LAB 3 – Application Layer

Name – Sati, Ankit Date – 10/13/2021

Section - 001

Total in points (Maximum 100 points)–

Professors Comments –

Affirmation of Independent Effort – Ankit Sati

a. The basic HTTP GET/response interaction

1. The Brower is running HTTP 1.1 and the server is running

[Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n][Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

2. It indicates that it can accept an en-US language (English).
Accept-Language: en-US,en;q=0.9\r\n

3. The IP of my computer is 10.0.0.220 and the server IP is 128.119.245.12. The source and destination IP address.
165 2021-10-11 19:53:05.576884 **10.0.0.220** **128.119.245.12** HTTP 640 GET

4. 200 OK is the status code returned to the browser.
2021-10-11 20:11:13.657673 128.119.245.12 10.0.0.220 HTTP 540
HTTP/1.1 200 OK (text/html)

5. The time appear to be only 1 minute earlier than I opened it due to the fact that we waited for a minute.
Last-Modified: Mon, 11 Oct 2021 05:59:02 GMT\r\n

6. 128 bytes is being returned to the browser
Content-Length: 128\r\n
[Content length: 128]

7. No, I do not see any other headers that are not displayed in packet window. This is because it will be a superset with all the information. Only time and date that is varied so that cannot be considered as a header.

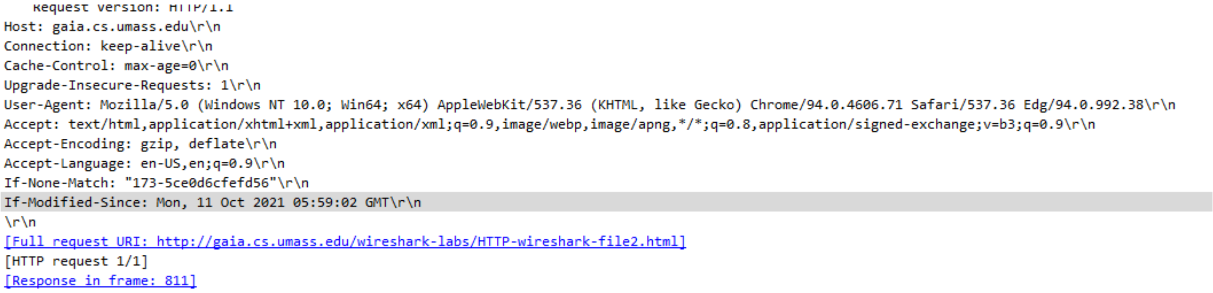
b. The HTTP conditional GET/response interaction

1. 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?
   Answer - No, there is no IF-MODIFIED-SINCE line in the first HTTP GET.

2. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
   Answer - Yes, the server did return the contents of the file as there is a "Line-based text data" line and under it is the text.

3. 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?
   Answer - Yes, there is an "IF-MODIFIED-SINCE" line in the second GET request and it follows with a date of Mon, 11 Oct 2021 05:59:02 GMT
   If-Modified-Since: Mon, 11 Oct 2021 05:59:02 GMT\r\n

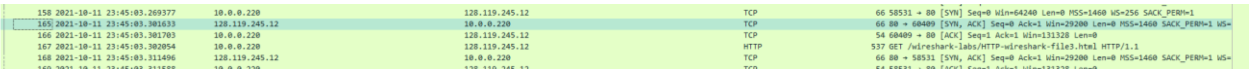
4. HTTP GET? Did the server explicitly return the contents of the file? Explain.
   Answer - The status code is 304 Not Modified and this time it did not return the contents of the file. The reason is that since the file was not modified there is no new content that needs to be passed and so there is no need to download the file again.



```
    Request Version: HTTP/1.1
Host: gaia.cs.umass.edu\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36 Edg/94.0.992.38\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
If-None-Match: "173-5ce0d6cfefd56"\r\n
If-Modified-Since: Mon, 11 Oct 2021 05:59:02 GMT\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
[HTTP request 1/1]
[Response in frame: 811]
```

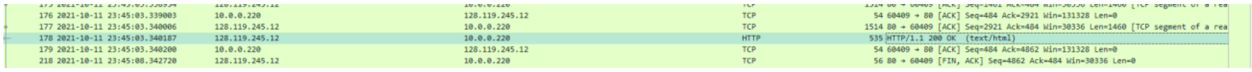
c. Retrieving long documents

1. My browser only sent 1 HTTP GET request to the server. The Packet that contained the GET message was packet number 167.
   167 2021-10-11 23:45:03.302054 10.0.0.220 128.119.245.12 HTTP 537
   GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1



```
158 2021-10-11 23:45:03.269377   10.0.0.220       128.119.245.12   TCP    66 58531 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
165 2021-10-11 23:45:03.301633   128.119.245.12   10.0.0.220       TCP    66 80 → 60409 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=
166 2021-10-11 23:45:03.301703   10.0.0.220       128.119.245.12   TCP    54 60409 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
167 2021-10-11 23:45:03.302054   10.0.0.220       128.119.245.12   HTTP  537 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
168 2021-10-11 23:45:03.311496   128.119.245.12   10.0.0.220       TCP    66 80 → 58531 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=
```

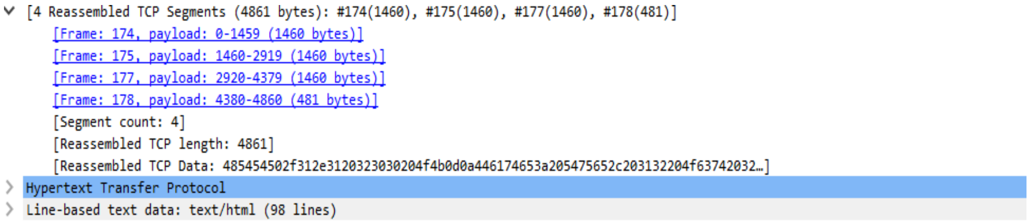
2. The packet that contains the status code and phrase which the server sent in response to the GET message was packet number 178.
   178 2021-10-11 23:45:03.340187 128.119.245.12 10.0.0.220 HTTP 535
   HTTP/1.1 200 OK (text/html)



```
176 2021-10-11 23:45:03.339003   10.0.0.220       128.119.245.12   TCP    54 60409 → 80 [ACK] Seq=484 Ack=2921 Win=131328 Len=0
177 2021-10-11 23:45:03.340006   128.119.245.12   10.0.0.220       TCP  1514 80 → 60409 [ACK] Seq=2921 Ack=484 Win=30336 Len=1460 [TCP segment of a rea
178 2021-10-11 23:45:03.340187   128.119.245.12   10.0.0.220       HTTP  535 HTTP/1.1 200 OK  (text/html)
179 2021-10-11 23:45:03.340200   10.0.0.220       128.119.245.12   TCP    54 60409 → 80 [ACK] Seq=484 Ack=4862 Win=131328 Len=0
218 2021-10-11 23:45:08.342720   128.119.245.12   10.0.0.220       TCP    56 80 → 60409 [FIN, ACK] Seq=4862 Ack=484 Win=30336 Len=0
```

3. As mentioned in the question above, the status code that is returned **is 200 OK**. (refer to the screenshot above)

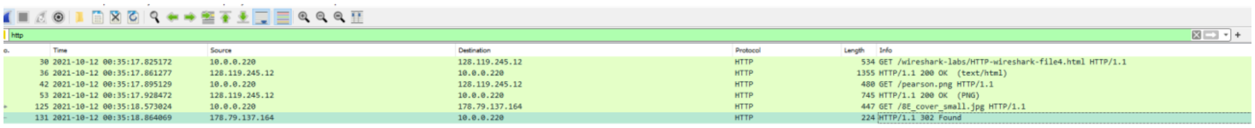
4. The data was sent over 4 packets as mentioned in the screenshot below.
   This is because the packet size exceeded the normal size and we had to break the packets and arrange them back in order.



```
v [4 Reassembled TCP Segments (4861 bytes): #174(1460), #175(1460), #177(1460), #178(481)]
     [Frame: 174, payload: 0-1459 (1460 bytes)]
     [Frame: 175, payload: 1460-2919 (1460 bytes)]
     [Frame: 177, payload: 2920-4379 (1460 bytes)]
     [Frame: 178, payload: 4380-4860 (481 bytes)]
     [Segment count: 4]
     [Reassembled TCP length: 4861]
     [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205475652c203132204f63742032…]
> Hypertext Transfer Protocol
> Line-based text data: text/html (98 lines)
```

d. HTML Documents with Embedded Objects

1. There were 3 HTTP GET requests sent from my browser to the server (128.119.245.12). It sent to the internet address of the main html page, and also the locations of the images.

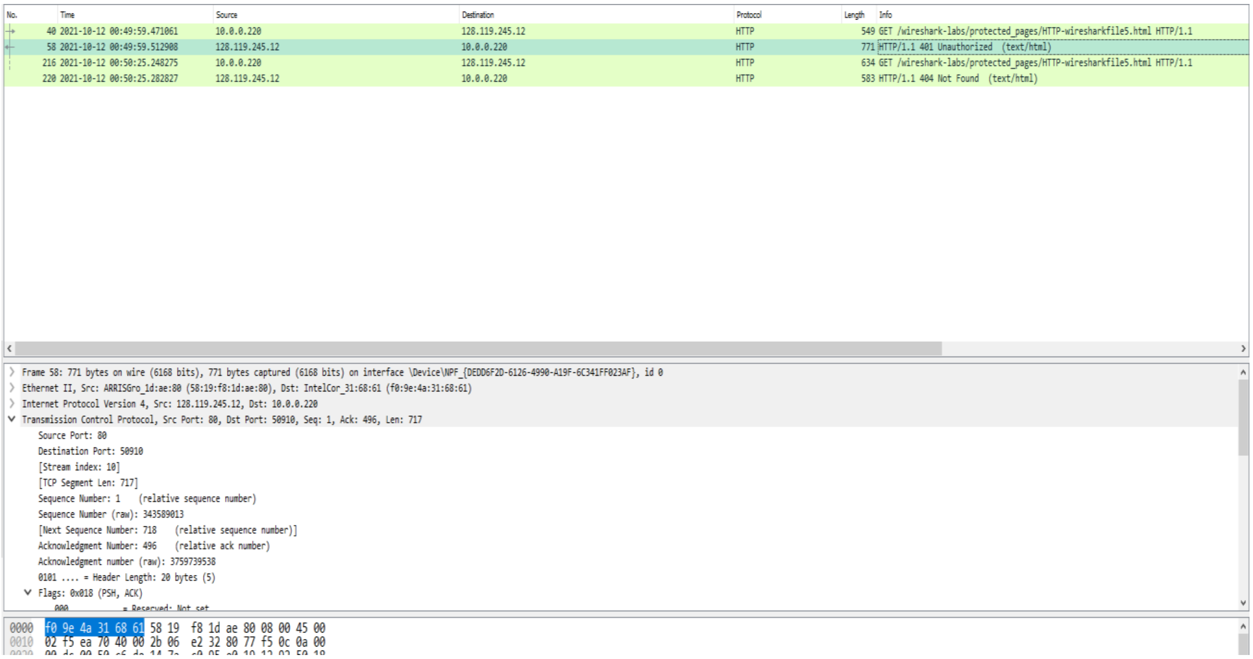


```
http
o.   Time                        Source           Destination      Protocol  Length  Info
  30 2021-10-12 00:35:17.825172  10.0.0.220       128.119.245.12   HTTP         534  GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
  36 2021-10-12 00:35:17.861277  128.119.245.12   10.0.0.220       HTTP        1355  HTTP/1.1 200 OK  (text/html)
  42 2021-10-12 00:35:17.895129  10.0.0.220       128.119.245.12   HTTP         480  GET /pearson.png HTTP/1.1
  53 2021-10-12 00:35:17.928472  128.119.245.12   10.0.0.220       HTTP         745  HTTP/1.1 200 OK  (PNG)
 125 2021-10-12 00:35:18.573024  10.0.0.220       178.79.137.164   HTTP         447  GET /8E_cover_small.jpg HTTP/1.1
 131 2021-10-12 00:35:18.864069  178.79.137.164   10.0.0.220       HTTP         224  HTTP/1.1 302 Found
```

2. The two images were downloaded serially. The first image was requested and retrieved with a status of 200 OK. Then the browser tried to download the second image and had a response of 302 Found, which means the image location moved. The browser had to then send another request to the new destination to retrieve the second image, and it came back with a 200 OK.

e. HTTP Authentication

1. The servers intial response was “401 Unauthorized ”
   58 2021-10-12 00:49:59.512908 128.119.245.12 10.0.0.220 HTTP 771
   HTTP/1.1 **401 Unauthorized (text/html)**
2. The new field that is now included is the authorization field. This is included because we sent the server a username and password along with our request stating that we were authorized to receive the page.
   Having said that the page was long lost and hence we were not able to view anything but we could easily see the authentication requests.



```
No.  Time                        Source          Destination     Protocol  Length Info
 40  2021-10-12 00:49:59.471061  10.0.0.220      128.119.245.12  HTTP      549 GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
 58  2021-10-12 00:49:59.512908  128.119.245.12  10.0.0.220      HTTP      771 HTTP/1.1 401 Unauthorized  (text/html)
216  2021-10-12 00:50:25.248275  10.0.0.220      128.119.245.12  HTTP      634 GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
220  2021-10-12 00:50:25.282827  128.119.245.12  10.0.0.220      HTTP      583 HTTP/1.1 404 Not Found  (text/html)
```

```
> Frame 58: 771 bytes on wire (6168 bits), 771 bytes captured (6168 bits) on interface \Device\NPF_{DEDD6F2D-6126-4990-A19F-6C341FF023AF}, id 0
> Ethernet II, Src: ARRISGro_1d:ae:80 (58:19:f8:1d:ae:80), Dst: IntelCor_31:68:61 (f0:9e:4a:31:68:61)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.220
v Transmission Control Protocol, Src Port: 80, Dst Port: 50910, Seq: 1, Ack: 496, Len: 717
    Source Port: 80
    Destination Port: 50910
    [Stream index: 10]
    [TCP Segment Len: 717]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 343589013
    [Next Sequence Number: 718    (relative sequence number)]
    Acknowledgment Number: 496    (relative ack number)
    Acknowledgment number (raw): 3759739538
    0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
```

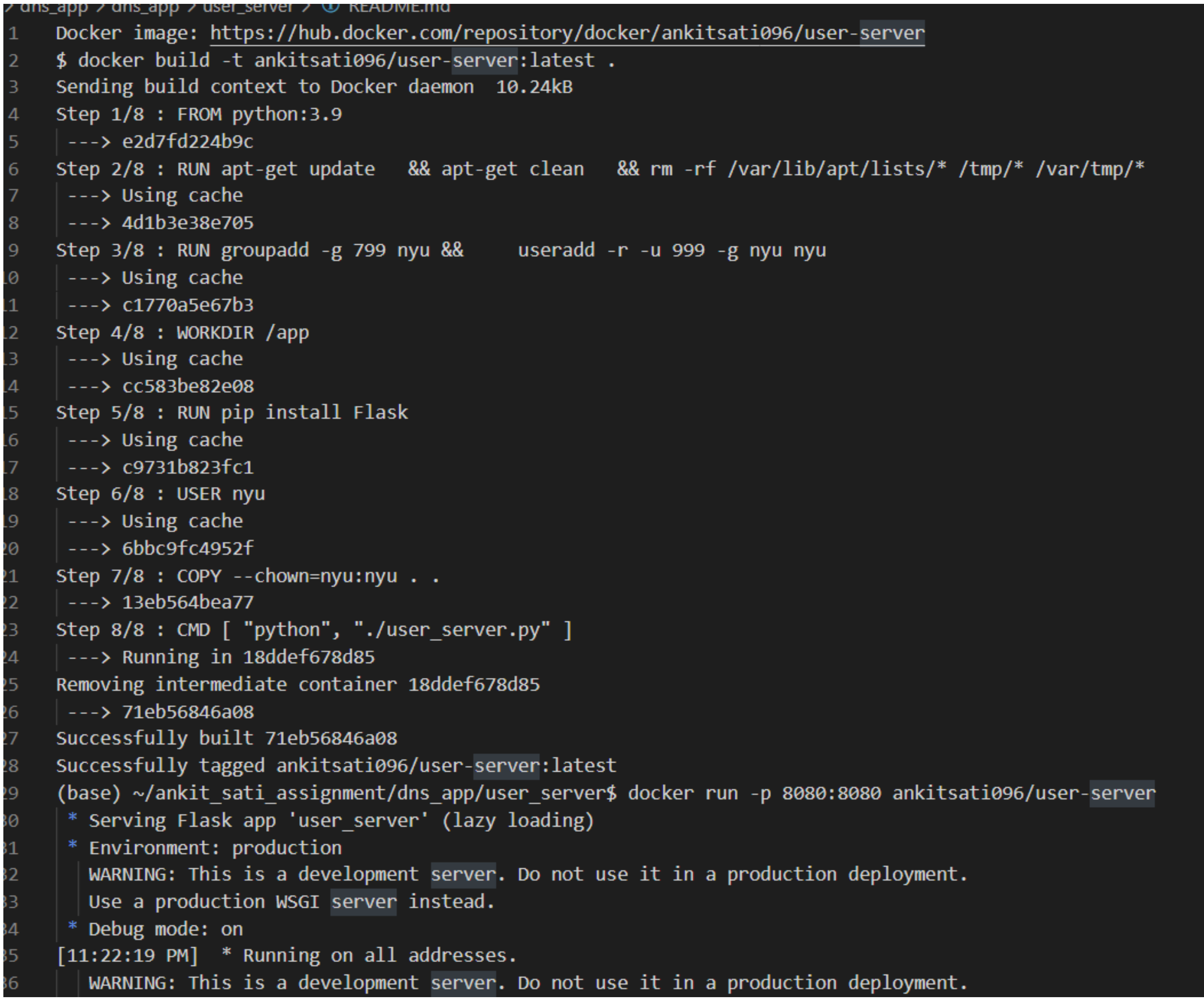
```
0000  f0 9e 4a 31 68 61 58 19  f8 1d ae 80 08 00 45 00
0010  02 f5 ea 70 40 00 2b 06  e2 32 80 77 f5 0c 0a 00
```

Question 2

Domain naming system.

Steps.

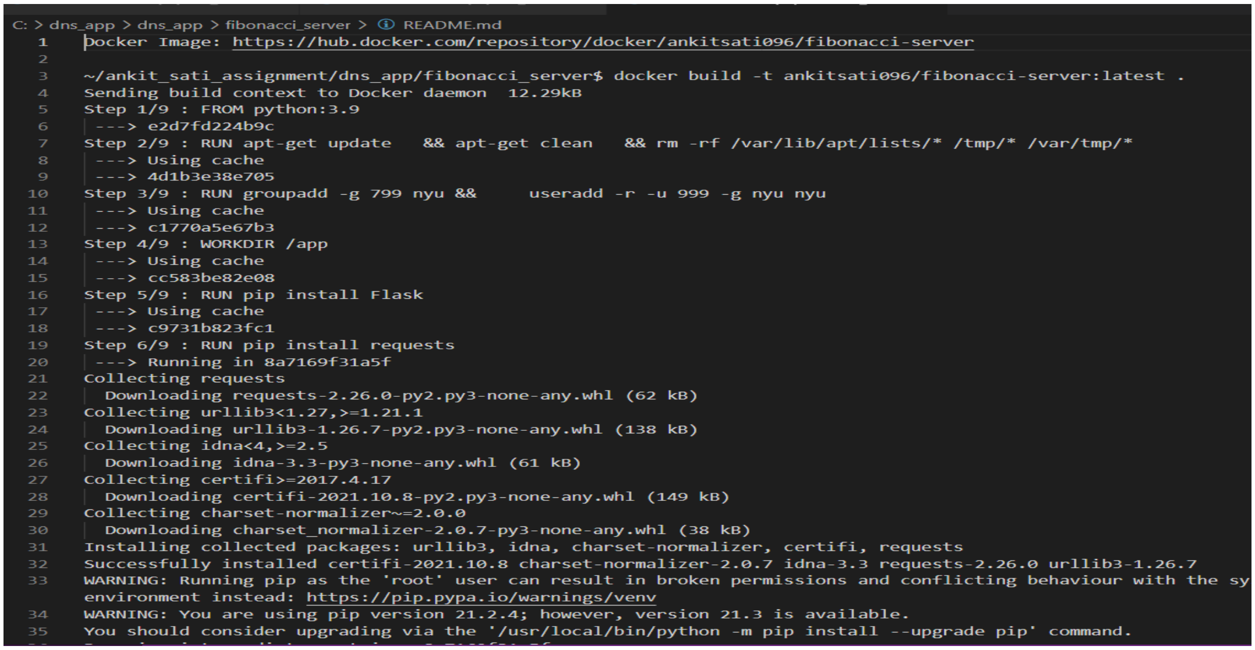
1. Creating the flask application for User Server.



> dns_app > dns_app > user_server > ⓘ README.md

```
Docker image: https://hub.docker.com/repository/docker/ankitsati096/user-server
$ docker build -t ankitsati096/user-server:latest .
Sending build context to Docker daemon  10.24kB
Step 1/8 : FROM python:3.9
 ---> e2d7fd224b9c
Step 2/8 : RUN apt-get update   && apt-get clean   && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
 ---> Using cache
 ---> 4d1b3e38e705
Step 3/8 : RUN groupadd -g 799 nyu &&     useradd -r -u 999 -g nyu nyu
 ---> Using cache
 ---> c1770a5e67b3
Step 4/8 : WORKDIR /app
 ---> Using cache
 ---> cc583be82e08
Step 5/8 : RUN pip install Flask
 ---> Using cache
 ---> c9731b823fc1
Step 6/8 : USER nyu
 ---> Using cache
 ---> 6bbc9fc4952f
Step 7/8 : COPY --chown=nyu:nyu . .
 ---> 13eb564bea77
Step 8/8 : CMD [ "python", "./user_server.py" ]
 ---> Running in 18ddef678d85
Removing intermediate container 18ddef678d85
 ---> 71eb56846a08
Successfully built 71eb56846a08
Successfully tagged ankitsati096/user-server:latest
(base) ~/ankit_sati_assignment/dns_app/user_server$ docker run -p 8080:8080 ankitsati096/user-server
 * Serving Flask app 'user_server' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
[11:22:19 PM]  * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
```

2. Create another app for Fibanacci server.

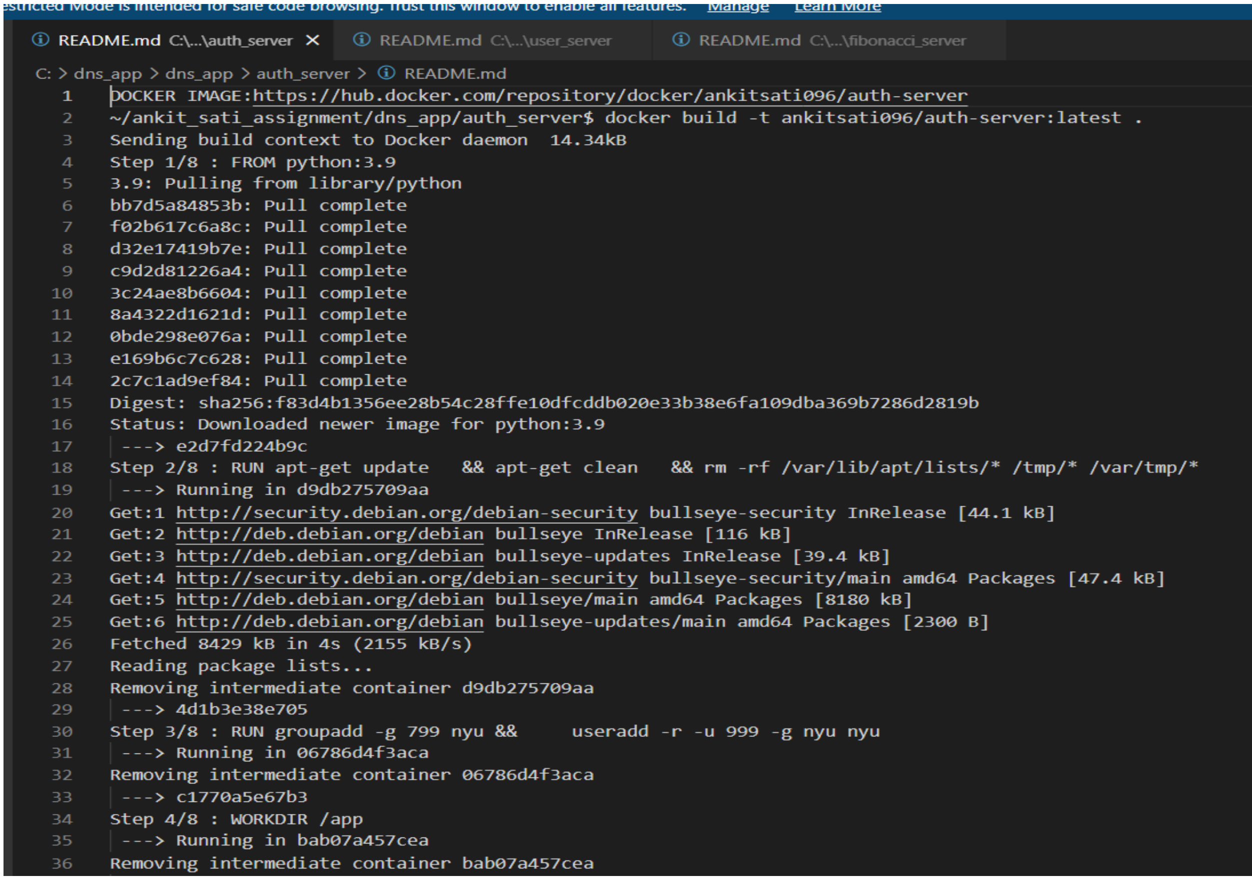


C: > dns_app > dns_app > fibonacci_server > ⓘ README.md

```
Docker Image: https://hub.docker.com/repository/docker/ankitsati096/fibonacci-server

~/ankit_sati_assignment/dns_app/fibonacci_server$ docker build -t ankitsati096/fibonacci-server:latest .
Sending build context to Docker daemon  12.29kB
Step 1/9 : FROM python:3.9
 ---> e2d7fd224b9c
Step 2/9 : RUN apt-get update   && apt-get clean   && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
 ---> Using cache
 ---> 4d1b3e38e705
Step 3/9 : RUN groupadd -g 799 nyu &&     useradd -r -u 999 -g nyu nyu
 ---> Using cache
 ---> c1770a5e67b3
Step 4/9 : WORKDIR /app
 ---> Using cache
 ---> cc583be82e08
Step 5/9 : RUN pip install Flask
 ---> Using cache
 ---> c9731b823fc1
Step 6/9 : RUN pip install requests
 ---> Running in 8a7169f31a5f
Collecting requests
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
Collecting idna<4,>=2.5
  Downloading idna-3.3-py3-none-any.whl (61 kB)
Collecting certifi>=2017.4.17
  Downloading certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Collecting charset-normalizer~=2.0.0
  Downloading charset_normalizer-2.0.7-py3-none-any.whl (38 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.10.8 charset-normalizer-2.0.7 idna-3.3 requests-2.26.0 urllib3-1.26.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the sy
environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 21.3 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
```

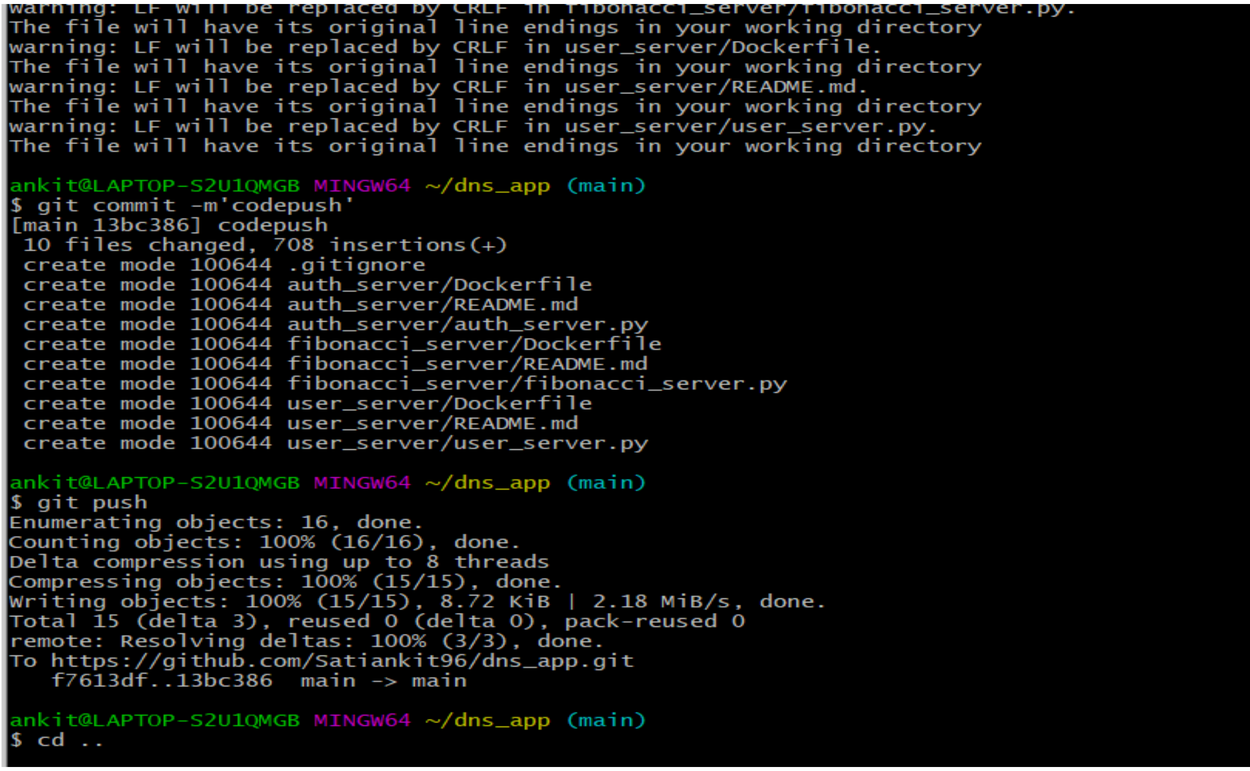
3. Finally build the socket.


estricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
README.md C:\...\auth_server
README.md C:\...\user_server
README.md C:\...\fibonacci_server
C: > dns_app > dns_app > auth_server > README.md

```
DOCKER IMAGE:https://hub.docker.com/repository/docker/ankitsati096/auth-server
~/ankit_sati_assignment/dns_app/auth_server$ docker build -t ankitsati096/auth-server:latest .
Sending build context to Docker daemon  14.34kB
Step 1/8 : FROM python:3.9
3.9: Pulling from library/python
bb7d5a84853b: Pull complete
f02b617c6a8c: Pull complete
d32e17419b7e: Pull complete
c9d2d81226a4: Pull complete
3c24ae8b6604: Pull complete
8a4322d1621d: Pull complete
0bde298e076a: Pull complete
e169b6c7c628: Pull complete
2c7c1ad9ef84: Pull complete
Digest: sha256:f83d4b1356ee28b54c28ffe10dfcddb020e33b38e6fa109dba369b7286d2819b
Status: Downloaded newer image for python:3.9
 ---> e2d7fd224b9c
Step 2/8 : RUN apt-get update   && apt-get clean   && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
 ---> Running in d9db275709aa
Get:1 http://security.debian.org/debian-security bullseye-security InRelease [44.1 kB]
Get:2 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [39.4 kB]
Get:4 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [47.4 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 Packages [8180 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [2300 B]
Fetched 8429 kB in 4s (2155 kB/s)
Reading package lists...
Removing intermediate container d9db275709aa
 ---> 4d1b3e38e705
Step 3/8 : RUN groupadd -g 799 nyu &&     useradd -r -u 999 -g nyu nyu
 ---> Running in 06786d4f3aca
Removing intermediate container 06786d4f3aca
 ---> c1770a5e67b3
Step 4/8 : WORKDIR /app
 ---> Running in bab07a457cea
Removing intermediate container bab07a457cea
```

4. As shown in the screenshots above the images of all the 3 files were mounted on docker.
5. Checked the simultaneous files on dockerhub.
6. Pushed the file to Git.
7. Exercise complete.

- File on Git - https://github.com/Satiankit96/dns_app.git
- Another Zip file attached on the homework.



```
warning: LF will be replaced by CRLF in fibonacci_server/fibonacci_server.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in user_server/Dockerfile.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in user_server/README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in user_server/user_server.py.
The file will have its original line endings in your working directory

ankit@LAPTOP-S2U1QMGB MINGW64 ~/dns_app (main)
$ git commit -m'codepush'
[main 13bc386] codepush
 10 files changed, 708 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 auth_server/Dockerfile
 create mode 100644 auth_server/README.md
 create mode 100644 auth_server/auth_server.py
 create mode 100644 fibonacci_server/Dockerfile
 create mode 100644 fibonacci_server/README.md
 create mode 100644 fibonacci_server/fibonacci_server.py
 create mode 100644 user_server/Dockerfile
 create mode 100644 user_server/README.md
 create mode 100644 user_server/user_server.py

ankit@LAPTOP-S2U1QMGB MINGW64 ~/dns_app (main)
$ git push
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 8.72 KiB | 2.18 MiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Satiankit96/dns_app.git
   f7613df..13bc386  main -> main

ankit@LAPTOP-S2U1QMGB MINGW64 ~/dns_app (main)
$ cd ..
```