**CS 6390 – Advanced Computer Networks**

**Final Project**

**Synchronize information between laptop, cloud and smartphone**

**Submitted by:**

**Group 5**

Apoorva D Ajay – ada140530

Nischitha G K Reddy – nxr142430

Chitra Harihara Pranadarthan – chx141330

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Topic** | **Page No** |
| 1 | Overview of the project | 3 |
| 2 | Introduction | 4 |
| 3 | Software requirements | 7 |
| 4 | Design | 8 |
| 5 | Cloud setup | 9 |
| 6 | Test case 3: Communication between Smartphone and Laptop | 14 |
| 7 | References | 17 |

**Overview of the project:**

***Stage1:***

1. Download netbeans or eclipse IDE, (for Java and c/c++)
   1. run a demo program to make sure it works
   2. Capture screen shot to show it.
2. Download Android software and the necessary development libraries.
   1. Run a demo program to make sure it works
   2. Capture screen shot(s) to show it works

Stage 1 part of the project was submitted on 09/29/2014.

***Stage2:***

To develop an application to synchronize information between laptop/computer, smartphone and a cloud. This sync could be appointments, some other info file. The main storage is on computer and cloud. Smartphone can be used to update info on other devices.

Two way communication between the devices should be established:

Client device is the device that initiates update.

Server device is going to accept the update.

1. Laptop initiates TCP or UDP communication with cloud.

    Send a file to the cloud.

    The cloud receives it and updates.

2. Cloud initiates TCP or UDP communication with the laptop.

    Send a file to the laptop.

    Laptop receives and updates the info.

3. Smartphone initiates TCP or UDP communication with the laptop.

    Send a file to the laptop.

    The laptop receives it and updates it.

4. Laptop initiates TCP or UDP communication with Smartphone.

Send a file to the laptop.

    The laptop receives it and updates it.

5. Smartphone initiates TCP or UDP communication with cloud.

    Send a file to cloud.

    Cloud receives it and updates it.

6. Cloud initiates TCP or UDP communication with the Smartphone.

    Send a file to the Smartphone.

    Smartphone receives and updates the info.

**INTRODUCTION**

TCP

The Transmission Control Protocol (TCP) is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks [1]

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from hard-wired connections to packet-switched or circuit-switched networks [1].

TCP Connection establishment:

To establish a connection, TCP uses a three-way [handshake](http://en.wikipedia.org/wiki/Handshake_(computing)). Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. To establish a connection, the three-way (or 3-step) handshake occurs:

1. SYN: The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A.

2. SYN-ACK: In response, the server replies with a SYN-ACK. The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B.

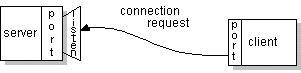
3. ACK: Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A+1, and the acknowledgement number is set to one more than the received sequence number i.e. B+1.

At this point, both the client and server have received an acknowledgment of the connection. The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

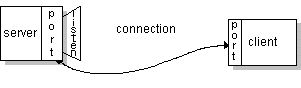
Socket

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

On the client-side: The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.



If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.



On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server.

The client and server can now communicate by writing to or reading from their socket [3].

Socket Programming in Java:

Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket.

The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

The following steps occur when establishing a TCP connection between two computers using sockets:

The server instantiates a ServerSocket object, denoting which port number communication is to occur on.

The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.

After the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.

The constructor of the Socket class attempts to connect the client to the specified server and port number. If communication is established, the client now has a Socket object capable of communicating with the server.

On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

After the connections are established, communication can occur using I/O streams. Each socket has both an OutputStream and an InputStream. The client's OutputStream is connected to the server's InputStream, and the client's InputStream is connected to the server's OutputStream [4].

Cloud

Cloud storage sounds like it has something to do with weather fronts and storm systems, it really refers to saving data to an off-site storage system maintained by a third party. Instead of storing information to your computer's hard drive or other local storage device, you save it to a remote database. The [Internet](http://computer.howstuffworks.com/internet/basics/internet-infrastructure.htm) provides the connection between your computer and the database.

On the surface, cloud storage has several advantages over traditional data storage. For example, if you store your data on a cloud storage system, you'll be able to get to that data from any location that has Internet access. You wouldn't need to carry around a physical storage device or use the same computer to save and retrieve your information. With the right storage system, you could even allow other people to access the data, turning a personal project into a collaborative effort, so cloud storage is convenient and offers more flexibility.

There are hundreds of different cloud storage systems. Some have a very specific focus, such as storing Web [e-mail](http://computer.howstuffworks.com/e-mail-messaging/email.htm) messages or [digital pictures](http://electronics.howstuffworks.com/cameras-photography/digital/digital-photography.htm). Others are available to store all forms of digital data. Some cloud storage systems are small operations, while others are so large that the physical equipment can fill up an entire warehouse. The facilities that house cloud storage systems are called data centers.

At its most basic level, a cloud storage system needs just one data server connected to the [Internet](http://computer.howstuffworks.com/internet/basics/internet-infrastructure.htm). A client (e.g., a computer user subscribing to a cloud storage service) sends copies of files over the Internet to the data server, which then records the information. When the client wishes to retrieve the information, he or she accesses the data server through a Web-based interface. The server then either sends the files back to the client or allows the client to access and manipulate the files on the server itself.

Cloud storage systems generally ­rely on hundreds of data servers. Because computers occasionally require maintenance or repair, it's important to store the same information on multiple machines. This is called redundancy. Without redundancy, a cloud storage system couldn't ensure clients that they could access their information at any given time. Most systems store the same data on [servers](http://computer.howstuffworks.com/web-server.htm) that use different [power supplies](http://computer.howstuffworks.com/power-supply.htm). That way, clients can access their data even if one power supply fails.

Not all cloud storage clients are worried about running out of storage space. They use cloud storage as a way to create backups of data. If something happens to the client's computer system, the data survives off-site [5].

Smartphone – Android

Android is a [mobile operating system](http://en.wikipedia.org/wiki/Mobile_operating_system) (OS) based on the [Linux kernel](http://en.wikipedia.org/wiki/Linux_kernel) and currently developed by [Google](http://en.wikipedia.org/wiki/Google). With a [user interface](http://en.wikipedia.org/wiki/User_interface) based on [direct manipulation](http://en.wikipedia.org/wiki/Direct_manipulation_interface), Android is designed primarily for [touchscreen](http://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](http://en.wikipedia.org/wiki/Smartphone) and [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer)

Android's [source code](http://en.wikipedia.org/wiki/Source_code) is released by Google under [open source](http://en.wikipedia.org/wiki/Open_source) licenses, although most Android devices ultimately ship with a combination of open source and proprietary software.[[3]](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-ars-irongrip-3) Initially developed by Android, Inc., which Google backed financially and later bought in 2005,[[23]](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-AndroidInc-23) Android was unveiled in 2007 along with the founding of the [Open Handset Alliance](http://en.wikipedia.org/wiki/Open_Handset_Alliance)—​a consortium of [hardware](http://en.wikipedia.org/wiki/Computer_hardware), software, and telecommunication companies devoted to advancing [open standards](http://en.wikipedia.org/wiki/Open_standard) for mobile devices.[[2]](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-AndroidAnnouncement-24)

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for [high-tech](http://en.wikipedia.org/wiki/High-tech) devices.[[25]](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-ars5th-25) Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users[[26]](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-apolroms-26) or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "[smartphone wars](http://en.wikipedia.org/wiki/Smartphone_wars)" between technology companies [2].

**Software requirements**

1. Java 8
2. Android sdk
3. Eclipse – ADT bundle
4. Emulator – Nexus 4 – API 21
5. Android studio - .8.14
6. AWS cloud
7. Putty and Winscp to connect and transfer files to AWS

**DESIGN**

The main storage is on computer and cloud. Smartphone is used to update info on other devices.

Two way communication between the devices is established:

Client device is the device that initiates update.

Server device is going to accept the update.

1. Laptop initiates TCP communication with cloud.

    Send a file to the cloud.

    The cloud receives it and updates.

2. Cloud initiates TCP communication with the laptop.

    Send a file to the laptop.

    Laptop receives and updates the info.

3. Smartphone initiates TCP communication with the laptop.

    Send a file to the laptop.

    The laptop receives it and updates it.

4. Laptop initiates TCP communication with Smartphone.

Send a file to the laptop.

    The laptop receives it and updates it.

5. Smartphone initiates TCP communication with cloud.

    Send a file to cloud.

    Cloud receives it and updates it.

6. Cloud initiates TCP communication with the Smartphone.

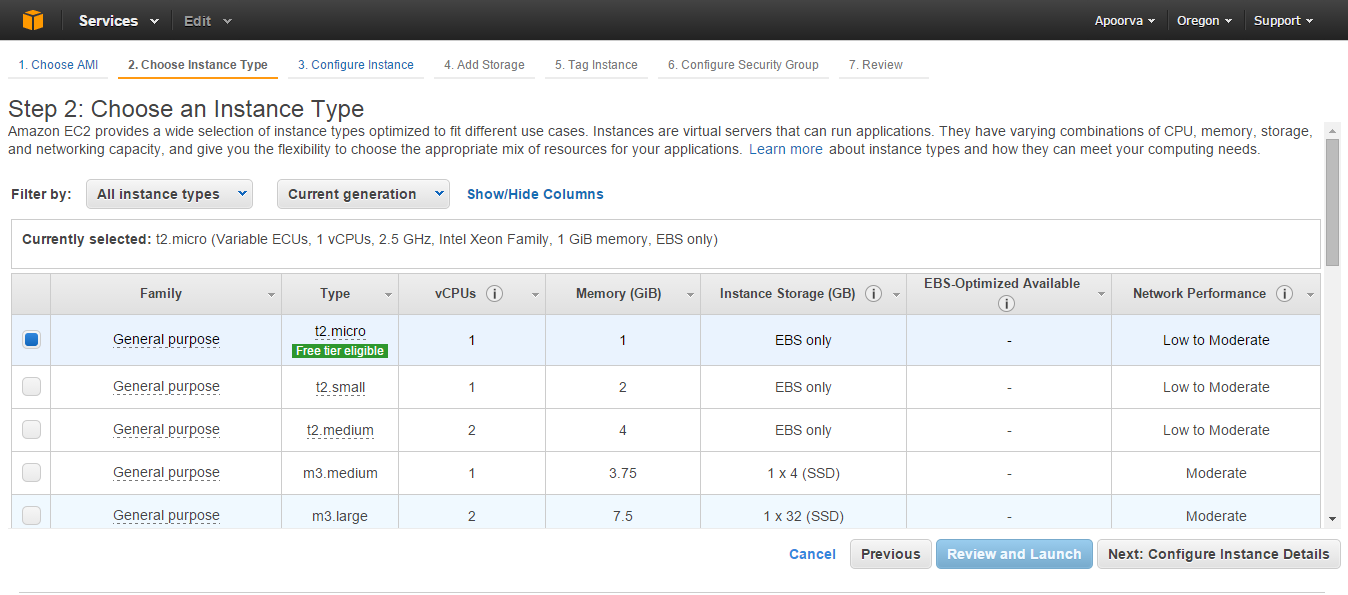
    Send a file to the Smartphone.

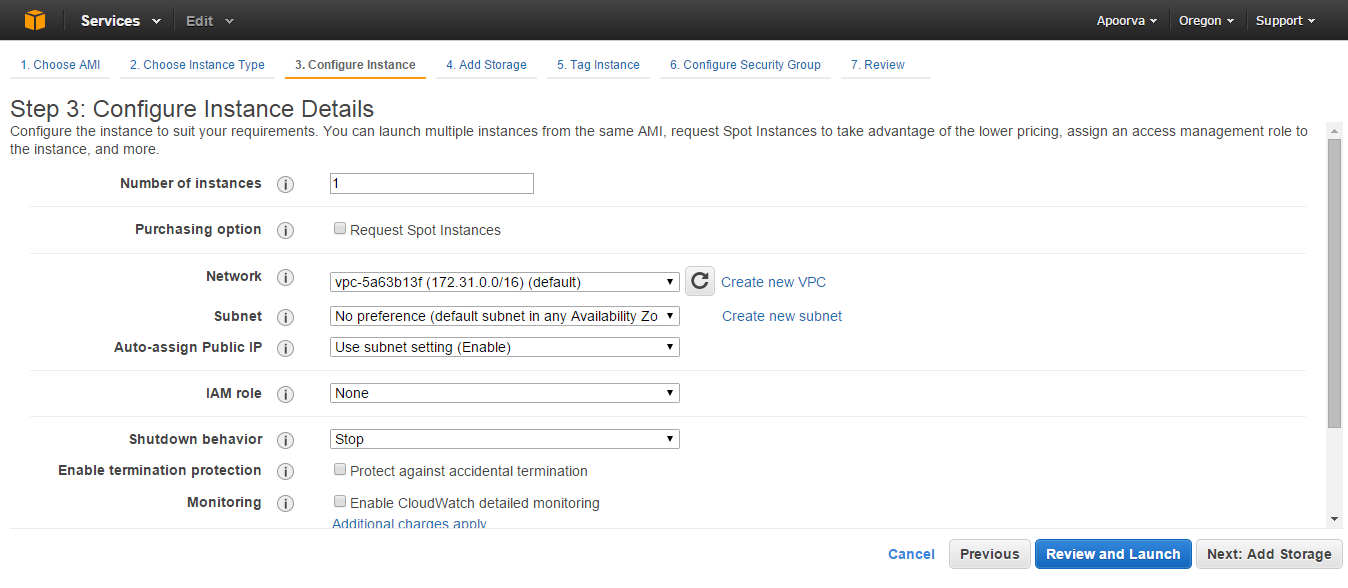
    Smartphone receives and updates the info.

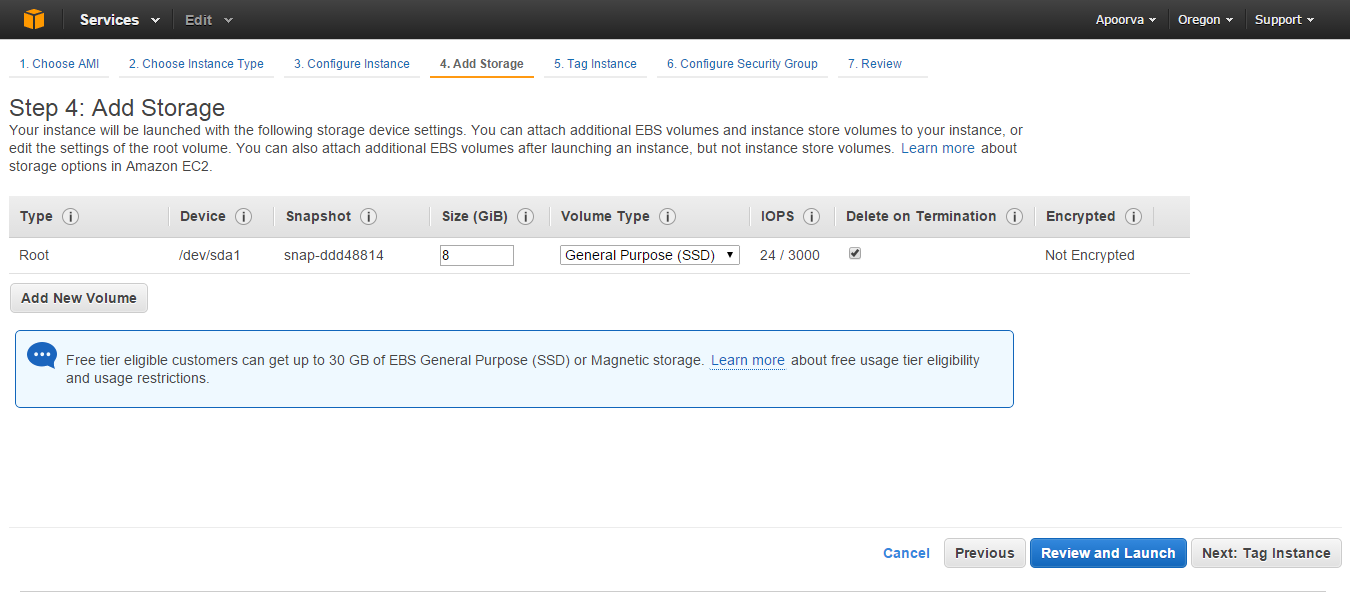
Cloud

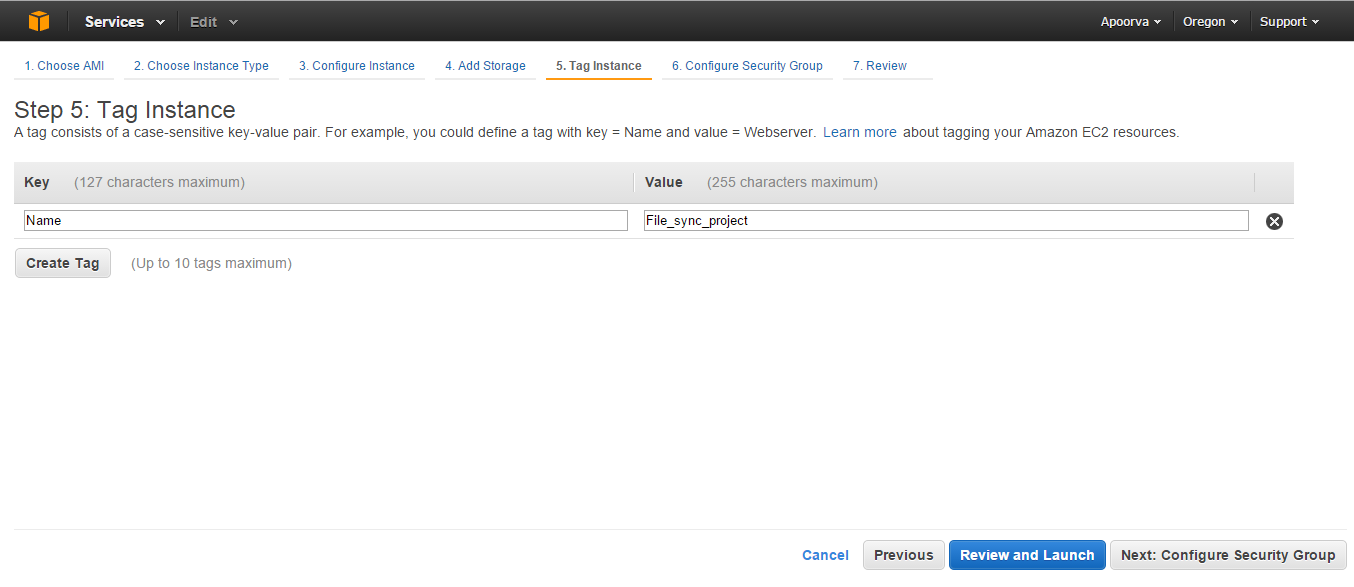
Laptop Smartphone

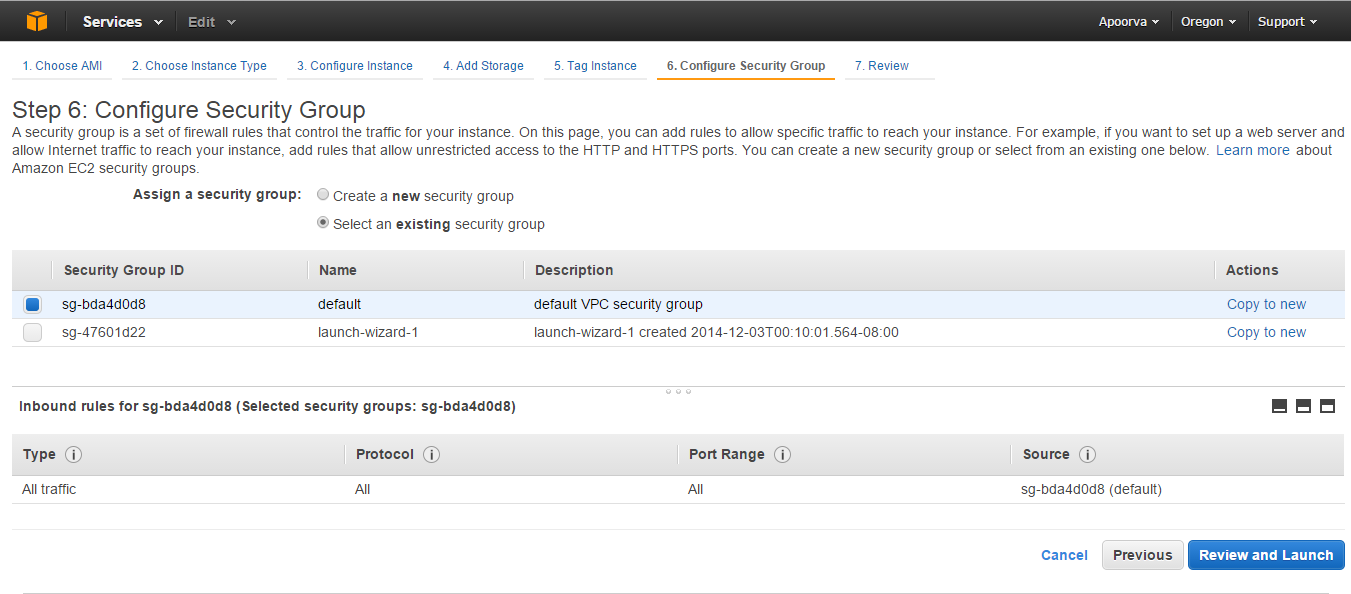
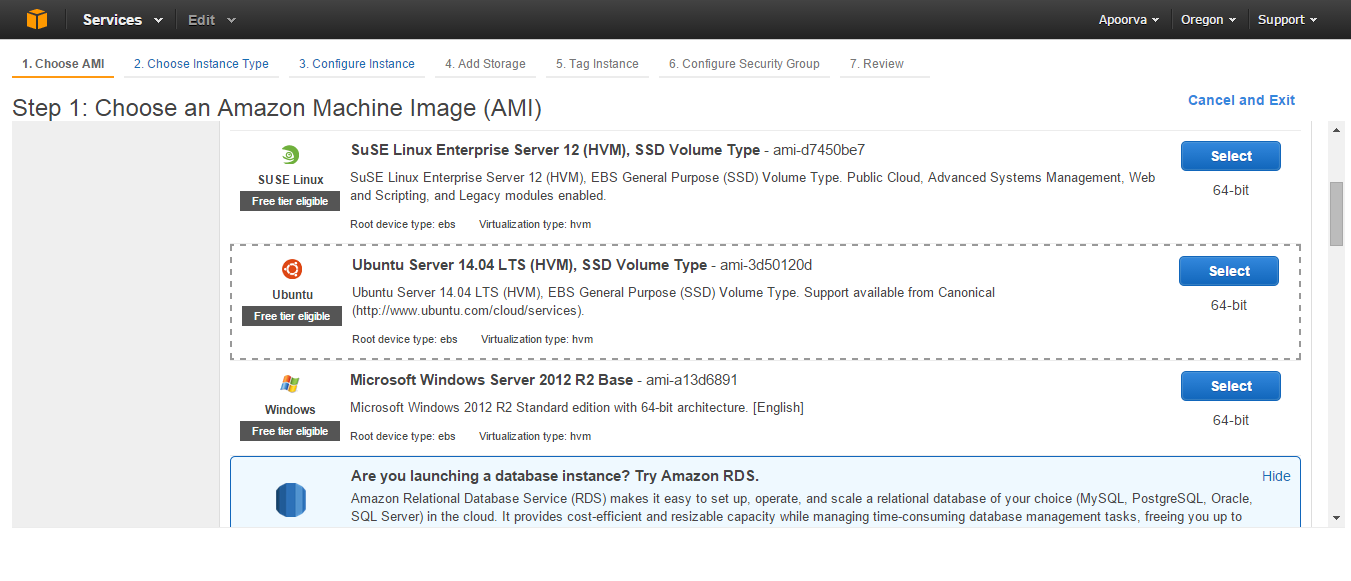
Cloud Setup

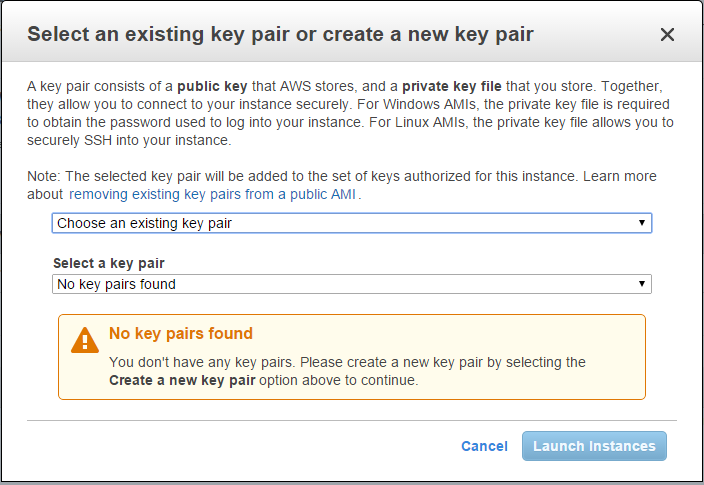


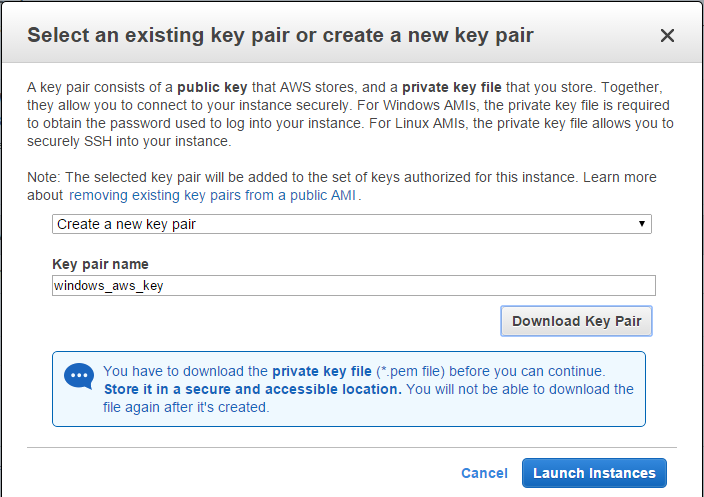


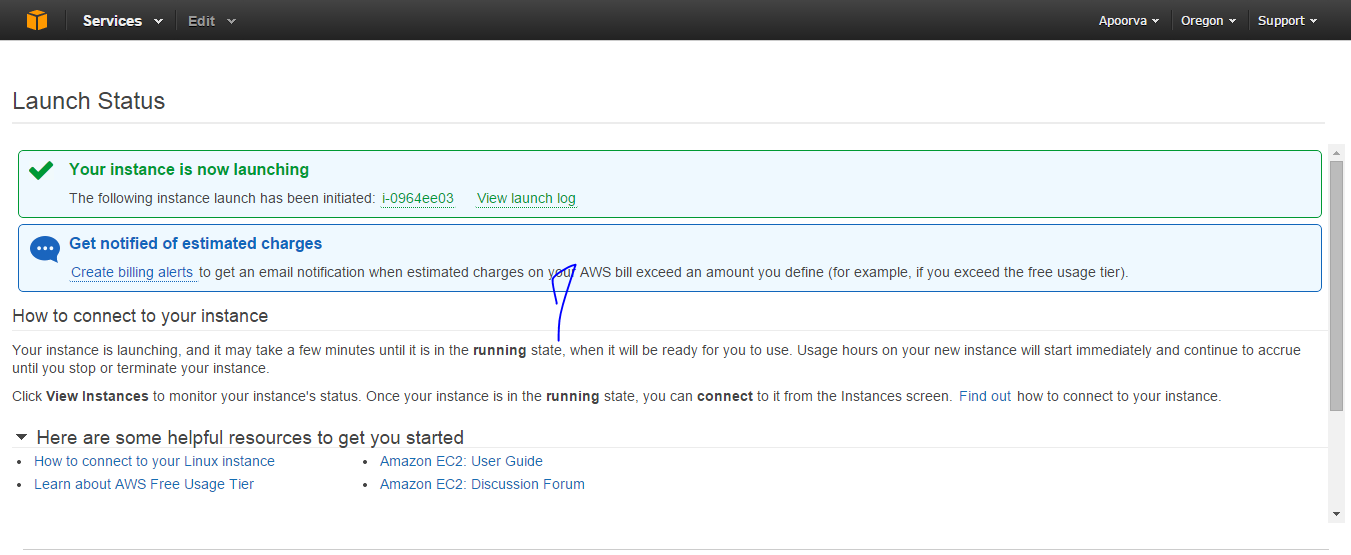


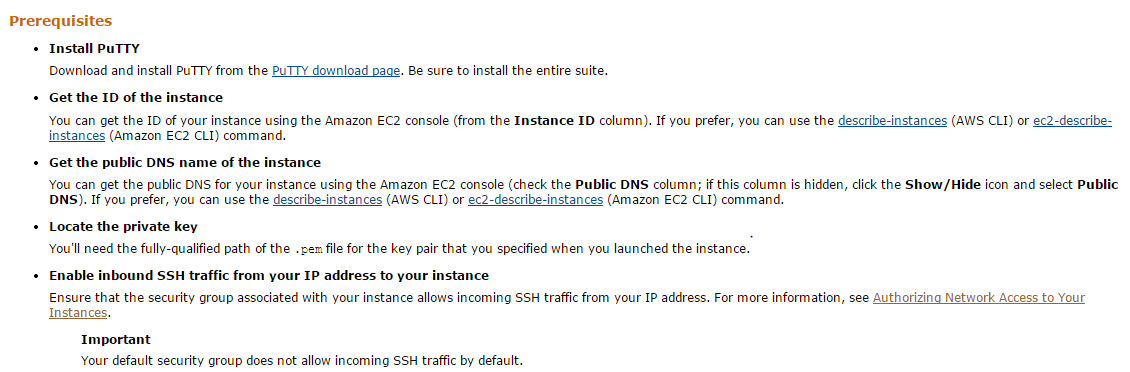












Test case 2: Communication between Smartphone and cloud

* Smartphone initiates TCP communication with cloud.

    Send a file to cloud.

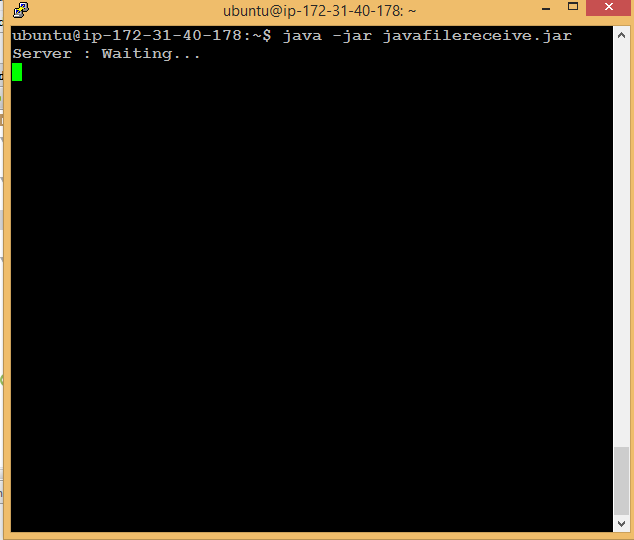
    Cloud receives it and updates it.

* Cloud initiates TCP communication with the Smartphone.

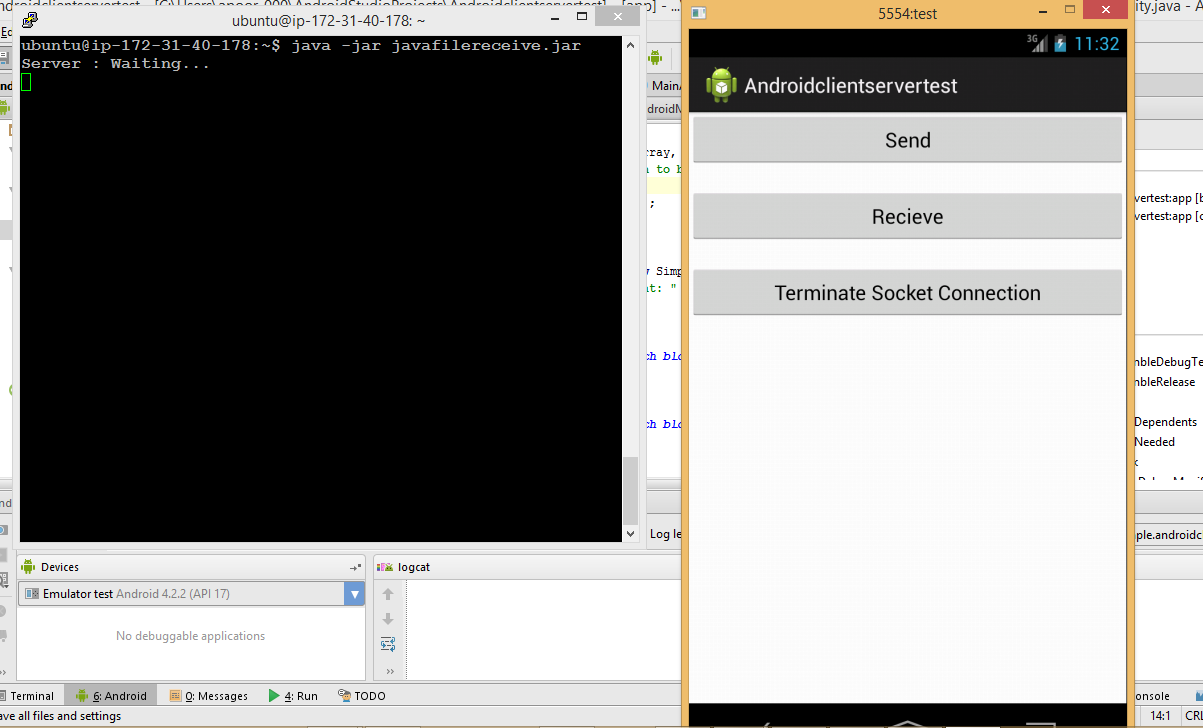
    Send a file to the Smartphone.

    Smartphone receives and updates the info.

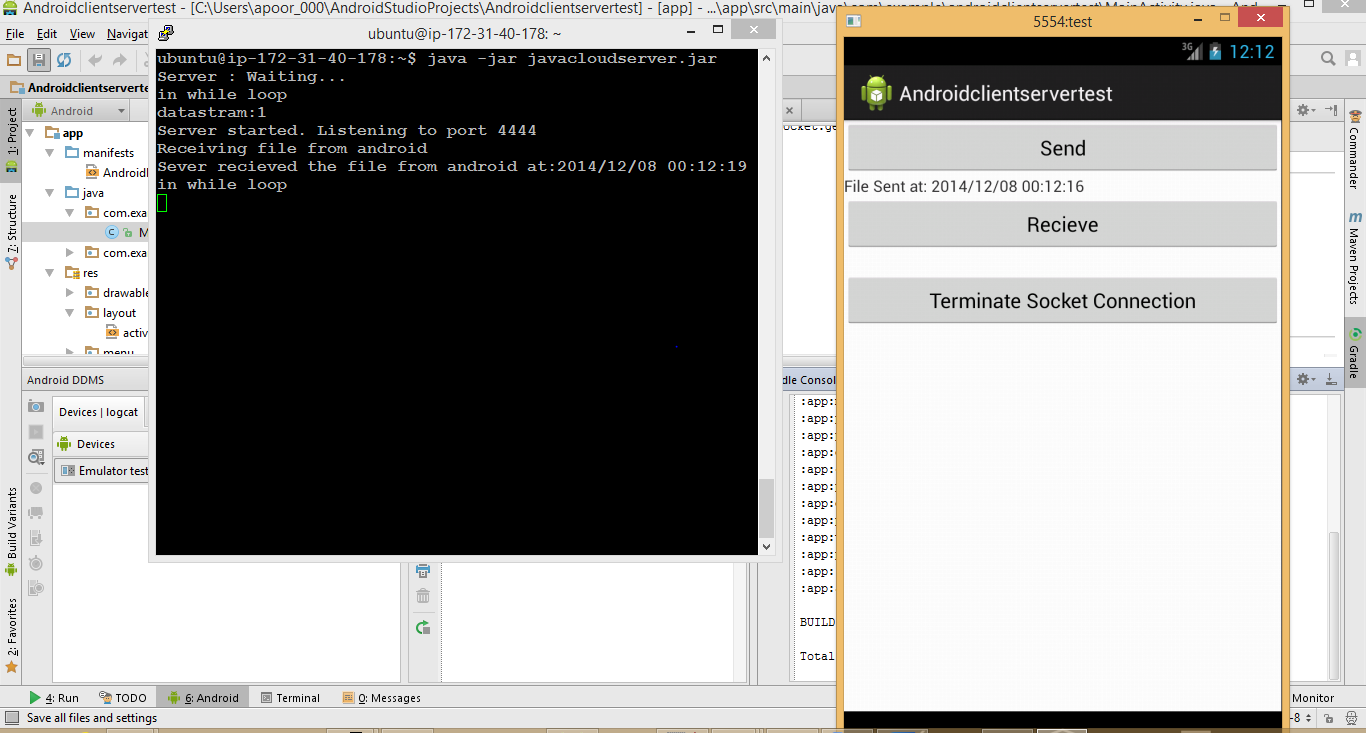
Run the server on the cloud first



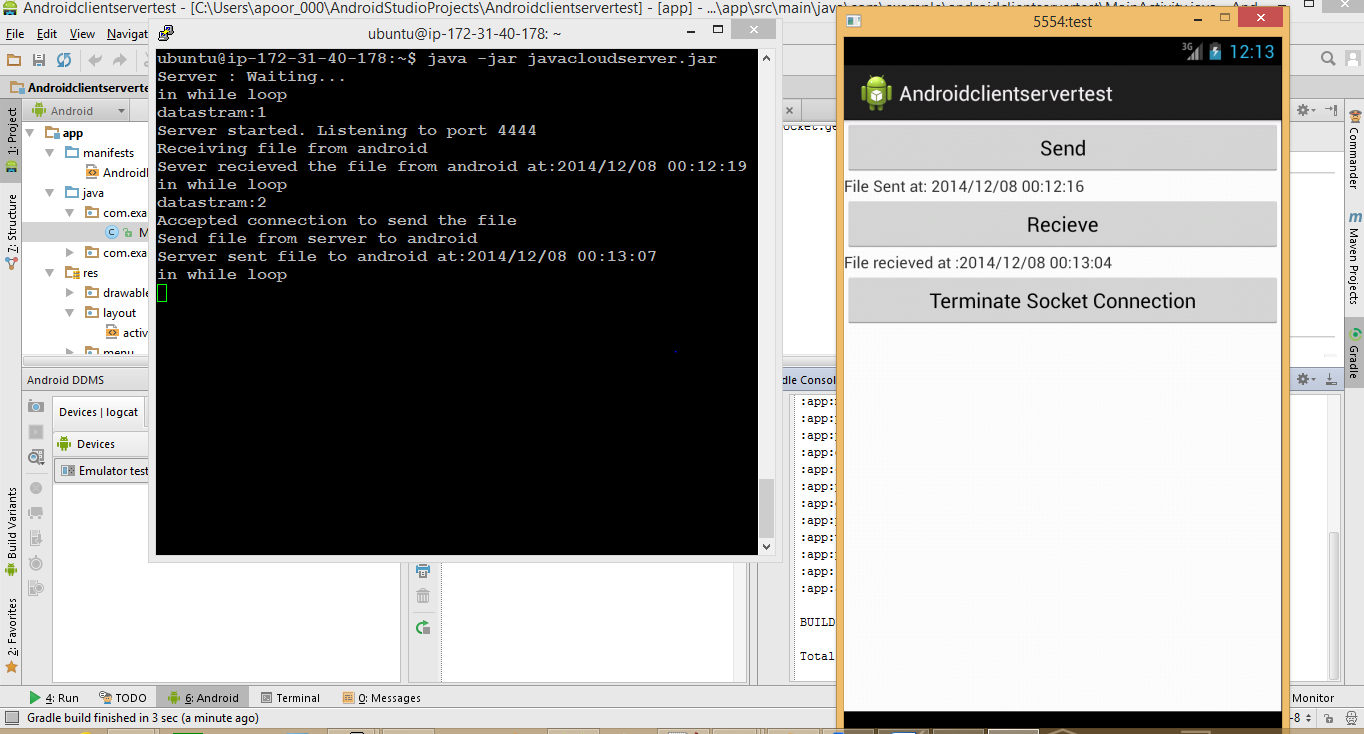
Run the android application



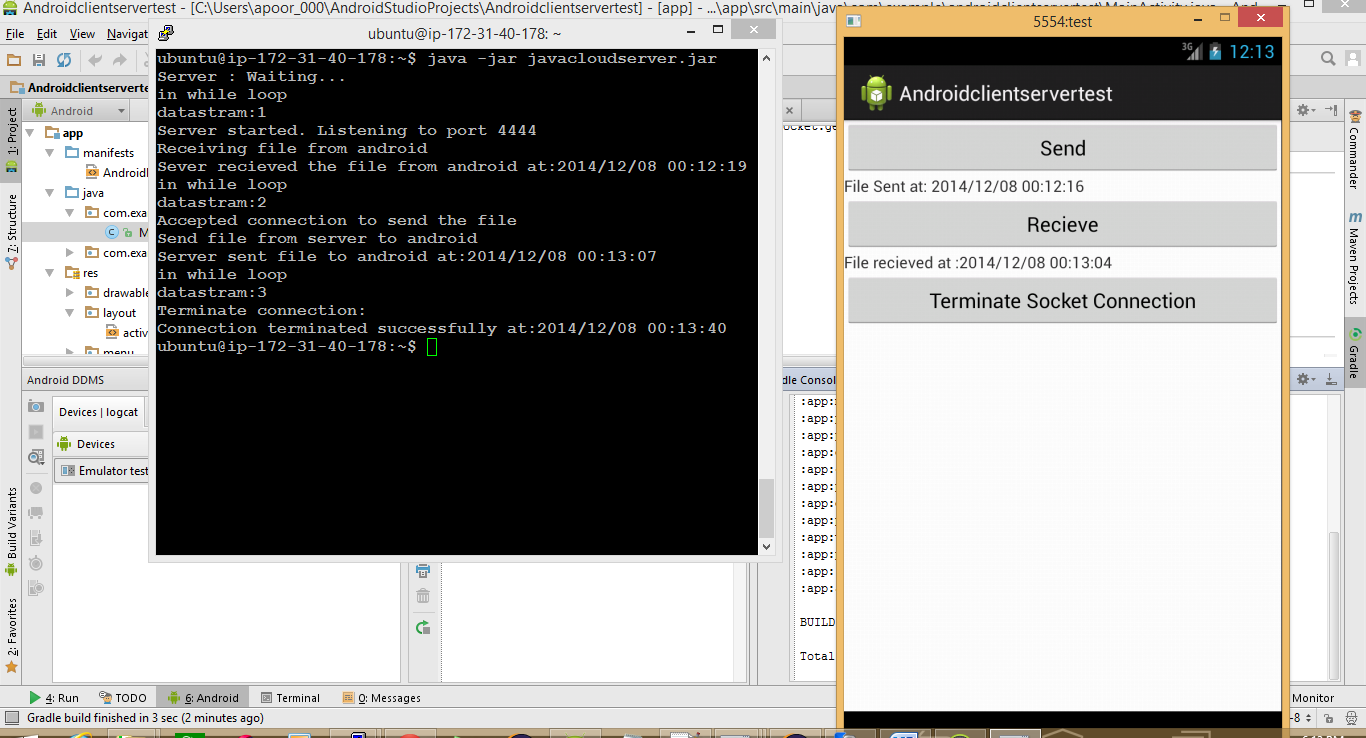
Click send button: Smartphone acts as client and cloud acts as server



Click receive button: Cloud acts as client and Smartphone acts as server



Click Terminate socket connection: Connection terminated



Test case 3: Communication between Smartphone and Laptop

* Smartphone initiates TCP communication with the laptop.

    Send a file to the laptop.

    The laptop receives it and updates it.

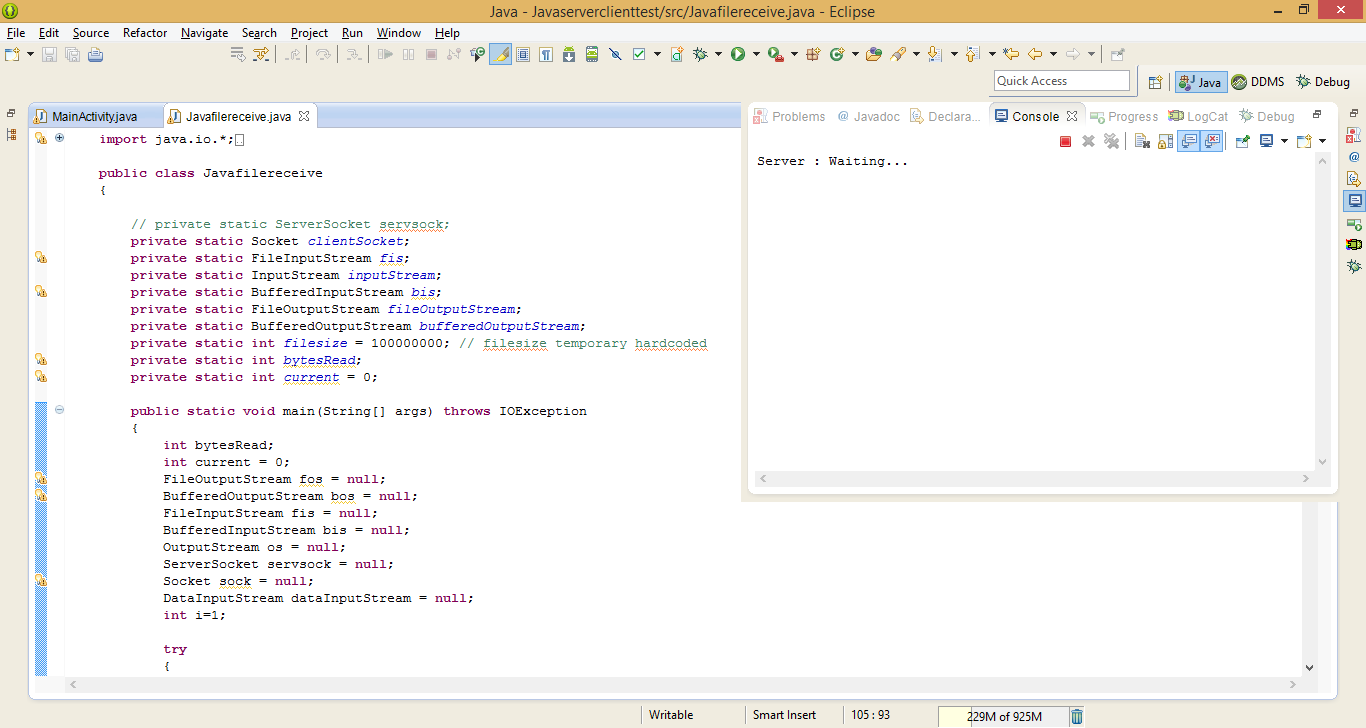
* Laptop initiates TCP communication with Smartphone.

Send a file to the laptop.

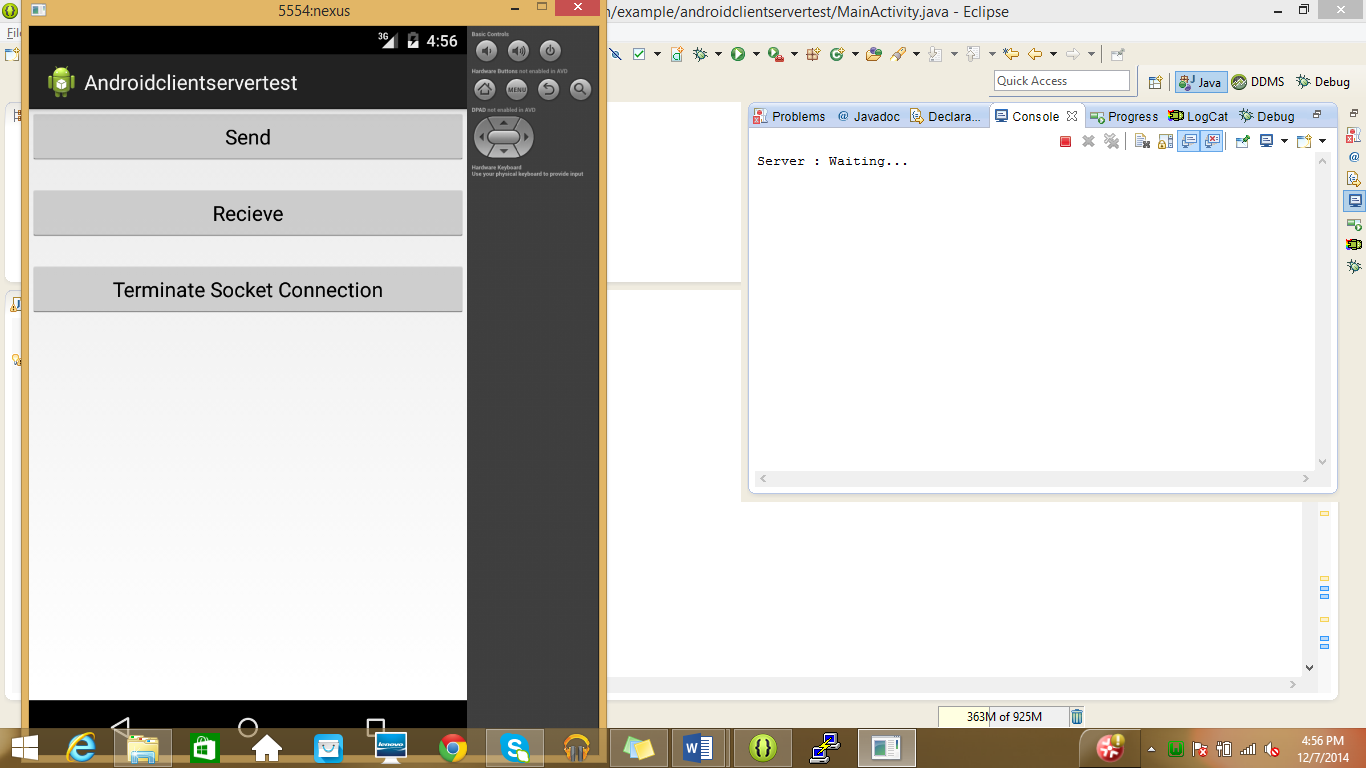
    The laptop receives it and updates it.

Test Results:

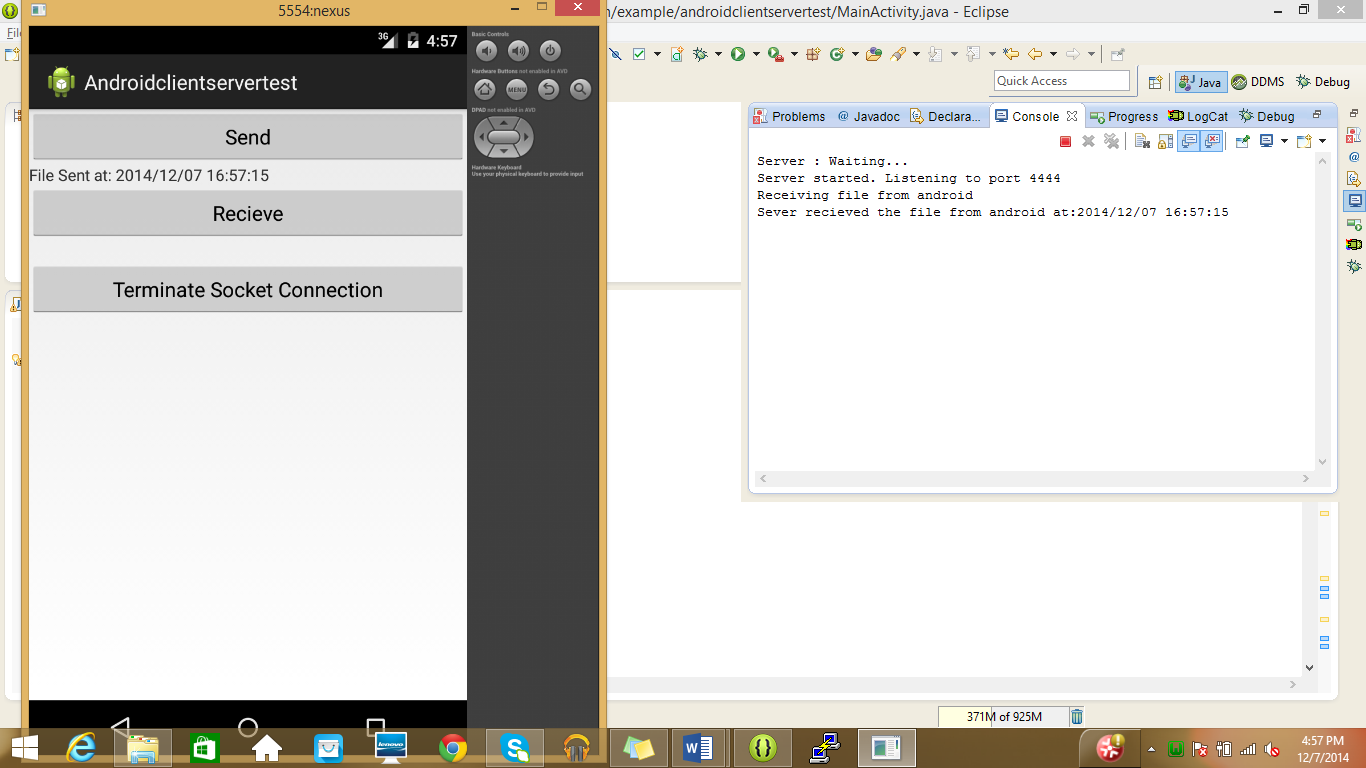
Run the server first



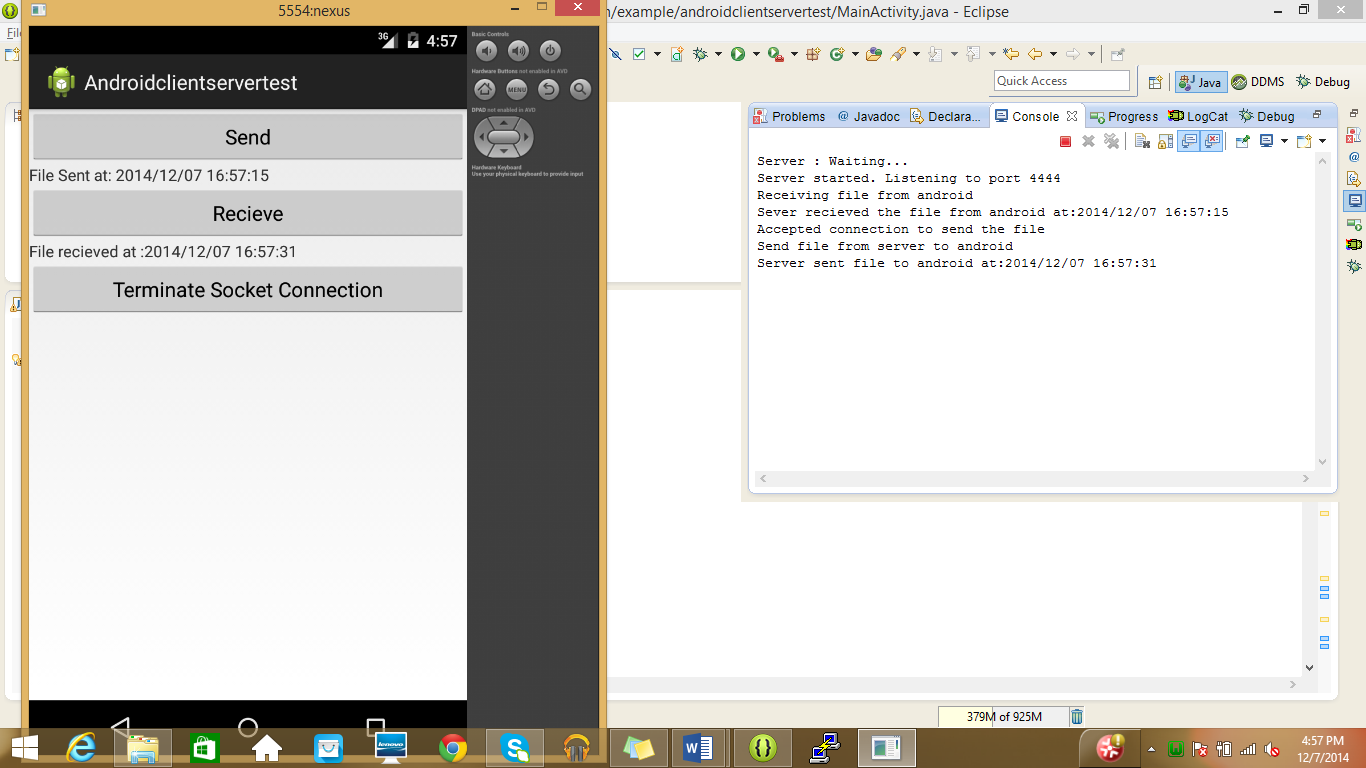
Run Android application



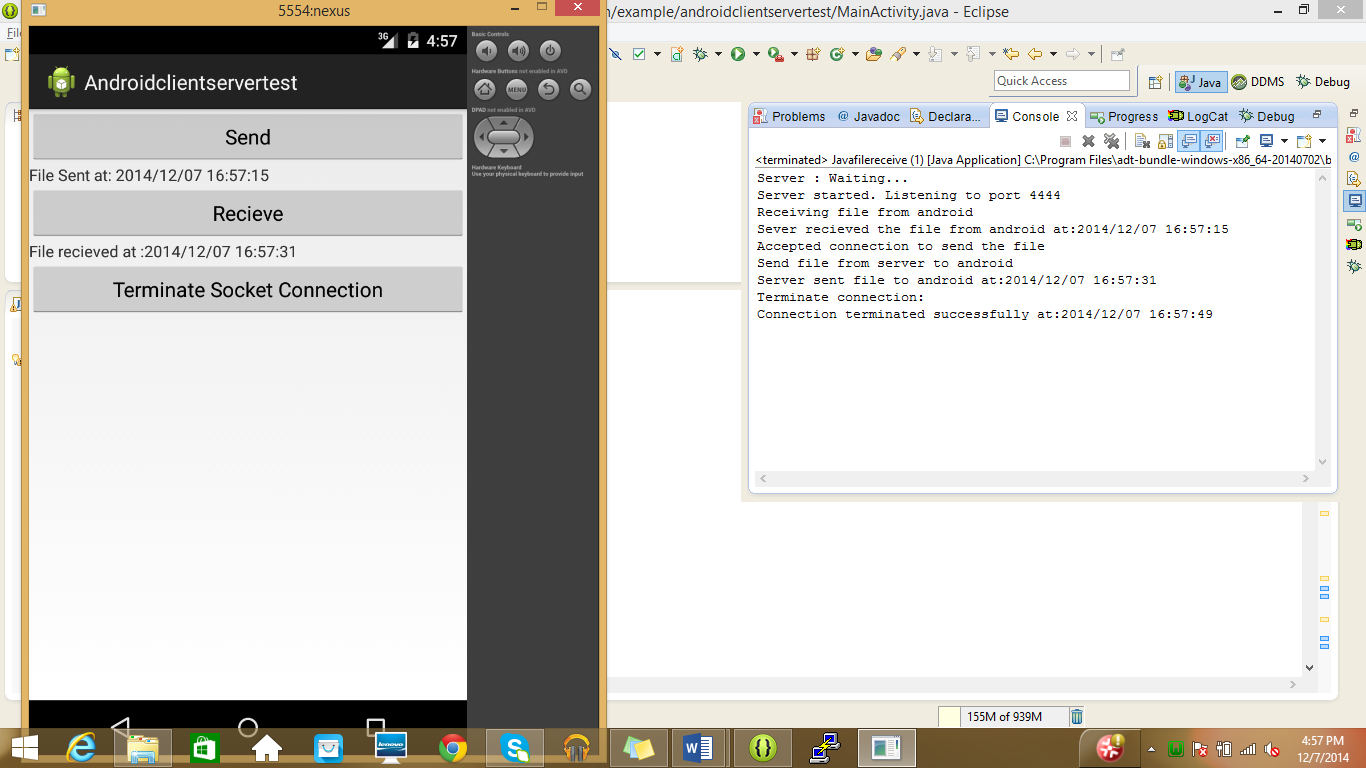
Click send button on android: Smartphone acts as client and laptop acts as server



Click receive button on android: Laptop acts as client and Smartphone acts as server



Click Terminate socket connection button on android: Connect will be terminated



**REFERENCES**

[1] RFC 793 Transmission control protocol Darpa internet program protocol specification

[2] Wikipedia – TCP, Android

[3] <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>

[4] <http://www.tutorialspoint.com/java/java_networking.htm>

[5] <http://computer.howstuffworks.com/cloud-computing/cloud-storage1.htm>