

Lab #3

Sriram R

SRN : PES1UG20CS435

Section : H

(Note : Answers on pages 10-16 & 25-28)

1. Understanding Persistent and Non-persistent HTTP Connections

To understand persistent and non-persistent HTTP connections and corresponding performance impact.

Create a web page with N (e.g. 10) embedded images. Each image should be of minimum 2 MB size. Configure your browser (Firefox) with following settings (each setting requires repeat of experiment)

- Non persistent connection
- 2 persistent connections
- 4 persistent connections
- 6 persistent connections
- 10 persistent connections.

Observation: Note down the time taken to display the entire page in each of the settings. Ensure that (cache is cleared before starting the web request). Explain the response time differences. What is the optimal number of persistent connections for best performance? Explain your answer.

Introduction

The Apache HTTP server is the most widely used web server in the world. It provides many powerful features including dynamically loadable modules, robust media support, and extensive integration with other popular software.

Objective: Understand persistent and non-persistent HTTP connections and corresponding performance impact.

Experiment: Create a web page with N (e.g. 10) embedded images. Each image should be of minimum 2 MB size. Configure your browser (Firefox) with following settings (each setting requires repeat of experiment)

- a) Non-persistent connection
- b) 2 persistent connections
- c) 4 persistent connections
- d) 6 persistent connections
- e) 10 persistent connections

Note down the time taken to display the entire page in each of the settings. **Ensure that cache is cleared before starting the web request.** Explain the response time differences. What is the optimal number of persistent connections for best performance? Explain your answer.

Note: To install Apache server, use the following command,

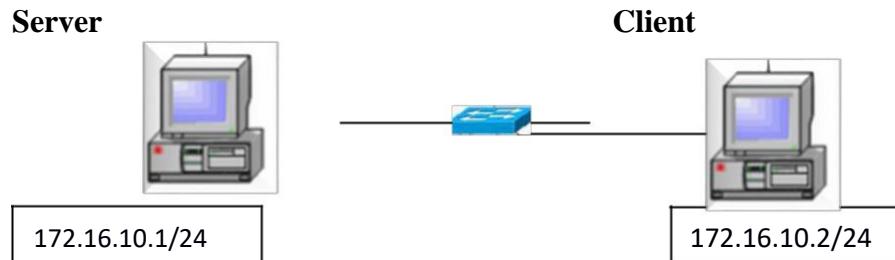
```
sudo apt-get install apache2
```

If there is any error during installation, update the package manager by issuing the command,

```
sudo apt-get update
```

EXECUTION STEPS

Step 1: Connect 2 desktops using switch and cables as shown below. (Use 2 VMs on Virtualbox or VMware instead of physical connections.)



Server Side:

Step 2: Check your Web Server

At the end of the installation process, Ubuntu 16.04 starts Apache. The web server should already be up and running. We can check with the `systemctl` command to make sure the service is running by typing:

`sudo systemctl status apache2`

or

`sudo service apache2 status`

```
system:~ netlab@system:~$ sudo systemctl status apache2
apache2.service - LSB: Apache2 web server
  Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
  Active: active (running) since Tue 2017-06-20 10:44:34 IST; 9min ago
    Docs: man:systemd-sysv-generator(8)
  Group: /system.slice/apache2.service
         ├─5548 /usr/sbin/apache2 -k start
         ├─5551 /usr/sbin/apache2 -k start
         ├─5552 /usr/sbin/apache2 -k start
         └─5553 /usr/sbin/apache2 -k start

Jun 20 10:44:32 system systemd[1]: Starting LSB: Apache2 web server...
Jun 20 10:44:32 system apache2[5525]: * Starting Apache httpd web server apache2
Jun 20 10:44:33 system apache2[5525]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1.
Jun 20 10:44:34 system apache2[5525]: *
Jun 20 10:44:34 system systemd[1]: Started LSB: Apache2 web server.
[lines 1-16/16 (END)]
```

As you can see above, the service appears to have started successfully. However, the best way to test this is to actually request a page from Apache. You can access the default Apache landing page to confirm that the software is running properly. You can access this through your server's domain name or IP address.

Step 3: Server IP address can be set by the following command

```
$sudo ip addr add 172.16.10.1/24 dev enps0
$sudo ip addr
```

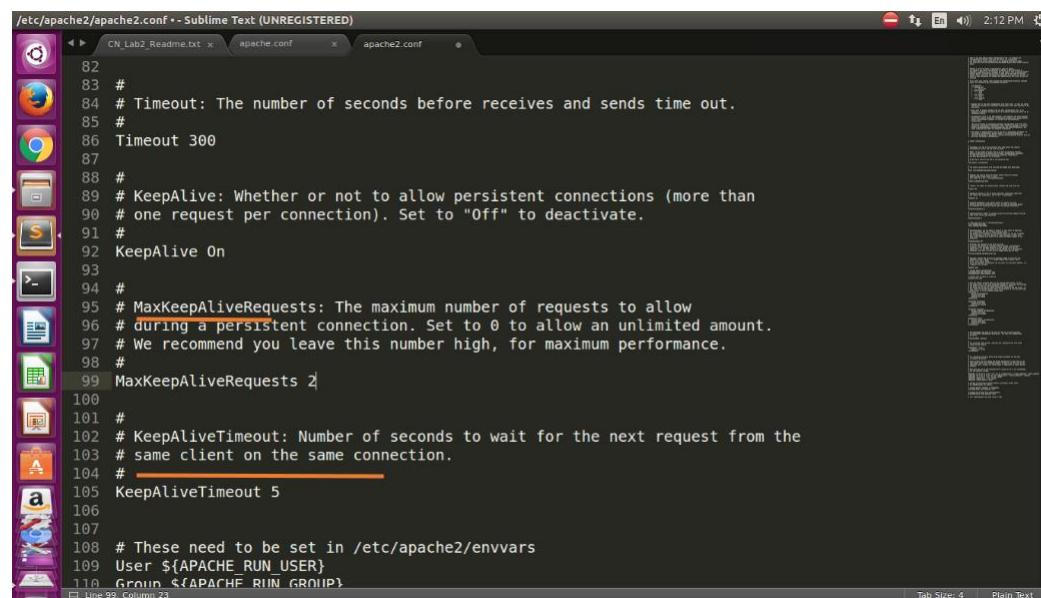
Note: If IP address fluctuates, kindly setup the IP address manually using ‘Edit connections’.

```
student@student-H81H3-I:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether b8:ae:ed:a5:a5:a9 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.1/24 brd 172.16.10.255 scope global enp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::c901:c994:4cf5:f837/64 scope link
        valid_lft forever preferred_lft forever
student@student-H81H3-I:~$
```

Step 4: The **apache2.conf** file present in the **/etc/apache2** directory is modified as:

- The **keep-alive** option was set (i.e. value was made **ON**)
- The **MaximumKeepAliveRequests** were set to

2 \$sudo nano /etc/apache2/apache2.conf

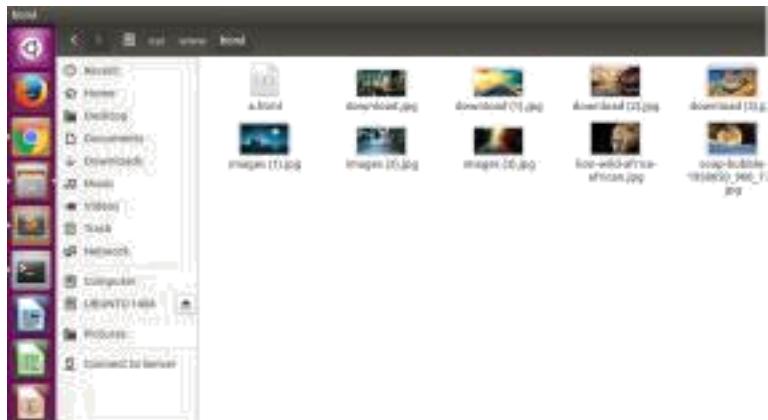


A screenshot of the Sublime Text editor showing the `/etc/apache2/apache2.conf` file. The file contains configuration directives for Apache 2.0. Key modifications shown are:

```
82 #
83 #
84 # Timeout: The number of seconds before receives and sends time out.
85 #
86 Timeout 300
87 #
88 #
89 # KeepAlive: Whether or not to allow persistent connections (more than
90 # one request per connection). Set to "Off" to deactivate.
91 #
92 KeepAlive On
93 #
94 #
95 # MaxKeepAliveRequests: The maximum number of requests to allow
96 # during a persistent connection. Set to 0 to allow an unlimited amount.
97 # We recommend you leave this number high, for maximum performance.
98 #
99 MaxKeepAliveRequests 2
100 #
101 #
102 # KeepAliveTimeout: Number of seconds to wait for the next request from the
103 # same client on the same connection.
104 #
105 KeepAliveTimeout 5
106 #
107 #
108 # These need to be set in /etc/apache2/envvars
109 User ${APACHE_RUN_USER}
110 Group ${APACHE_RUN_GROUP}
```

Step 5: Store images in the server path. A html page consisting of 10 images having size > 2MB were placed and accessed by the client. This html page is stored in the location - **/var/www/html/file_name.html**.

Note: Use the images provided by faculty incharges.



Step 6: Prepare a web page as shown below. The html file needs to add 10 images. (Kindly skip the style attribute in the below image)

```
a.html [Read-Only] (/var/www/html) - gedit
Open ▾ F
<!DOCTYPE html>
<html>
<body>
<h2>Spectacular Mountain</h2>










</body>
</html>
```

Client side:

Client IP address can be set by the following command.

```
$sudo ip addr add 172.16.10.2/24 dev enps0
$ sudo ip addr
```

Note: If IP address fluctuates, kindly setup the IP address manually using ‘Edit connections’.

```
student@student-H81H3-I:~$ sudo ip addr add 172.16.10.2/24 dev enp2s0
student@student-H81H3-I:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    group default qlen 1000
    link/ether b8:ae:ed:a5:a6:32 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.2/24 scope global enp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::8bf0:837a:849e:a79f/64 scope link
        valid_lft forever preferred_lft forever
student@student-H81H3-I:~$
```

There are broadly two parts of execution:

1. Dealing with non-persistent connections
2. Dealing with persistent connections

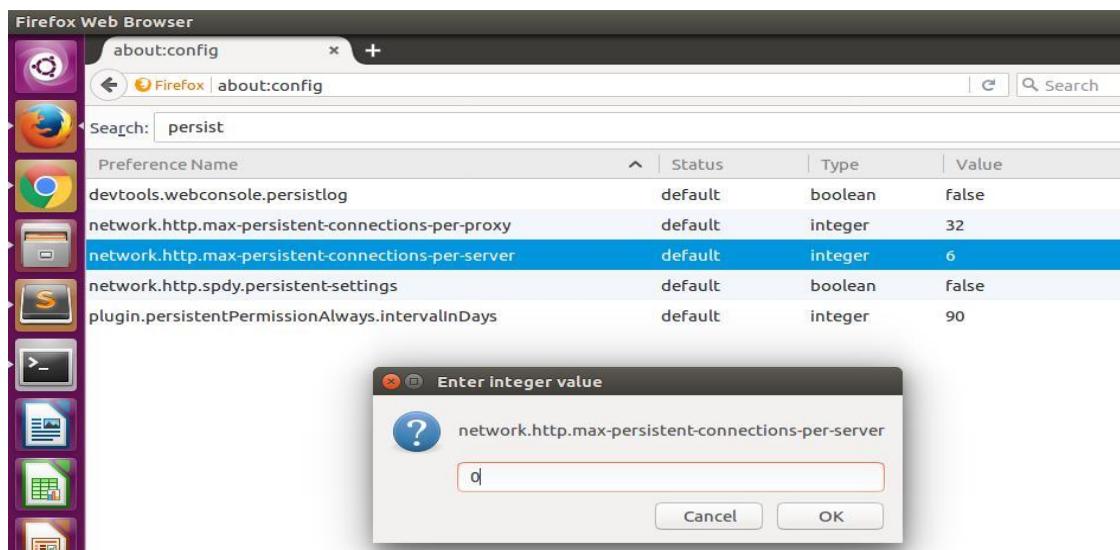
Open Firefox browser to configure for persistent option. Go to browser and type

about:config and search for the term '**persistent**'

- While using non-persistent connection experiment, the **max-persistent-connections-per-server** has the value set to **0** and **persistent-settings** value set to false.
- While using persistent connection experiment, the **max-persistent-connections-per-server** should have value greater than 0 (depending on the number of persistent connections needed) and **persistent-settings** value set to true.

PART 1: NON-PERSISTENT CONNECTION

Step 1: This is done by setting the value of max-persistent-connection-per-server to 0 in the client computer.



Step 2: Access web page on client-side browser (Firefox)

The client could access the file as:

172.16.10.1/file_name.html where--> **172.16.10.1** is Server's IP

Here the file name is **a.html** present in server. So, by tying **172.16.10.1/a.html** in client browser, we will be able to open the requested web page.

Note 1: The wireshark should capture the packets between the client and the server while the file is accessed.

Note 2: The images in the HTML page should have all the permissions specified through the server for the proper access.

Step 3: Use wireshark. Open wireshark in the server computer while client is trying to access the server's local host webpage. Apply 'http' filter and note the time to capture all the 10 images.

No.	Time	Source	Destination	Protocol	Length	Info
25	0.211530105	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
27	2.070581279	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
28	2.070866155	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
30	2.117160769	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
35	2.117571913	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG JFIF image)
36	2.117753115	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
45	2.117944288	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
51	2.118574057	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
63	2.119058490	172.16.10.1	172.16.10.2	HTTP	242	HTTP/1.1 200 OK (JPEG JFIF image)
65	2.119487932	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
77	2.119784374	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
79	2.120323770	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
94	2.121263792	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
110	2.122045168	172.16.10.1	172.16.10.2	HTTP	1226	HTTP/1.1 200 OK (JPEG JFIF image)
117	2.122719543	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
138	2.123847115	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
160	2.124700199	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_960_720.jpg HTTP/1.1
164	2.124733805	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG JFIF image)
171	2.125125151	172.16.10.1	172.16.10.2	HTTP	711	HTTP/1.1 200 OK (JPEG JFIF image)
184	2.126599573	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
252	2.131056667	172.16.10.1	172.16.10.2	HTTP	114	HTTP/1.1 200 OK (JPEG JFIF image)
529	2.151487483	172.16.10.1	172.16.10.2	HTTP	73	HTTP/1.1 200 OK (JPEG JFIF image)
3834	2.429637133	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $2.429637133 - 2.070581279 = 0.359055854$

PART 2: PERSISTENT CONNECTIONS

Step 1: For 2 persistent connections, set the value of **max-persistent-connection-per-server** **to 2** in the client computer.

Step 2: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
28	0.158495832	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
30	2.685888334	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
31	2.686488793	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
33	2.734091058	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
38	2.734592637	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
39	2.734696958	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG JFIF image)
48	2.735025557	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
49	2.735180365	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
66	2.736079156	172.16.10.1	172.16.10.2	HTTP	243	HTTP/1.1 200 OK (JPEG JFIF image)
68	2.736374643	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
82	2.736755733	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
85	2.737381832	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
92	2.737840608	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
101	2.738335480	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
119	2.738809142	172.16.10.1	172.16.10.2	HTTP	1226	HTTP/1.1 200 OK (JPEG JFIF image)
121	2.739075438	172.16.10.1	172.16.10.2	HTTP	1016	HTTP/1.1 200 OK (JPEG JFIF image)
139	2.740900738	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
143	2.741014891	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_960_720.jpg HTTP/1.1
148	2.741205777	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
179	2.742807473	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
190	2.743723330	172.16.10.1	172.16.10.2	HTTP	712	HTTP/1.1 200 OK (JPEG JFIF image)
402	2.764054977	172.16.10.1	172.16.10.2	HTTP	72	HTTP/1.1 200 OK (JPEG JFIF image)
3774	3.042252027	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $3.042252027 - 2.685888334 = 0.356363$

Step 3: For 4 persistent connections, Set the value of **max-persistent-connection-per-server to 4** in the client computer.

Step 4: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
28	0.152642908	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
30	1.667969551	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
31	1.668311781	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
33	1.699473631	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
35	1.6999692009	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
45	1.699908042	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
46	1.699913003	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG JFIF image)
47	1.700012712	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
63	1.7009001747	172.16.10.1	172.16.10.2	HTTP	242	HTTP/1.1 200 OK (JPEG JFIF image)
69	1.701341018	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
70	1.701432635	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
86	1.701888908	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
93	1.702192885	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
95	1.702219175	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
97	1.702228220	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
98	1.702233130	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_960_720.jpg HTTP/1.1
122	1.703328136	172.16.10.1	172.16.10.2	HTTP	711	HTTP/1.1 200 OK (JPEG JFIF image)
126	1.703773424	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
157	1.705498971	172.16.10.1	172.16.10.2	HTTP	1227	HTTP/1.1 200 OK (JPEG JFIF image)
159	1.705614894	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
167	1.706637782	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG JFIF image)
414	1.724541388	172.16.10.1	172.16.10.2	HTTP	73	HTTP/1.1 200 OK (JPEG JFIF image)
3825	2.005934395	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here is it $2.005934395 - 1.667969557 = 0.337964838$

Step 5: For 6 persistent connections, set the value of **max-persistent-connection-per-server to 6** in the server computer.

Step 6: Repeat the **steps 1-3** in the previous section.

No.	Time	Source	Destination	Protocol	Length	Info
21	0.100232302	172.16.10.2	172.16.10.1	HTTP	306	GET /favicon.ico HTTP/1.1
22	0.100476138	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
24	0.184514911	172.16.10.2	172.16.10.1	HTTP	366	GET /favicon.ico HTTP/1.1
25	0.184789474	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
27	3.915242469	172.16.10.2	172.16.10.1	HTTP	421	GET /.html HTTP/1.1
28	3.915930950	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
30	3.934519286	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
31	3.934703623	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
44	3.935084209	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG JFIF image)
45	3.935091751	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
50	3.935485109	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
68	3.936344013	172.16.10.1	172.16.10.2	HTTP	243	HTTP/1.1 200 OK (JPEG JFIF image)
74	3.936634551	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
75	3.936649737	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
76	3.936654620	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
78	3.936684823	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
80	3.936696984	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
122	3.937371850	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_960_720.jpg HTTP/1.1
123	3.937539442	172.16.10.1	172.16.10.2	HTTP	1227	HTTP/1.1 200 OK (JPEG JFIF image)
160	3.939256627	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG JFIF image)
167	3.940125154	172.16.10.1	172.16.10.2	HTTP	712	HTTP/1.1 200 OK (JPEG JFIF image)
183	3.941778538	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
229	3.946434434	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
233	3.946891865	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
441	3.964535410	172.16.10.1	172.16.10.2	HTTP	72	HTTP/1.1 200 OK (JPEG JFIF image)
3771	4.241013689	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $4.241013689 - 3.915242469 = 0.325771229$

Step 7: For 10 persistent connections, set the value of **max-persistent-connection-per-server to 10** in the client computer.

Step 8: Repeat the **steps 1-3** in the previous section.

No.	Time	Source	Destination	Protocol	Length	Info
27	0.192665375	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
29	1.556964626	172.16.10.2	172.16.10.1	HTTP	421	GET /.html HTTP/1.1
30	1.557214715	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
32	1.575716934	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
33	1.575953704	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
46	1.576334520	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG JFIF image)
47	1.576343533	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
52	1.576760416	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
70	1.577515601	172.16.10.1	172.16.10.2	HTTP	243	HTTP/1.1 200 OK (JPEG JFIF image)
76	1.577834686	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
77	1.577847379	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
78	1.577855269	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
80	1.577886802	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
82	1.577905312	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
118	1.578606528	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_960_720.jpg HTTP/1.1
119	1.578639337	172.16.10.1	172.16.10.2	HTTP	1227	HTTP/1.1 200 OK (JPEG JFIF image)
146	1.580341669	172.16.10.1	172.16.10.2	HTTP	712	HTTP/1.1 200 OK (JPEG JFIF image)
169	1.582240704	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
187	1.583749770	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG JFIF image)
219	1.586862673	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
222	1.587108849	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
455	1.606226568	172.16.10.1	172.16.10.2	HTTP	72	HTTP/1.1 200 OK (JPEG JFIF image)
3814	1.882459413	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $1.882459413 - 1.556964626 = 0.325494787$

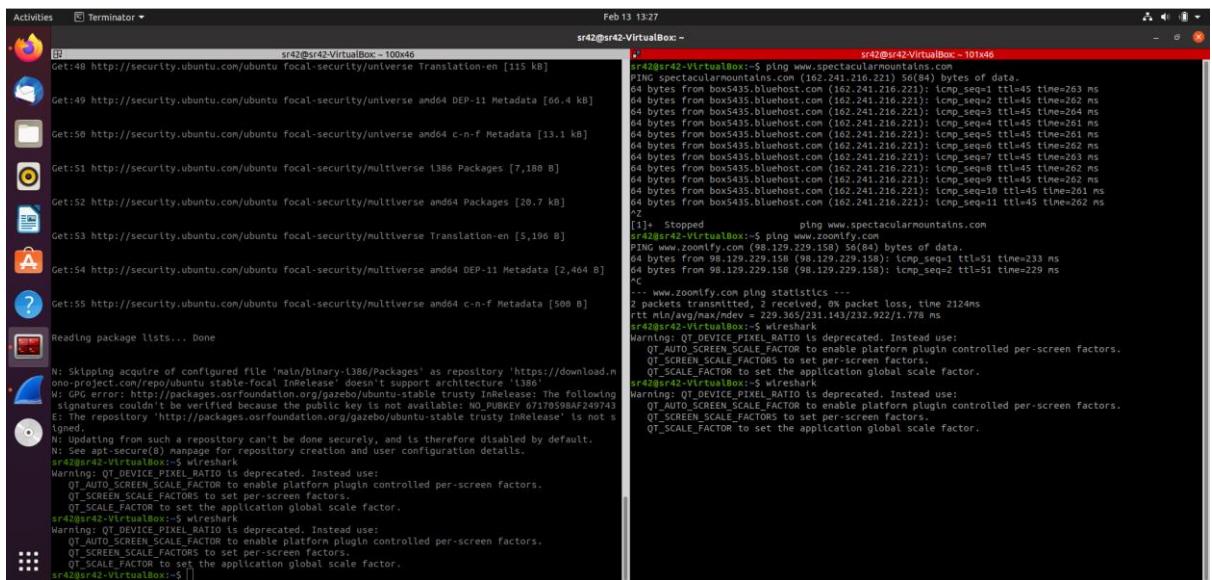
OBSERVATIONS REQUIRED ON EDMODO:

Calculate the time taken to load objects from the server for non-persistent and persistent connections (2, 4, 6, 8 & 10). Find out the optimal number of HTTP persistent connections based on your observations.

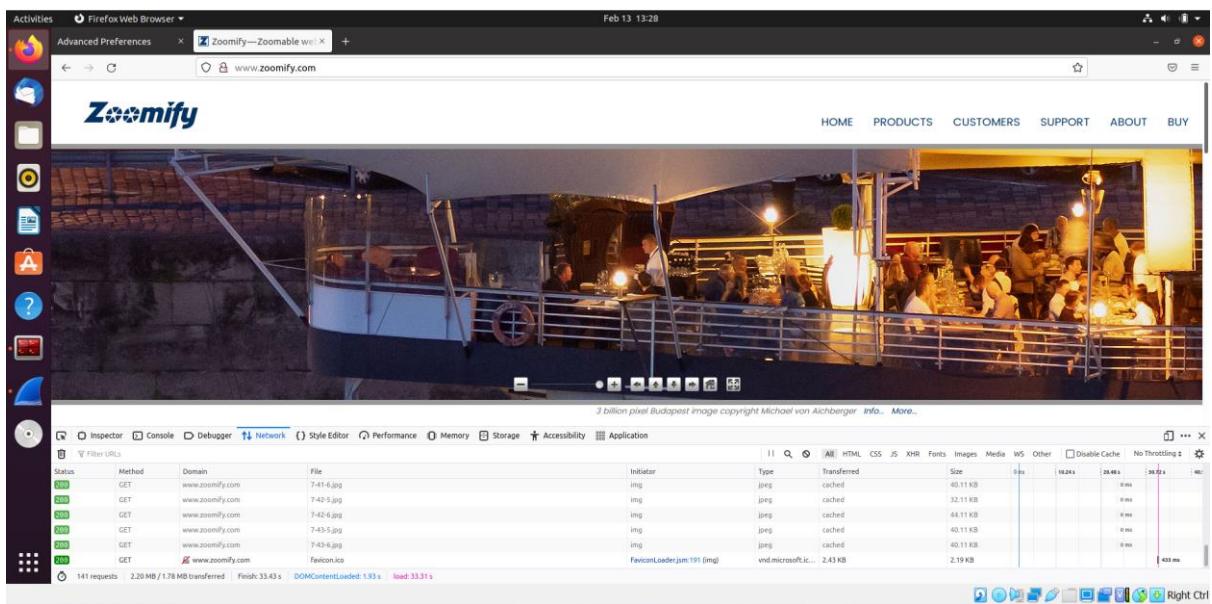
SCREENSHOTS REQUIRED FOR EDMODO:

- 1) Non-persistent connection wireshark capture (should include all 10 images)
- 2) Persistent connections wireshark capture – 2, 4, 6, 8 & 10 respectively (should include all 10 images).

Initialization :



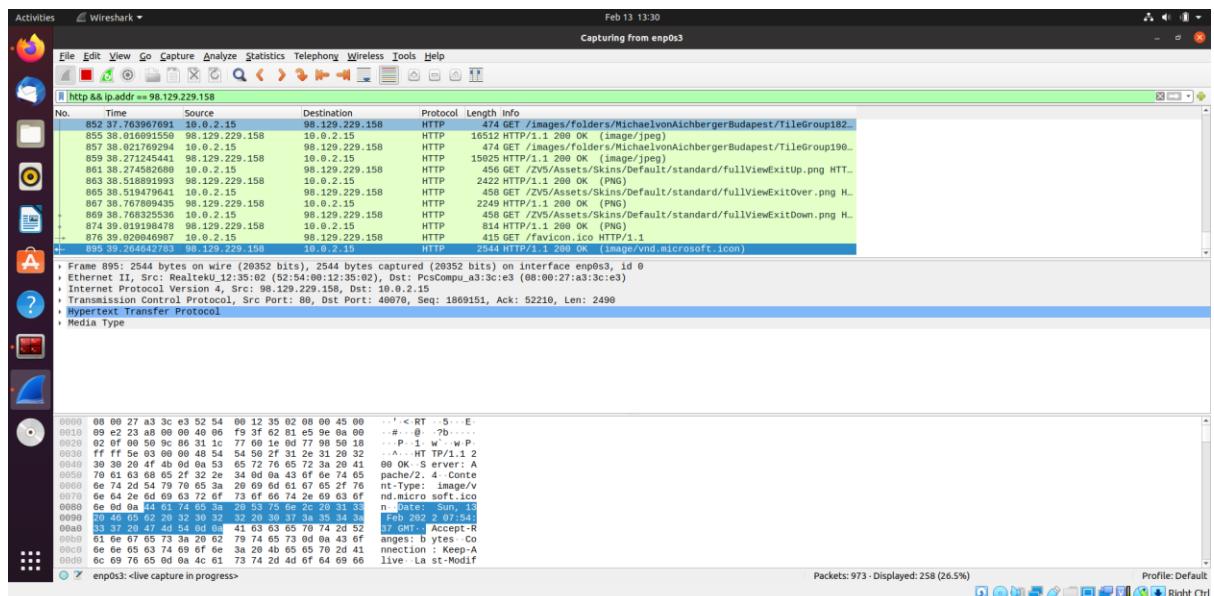
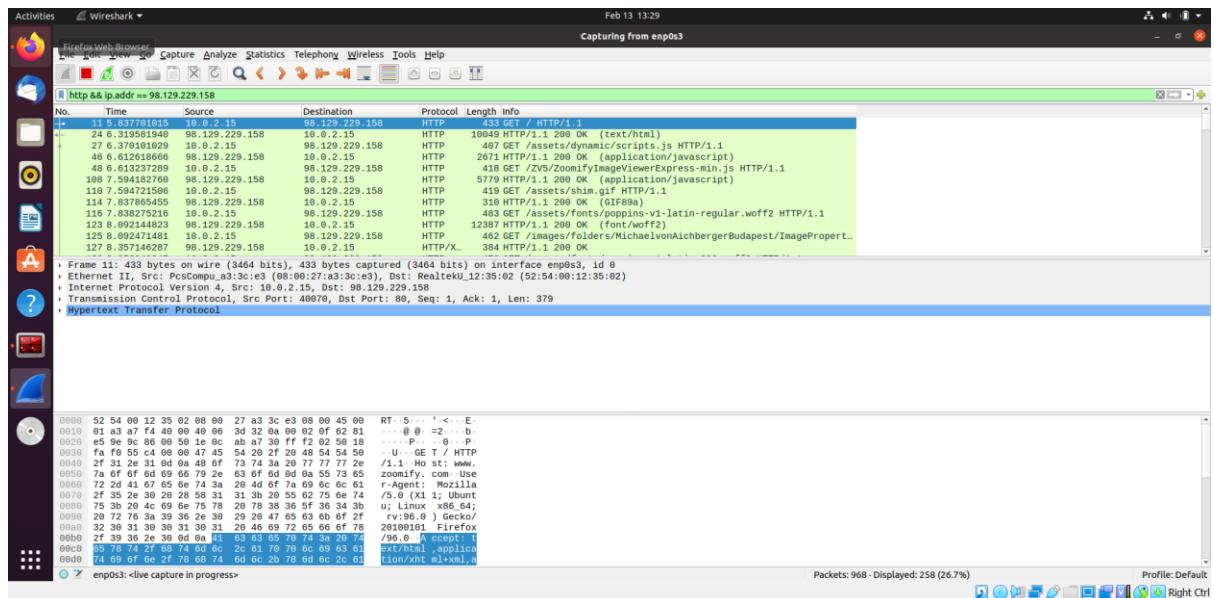
```
Activities Terminator Feb 13 13:27
sr42@sr42-VirtualBox ~ 100x46
Get:48 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [115 kB]
Get:49 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [66.4 kB]
Get:50 http://security.ubuntu.com/ubuntu focal-security/universe and04 c-n-f Metadata [13.1 kB]
Get:51 http://security.ubuntu.com/ubuntu focal-security/multiverse Packages [7,180 B]
Get:52 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [20.7 kB]
Get:53 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5,196 B]
Get:54 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:55 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [500 B]
Reading package lists... done
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository 'https://download.mono-project.com/repo/ubuntu stable-focal InRelease' doesn't support architecture 'i386'
W: GPG error: http://packages.osrfoundation.org/gazebo/ubuntu-stable trusty InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 6717059BAF249743
E: The repository 'http://packages.osrfoundation.org/gazebo/ubuntu-stable trusty InRelease' is not signed.
Ign:56 http://packages.osrfoundation.org/gazebo/ubuntu-stable trusty InRelease
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
sr42@sr42-VirtualBox:~$ ping www.spectacularmountains.com
PING www.spectacularmountains.com (162.241.216.221) 56(84) bytes of data.
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=1 ttl=45 time=263 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=2 ttl=45 time=262 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=3 ttl=45 time=264 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=4 ttl=45 time=261 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=5 ttl=45 time=262 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=6 ttl=45 time=263 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=7 ttl=45 time=263 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=8 ttl=45 time=262 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=9 ttl=45 time=262 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=10 ttl=45 time=261 ms
64 bytes from box5435.bluehost.com (162.241.216.221): icmp_seq=11 ttl=45 time=262 ms
...
[1]: Stopped ping www.spectacularmountains.com
sr42@sr42-VirtualBox:~$ ping www.zoomify.com
PING www.zoomify.com (98.129.229.158) 56(84) bytes of data.
64 bytes from 98.129.229.158 (98.129.229.158): icmp_seq=1 ttl=51 time=233 ms
64 bytes from 98.129.229.158 (98.129.229.158): icmp_seq=2 ttl=51 time=229 ms
...
```
-- www.zoomify.com ping statistics --
2 packets transmitted, 2 received, 0% packet loss, time 2124ms
rtt min/avg/max = 231.143/232.922/1.778 ms
sr42@sr42-VirtualBox:~$ wireshark
Warning: QT_DEVICE_PIXEL_RATIO is deprecated. Instead use:
QT_SCREEN_SCALE_FACTOR to enable platform plugin controlled per-screen factors.
QT_SCREEN_SCALE_FACTORS to set per-screen factors.
QT_SCALE_FACTOR to set the application global scale factor.
sr42@sr42-VirtualBox:~$ wireshark
Warning: QT_DEVICE_PIXEL_RATIO is deprecated. Instead use:
QT_AUTO_SCREEN_SCALE_FACTOR to enable platform plugin controlled per-screen factors.
QT_SCREEN_SCALE_FACTORS to set per-screen factors.
QT_SCALE_FACTOR to set the application global scale factor.
QT_SCALE_FACTOR to set the application global scale factor.
```



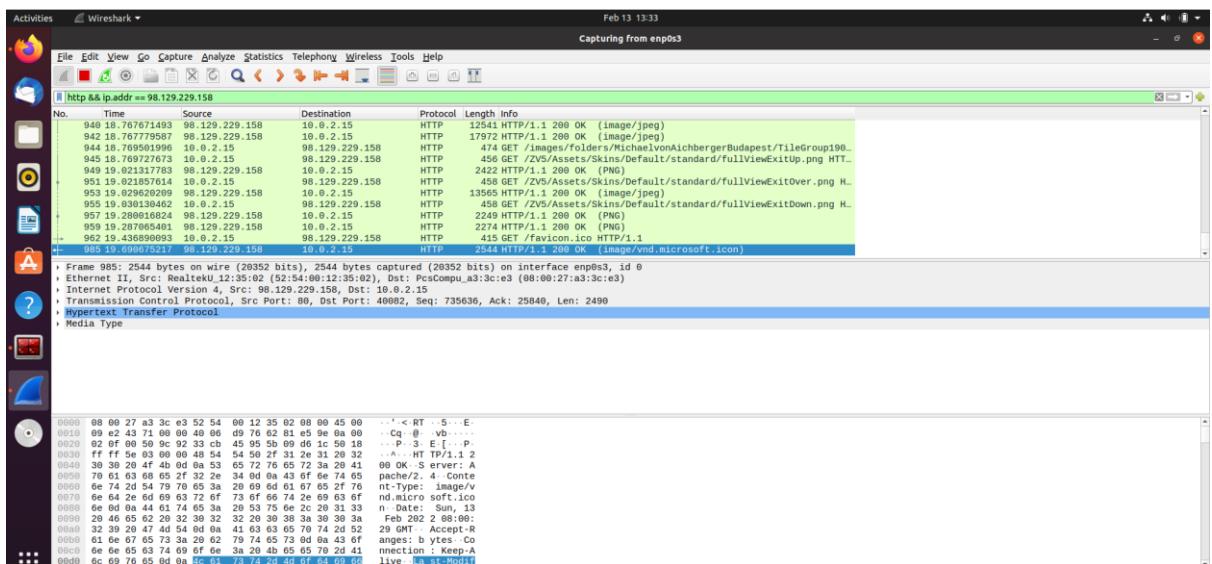
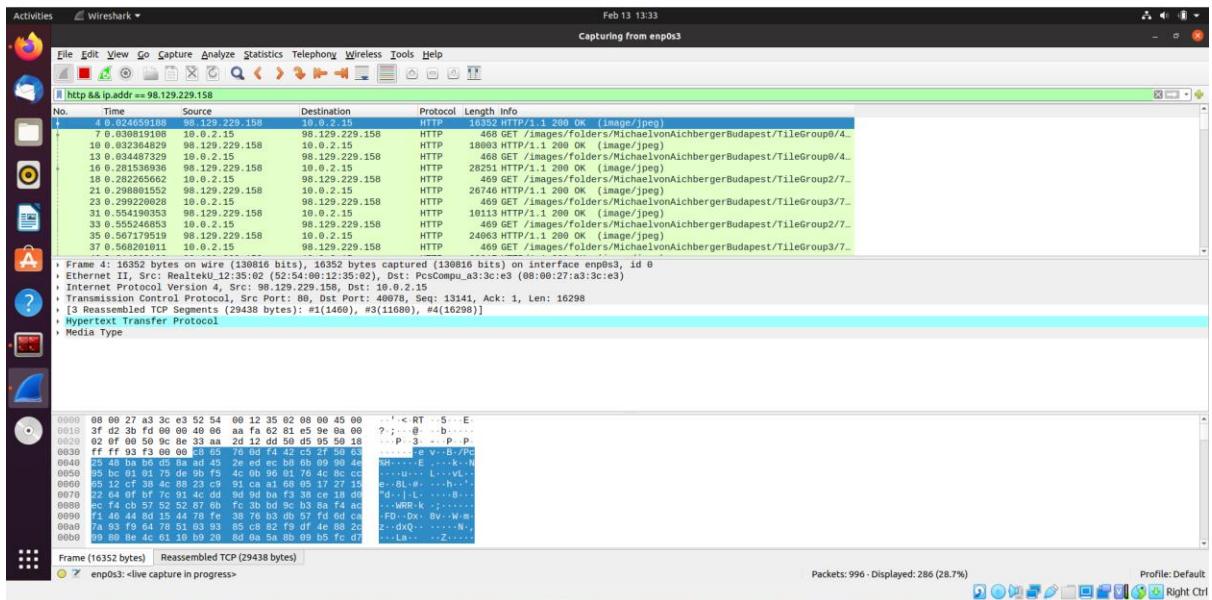
The screenshot shows a Linux desktop environment with a terminal window and a Firefox browser window. The terminal window displays a ping session to www.spectacularmountains.com. The Firefox browser window shows the Zoomify website, which features a large, zoomable image of a night scene at a restaurant. Below the browser, a NetworkMiner tool is running, showing a list of network requests and their details, including file types like jpg and gif, and sizes ranging from 2.19 KB to 40.11 KB.

## Observation notes :

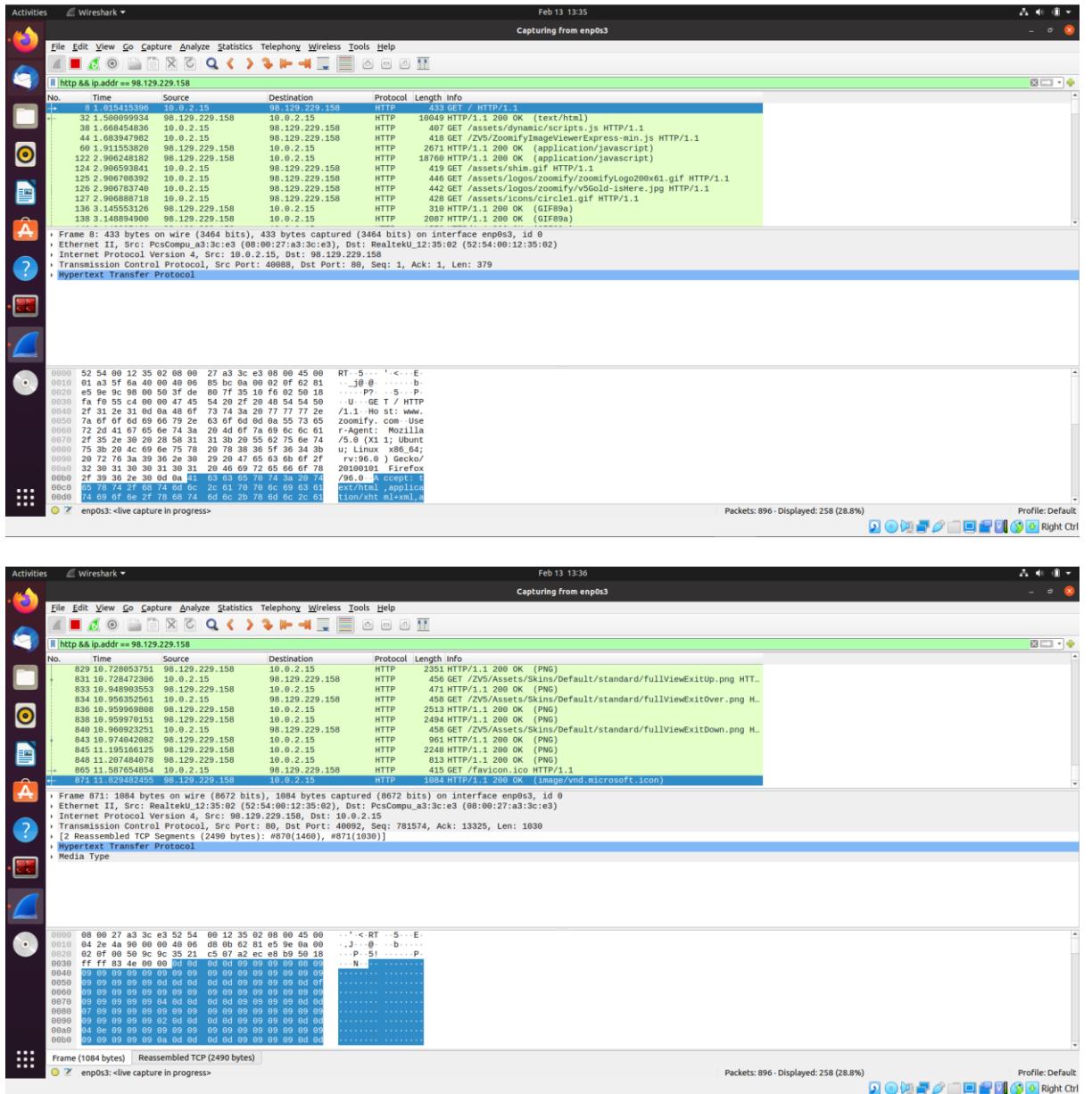
- Non persistent connection time for connection = last HTTP response time - first GET request transmission time =  $39.26 - 5.84 = 33.42\text{s}$



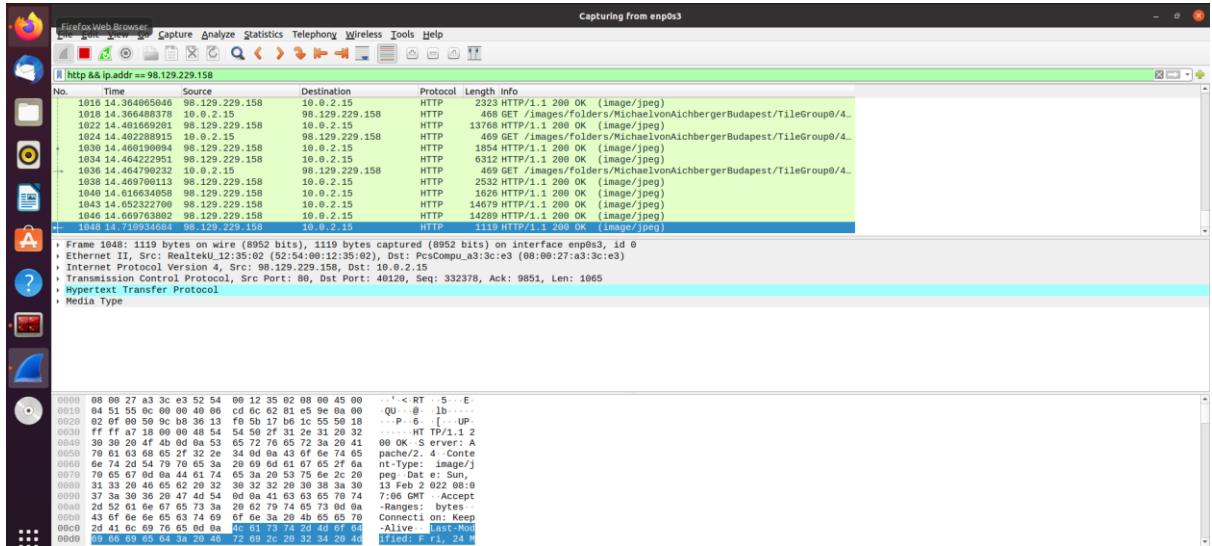
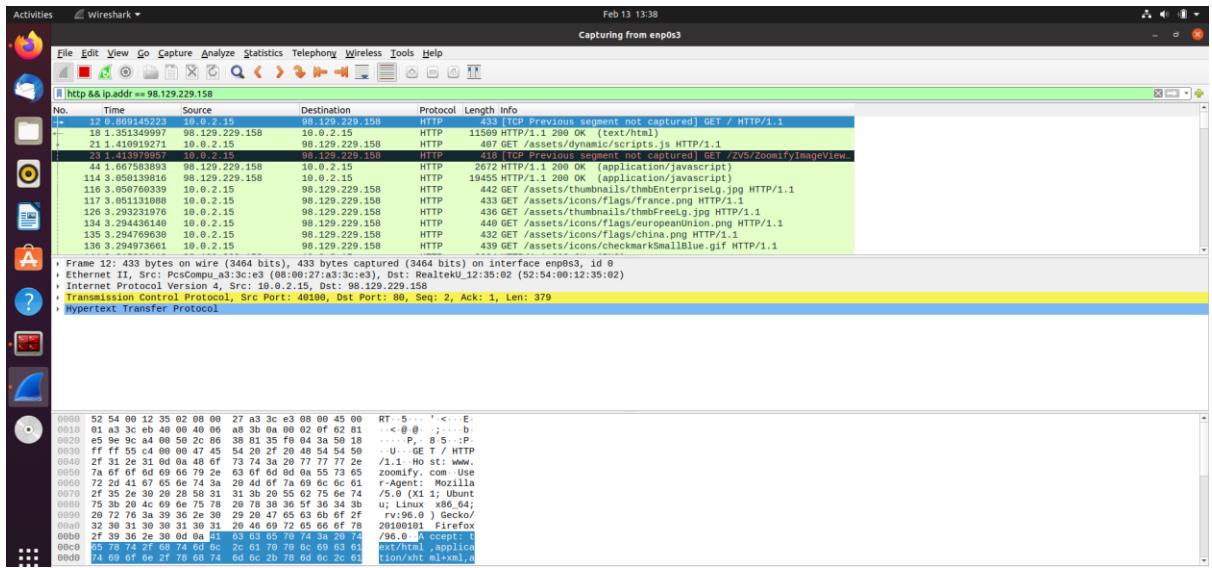
- Persistent connection time (2 connections) for connection = last HTTP response time - first GET request transmission time =  $19.69 - 0.02 = 19.67\text{s}$



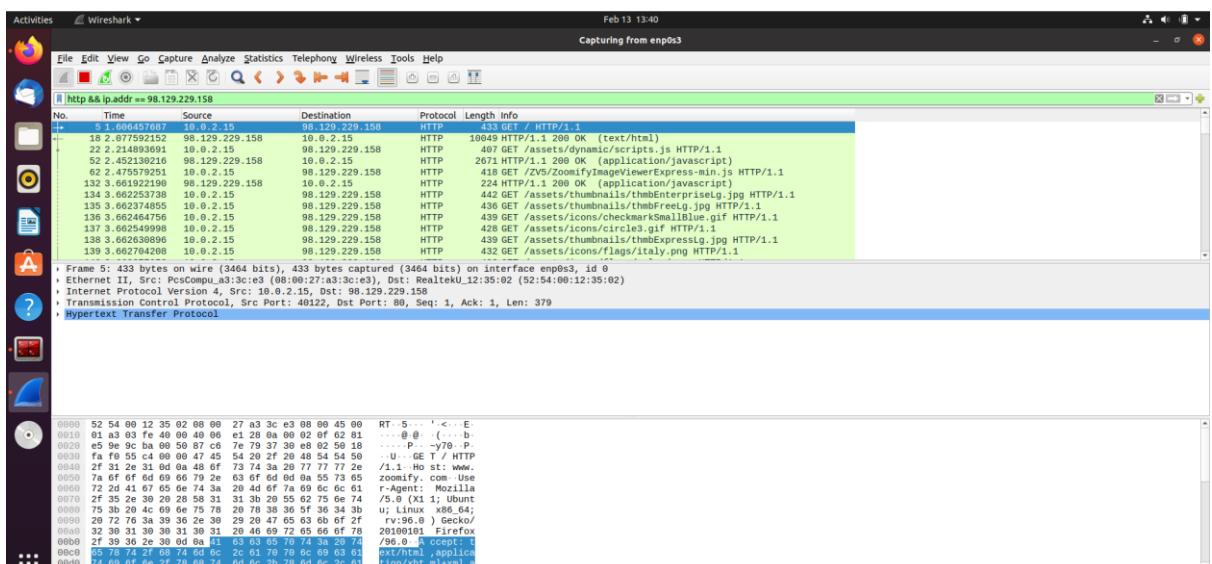
- Persistent connection time (4 connections) for connection = last HTTP response time - first GET request transmission time =  $11.83 - 1.01 = 10.82$ s

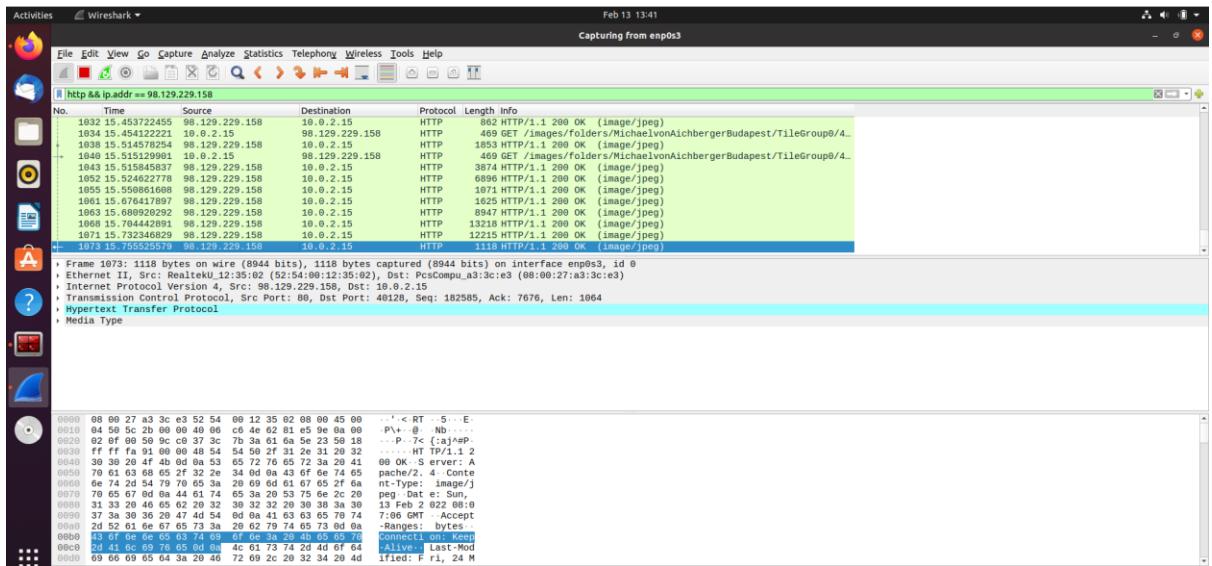


- Persistent connection time (6 connections) for connection = last HTTP response time - first GET request transmission time =  $14.71 - 0.87 = 13.84s$

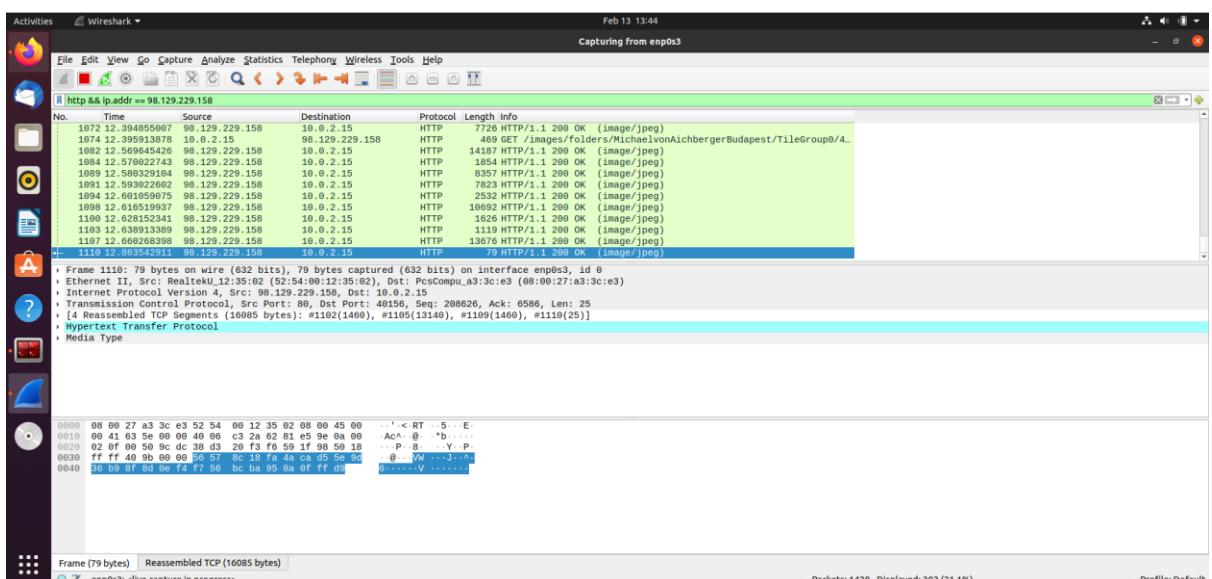
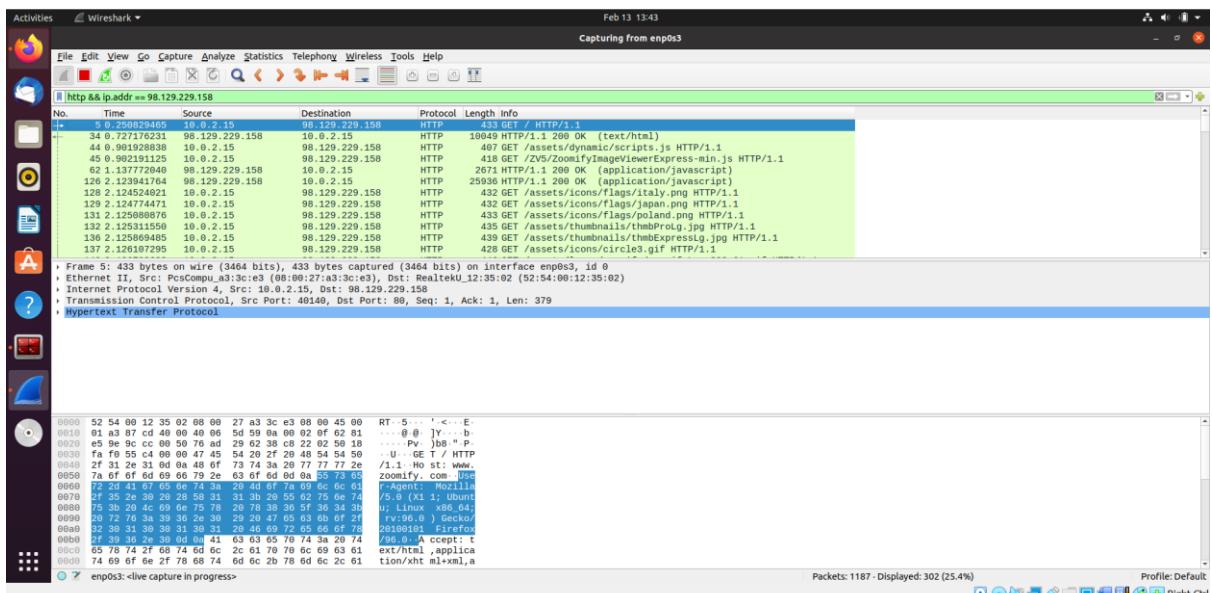


- Persistent connection time (8 connections) for connection = last HTTP response time - first GET request transmission time =  $15.75 - 1.61 = 14.14$ s





- Persistent connection time (10 connections) for connection = last HTTP response time - first GET request transmission time =  $12.86 - 0.25 = 12.41$ s



Conclusion :

Therefore, the best load time for this particular site was obtained with 4 persistent connections.

---

## 2. Understand working of HTTP Headers

### **Understand working of HTTP headers:**

Conditional Get: If-Modified-Since

HTTP Cookies: Cookie and Set-Cookie

Authentication: Auth-Basic

Design a web page that has one embedded page (e.g. image) and sets a cookie and enables authentication. You are required to configure the web server (e.g. apache) with authentication mechanism.

Show the behavior of conditional get when embedded objects is modified and when it is not (you can just change the create date of the embedded object). Decode the Basic-Auth header using Base64 mechanism as per the password setup.

**Observation:** Show the behavior of browser when is cookie is set and when cookie is removed.

# Lab: 3

## Understanding Working of HTTP Headers

**Question:** Understand working of HTTP headers

Conditional Get: If-Modified-Since

HTTP Cookies: Cookie and Set-Cookie

Authentication: Auth-Basic

Design a web page that has one embedded page (e.g. image) and sets a cookie and enables authentication. You are required to configure the web server (e.g. apache) with authentication mechanism. Show the behavior of conditional get when embedded objects are modified and when it is not (you can just change the create date of the embedded object). Decode the Basic-Auth header using Base64 mechanism as per the password setup.

**Observation:** Show the behavior of browser when is cookie is set and when cookie is removed.

**Solution:** Analyzing Basic Authentication and Cookies

The three parts of experiment are:

1. Password Authentication
2. Cookie Setting
3. Conditional get

### **Steps of Execution (for Password Authentication)**

1. Executing the below commands on the terminal.

--> To update and integrate the existing  
softwares **sudo apt-get update**

--> To install the apache utility  
**sudo apt-get install apache2 apache2-utils**

```
osboxes@osboxes:~$ sudo apt-get install apache2-utils
[sudo] password for osboxes:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
 apache2-utils
1 upgraded, 0 newly installed, 0 to remove and 247 not upgraded.
Need to get 81.7 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-utils amd64 2.4.18-2ubuntu3.10 [81.7 kB]
Fetched 81.7 kB in 1s (45.5 kB/s)
(Reading database ... 217540 files and directories currently installed.)
Preparing to unpack .../apache2-utils_2.4.18-2ubuntu3.10_amd64.deb ...
Unpacking apache2-utils (2.4.18-2ubuntu3.10) over (2.4.18-2ubuntu3.9) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up apache2-utils (2.4.18-2ubuntu3.10) ...
osboxes@osboxes:~$
```

--> Provide username and password to set authentication

```
sudo htpasswd -c /etc/apache2/.htpasswd ANY_USERNAME
```

```
osboxes@osboxes:~$ sudo htpasswd -c /etc/apache2/.htpasswd netwo
New password:
Re-type new password:
Adding password for user netwo
osboxes@osboxes:~$ sudo cat /etc/apache2/.htpasswd
netwo:$apr1$6YdDa0Ti$ELrUaOlQ/jun9TTU1PYKu/
osboxes@osboxes:~$
```

Here “netwo” is the username. Also, password is entered twice.

--> View the authentication

```
sudo cat /etc/apache2/.htpasswd
```

2. To setup the authentication phase, execute the following commands. Configuring Access control within the Virtual Host Definition.

--> Opening the file for setting authentication

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost*:80>
 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/html
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log
 combined <Directory "/var/www/html">
 AuthType Basic
 AuthName "RESTRICTED"
 AuthUserFile /etc/apache2/.htpasswd
 Require valid-user
 </Directory>
</VirtualHost>
```

```
GNU nano 2.5.3 File: /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>

 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/html

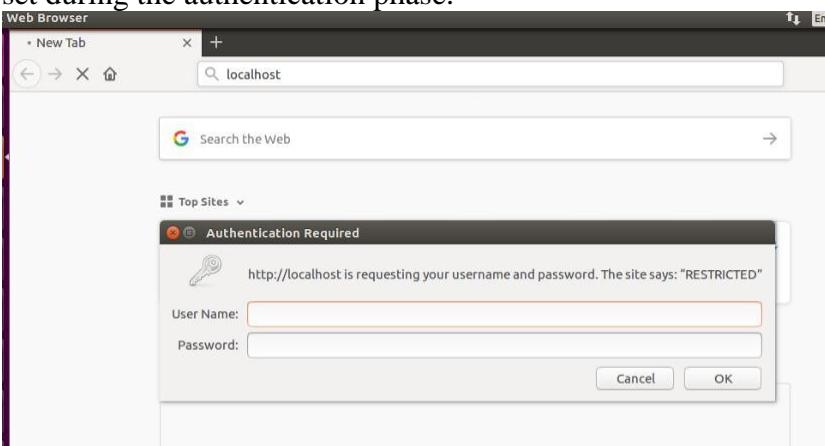
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined

 <Directory "/var/www/html">
 AuthType Basic
 AuthName "RESTRICTED"
 AuthUserFile /etc/apache2/.htpasswd
 Require valid-user >
 </Directory>
</VirtualHost>
```

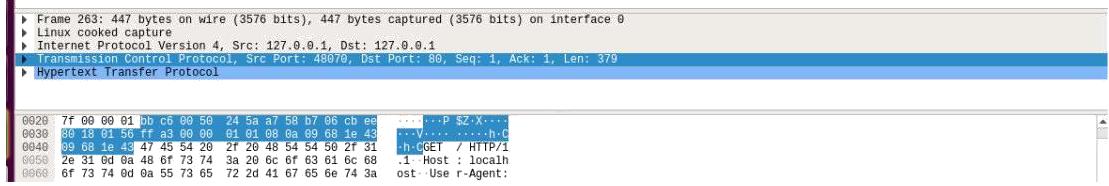
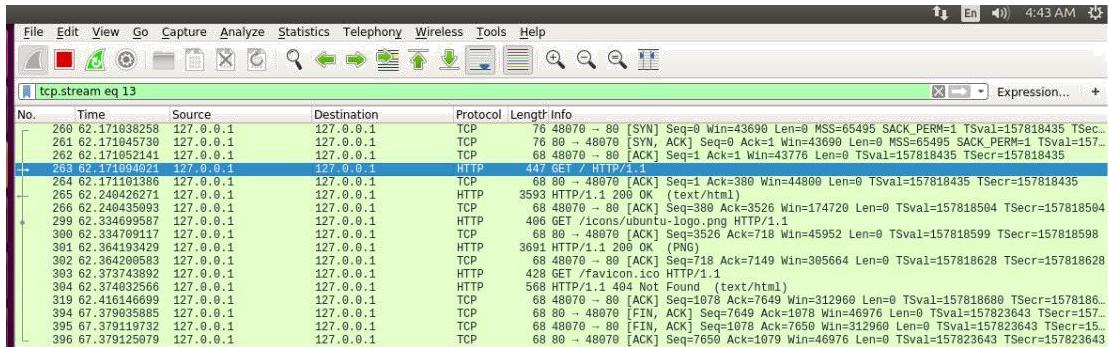
3. Password policy implementation is done by restarting the server  
as: **sudo service apache2 restart**

```
osboxes@osboxes:~$ sudo service apache2 restart
osboxes@osboxes:~$
```

4. The localhost is then accessed using the Firefox browser requiring a username and a password set during the authentication phase.



5. Wireshark is used to capture the packets sent upon the network.



6. Using the “follow TCP stream” on the HTTP message segment the password was retrieved which was encrypted by the base64 algorithm and decryption could be done with same algorithm.

```
ark · Follow TCP Stream (tcp.stream eq 13) · any

GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic bmV0d286bmV0d28xMjM=

HTTP/1.1 200 OK
Date: Mon, 29 Jul 2019 08:38:04 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Thu, 21 Mar 2019 05:57:12 GMT
ETag: "2c39-5849468d1860-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3186
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

.....Z.s.6.....U.$'....".&c.....!".....P.....].....MN.G".....}HG.....^M...!K..
[...p.....@E8}>".....In.....2.M..<X..+..2.s.....2.j0.7B..1.s....+b..`......./8.'s..//,1...jC.
[px2.....d4..[+f..`e}].a.L.....]..{4!*.DK.43..Lo....=%..../S.....L.rG..1.z..
F..1".{d.bN/z.R.%z..{a.e..hL.....y.h}....7.....}u.).,.8.g4zX(%..B.1..zt....?....s.b;{
2.L..*..#..q7Q..|.D.e.
)C..V..|.T:..5|..TP.n.L0..f9&..B.g..L.#Wc.b..`s&43.r..rY.....^R.L
<.....dx.....h)x.....0..)o.%s...ry9....t0.....u
[...w%..|..N.F.....r.E.i.....|[#.ZW....{.....
}!.....0..A.....a..600..y....y.l..H.Pyi....J.....09....5.....h....{.00...MwsJ...8.
7.....a.A.....w.....F7.W.N1~..^..%.o>{.UE.WP+..'|.....m..b....o.0Gq....$.da...[.h....]
```

## Steps of Execution (Cookie Setting)

1. A PHP file to set the cookie is created which also contains an image in it (placed under the HTML directory) to be accessed once the cookie is set. The following code helped to set the cookie:

```
<html>
<?php
```

```

setcookie("namecookie","netqwerty",time()+123);
setcookie("nickname","work");
?>

</html>

```

GNU nano 2.5.3                                  File: abc.php

```

<html>
<?php
setcookie("namecookie","netqwerty",time()+123);
setcookie("nickname","work");
?>

</html>

```

Note: Here you can add any image if required

**Note: You can capture Cookies mostly during the first time of web access. Hence keep wireshark capture ready before executing the task for the first time.**

2. The combined file saved with a .php extension is placed under /var/www/html for accessing.



3. The packets are captured using Wireshark and using the “follow TCP stream” which checks for the set-cookie field whether the cookie is set or not set.

No.	Time	Source	Destination	Protocol	Length	Info
309	23.481763156	127.0.0.1	127.0.0.1	TCP	76	48404 → 80 [SYN] Seq=0 Win=43690 Len=0 MSS=65495
310	23.481770358	127.0.0.1	127.0.0.1	TCP	76	80 → 48404 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0
311	23.481776479	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSV
312	23.481809760	127.0.0.1	127.0.0.1	HTTP	454	GET /abc.php HTTP/1.1
313	23.481816171	127.0.0.1	127.0.0.1	TCP	68	80 → 48404 [ACK] Seq=1 Ack=387 Win=44800 Len=0
314	23.482261152	127.0.0.1	127.0.0.1	HTTP	524	HTTP/1.1 200 OK (text/html)
315	23.482299412	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=387 Ack=457 Win=44800 Len=0
327	23.581439939	127.0.0.1	127.0.0.1	HTTP	448	GET /highres.jpg HTTP/1.1
328	23.581712586	127.0.0.1	127.0.0.1	TCP	22468	80 → 48404 [ACK] Seq=457 Ack=767 Win=45952 Len=2
329	23.581718698	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=22857 Win=175744 L
330	23.581735122	127.0.0.1	127.0.0.1	TCP	65551	80 → 48404 [PSH, ACK] Seq=22857 Ack=767 Win=4595
331	23.581740280	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=88340 Win=306816 L
332	23.581756133	127.0.0.1	127.0.0.1	TCP	65551	80 → 48404 [ACK] Seq=88340 Ack=767 Win=45952 Len
333	23.581764076	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=153823 Win=437760 L
334	23.581780285	127.0.0.1	127.0.0.1	TCP	65551	80 → 48404 [ACK] Seq=153823 Ack=767 Win=45952 L
335	23.581785729	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=219396 Win=419840 L
336	23.581801828	127.0.0.1	127.0.0.1	TCP	65551	80 → 48404 [ACK] Seq=219396 Ack=767 Win=45952 L
337	23.581806654	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=284789 Win=386560 L
338	23.581820136	127.0.0.1	127.0.0.1	HTTP	61937	HTTP/1.1 200 OK (JPEG JFIF image)
339	23.581824816	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=767 Ack=346658 Win=355200 L
341	23.607036646	127.0.0.1	127.0.0.1	HTTP	473	GET /Favicon.ico HTTP/1.1
342	23.607286489	127.0.0.1	127.0.0.1	HTTP	568	HTTP/1.1 404 Not Found (text/html)
342	23.641216059	127.0.0.1	127.0.0.1	TCP	68	48404 → 80 [ACK] Seq=1172 Ack=347158 Win=568704

Frame 312: 454 bytes on wire (3632 bits), 454 bytes captured (3632 bits) on interface 0  
 ▶ Linux cooked capture  
 ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1  
 ▶ Transmission Control Protocol, Src Port: 48404, Dst Port: 80, Seq: 1, Ack: 1, Len: 386  
 ▶ Hypertext Transfer Protocol

```
Wireshark - Follow TCP Stream (tcp.stream eq 11) · any

GET /abc.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic bmV0d286bmV0d28xMjM=

HTTP/1.1 200 OK
Date: Mon, 29 Jul 2019 09:10:23 GMT
Server: Apache/2.4.18 (Ubuntu)
Set-Cookie: namecookie=netqwerty; expires=Mon, 29-Jul-2019 09:12:26 GMT; Max-Age=123
Set-Cookie: nickname=work
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 92
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

.....(.....MW(.J.U..L.(J..-*HWR(.L)..U260PR.H..@9%.%9..J.....E)J.@#.!Fq.... .S...GET /highres.jpg HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: image/webp,*/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/abc.php
DNT: 1
Authorization: Basic bmV0d286bmV0d28xMjM=
Connection: keep-alive
Cookie: namecookie=netqwerty; nickname=work
```

The cookie is set as shown in the above screenshot.

**Observation:** Understand and work out base 64 algorithm and write in your observation.  
 Observe various parameters associated with Cookie in the wireshark capture.

### Conditional Get: If-Modified-Since

Before performing the steps below, make sure your browser's cache is empty. (To do this under Firefox, select Tools -> Clear Recent History and check the Cache box). Now do the following:

- Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
- Start up the Wireshark packet sniffer.
- Enter the following URL into your browser  
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>
- Your browser should display a very simple five-line HTML file.
- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
- Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

### **Observations:**

- ✓ Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?
- ✓ Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
- ✓ Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?
- ✓ What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

**Repeat the above task with some images on the server.**

**Attach screenshots wherever necessary.**

---

## Screenshots & observations :

- Password authentication :
  - o Terminal initialization :

The screenshot shows a terminal window titled "sr42@sr42-VirtualBox ~" with the command "apt update" running. It lists numerous package updates from the "In.archive.ubuntu.com/ubuntu" repository, including "focal-security" and "focal-updates" InRelease files. The output includes several warning messages about GPG keys and repository signatures. After the update, the user runs "sudo apt-get install apache2 apache2-utils". The Apache2 configuration file is edited to set the server name to "www.example.com". Finally, the service is restarted with "sudo service apache2 restart".

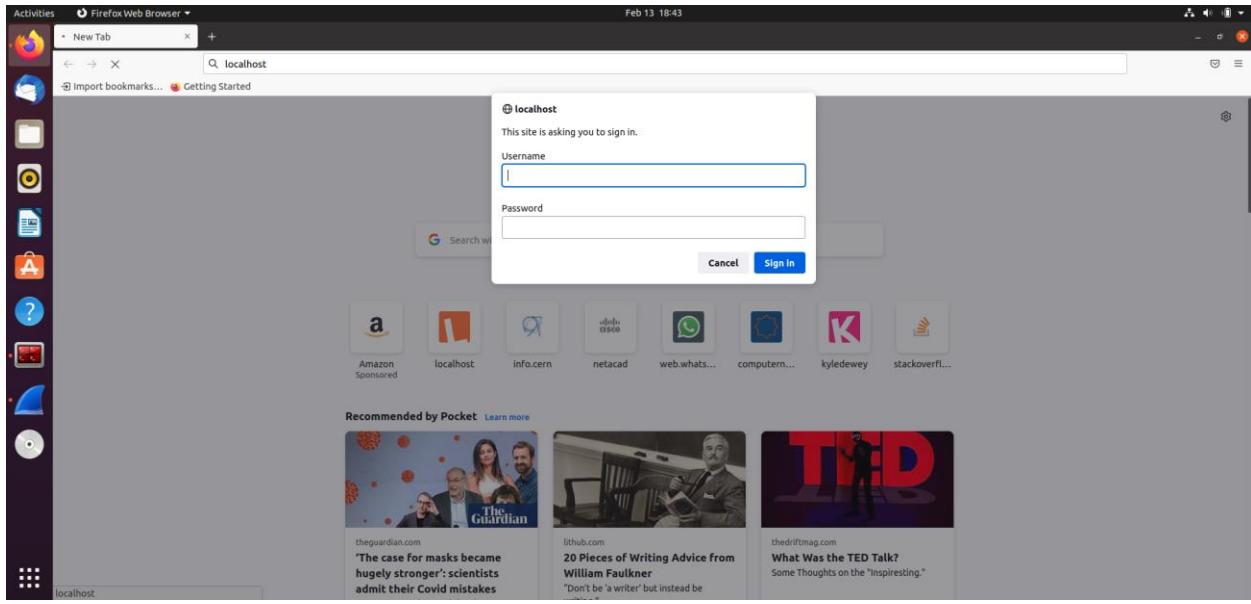
```
sr42@sr42-VirtualBox ~$ apt update
Fetched 1000 kB in 1s (1000 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading status information... Done
Get:2 http://in.archive.ubuntu.com/ubuntu focal-security InRelease [100 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://packages.osrfoundation.org/gazebo/ubuntu-stable trusty InRelease [4,238 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:7 http://packages.ros.org/ros/ubuntu trusty InRelease
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [282 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [390 kB]
Err:10 http://packages.osrfoundation.org/gazebo/ubuntu-stable trusty InRelease
 The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 6717059BAF249743
717059BAF249743
Get:10 https://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [948 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [8,000 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [23.7 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.6 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [60.3 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 kB]
Reading package lists... Done
Building dependency tree... Done
Reading status information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.9).
apache2-utils is already the newest version (2.4.41-4ubuntu3.9).
0 packages upgraded, 0 newly installed, 0 to remove and 129 not upgraded.
sr42@sr42-VirtualBox:~$ sudo htpasswd -c /etc/apache2/.htpasswd user1
New password:
Re-type new password:
Adding password for user user1
sr42@sr42-VirtualBox:~$ sudo nano /etc/apache2/sites-available/000-default.conf
sr42@sr42-VirtualBox:~$ sudo service apache2 restart
sr42@sr42.VIRTUALBOX:~$
```

- o config file :

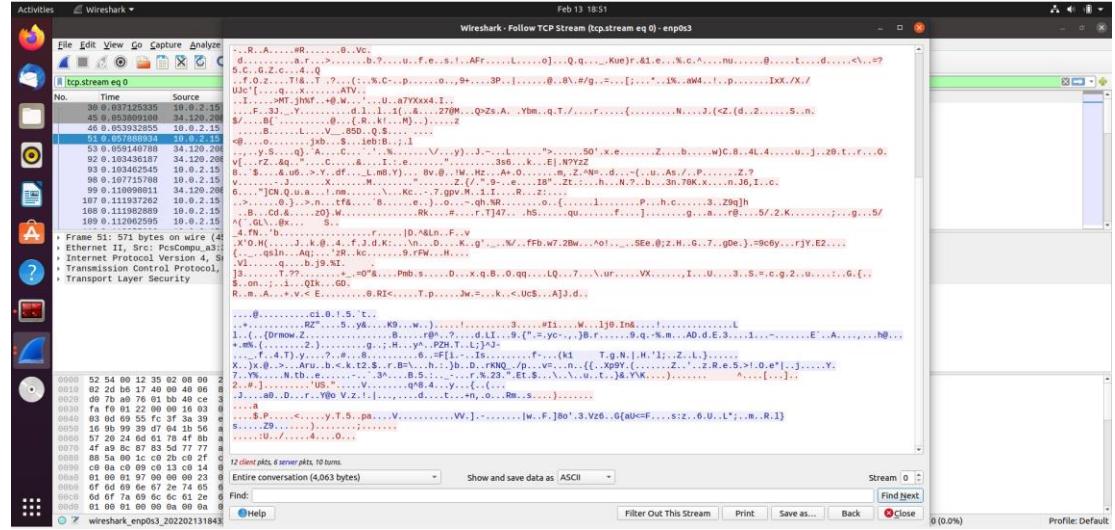
The screenshot shows a text editor window titled "000-default.conf" with the Apache2 configuration file open. The file contains various directives for the "www.example.com" virtual host, including "ServerName", "DocumentRoot", "LogLevel", "ErrorLog", "CustomLog", and "Directory" blocks. It also includes "AuthType Basic", "AuthName", "AuthUserFile", and "Require valid-user" directives for a specific directory. A note at the bottom of the file explains the use of "conf-available" files.

```
1</VirtualHost>
2 # The ServerName directive sets the request scheme, hostname and port that
3 # the server uses to identify itself. This is used when creating
4 # redirection URLs. In the context of virtual hosts, the ServerName
5 # specifies what hostname must appear in the request's Host: header to
6 # match this virtual host. For the default virtual host (this file), this
7 # value is not decisive as it is used as a last resort host address.
8 # However, you must set it for any further virtual host explicitly.
9 #ServerName www.example.com
10
11 ServerAdmin webmaster@localhost
12 DocumentRoot /var/www/html
13
14 # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15 # error, crit, alert, emerg.
16 # It is also possible to configure the loglevel for particular
17 # modules, e.g.
18 #LogLevel info ssl:warn
19
20 ErrorLog ${APACHE_LOG_DIR}/error.log
21 CustomLog ${APACHE_LOG_DIR}/access.log combined
22
23 <Directory "/var/www/html">
24
25 AuthType Basic
26 AuthName "RESTRICTED"
27 AuthUserFile /etc/apache2/.htpasswd
28 Require valid-user
29
30 </Directory>
31
32 # For most configuration files from conf-available/, which are
33 # enabled or disabled at a global level, it is possible to
34 # include or exclude them on a per-virtual-host basis. For example the
35 # following line enables the CGI configuration for this host only
36 # after it has been globally disabled with "a2disconf".
37 #Include conf-available/serve-cgi-bin.conf
38</VirtualHost>
39
40# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- o Login page :

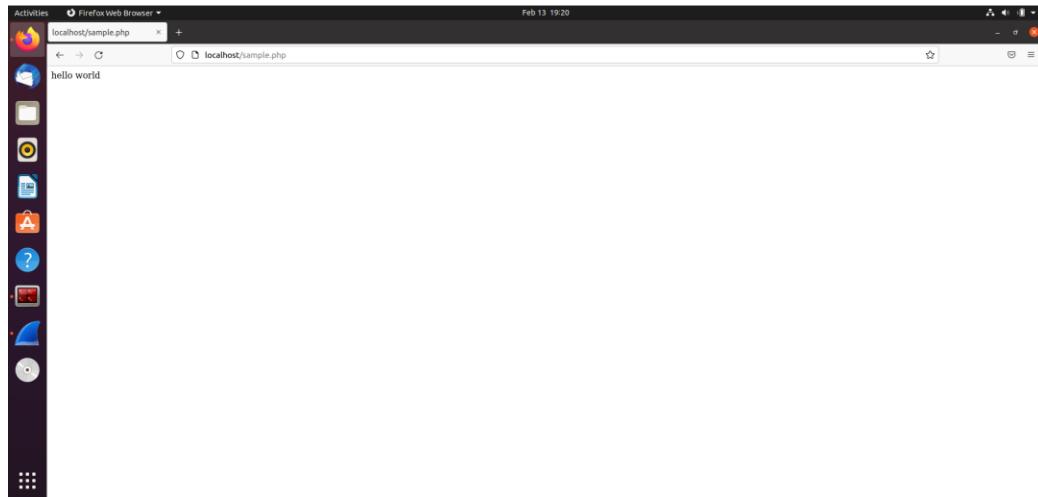


## o Wireshark :

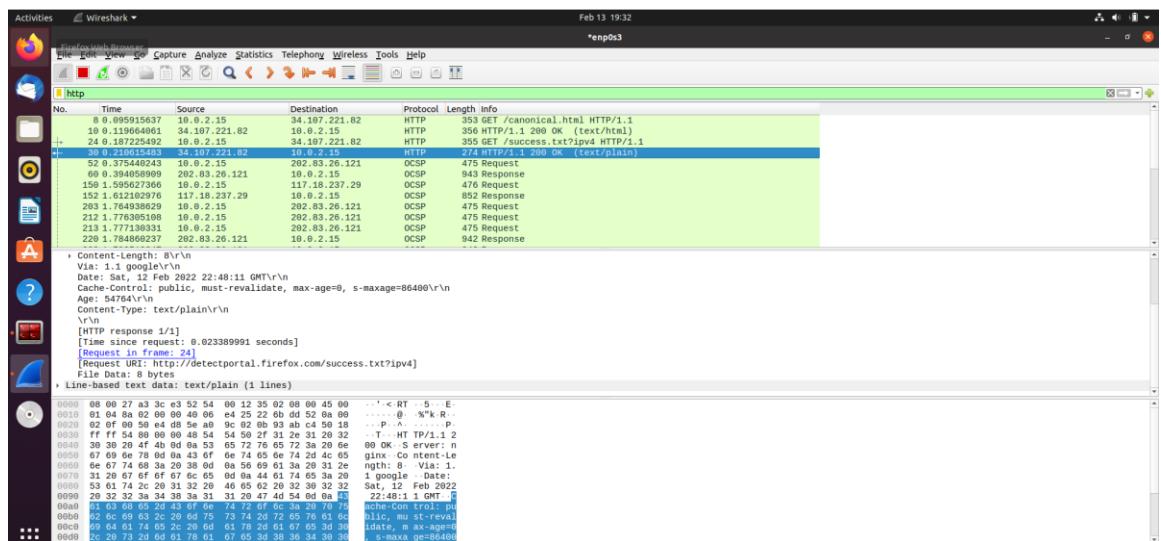


## - Cookie setting :

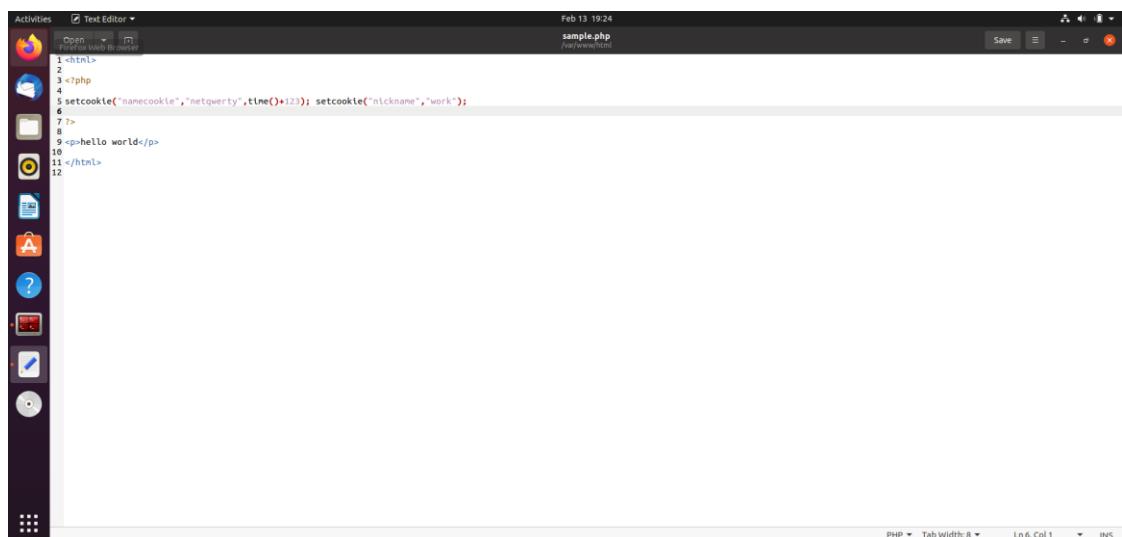
### o Output :



- Wireshark :

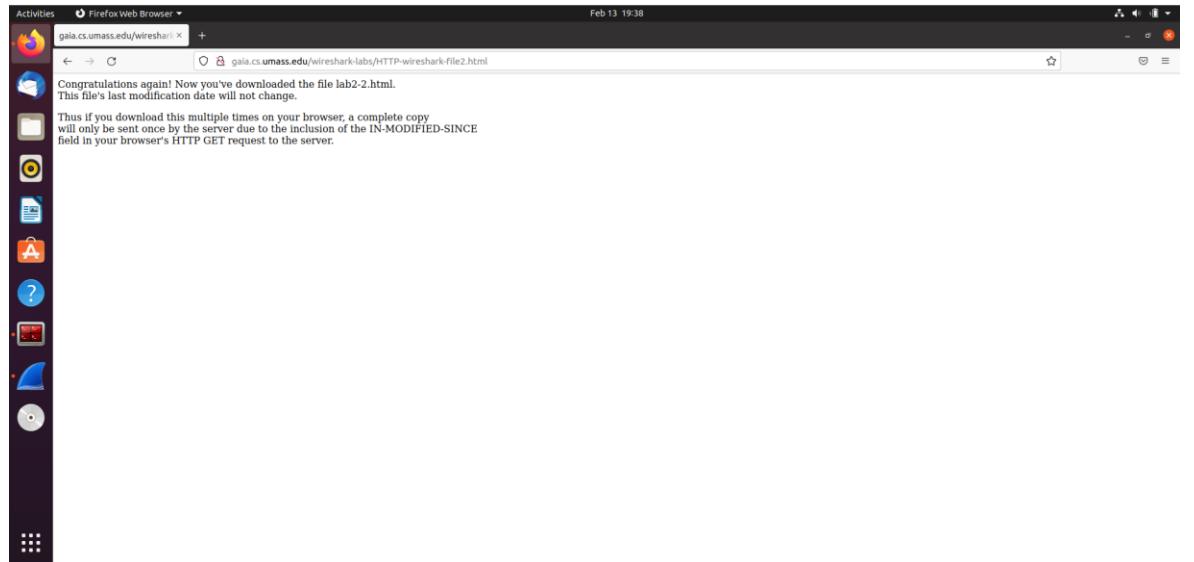


- sample.php :



- Conditional get :

- o Output :



- o Wireshark :

- If-modified-since is a visible field in the get request
- The server explicitly returned the contents of the file. This was visible by following the TCP stream.
- The if-modified-since field contains the text ‘Sun, 13 Feb 2022 06:59:02 GMT’
- HTTP status code in the response is 304 and the response phrase says ‘Not Modified’. The server did not explicitly return the contents of the file in the second response.

