-Persistent, Authentication, Cookies and Conditional GET**

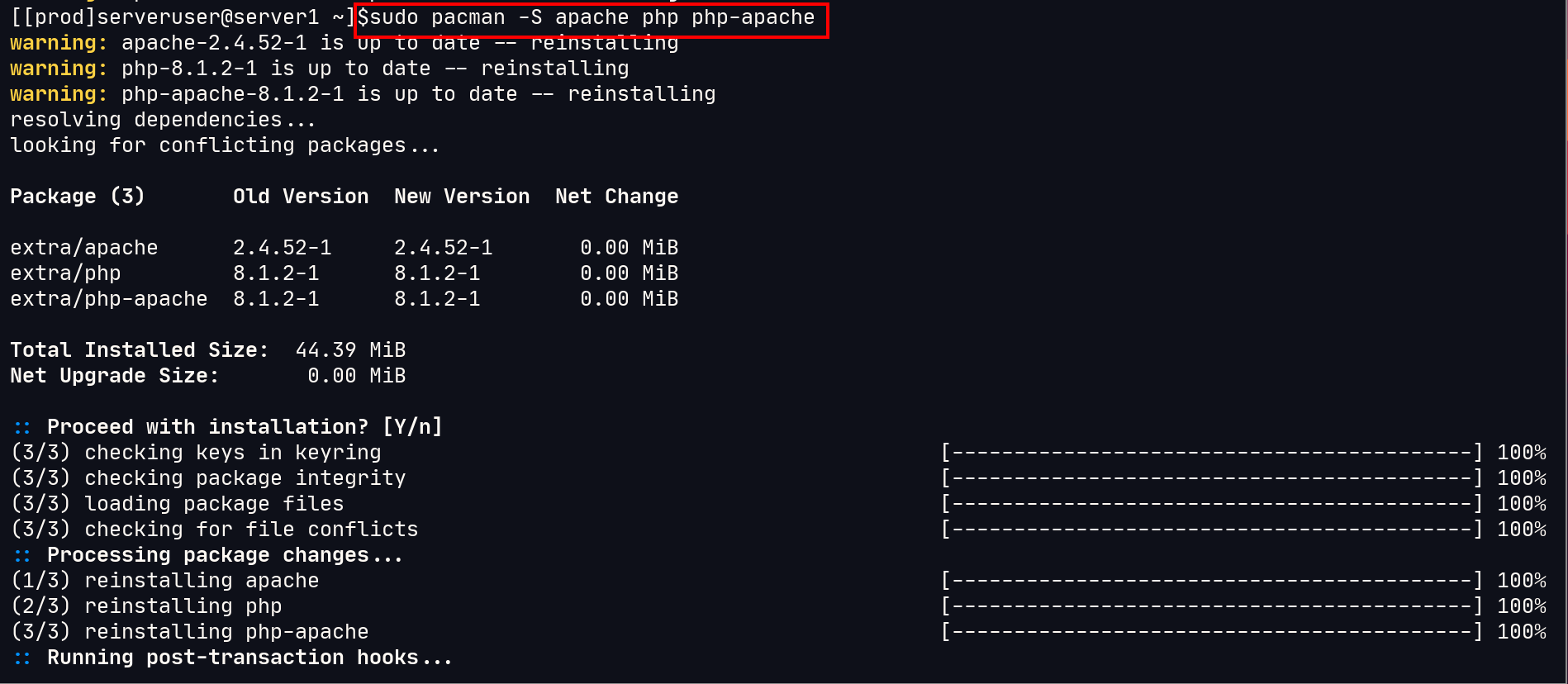
**SRN : PES2UG20CS237**

**Name : P K Navin Shrinivas**

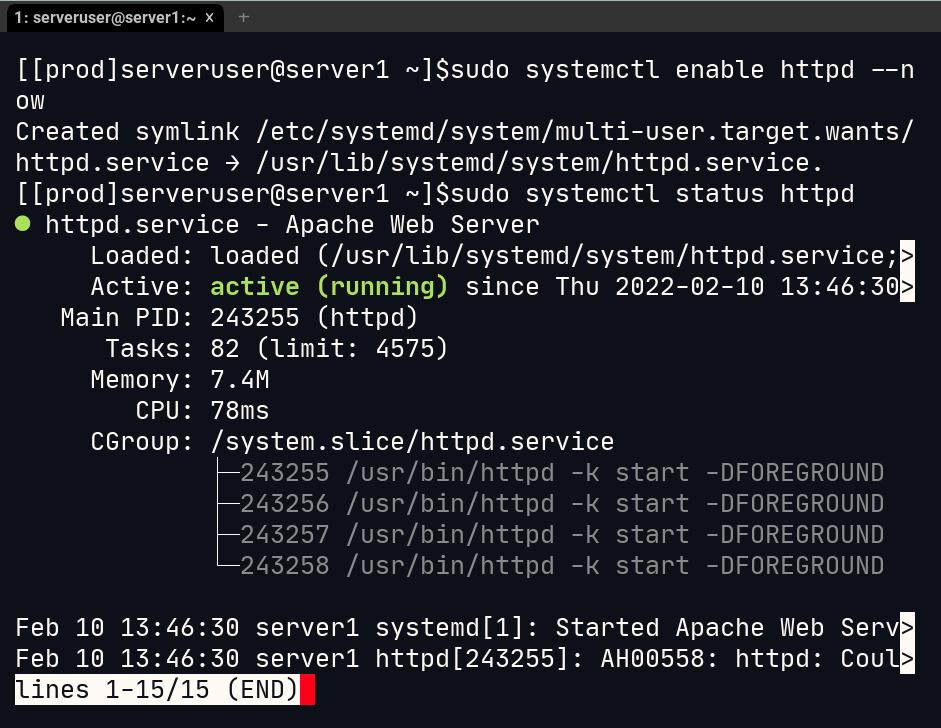
**Section : D**

## Task 1 : Setup a apache server

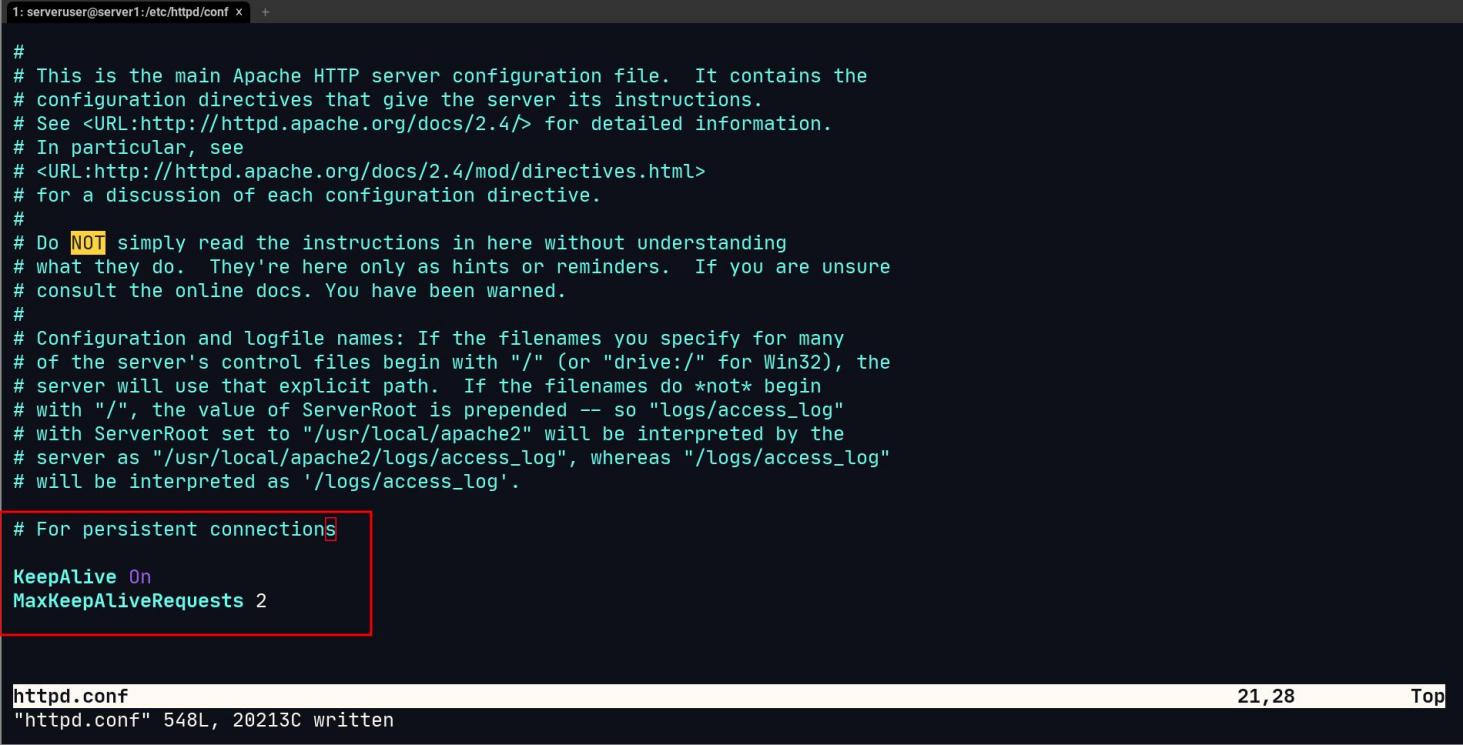
* My setup includes 2 sepereate linux machines within my netowrk, both runnning arch linux.
* First packages needed were installed using the following command :



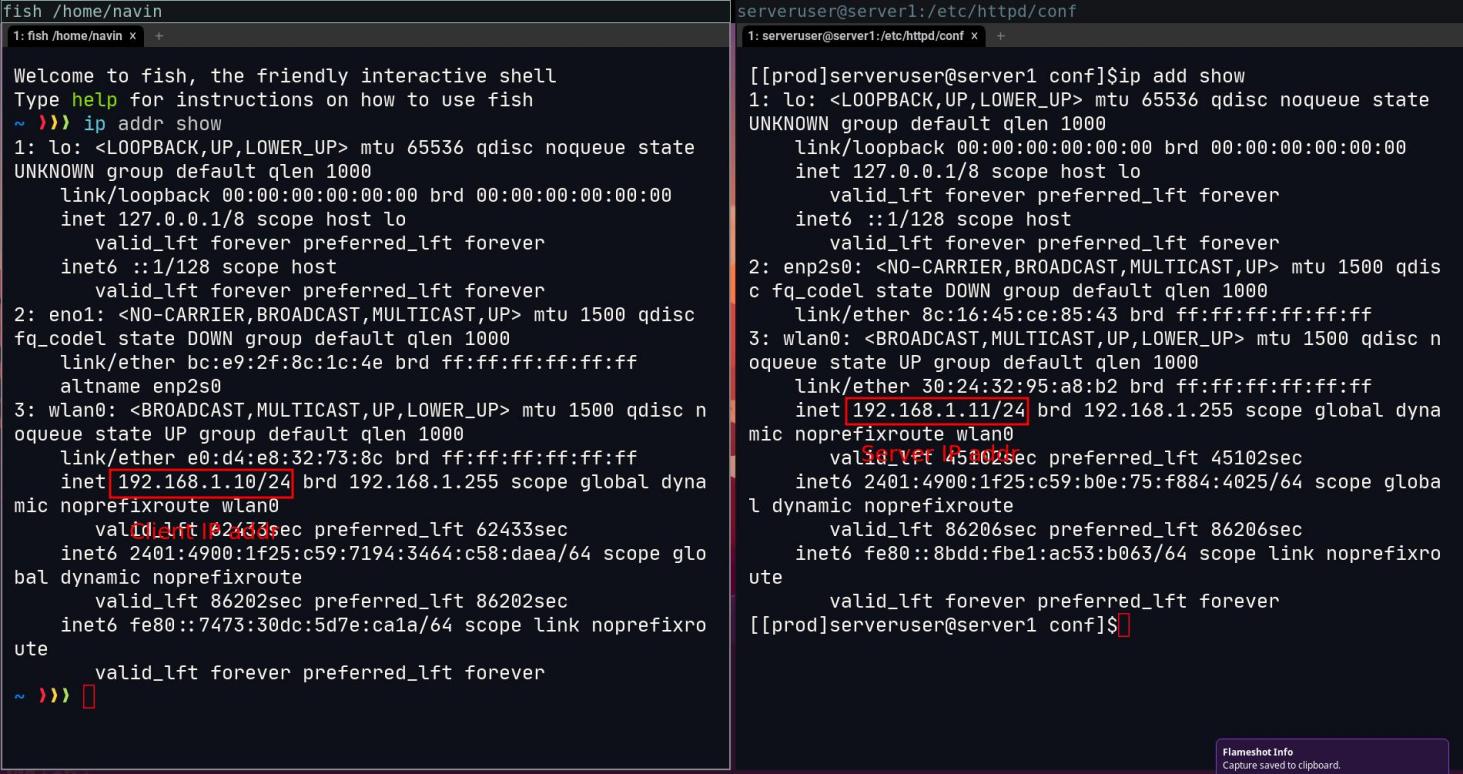
* Apache server was then turned on and its status was checked to be “active” :



* Apache server was configured to allow persistent connections by modifying the /etc/httpd/conf/httpd.cong (specific to distro) was modified like so :



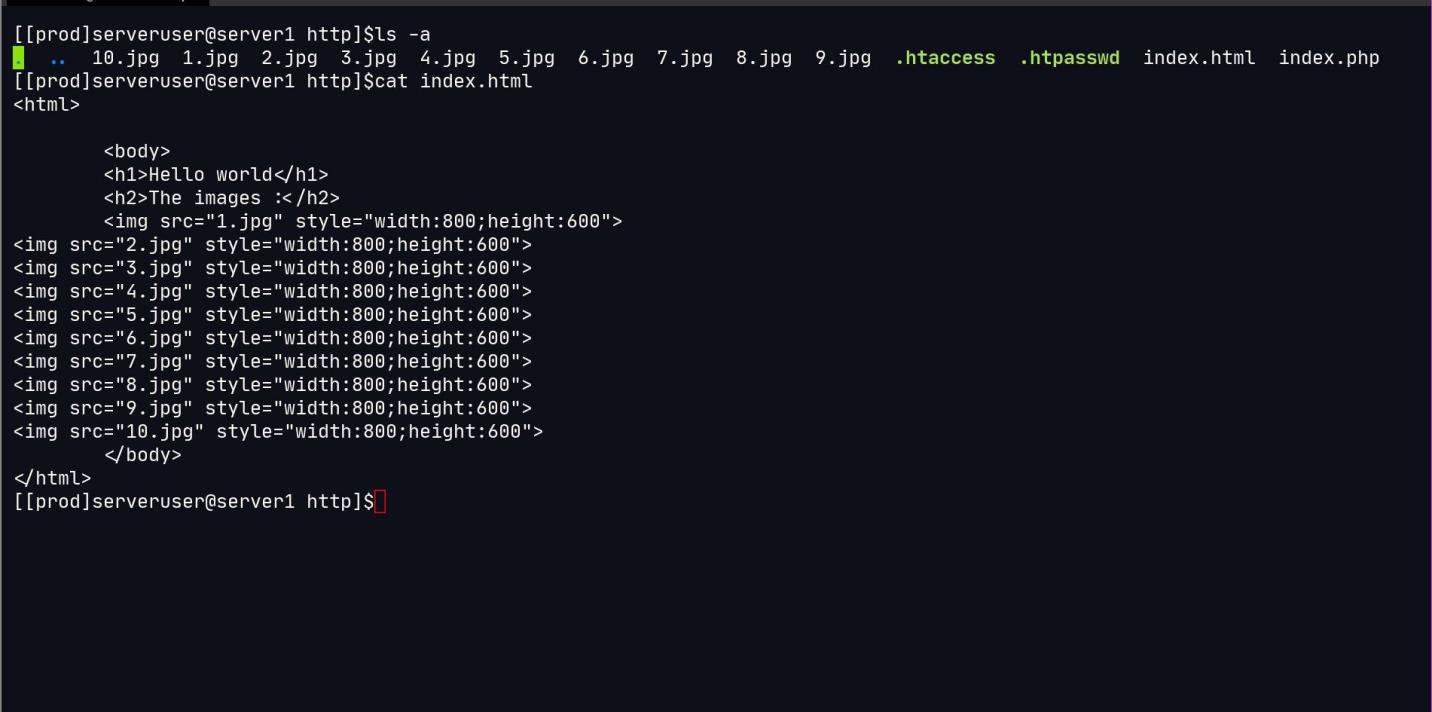
* The apache server was restarted after saving changes to the above file suing : sudo systemctl reststart httpd
* Both the system were assigned status ip local addresses from the router itself :



Client : 192.168.1.10

Server : 192.168.1.11

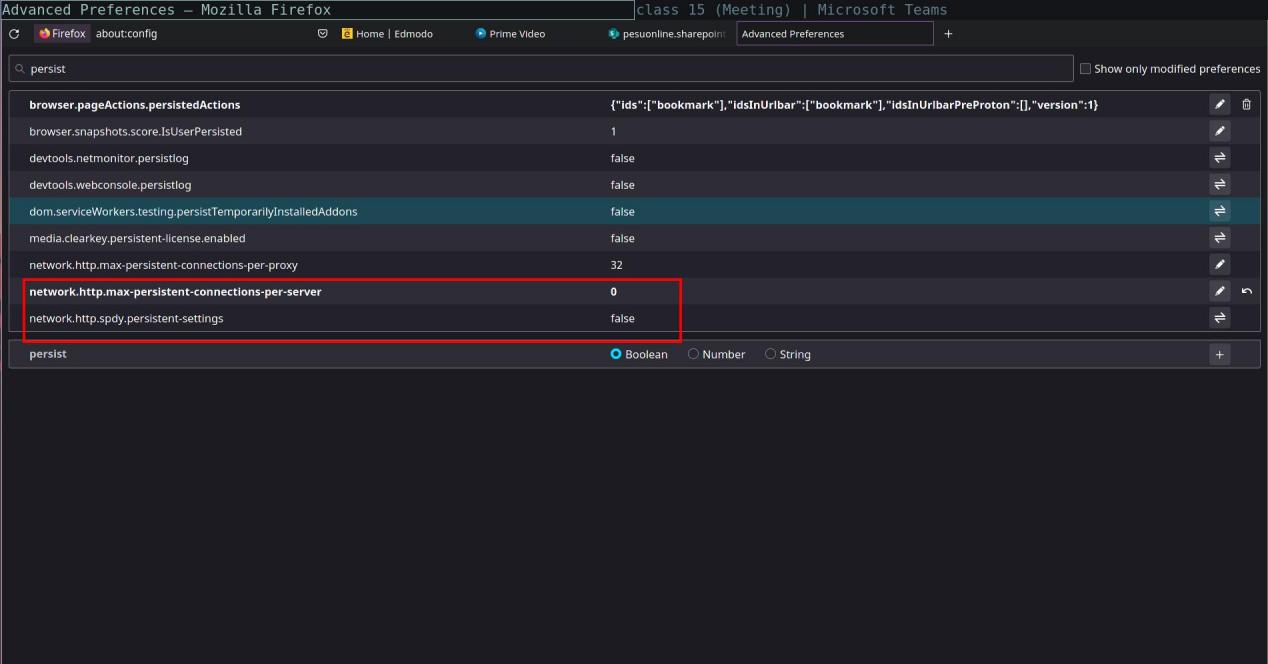
* The Webpage is hosted by copying over the provided photos to the respective folder and writing an index.html file :



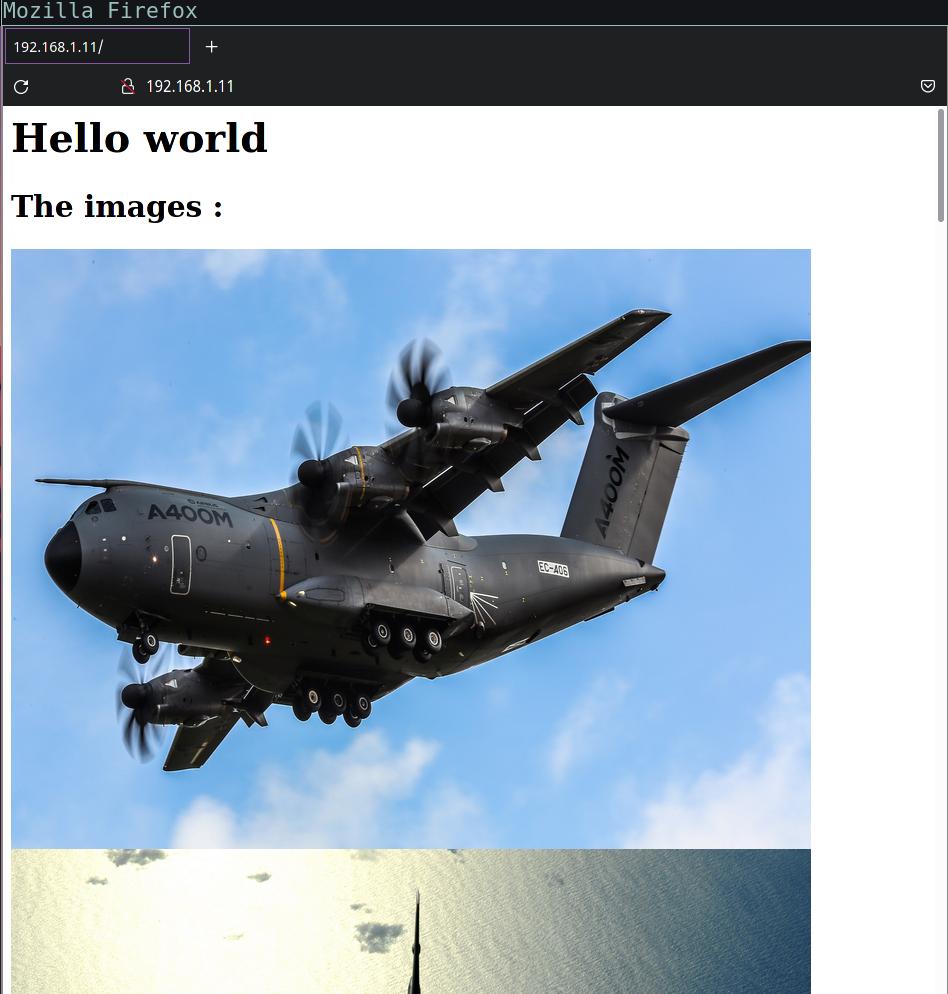
The extra folder are not to confused as they are used in later tasks, only the .jpg and index.html folder is needed for task 1.

## **Task 2 :Non-Persistent Connections**

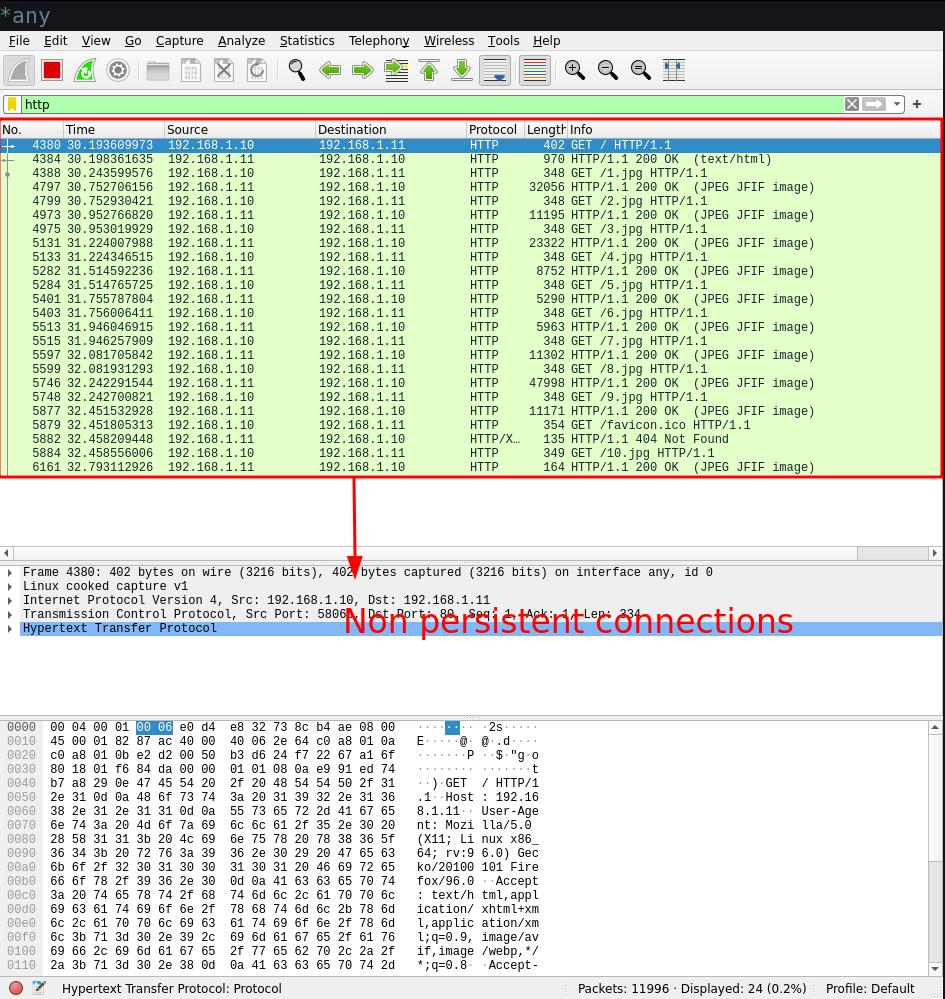
* Open firefox and enter address about:config , and press accpet risk and continue if asked.
* Once done, search for persist and set : **max-persistent-connections-per-server**  to 0 and set **persistent-settings** to false, like so :



* Now, ping your server from firefox by simply entering the ip address, like so :



* Analysing the packets in wirehark make it clear that we are handling non persistent connections :



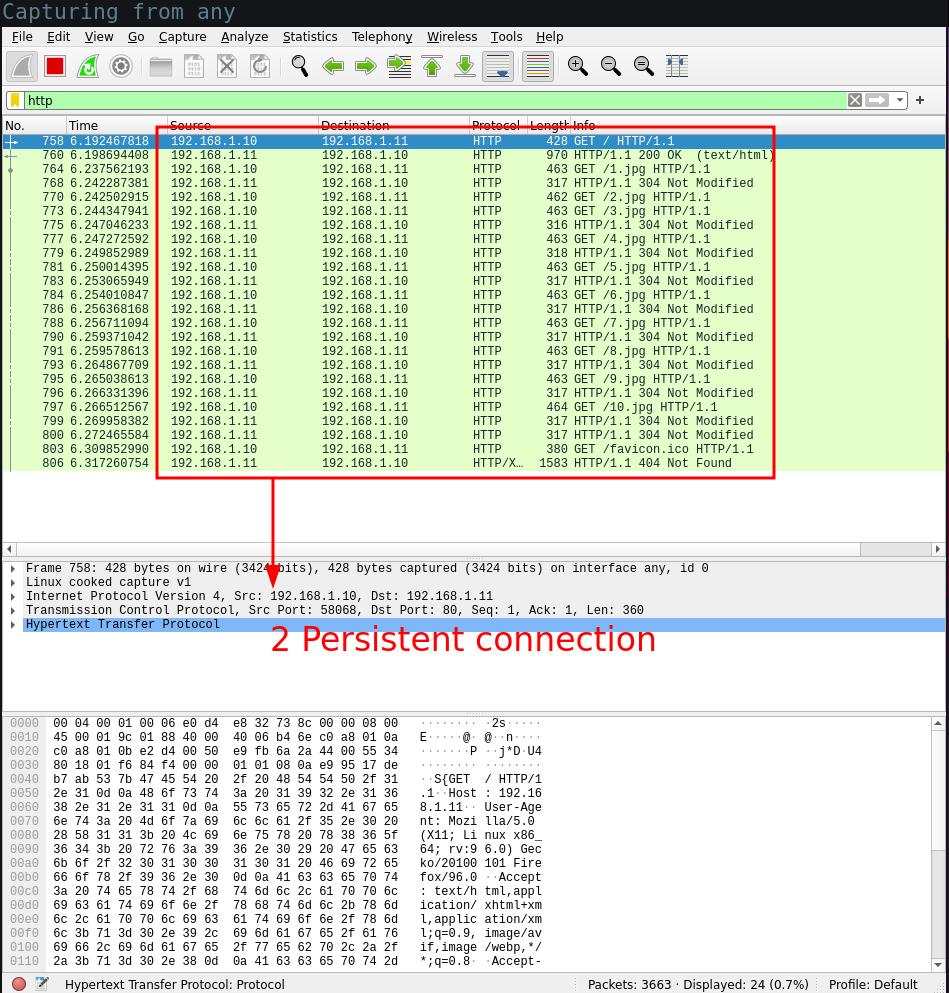
## **Task 3 : Persistent Connections**

* Open firefox again and got to about:config and set:

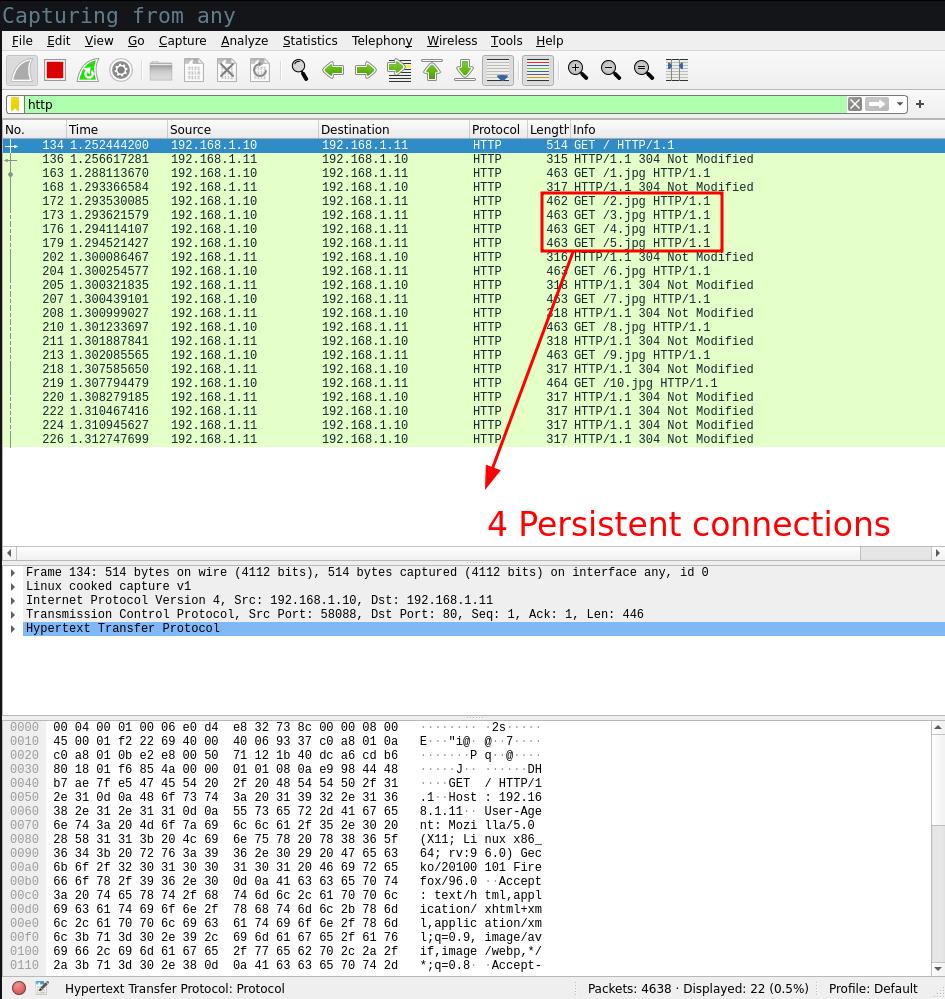
**max-persistent-connections-per-server** to 2,4,6..

**persistent-settings** to true

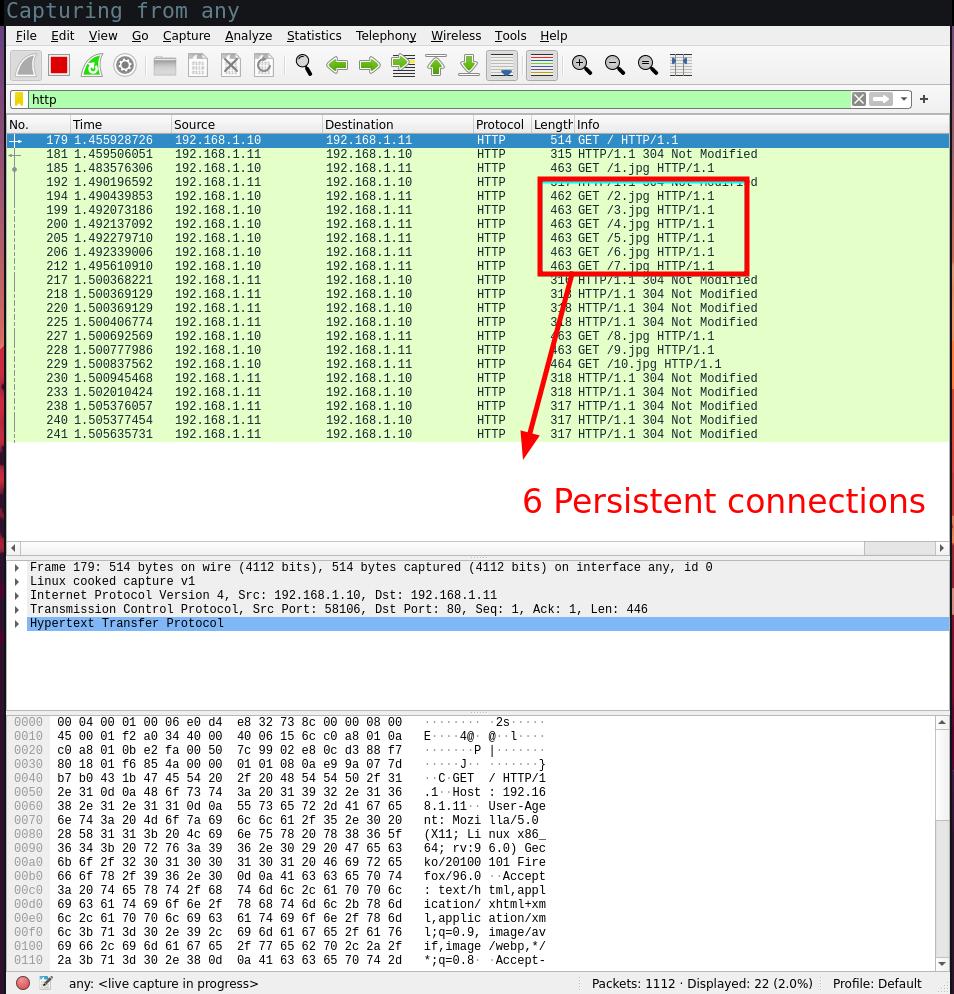
* 2 persist connections



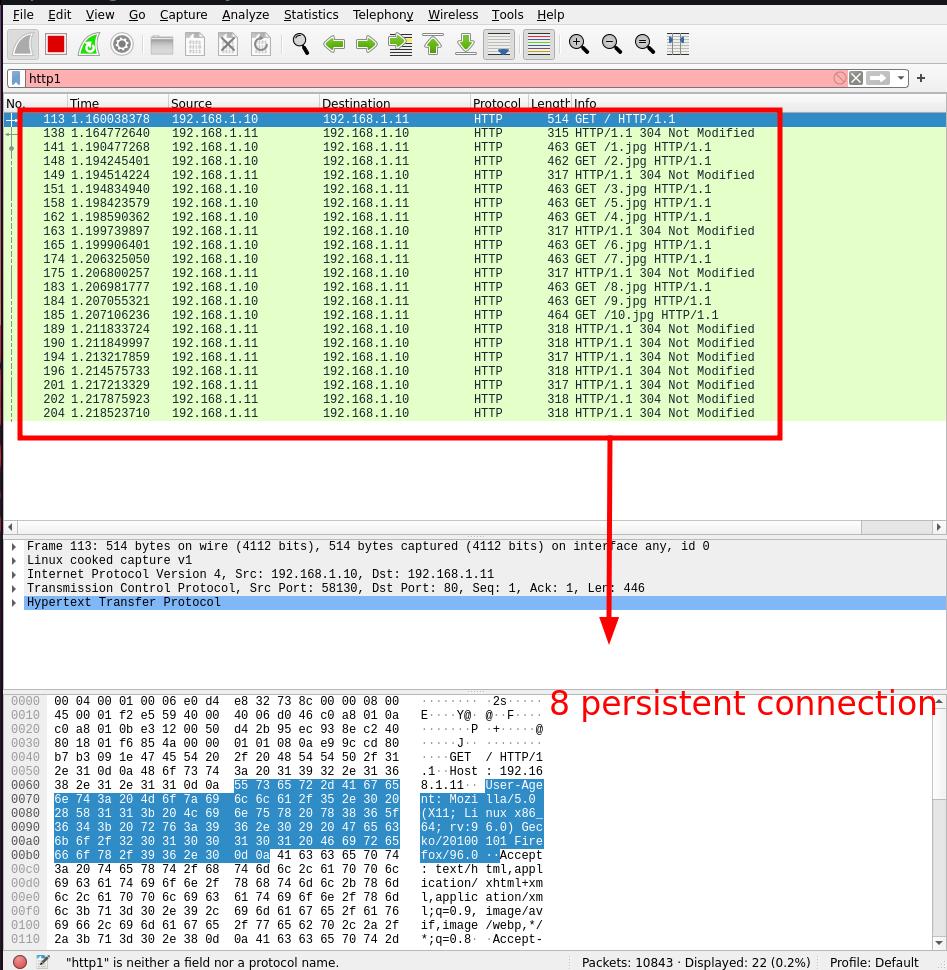
* 4 persist connections

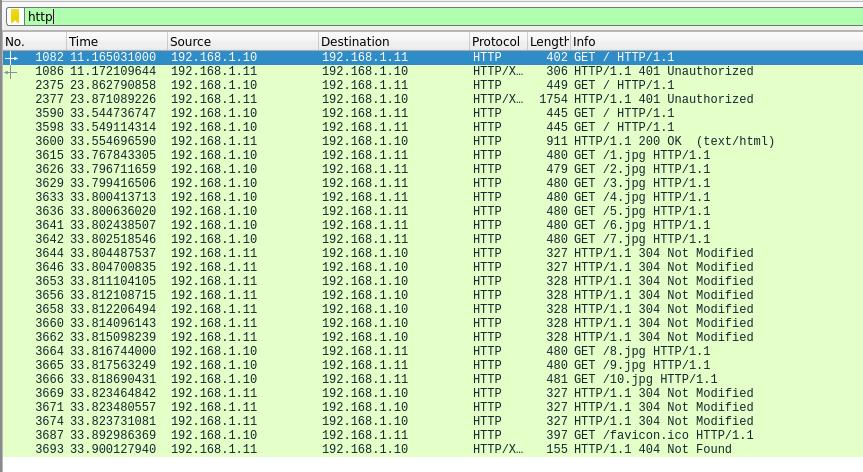


* 6 persist connections



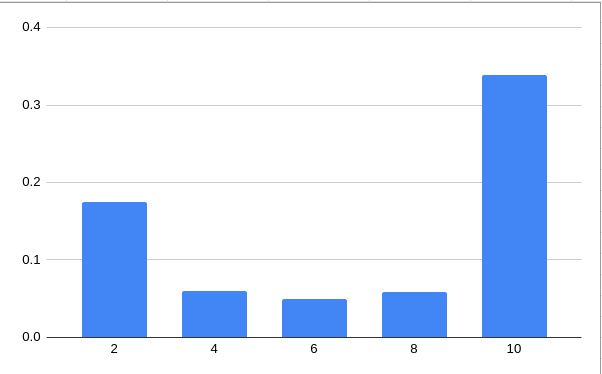
* 8 persist connections



* 10connections
* Obsevations from task 3 :

Calcuations and tabluating load time using wireshark :

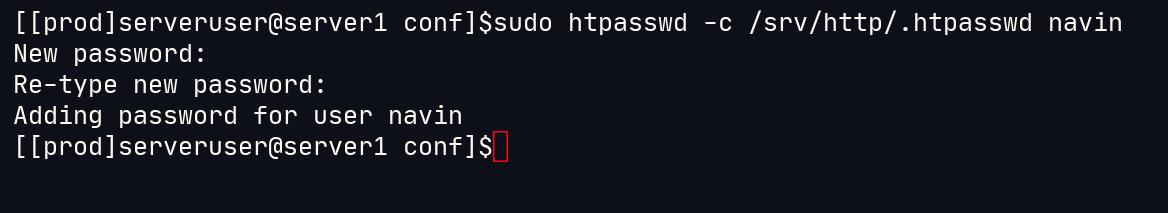
|  |  |  |  |
| --- | --- | --- | --- |
| **Persistence** | **First GET time** | **Last Response** | **Load time** |
| **0** | 30.19360 | 32.71399 | 2.52039 |
| **2** | 6.14296 | 6.31726 | 0.1743 |
| **4** | 1.25244 | 1.31274 | 0.0603 |
| **6** | 1.45592 | 1.50563 | 0.04971 |
| **8** | 1.16003 | 1.21852 | 0.05849 |
| **10** | 33.55469 | 33.89298 | 0.33829 |



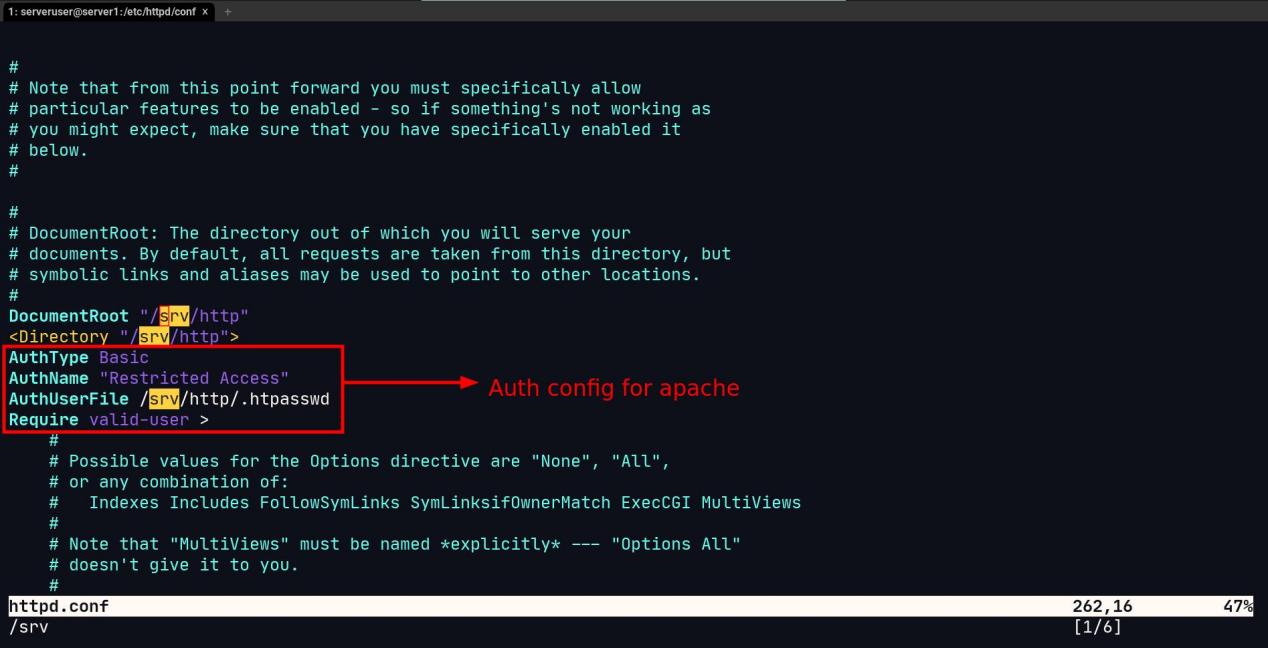
It is observed that the best number of persistent connection is 6.

## **Task 4 : Authentication in Apache server**

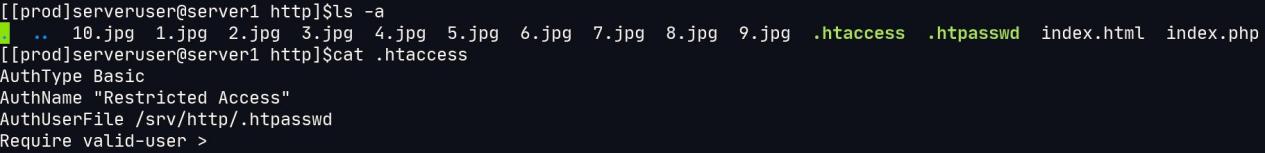
* Create a .htpasswd file in any place in the system preferably outside the server root. Like so :



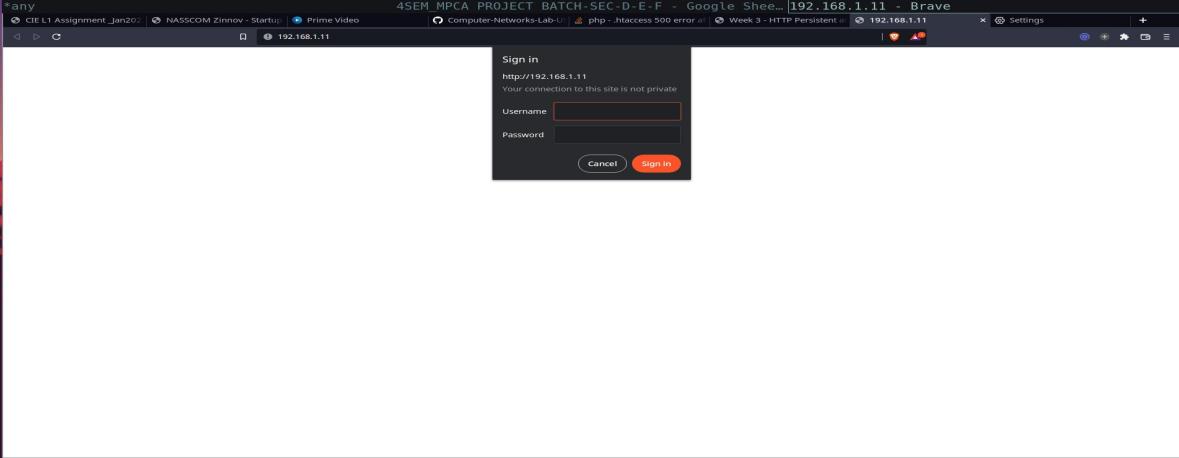
* Edit the apache config file in /etc/httpd/conf/httpd.conf file as so :



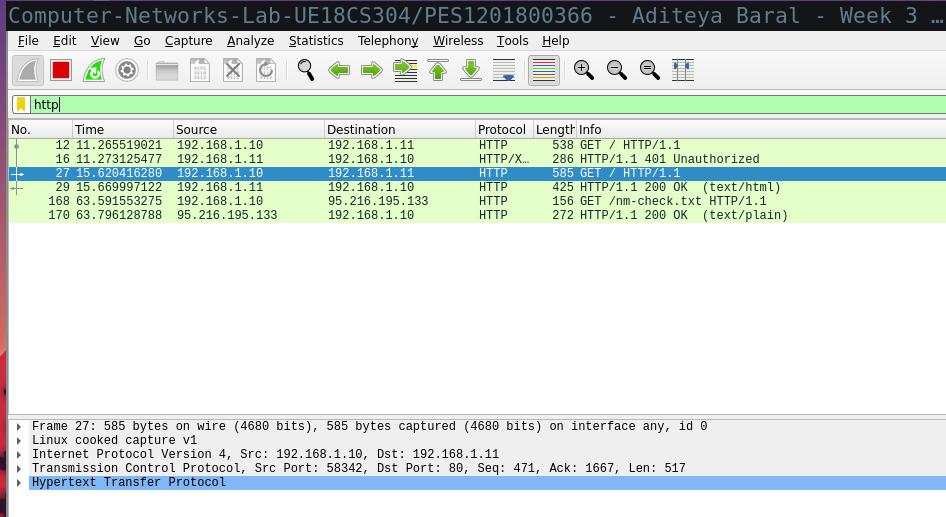
* Create an .htaccess file in the server route with the following contents :

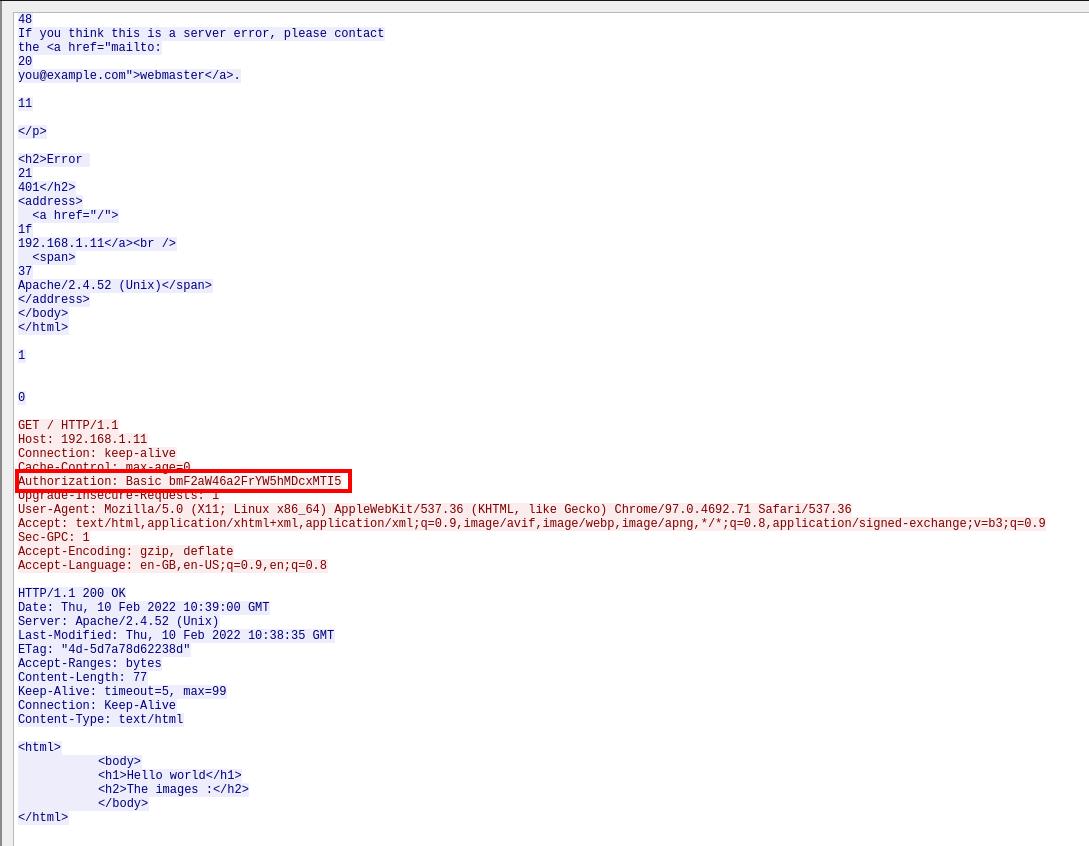


* Opening the webpage should lead to prompt :



* Analyzing the packets in wireshark :





* Analyzing base64 Auth cipher text :

As the Authneitcation follows a base 64 addressing, each letter in cipher is of 6 bits and which can later be coverted to ascii of 8 bits each, this goves us :

011011 100110 000101 110110 011010 010110 111000 111010 011011 100110 000101 110110 011010 010110 111000 110001 001100 100011 001100 110100

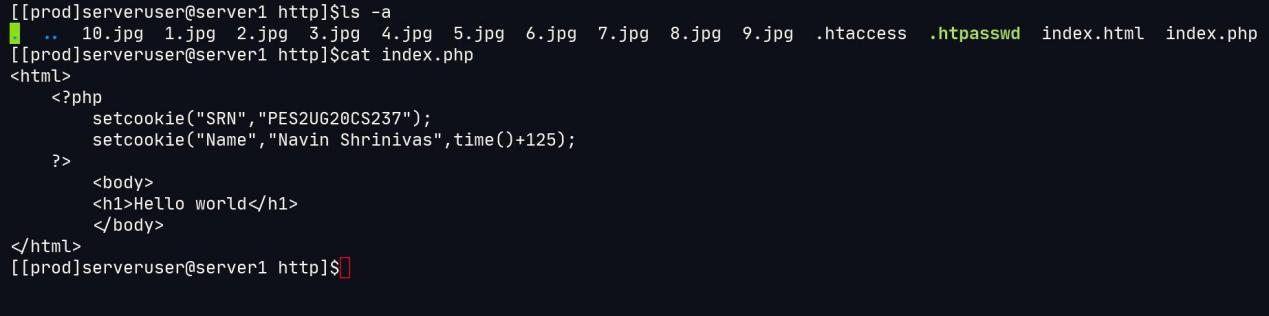
rearraging to ascii 8 bits :

01101110 01100001 01110110 01101001 01101110 00111010 01101110 01100001 01110110 01101001 01101110 00110001 00110010 00110011 00110100

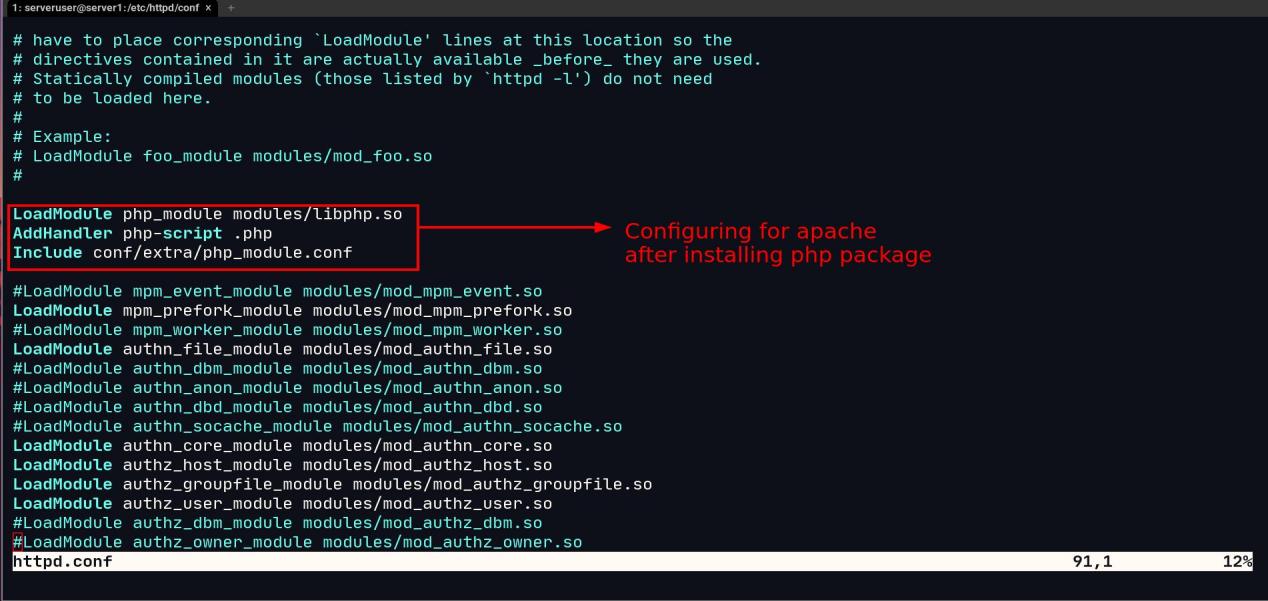
This leads to : navin:navin1234, whcih was indeed my username:password

## **Task 5 : Cookies using PHP**

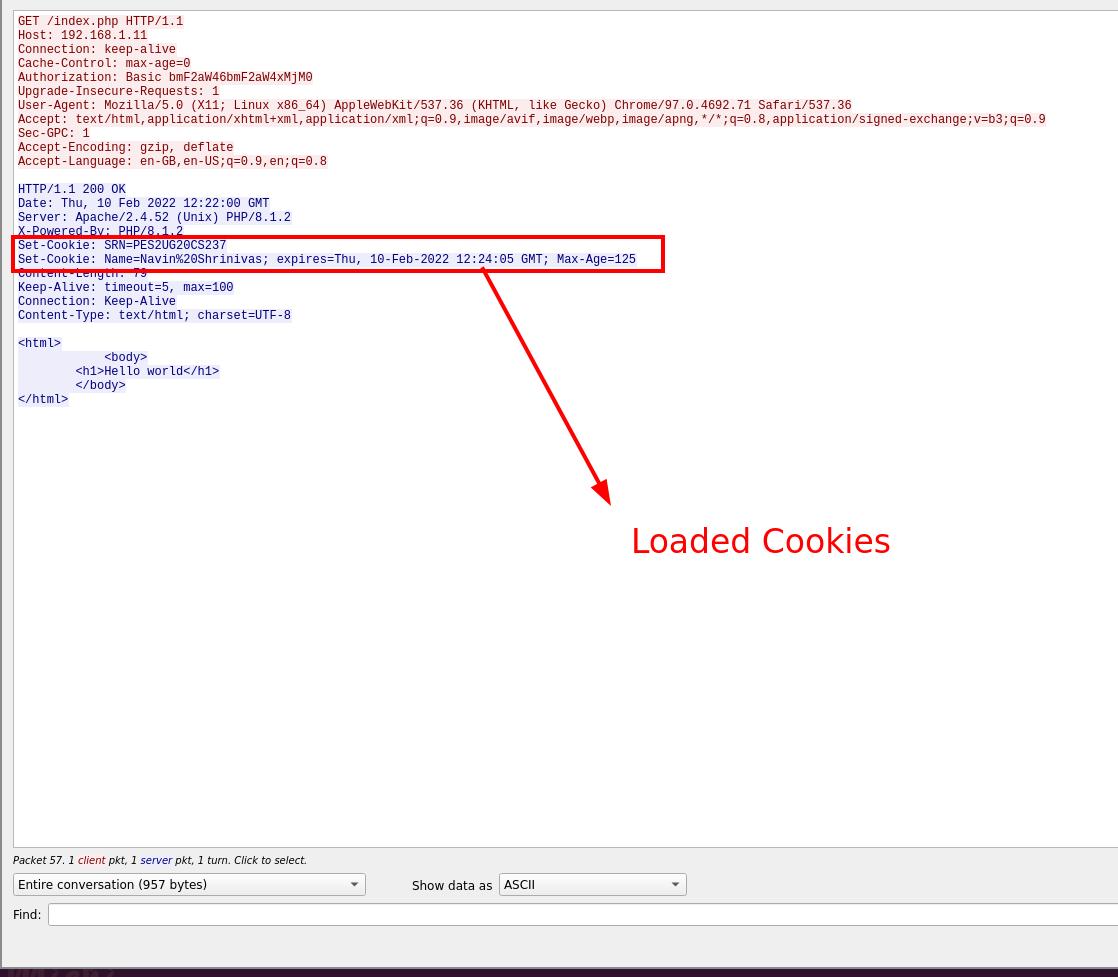
* For this a new index.php file is created in the server root, which has html content with php embedded, like so :



* PHP is enabled in the apache server by editing the conf file, like so :

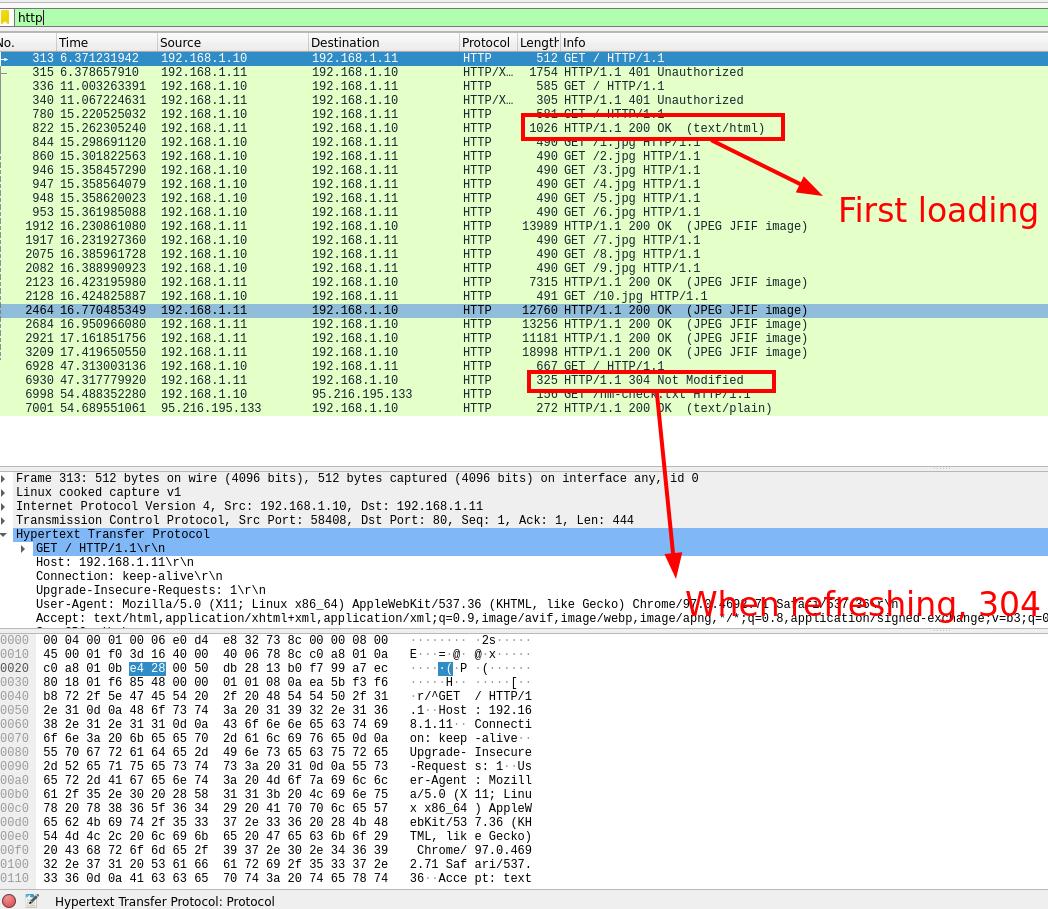


* The PHP file is then accessed from the client and wireshark packets are analysed, the **set-cookie** can be seen :

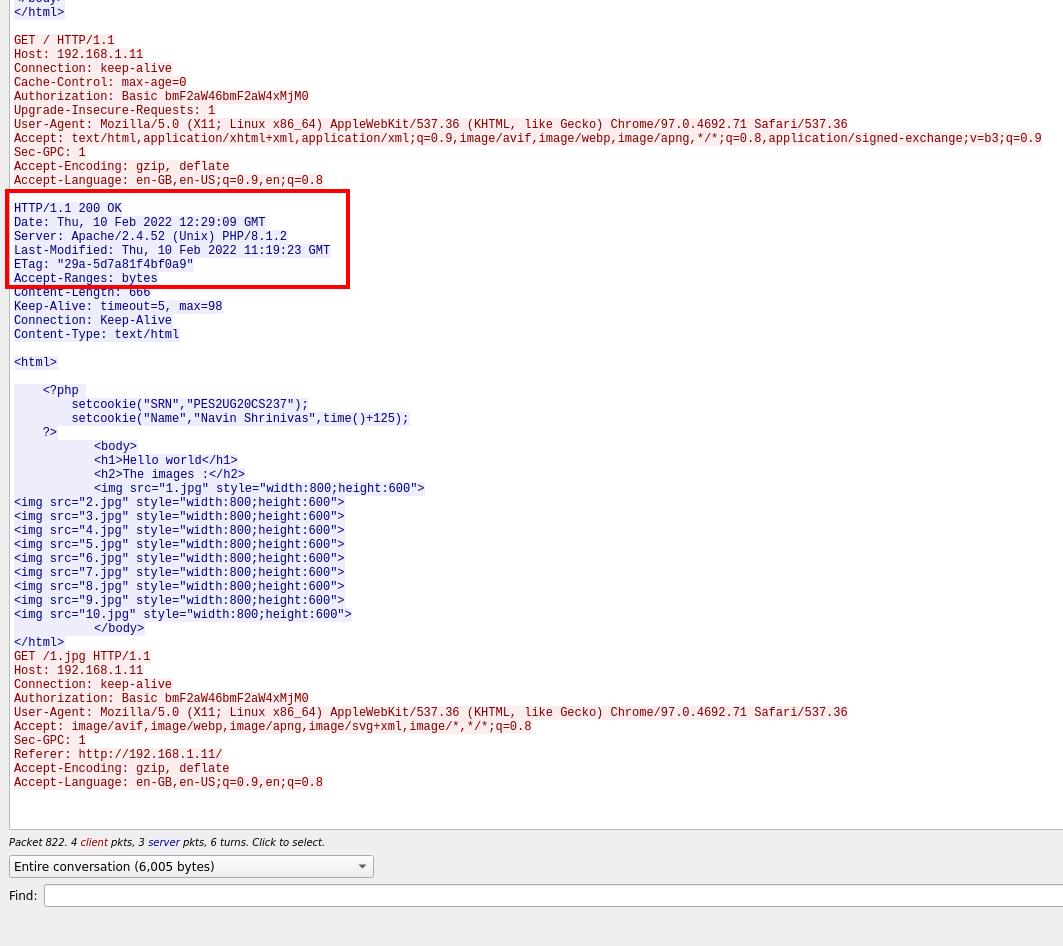


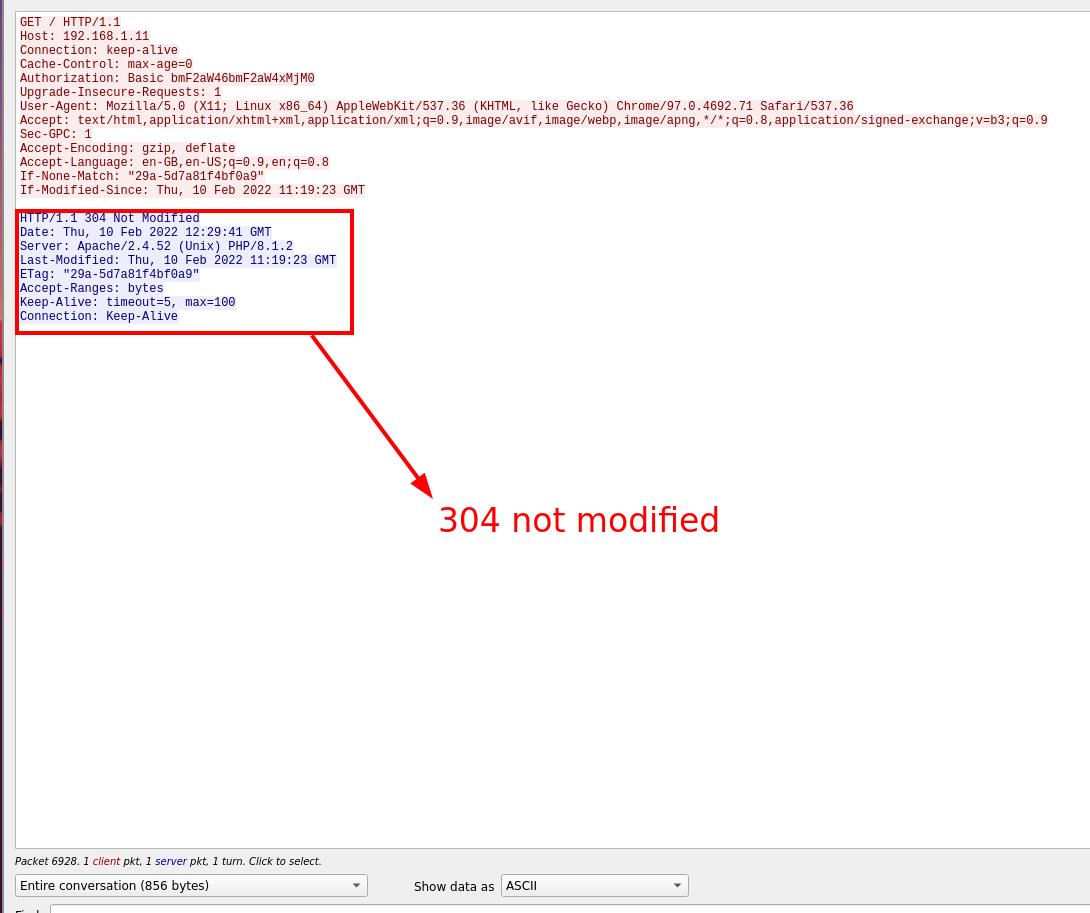
## **Task 6 : Conditional get**

* A conditional HTTP request that carries data only if data has been modified. Apache servers are configured by default to do this
* Code 304 is for not modified, 200 is when the data is actually fetched successfully.
* Below is a demonstration of Conditional get using refreshing pages and wireshark :



Wire shark overlook

 200 status on first load after clearing cache

 304 not modified status after refreshing