

Code can be also viewed on this link to get the basic idea although this doc contains the code also - <https://github.com/snehasurna1875/Task>

I have explained the procedure of adding database creation and adding in docker container below

TASK1: What is Docker? Explain in detail.

Docker is a platform that is used for building software application with the help of containers. Container or docker container is image i.e., light weight, standalone that will provide a environment for the execution of application and isolated from other container. Earlier we use to create a virtual machine to run application with certain configuration now we use docker in any operation and run our application.

Docker file

To make a container early step is to create a docker file. It is text file that have easy syntax to understand and learn to build the docker image of particular application. This file generally contains the version of operating system to be used along with languages, environment variables, location, network ports for certain application.

Docker image

After creating docker file, we invoke docker to build utility to let image to be created on that dockefile. Dockerfile is basically a order-wise groups of instructions that instruct how to create and build image. And best feature of docker image is that it is portable and can run on system. The image created by docker is static.

Docker run

It is the utility that leads to launch of a container. Each container is the basically instance of the image. Containers can be restated and stopped, these features make them transient and temporary. We can run same container instances simultaneously conditioned the container name should be different

Docker Engine

It is the heart or core of docker that creates, build and run docker containers. Docker engines comes in different version named Engine Enterprise and Docker Engine Community.

Docker Compose

It is utility to define and run multi container docker applications by using YAML file basically. By using command **docker-compose up** to create, build and start all the configuration

To run compose three steps to be followed :

1. Define your Dockerfile with environment needed for application
2. Define all the services needed for application to run in docker-compose.yaml file.
3. Then run the command docker-compose up to execute entire app.

Docker Swarm

It is a group of physical or virtual machines that runs the docker application and are combined to form a cluster. To manage such kinds of activities a swarm manager is there that manage all nodes.

Docker Kubernetes

It is a container that orchestrate system for the containers of docker which is more extensive than Docker swarm and it is used for coordinates clusters of nodes at scale in a efficient way. It automate the management scaling and routing of container.

References:

1. [What is Docker? The spark for the container revolution | InfoWorld](#)
2. [What is Docker Swarm? | Sumo Logic](#)
3. [Kubernetes Vs Docker | Sumo Logic](#)
4. [Overview of Docker Compose | Docker Documentation](#)

Create an HTTP Docker Container and host any customized dynamic website inside it. The website should be developed using the LAMP Stack. You should operate the Docker container using docker-compose and also download the logs(Apache Logs) from inside the docker to any folder inside your machine in real-time.

LAMP stack means Linux as operating system Apache as server, Mysql as Database and PHP as server side language.

We will install mysql-server and php-mysql in our machine to create database and table.

```
sneha@sneha-VirtualBox:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinypa
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 16 newly installed, 0 to remove and 344 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 249 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
sneha@sneha-VirtualBox:~$ sudo apt-get install php-mysql
[sudo] password for sneha:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  php7.4-mysql
The following NEW packages will be installed:
  php-mysql php7.4-mysql
0 upgraded, 2 newly installed, 0 to remove and 344 not upgraded.
Need to get 123 kB of archives.
After this operation, 487 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Now open mysql with below command but this will open mysql without password that means no user is created

```
sneha@sneha-VirtualBox:~$ sudo mysql
[sudo] password for sneha:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

But for dynamic page we want to create user so we will use the below mentioned command

We will create a database using mysql command i.e. **create database details;**

```
sneha@sneha-VirtualBox:~$ sudo mysql -u surana -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

To switch to our created database we will use mentioned commands i.e. **use details.**

After that we create a table using mysql i.e.

```
create table logininfo (  
username varchar(30);  
password varchar(30)  
);
```

Once table is created we will insert values using this mysql commands

```
insert into logininfo values('sneha','12345');
```

```
mysql> use details
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select *from logininfo;
+-----+-----+-----+
| id   | username | password |
+-----+-----+-----+
| 1    | sneha    | 12345    |
+-----+-----+-----+
1 row in set (0.03 sec)

mysql>
```

```
sneha@sneha-VirtualBox:/var/www/html$ ls
login_d.php  login.html
sneha@sneha-VirtualBox:/var/www/html$
```

After that we will create a html page for login and create a form .

```
GNU nano 4.8                               login.html
<html>
<head>
  <title>web application using LAMP </title></head>
<body>
  <b><h1>Sign-In</h1></b>
  <form action="login_d.php" method="POST">
    <h6>Username:</h6><input type="text" id="user" name="user"><br><br>
    <h6>Password:</h6><input type="password" id="pass" name="pass"><br><br>
    <input type="submit" name="sub" value="login">
  </form>
</body>
</html>
```

After that we will create a server side page using php We will connect php to our database and fetch information from database

```
GNU nano 4.8                                login_d.php
<?php
$servername = "localhost";
$username = "surana";
$password = "surana";
$db_name="details";
$tablename="logininfo";

// Create connection
$conn = new mysqli($servername,$username, $password,$db_name);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
$myusername=$_POST['user'];
$mypassword=$_POST['pass'];

$sql="select * from $tablename where username='$myusername' and password='$mypassword'";
$result=mysqli_query($conn,$sql);
$count=mysqli_num_rows($result);

if ($count == 1)
{
echo ":) :) LOGIN SUCCESS :) :) ";
}

else
{
echo ":( :( AUTHETICATION FAILURE :( :( ";
}
?>
```

Command to download the current stable version of Docker Compose:

```
sneha@sneha-VirtualBox:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.28.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0     0     0  --:--:-- --:--:-- --:--:-- 100    633   100    6
33     0    0  2335    0     0     0     0  --:--:-- --:--:-- --:--:-- 2335
18 11.6M  18 2242k    0     0  2421k    0  0:00:04 --:--:-- 0:00:04 75 11.6M   75 904
2k     0    0  4704k    0     0  0:00:02 0:00:01 0:00:01 100 11.6M  100 11.6M    0    516
8k     0  0:00:02 0:00:02 --:--:-- 7009k
sneha@sneha-VirtualBox:~$
```

Apply executable permissions to the binary that was downloaded

```
sneha@sneha-VirtualBox:~$ sudo chmod +x /usr/local/bin/docker-compose
sneha@sneha-VirtualBox:~$
```

To check the installation run the below mentioned commands

```
sneha@sneha-VirtualBox:~$ docker-compose --version
docker-compose version 1.28.4, build cabd5cfb
sneha@sneha-VirtualBox:~$
```

To install docker and check whether docker compose is install successfully we run below mentioned command

```
sneha@sneha-VirtualBox:~$ sudo apt-get install docker docker-compose
[sudo] password for sneha:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

We will make docker directory to run our code and manage files

```
sneha@sneha-VirtualBox:~$ mkdir -p dockerized-lamp/DocumentRoot
sneha@sneha-VirtualBox:~$ ls
Desktop  dockerized-lamp  Documents  Downloads  Music  Pictures  Public  Templates  Videos
sneha@sneha-VirtualBox:~$
```

And php folder to create php Docker file install php dependencies

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ mkdir php-apache2
sneha@sneha-VirtualBox:~/dockerized-lamp$ cd php-apache2/
sneha@sneha-VirtualBox:~/dockerized-lamp/php-apache2$ nano Dockerfile
```

File structure of the http container

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ ls
docker-compose.yaml  DocumentRoot  php-apache2
sneha@sneha-VirtualBox:~/dockerized-lamp$ cd DocumentRoot/
sneha@sneha-VirtualBox:~/dockerized-lamp/DocumentRoot$ ls
index.php  login_d.php  login.html
sneha@sneha-VirtualBox:~/dockerized-lamp/DocumentRoot$ cd ..
sneha@sneha-VirtualBox:~/dockerized-lamp$ cd php-apache2/
sneha@sneha-VirtualBox:~/dockerized-lamp/php-apache2$ ls
Dockerfile
sneha@sneha-VirtualBox:~/dockerized-lamp/php-apache2$
```

Content of docker file named docker-compose.yaml

version: '3'

services:

php-apache2:

image: php:7.4.3-apache

ports:

- 80:80

volumes:

- ./DocumentRoot:/var/www/html:z

links:

- 'mysql'

mysql:

image: mysql

environment:

TZ: "INDIA"

MYSQL_ALLOW_EMPTY_PASSWORD: "no"

MYSQL_ROOT_PASSWORD: "sneha"

MYSQL_USER: 'surana'

MYSQL_PASSWORD: 'surana'

MYSQL_DATABASE: 'details'

ports:

- '3306:3306'

expose :

- '3306'

volumes:

- mysql:/var/lib/mysql

- mysql:/var/log/mysql

volumes:

mysql:

Contents of Dockerfile

FROM php:7.4.3-apache

MAINTAINER egidio docilex

RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli

Content of index.php

<?php phpinfo(); ?>

Content of login.html

<html>

<head>

<title>web application using LAMP </title></head>

<body>

<h1>Sign-In</h1>

<form action="login_d.php" method="POST">

<h6>Username:</h6><input type="text" id="user" name="user">

<h6>Password:</h6><input type="password" id="pass" name="pass">


```
        <input type="submit" name="sub" value="login">
    </form>
</body>

</html>
```

Content of login_d.php

```
<?php

$servername = "dockerized-lamp_mysql_1";
$username = "surana";
$password = "surana";
$db_name="details";
$tablename="logininfo";

// Create connection

$conn = new mysqli($servername,$username, $password,$db_name);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo "Database Connected successfully";
echo "<br>";

$myusername=$_POST['user'];
$mypassword=$_POST['pass'];
```

```
$sql="select * from $tablename where username='$myusername' and  
password='$mypassword';
```

```
$result=mysqli_query($conn,$sql);
```

```
$count=mysqli_num_rows($result);
```

```
if ($count == 1)
```

```
{
```

```
echo "YOU ARE LOGGED IN SUCCESSFULLY ";
```

```
}
```

```
else
```

```
{
```

```
echo "MAY BE YOUR USERNAME OR PASSWORD IS NOT CORRECT ";
```

```
}
```

```
?>
```

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo systemctl restart docker  
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker-compose up  
Building with native build. Learn about native build in Compose here: https://docs.docker.com/  
go/compose-native-build/  
Pulling mysql (mysql:)...  
latest: Pulling from library/mysql  
45b42c59be33: Already exists  
b4f790bd91da: Already exists  
325ae51788e9: Already exists  
adcb9439d751: Already exists  
174c7fe16c78: Already exists  
698058ef136c: Already exists  
4690143a669e: Already exists  
f7599a246fd6: Pull complete  
35a55bf0c106: Pull complete
```

To install mysqli in our container

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo !!
sudo docker ps -a
[sudo] password for sneha:
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
0b94d6582bde       php:7.4.3-apache   "docker-php-entryp... 30 minutes ago
Up 30 minutes      0.0.0.0:80->80/tcp  dockerized-lamp_php-apache2_1
254d89b9bc30       mysql              "docker-entrypoint.s..." 30 minutes ago
Up 30 minutes      0.0.0.0:3306->3306/tcp, 33060/tcp  dockerized-lamp_mysql_1
```

To add mysql database to our mysql container first we will use MySQL dump

```
sneha@sneha-VirtualBox: ~
sneha@sneha-VirtualBox:~$ sudo mysqldump -u surana -p details<details.sql
Enter password:
```

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker exec -ti 0b94d6582bde sh
[sudo] password for sneha:
# docker-php-ext-install mysqli
Configuring for:
PHP Api Version:      20190902
Zend Module Api No:   20190902
Zend Extension Api No: 320190902
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
```

Now to add that dump to our mysql container we need to run below mentioned command

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker exec -i 254d89b9bc30 mysql
-u root -psneha details < /home/sneha/details.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
sneha@sneha-VirtualBox:~/dockerized-lamp$
```

To list out container we need to use the below command

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
0b94d6582bde       php:7.4.3-apache   "docker-php-entryp... 38 minutes ago
Up 38 minutes      0.0.0.0:80->80/tcp  dockerized-lamp_p
hp-apache2_1
254d89b9bc30       mysql              "docker-entrypoint.s..." 38 minutes ago
Up 38 minutes      0.0.0.0:3306->3306/tcp, 33060/tcp  dockerized-lamp_m
ysql_1
```

To checker whether details.sql file is added or not

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker exec -it 254d89b9bc30 bin/bash
root@254d89b9bc30:/# ls
bin boot dev docker-entrypoint-initdb.d entrypoint.sh etc home lib lib64 med
ia mnt opt proc root run sbin srv sys tmp usr var
root@254d89b9bc30:/# cd home/
root@254d89b9bc30:/home# ls
root@254d89b9bc30:/home# cd
root@254d89b9bc30:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
```

```
root@254d89b9bc30:/# mysql -u surana -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

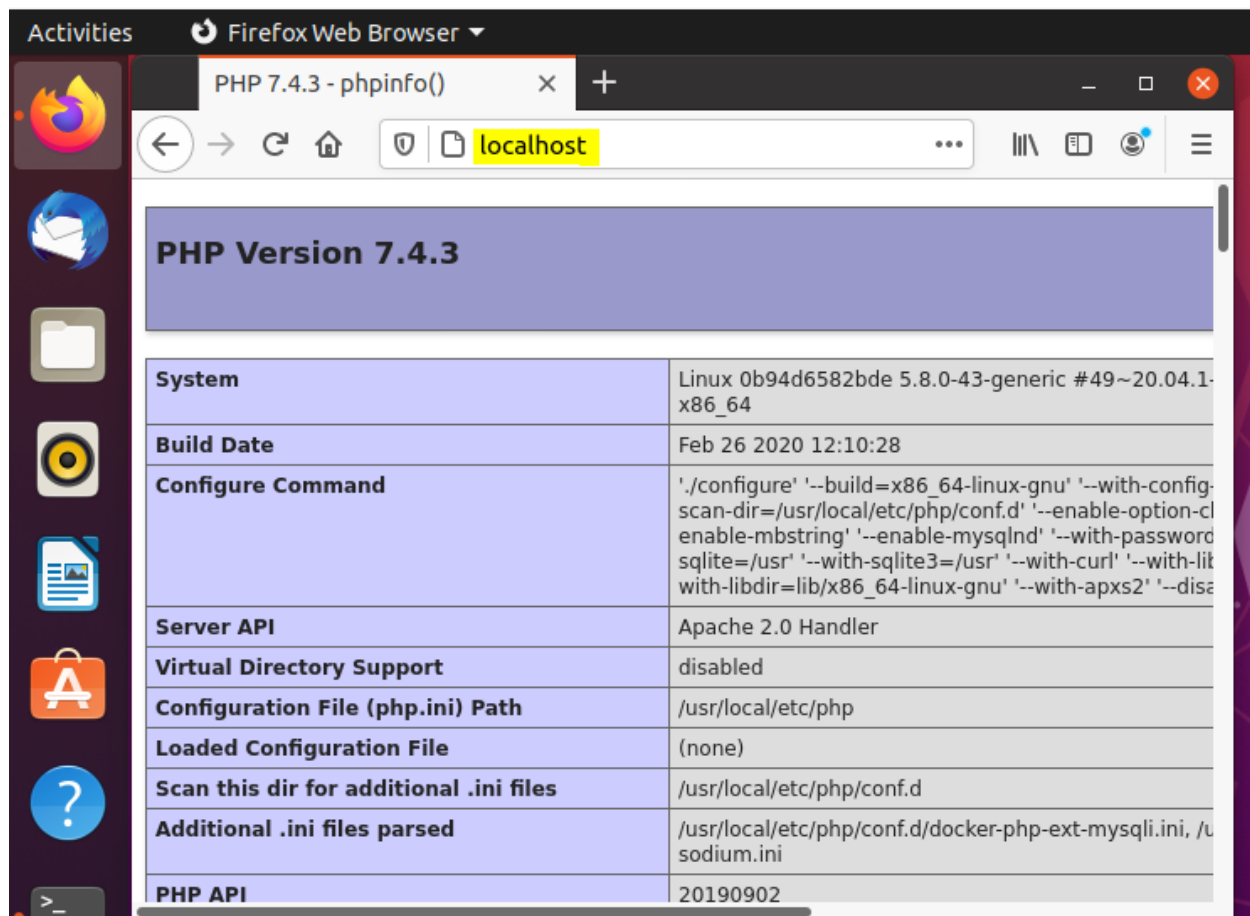
mysql> show databases;
+-----+
| Database          |
+-----+
| details            |
| information_schema |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> use details;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_details |
+-----+
| logininfo          |
+-----+
1 row in set (0.00 sec)

mysql> 
```

Open the website at localhost



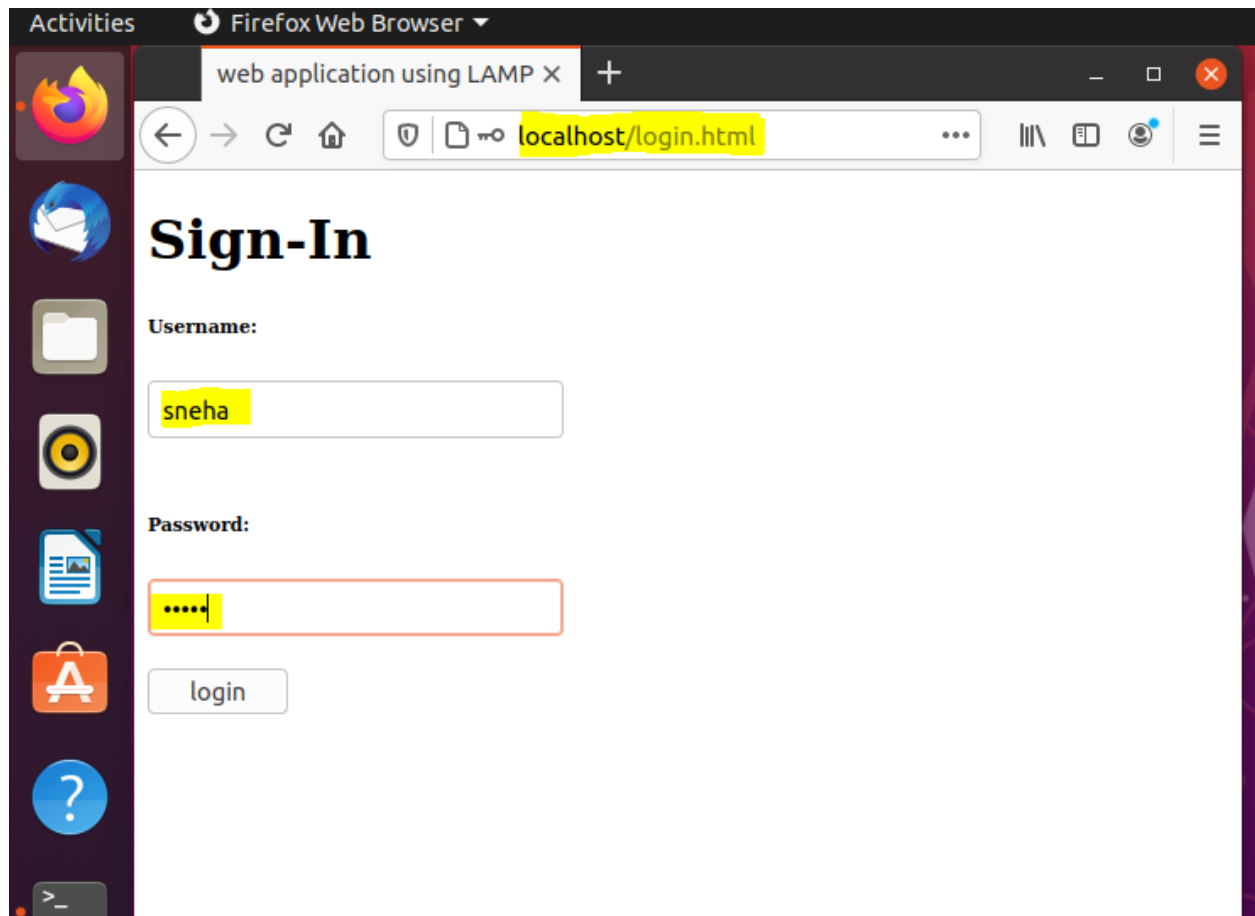
Activities Firefox Web Browser

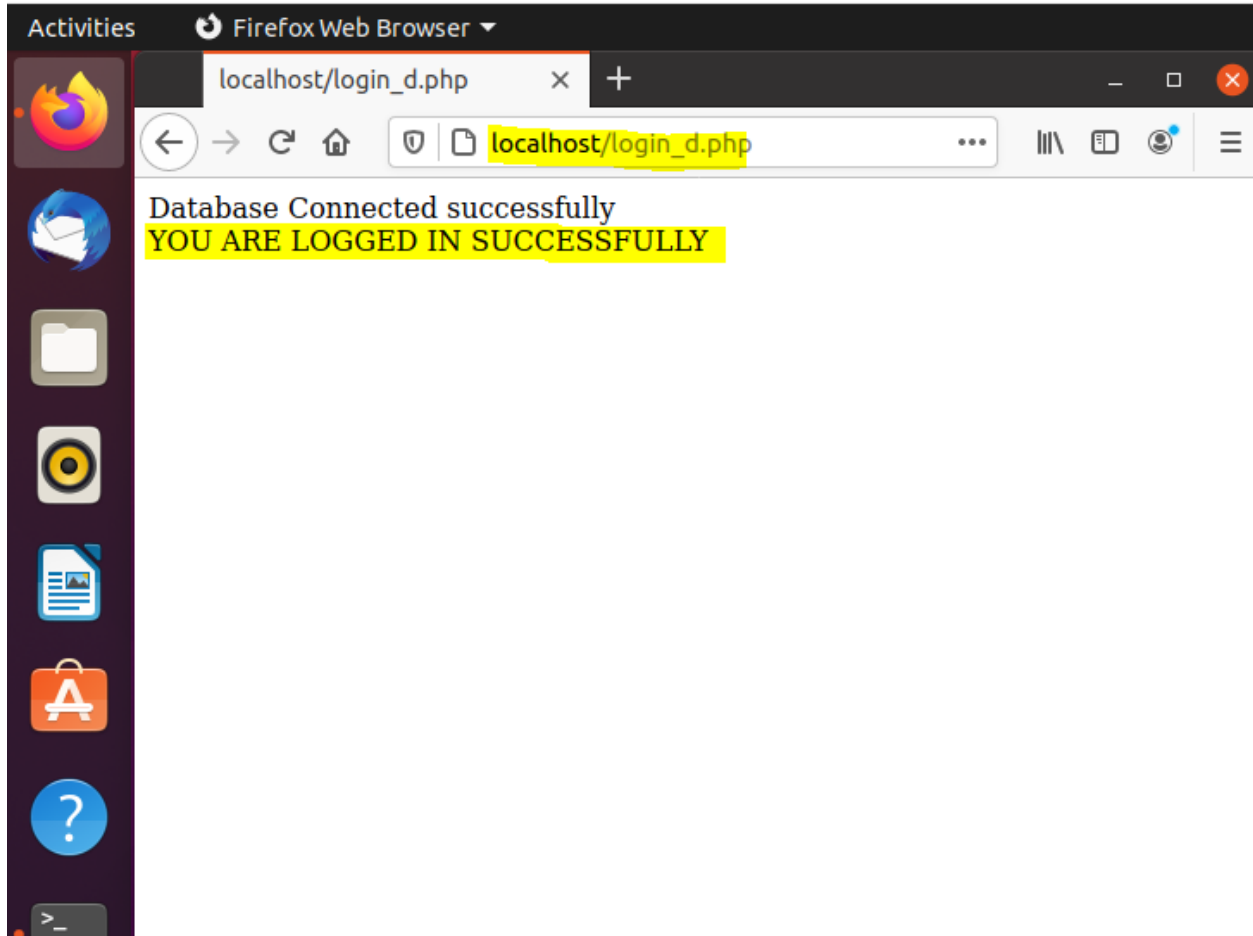
PHP 7.4.3 - phpinfo() x +

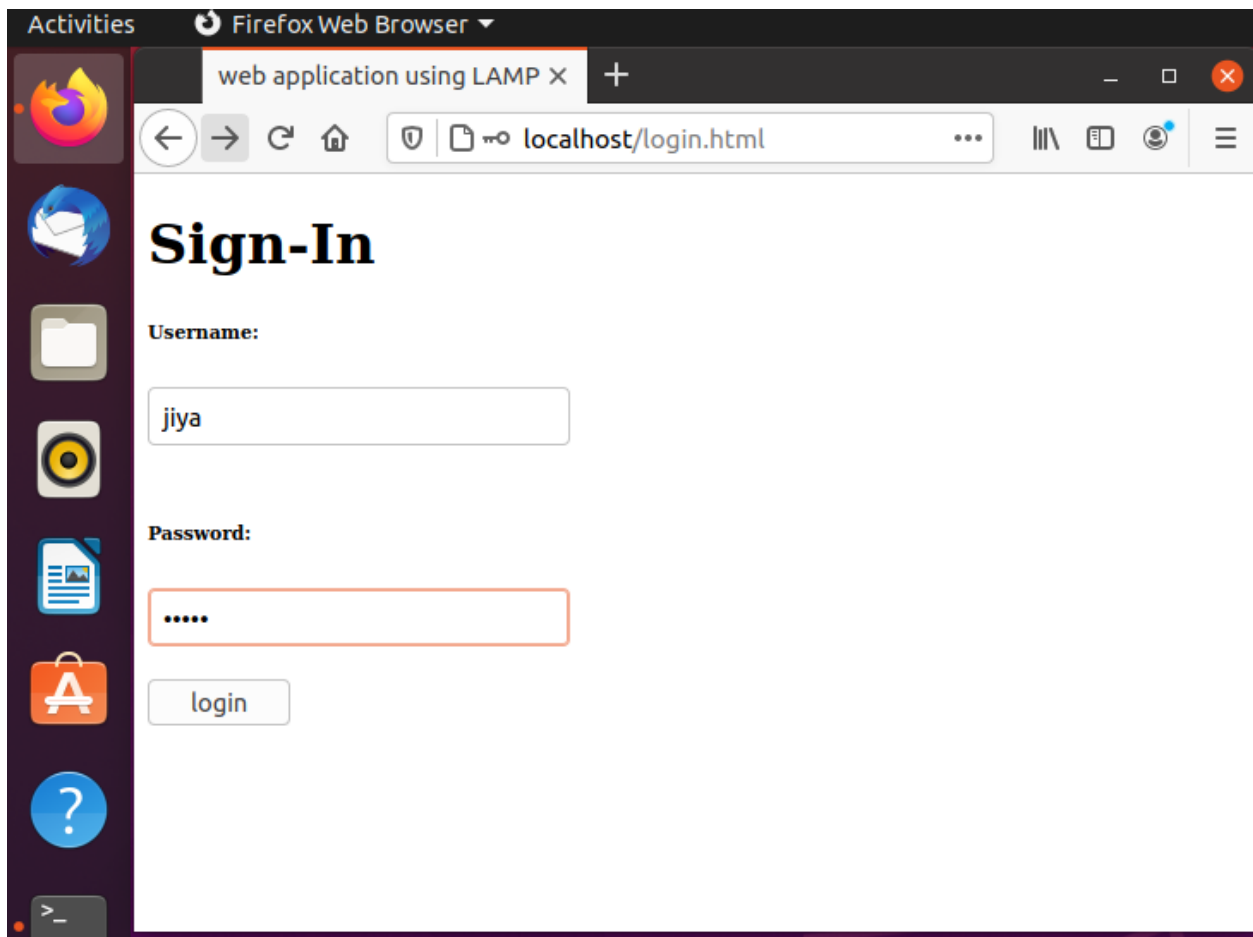
localhost

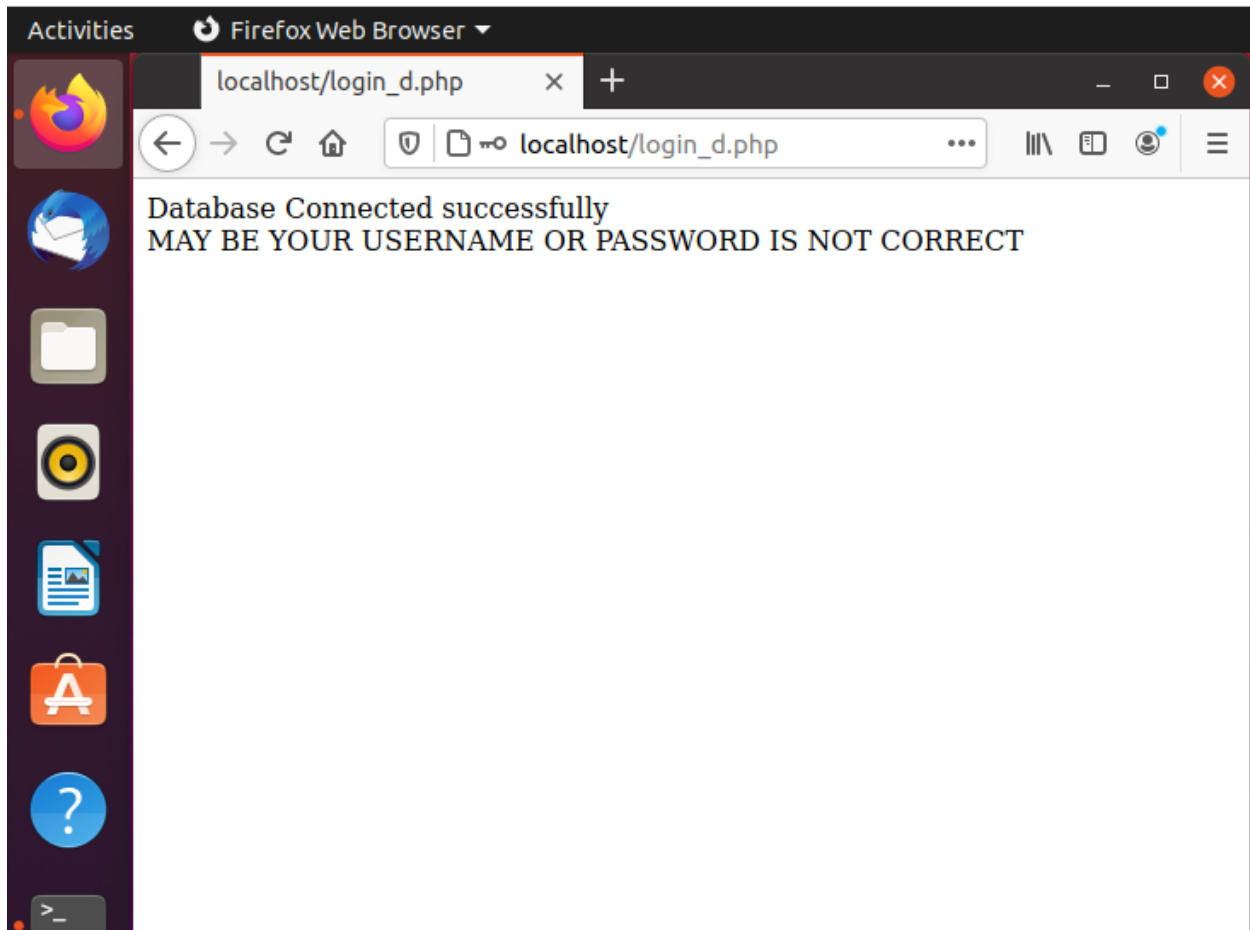
PHP Version 7.4.3

System	Linux 0b94d6582bde 5.8.0-43-generic #49~20.04.1-x86_64
Build Date	Feb 26 2020 12:10:28
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-clipboard' '--enable-mbstring' '--enable-mysqlnd' '--with-password-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disa
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /u sodium.ini
PHP API	20190902









To Download the logs from docker to normal system

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c458ac5ac510	php:7.4.3-apache	"docker-php-entrypoi..."	12 minutes ago	Up About a minute	0.0.0.0:80
->80/tcp		dockerized-lamp_php-apache2_1			
ec8c996847be	mysql	"docker-entrypoint.s..."	12 minutes ago	Up About a minute	0.0.0.0:33
06->3306/tcp, 33060/tcp		dockerized-lamp_mysql_1			

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker logs -f c458ac5ac510 > /home/sneha
```

bash: /home/sneha: Is a directory

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker logs -f c458ac5ac510 > /home/sneha/logs.txt
```

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c458ac5ac510	php:7.4.3-apache	"docker-php-entrypoi..."	12 minutes ago	Up About a mi
nute 0.0.0.0:80->80/tcp		dockerized-lamp_php-apache2_1		
ec8c996847be	mysql	"docker-entrypoint.s..."	12 minutes ago	Up About a mi
nute 0.0.0.0:3306->3306/tcp, 33060/tcp		dockerized-lamp_mysql_1		

```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker logs -f c458ac5ac510 > /home/sneha
```

bash: /home/sneha: Is a directory

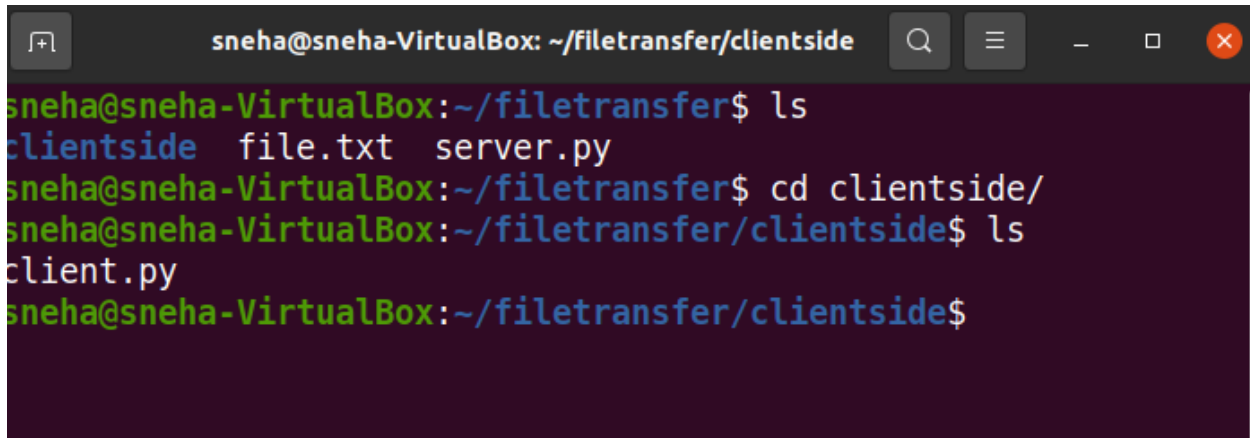
```
sneha@sneha-VirtualBox:~/dockerized-lamp$ sudo docker logs -f c458ac5ac510 > /home/sneha/logs.txt
```

```
sneha@sneha-VirtualBox:~$ ls
Desktop      dockerized-lamp  Downloads  login.html  Music      Public      Videos
details.sql  Documents        login_d.php logs.txt    Pictures   Templates
sneha@sneha-VirtualBox:~$
```

```
sneha@sneha-VirtualBox:~$ cat logs.txt
172.18.0.1 - - [25/Feb/2021:21:45:47 +0000] "GET / HTTP/1.1" 200 22235 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
172.18.0.1 - - [25/Feb/2021:21:46:04 +0000] "POST /login_d.php HTTP/1.1" 200 439 "http://localhost/login.html" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
172.18.0.1 - - [25/Feb/2021:21:47:57 +0000] "GET /login_d.php HTTP/1.1" 200 439 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
172.18.0.1 - - [25/Feb/2021:21:49:27 +0000] "GET /login_d.php HTTP/1.1" 200 378 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
172.18.0.1 - - [25/Feb/2021:21:49:46 +0000] "POST /login_d.php HTTP/1.1" 200 296 "http://localhost/login.html" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
172.18.0.1 - - [25/Feb/2021:21:50:08 +0000] "POST /login_d.php HTTP/1.1" 200 378 "http://localhost/login.html" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0"
sneha@sneha-VirtualBox:~$
```

TASK 2: Develop a basic FTP Server/Client in Python/C which should be able to do file transfer.

File structure

A terminal window titled 'sneha@sneha-VirtualBox: ~/filetransfer/clientside'. The terminal shows the following commands and output:

```
sneha@sneha-VirtualBox:~/filetransfer$ ls
clientside  file.txt  server.py
sneha@sneha-VirtualBox:~/filetransfer$ cd clientside/
sneha@sneha-VirtualBox:~/filetransfer/clientside$ ls
client.py
sneha@sneha-VirtualBox:~/filetransfer/clientside$
```

Content of server.py

```
import socket

PORT = 6060

d=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
SERVER =socket.gethostbyname(socket.gethostname())
d.bind((SERVER,PORT))
d.listen(10)

print ("SERVER STARTED LISTENING.....")

connection , address = d.accept()

print (address,"SEVER IS BEEN CONNECTED")

file_name= input(str("PLEASE ENTER THE NAME OF THE FILE TO BE TRANSFERED : "))

file_open = open(file_name , 'rb')

file_data = file_open.read(1024)

connection.send(file_data)

print("Data has been transmitted successfully")

connection.close()
```

Content of Client.py

```
import socket

PORT = 6060

d=socket.socket(socket.AF_INET, socket.SOCK_STREAM)

CLIENT =socket.gethostname(socket.gethostname())

d.connect((CLIENT,PORT))

print ("CLIENT IS SUCCESFULLY CONNECTED TO SERVER");


filename= input(str("Please enter the name"))

file = open (filename, 'wb')

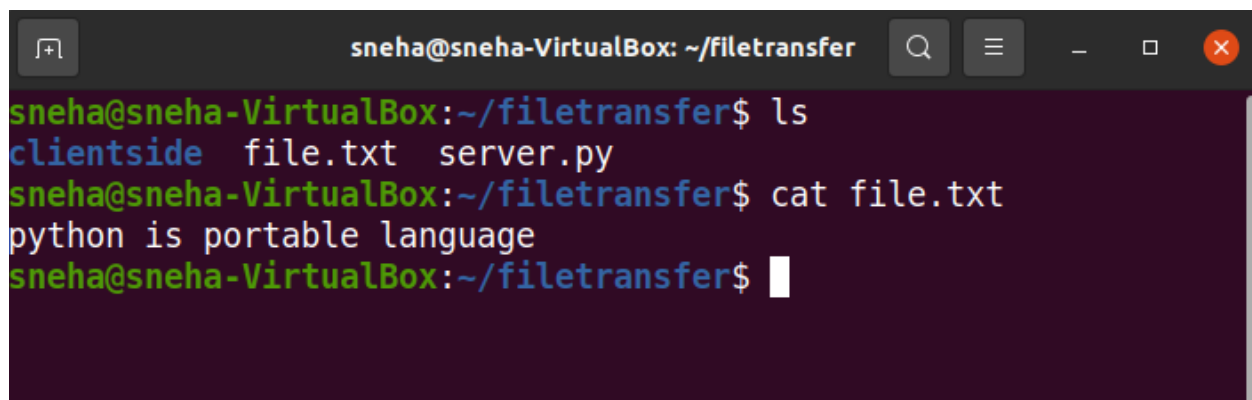
file_data = d.recv(1024)

file.write(file_data)

file.close()

print ("FILE RECIEVED SUCCESFULLY")
```

Content of file.txt

A terminal window titled 'sneha@sneha-VirtualBox: ~/filetransfer' with standard window controls. The terminal shows the following commands and output:

```
sneha@sneha-VirtualBox:~/filetransfer$ ls
clientside  file.txt  server.py
sneha@sneha-VirtualBox:~/filetransfer$ cat file.txt
python is portable language
sneha@sneha-VirtualBox:~/filetransfer$
```

Socket library provides interface of BSD socket. It can be used in all modern operating system like Unix system, Windows, MacOS etc.

Two parameter is passed in the socked instance named AF_INET and SOCK_STREAM.

- AF_INET :- The ipv4 address family. By passing this parameter we are telling server or client to connect to ipv4 address only.
- SOCK_STRAM :- It means that connection should be made using TCP protocol.

socket.gethostname(socket.gethostname())- To get the IP address of the local host. We used this since we want our program to run any system as it will automatically fetch the IP address of system.

Socket.connect(hostname,port) is connecting hostname on the port

Socket.bind() is used to host to the port

Socket.listen() is used to wait for client connection

Socket.accept() is used to accept connection of TCP client

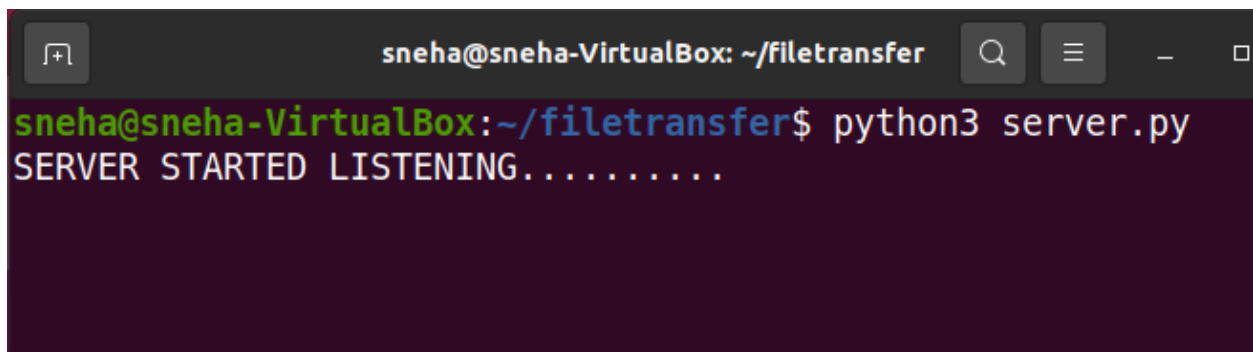
Socket.connect() is used by client side to initiate TCP connection to server

Socket.send() is used t transmit message

Socket.recv() is used to receives the message

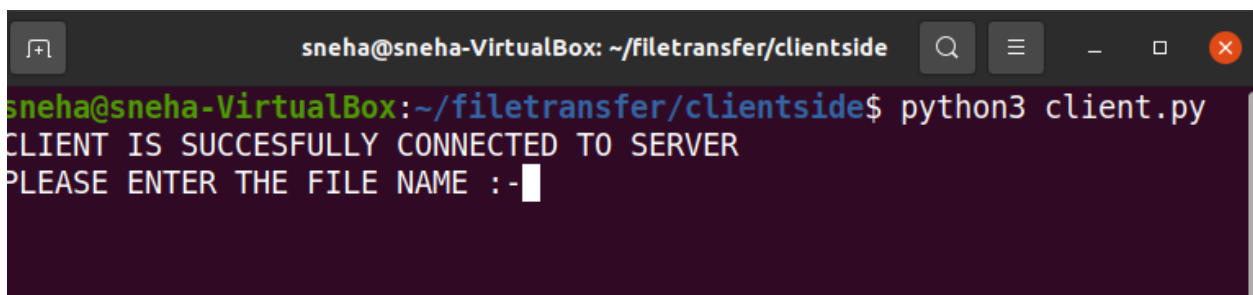
Socket .close() is used to close socket so that it does not accept the data.

First we will run server.py to let server listen on port

A terminal window titled 'sneha@sneha-VirtualBox: ~/filetransfer'. The prompt is 'sneha@sneha-VirtualBox:~/filetransfer\$'. The command 'python3 server.py' has been executed, and the output is 'SERVER STARTED LISTENING.....'.

```
sneha@sneha-VirtualBox: ~/filetransfer
sneha@sneha-VirtualBox:~/filetransfer$ python3 server.py
SERVER STARTED LISTENING.....
```

After that we will run client.py

A terminal window titled 'sneha@sneha-VirtualBox: ~/filetransfer/clientside'. The prompt is 'sneha@sneha-VirtualBox:~/filetransfer/clientside\$'. The command 'python3 client.py' has been executed, and the output is 'CLIENT IS SUCCESFULLY CONNECTED TO SERVER' followed by 'PLEASE ENTER THE FILE NAME :-' and a cursor.

```
sneha@sneha-VirtualBox: ~/filetransfer/clientside
sneha@sneha-VirtualBox:~/filetransfer/clientside$ python3 client.py
CLIENT IS SUCCESFULLY CONNECTED TO SERVER
PLEASE ENTER THE FILE NAME :-
```

Simultaneously server will run

```
sneha@sneha-VirtualBox: ~/filetransfer
sneha@sneha-VirtualBox:~/filetransfer$ python3 server.py
SERVER STARTED LISTENING.....
('127.0.0.1', 35342) SEVER IS BEEN CONNECTED
PLEASE ENTER THE NAME OF THE FILE TO BE TRANSFERED : 
```

```
sneha@sneha-VirtualBox: ~/filetransfer
sneha@sneha-VirtualBox:~/filetransfer$ python3 server.py
SERVER STARTED LISTENING.....
('127.0.0.1', 35342) SEVER IS BEEN CONNECTED
PLEASE ENTER THE NAME OF THE FILE TO BE TRANSFERED :

sneha@sneha-VirtualBox: ~/filetransfer/clientside
sneha@sneha-VirtualBox:~/filetransfer/clientside$ python3 client.py
CLIENT IS SUCCESFULLY CONNECTED TO SERVER
PLEASE ENTER THE FILE NAME :-
```

Now will specify the file name in client we want from server

```
sneha@sneha-VirtualBox: ~/filetransfer/clientside
sneha@sneha-VirtualBox:~/filetransfer/clientside$ python3 client.py
CLIENT IS SUCCESFULLY CONNECTED TO SERVER
PLEASE ENTER THE FILE NAME :-file.txt
```

Now we have to specify the name of file from server we want to transfer

```
sneha@sneha-VirtualBox: ~/filetransfer
sneha@sneha-VirtualBox:~/filetransfer$ python3 server.py
SERVER STARTED LISTENING.....
('127.0.0.1', 35342) SEVER IS BEEN CONNECTED
PLEASE ENTER THE NAME OF THE FILE TO BE TRANSFERED : file.txt
```

The client has successfully received the file

```
sneha@sneha-VirtualBox: ~/filetransfer
sneha@sneha-VirtualBox:~/filetransfer$ python3 server.py
SERVER STARTED LISTENING.....
('127.0.0.1', 35342) SEVER IS BEEN CONNECTED
PLEASE ENTER THE NAME OF THE FILE TO BE TRANSFERED : file.txt
Data has been transmitted successfully
sneha@sneha-VirtualBox:~/filetransfer$

sneha@sneha-VirtualBox: ~/filetransfer/clientside
sneha@sneha-VirtualBox:~/filetransfer/clientside$ python3 client.py
CLIENT IS SUCCESFULLY CONNECTED TO SERVER
PLEASE ENTER THE FILE NAME :-file.txt
FILE RECIEVED SUCCESFULLY
sneha@sneha-VirtualBox:~/filetransfer/clientside$
```

In the client-side folder file is received

```
sneha@sneha-VirtualBox: ~/filetransfer/clientside
sneha@sneha-VirtualBox:~/filetransfer/clientside$ ls
client.py  file.txt
sneha@sneha-VirtualBox:~/filetransfer/clientside$ cat file.txt
python is portable language
sneha@sneha-VirtualBox:~/filetransfer/clientside$
```