CS 549 Keyur Doshi – 10405923 – Assignment 1 – FTP

File transfer using FTP in Active and Passive in Local and Remote Server.

Explanations and Codes:
In POM file, default Ec2 server external domain name space is used to access ftp server and transfer files.
ec2-54-152-255-255.compute-1.amazonaws.com

To bind connection from client to server,

```
IServer server = null;
SERVER_MACHINE = serverMachine;
```

// Start by creating the string for the host

```
String host = "rmi://" + serverMachine + ":" + serverPort + "/" + serverName;
```

//Establish the Iserver for the Client and Server to communicate

```
Context con = new InitialContext();
IServerFactory fact = (IServerFactory) con.lookup(host);
System.out.println(con.lookup(host));
server = fact.createServer();
```

Active Mode:
In Active mode, Client acts like a server. It creates sockets and listen on socket and wait to hear from server. The server initiates the connection and connect to the socket then client accept the connection and finishes the connection.

Passive Mode:
In Passive mode, Client initiates server and server creates socket and reply back with port. Then client initiates data connection and connects to the socket on server. Server accepts the connection and complete the connection.

Here given below is the code that runs in Active and Passive mode.

For Client side: Active and Passive mode

```
private static class GetThread implements Runnable {
private ServerSocket dataChan = null;
private FileOutputStream file = null;
public GetThread(ServerSocket s, FileOutputStream f) {
dataChan = s;
file = f;
}
public void run() {
try {
/*
* TODO: Complete this thread.
*/
```

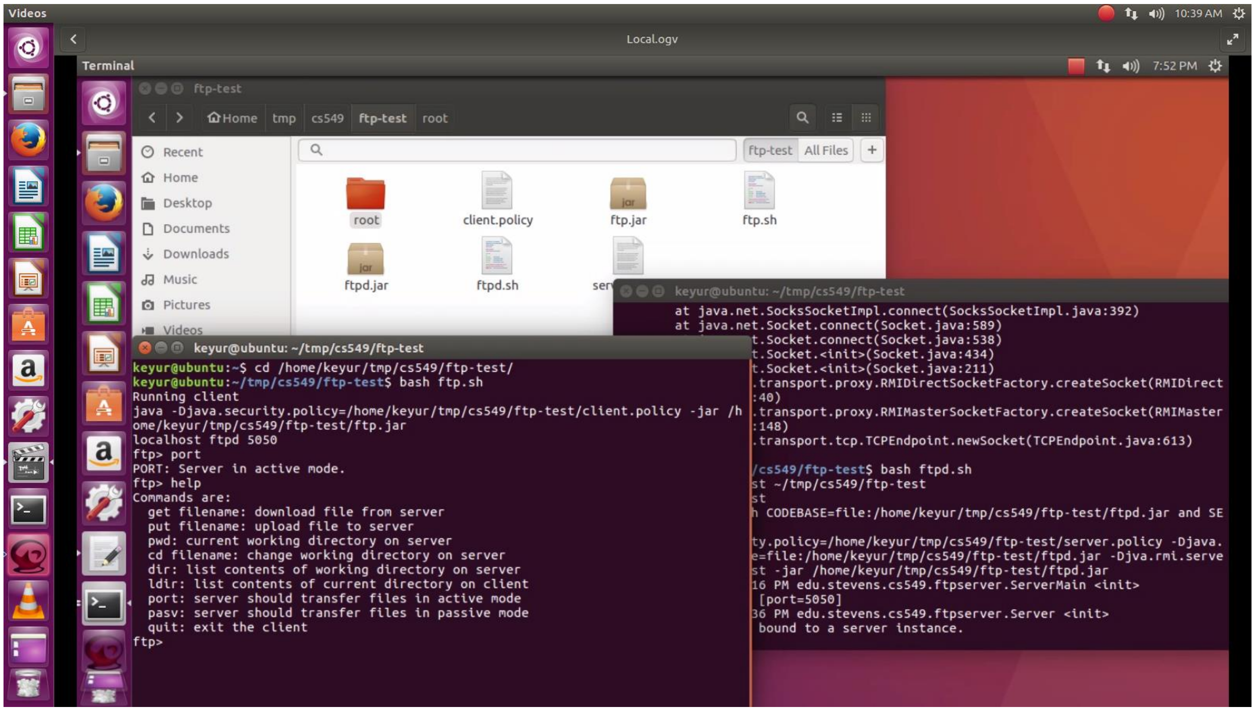
```
//The socket for the data transfer
Socket xfer = dataChan.accept();
//The input stream to read from the socket
byte[] bytes = new byte[1024];
int bytereads = 0;
while ((bytereads = buf.read(bytes)) > 0) {
f.write(bytes, 0, bytereads);
}f
.close();
if(buf!=null)
buf.close();
} /*
* End TODO
*/
} catch (IOException e) {
msg("Exception: " + e);
e.printStackTrace();
}}}
public void put(String[] inputs) {
if (inputs.length == 2) {
try {
/*
* TODO: Finish put.
*/
if (mode == Mode.PASSIVE) {
Socket xfer = new Socket(serverAddress,
serverSocket.getPort());
svr.put(inputs[1]);
BufferedOutputStream bout = new BufferedOutputStream(
xfer.getOutputStream());
byte[] bytes = new byte[1024];
int bytereads = 0;
InputStream f = new FileInputStream(inputs[1]);
BufferedInputStream bin = new BufferedInputStream(f);
while ((bytereads = bin.read(bytes)) != -1) {
bout.write(bytes, 0, bytereads);
} if(bout!=null)
bout.close();
if(bin!=null)
bin.close();
f.close();
```

Server Side Active and Passive

```
public void run() {
/*
* TODO: Process a client request to transfer a file.
*/
try {
```

```
Socket client = dataChan.accept();
BufferedOutputStream bout = new BufferedOutputStream(
client.getOutputStream());
BufferedInputStream bin = new BufferedInputStream(file);
byte[] contents = new byte[1024];
int bytereads = 0;
while ((bytereads = bin.read(contents)) != -1) {
bout.write(contents, 0, bytereads);
} if(bout!=null)
bout.close();
if(bin!=null)
bin.close();
file.close();
```

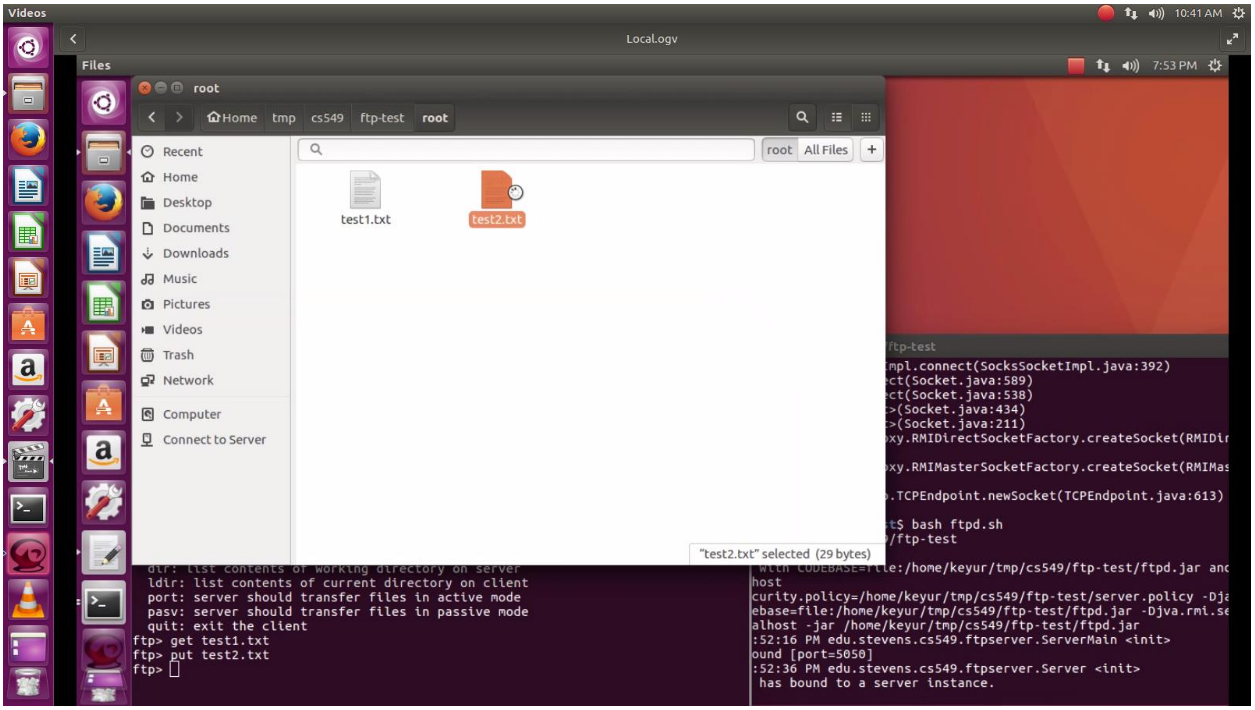
* Testing Locally...
Run ftpd.sh in the local machine.



```
keyur@ubuntu:~$ cd /home/keyur/tmp/cs549/ftp-test/
keyur@ubuntu:~/tmp/cs549/ftp-test$ bash ftp.sh
Running client
java -Djava.security.policy=/home/keyur/tmp/cs549/ftp-test/client.policy -jar /h
ome/keyur/tmp/cs549/ftp-test/ftp.jar
localhost ftpd 5050
ftp> port
PORT: Server in active mode.
ftp> help
Commands are:
  get filename: download file from server
  put filename: upload file to server
  pwd: current working directory on server
  cd filename: change working directory on server
  dir: list contents of working directory on server
  ldir: list contents of current directory on client
  port: server should transfer files in active mode
  pasv: server should transfer files in passive mode
  quit: exit the client
ftp>
```

```
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
        at java.net.Socket.connect(Socket.java:589)
t.Socket.connect(Socket.java:538)
t.Socket.<init>(Socket.java:434)
t.Socket.<init>(Socket.java:211)
.transport.proxy.RMIDirectSocketFactory.createSocket(RMIDirect
:40)
.transport.proxy.RMIMasterSocketFactory.createSocket(RMIMaster
:148)
.transport.tcp.TCPEndpoint.newSocket(TCPEndpoint.java:613)

/cs549/ftp-test$ bash ftpd.sh
st ~/tmp/cs549/ftp-test
st
h CODEBASE=file:/home/keyur/tmp/cs549/ftp-test/ftpd.jar and SE

ty.policy=/home/keyur/tmp/cs549/ftp-test/server.policy -Djava.
e=file:/home/keyur/tmp/cs549/ftp-test/ftpd.jar -Djva.rmi.serve
st -jar /home/keyur/tmp/cs549/ftp-test/ftpd.jar
16 PM edu.stevens.cs549.ftpserver.ServerMain <init>
 [port=5050]
36 PM edu.stevens.cs549.ftpserver.Server <init>
 bound to a server instance.
```

After get and put command executed successfully.

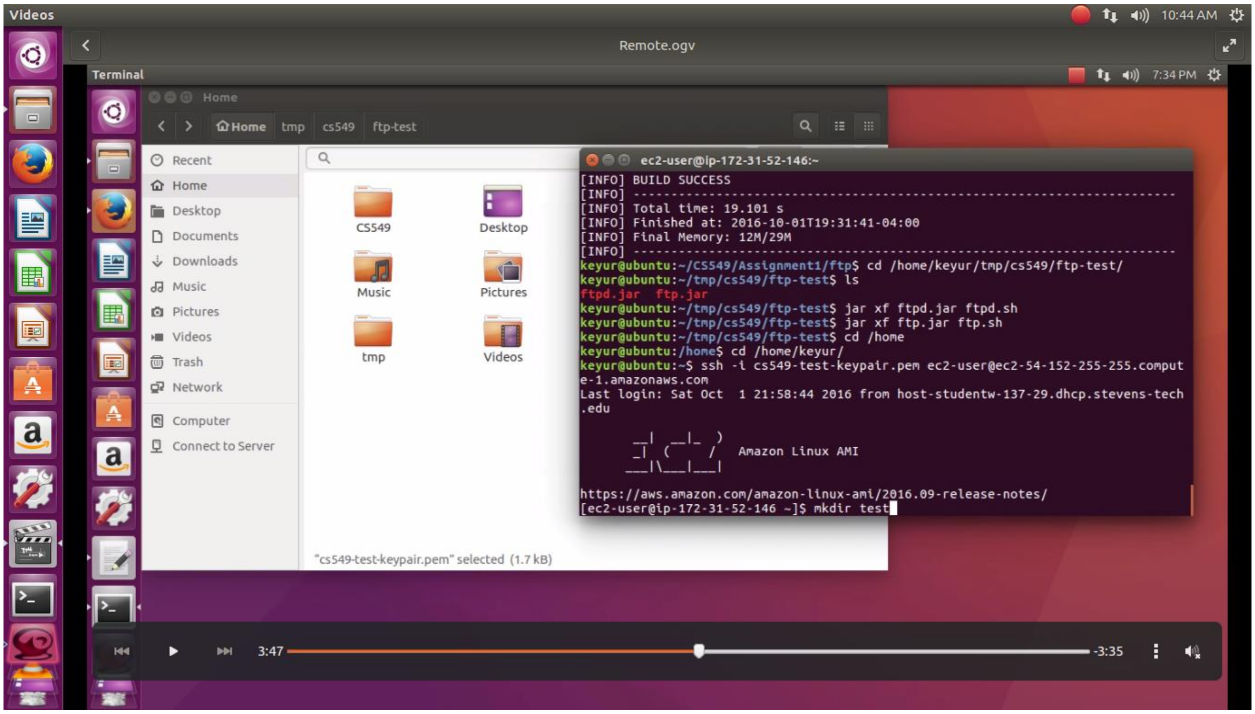


```
  dir: list contents of working directory on server
  ldir: list contents of current directory on client
  port: server should transfer files in active mode
  pasv: server should transfer files in passive mode
  quit: exit the client
ftp> get test1.txt
ftp> put test2.txt
ftp> 
```

```
Impl.connect(SocksSocketImpl.java:392)
ct(Socket.java:589)
ct(Socket.java:538)
>(Socket.java:434)
>(Socket.java:211)
xy.RMIDirectSocketFactory.createSocket(RMIDir

xy.RMIMasterSocketFactory.createSocket(RMIMas

.TCPEndpoint.newSocket(TCPEndpoint.java:613)

t$ bash ftpd.sh
/ftp-test

 with CODEBASE=file:/home/keyur/tmp/cs549/ftp-test/ftpd.jar and
host
curity.policy=/home/keyur/tmp/cs549/ftp-test/server.policy -Dja
ebase=file:/home/keyur/tmp/cs549/ftp-test/ftpd.jar -Djva.rmi.se
alhost -jar /home/keyur/tmp/cs549/ftp-test/ftpd.jar
:52:16 PM edu.stevens.cs549.ftpserver.ServerMain <init>
ound [port=5050]
:52:36 PM edu.stevens.cs549.ftpserver.Server <init>
 has bound to a server instance.
```

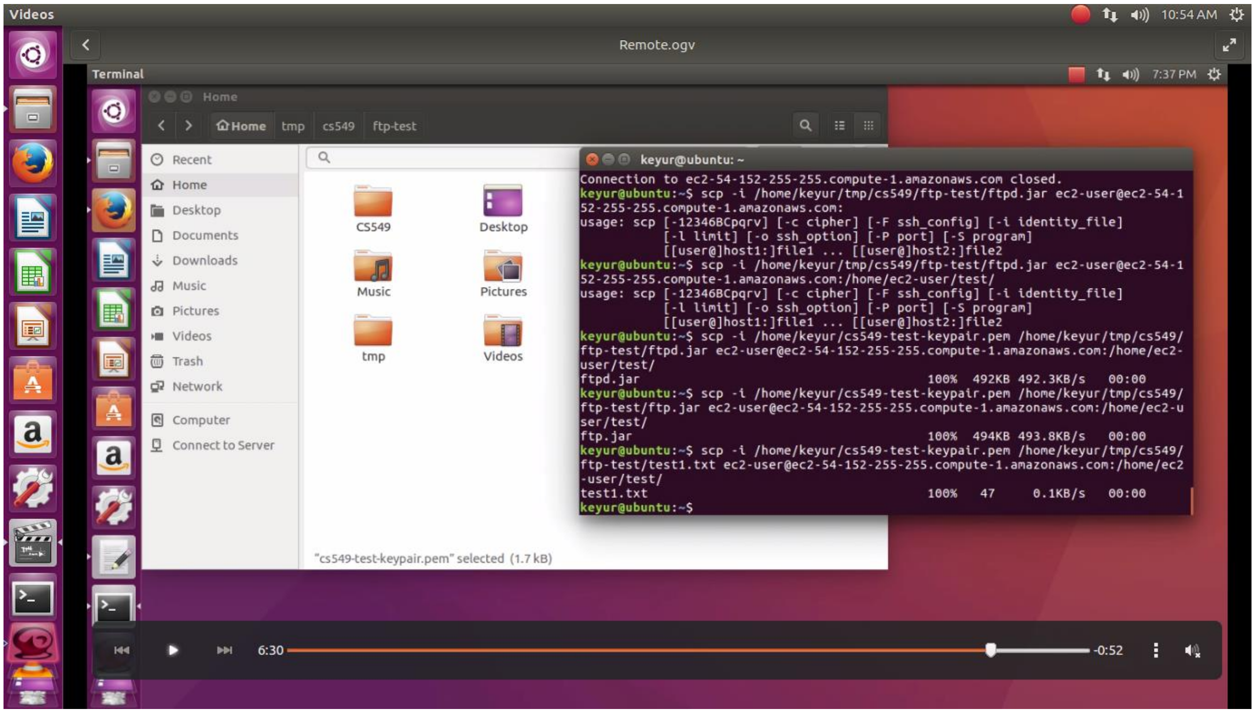
Testing on remote server.

Connecting to the Ec2 Instance via Secure shell script, Primary key and public DNS of the instance. First connecting instance via ssh command.



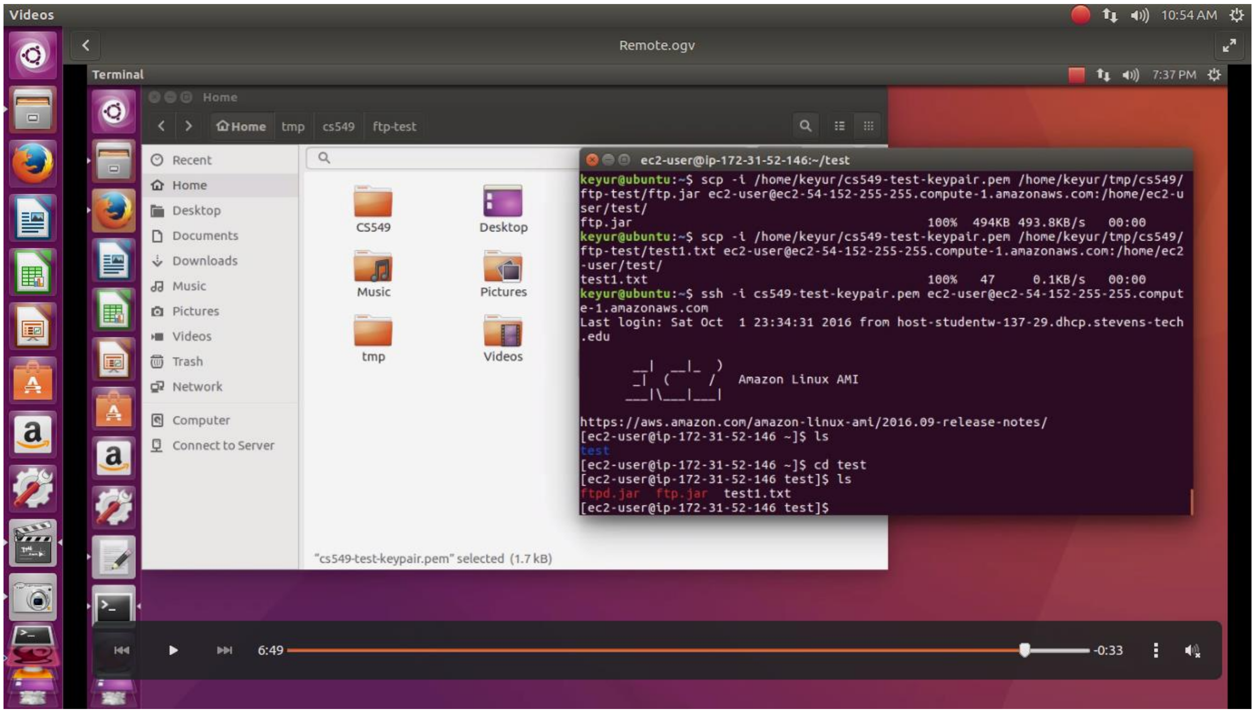
```
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 19.101 s
[INFO] Finished at: 2016-10-01T19:31:41-04:00
[INFO] Final Memory: 12M/29M
[INFO] ------------------------------------------------------------------------
keyur@ubuntu:~/CS549/Assignment1/ftp$ cd /home/keyur/tmp/cs549/ftp-test/
keyur@ubuntu:~/tmp/cs549/ftp-test$ ls
ftpd.jar  ftp.jar
keyur@ubuntu:~/tmp/cs549/ftp-test$ jar xf ftpd.jar ftpd.sh
keyur@ubuntu:~/tmp/cs549/ftp-test$ jar xf ftp.jar ftp.sh
keyur@ubuntu:~/tmp/cs549/ftp-test$ cd /home
keyur@ubuntu:/home$ cd /home/keyur/
keyur@ubuntu:~$ ssh -i cs549-test-keypair.pem ec2-user@ec2-54-152-255-255.comput
e-1.amazonaws.com
Last login: Sat Oct  1 21:58:44 2016 from host-studentw-137-29.dhcp.stevens-tech
.edu

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
[ec2-user@ip-172-31-52-146 ~]$ mkdir test
```

As on remoter server, passive mode should be activated and how to transfer jar and text file from client to server using Secure Copy (SCP) command.



```
Connection to ec2-54-152-255-255.compute-1.amazonaws.com closed.
keyur@ubuntu:~$ scp -i /home/keyur/tmp/cs549/ftp-test/ftpd.jar ec2-user@ec2-54-1
52-255-255.compute-1.amazonaws.com:
usage: scp [-12346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2
keyur@ubuntu:~$ scp -i /home/keyur/tmp/cs549/ftp-test/ftpd.jar ec2-user@ec2-54-1
52-255-255.compute-1.amazonaws.com:/home/ec2-user/test/
usage: scp [-12346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2
keyur@ubuntu:~$ scp -i /home/keyur/cs549-test-keypair.pem /home/keyur/tmp/cs549/
ftp-test/ftpd.jar ec2-user@ec2-54-152-255-255.compute-1.amazonaws.com:/home/ec2-
user/test/
ftpd.jar                                      100%  492KB 492.3KB/s   00:00
keyur@ubuntu:~$ scp -i /home/keyur/cs549-test-keypair.pem /home/keyur/tmp/cs549/
ftp-test/ftp.jar ec2-user@ec2-54-152-255-255.compute-1.amazonaws.com:/home/ec2-u
ser/test/
ftp.jar                                       100%  494KB 493.8KB/s   00:00
keyur@ubuntu:~$ scp -i /home/keyur/cs549-test-keypair.pem /home/keyur/tmp/cs549/
ftp-test/test1.txt ec2-user@ec2-54-152-255-255.compute-1.amazonaws.com:/home/ec2
-user/test/
test1.txt                                     100%   47     0.1KB/s   00:00
keyur@ubuntu:~$
```

List of files and directories on EC2 instance after transferring files from client machine.



```
keyur@ubuntu:~$ scp -i /home/keyur/cs549-test-keypair.pem /home/keyur/tmp/cs549/
ftp-test/ftp.jar ec2-user@ec2-54-152-255-255.compute-1.amazonaws.com:/home/ec2-u
ser/test/
ftp.jar                                       100%  494KB 493.8KB/s   00:00
keyur@ubuntu:~$ scp -i /home/keyur/cs549-test-keypair.pem /home/keyur/tmp/cs549/
ftp-test/test1.txt ec2-user@ec2-54-152-255-255.compute-1.amazonaws.com:/home/ec2
-user/test/
test1.txt                                     100%   47     0.1KB/s   00:00
keyur@ubuntu:~$ ssh -i cs549-test-keypair.pem ec2-user@ec2-54-152-255-255.comput
e-1.amazonaws.com
Last login: Sat Oct  1 23:34:31 2016 from host-studentw-137-29.dhcp.stevens-tech
.edu

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
[ec2-user@ip-172-31-52-146 ~]$ ls
test
[ec2-user@ip-172-31-52-146 ~]$ cd test
[ec2-user@ip-172-31-52-146 test]$ ls
ftpd.jar  ftp.jar  test1.txt
[ec2-user@ip-172-31-52-146 test]$
```