

Network Architecture-1

Project-2

Fall 2016



Submitted by:

Anusha Malineni (16233382)

Sri Sai Narayana Ram Gopal Mangena (16230843)

Sudhireddy Sainath Reddy (16232714)

Table of Contents

1. Introduction.....	2
2. About GENI and Putty.....	2
3. Project Description.....	7
3.1 Question-a.....	7
3.2 Question-b.....	11
3.3 Question-c	13
3.4 Question-d	16
4. References	19

1. Introduction

In this project, we are developing a simple Chat application by using TCP Client and Server. By using resources in GENI, we have connected several clients to single server. In Question a, we created a chat server that accepts a single client connection and displays messages from the client and the connection get terminated from both sides when the client types 'exit'. In Question b, when one client terminates, the server remains open for the other client's that wish to communicate. In Question c, the server will be able to handle multiple clients at a time and the server window displays the messages from the clients. In Question d, the server even echoes the messages from a client to all the remaining clients in the network.

2. About GENI and Putty

We need to first create a GENI account and a slice where we can reserve our resources on which we can work.

GENI Account Creation:

GENI provides a virtual laboratory for networking and distributed systems research and education.

1. Login to portal.geni.net
2. Activate the GENI account with the required credentials.
3. Download the SSH keys (Putty) for authentication process.
4. Using the Putty key generator, generate a private key which will be used to open the Client and Server windows.

Steps involved:

Step 1:

After logging in to GENI Portal, we have downloaded the SSH keypairs for authentication purposes.

SSH Keys

Name	Description	Public Key	Private Key	PuTTY	Edit	Delete
id_geni_ssh_rsa 4d:0f:f0:a8:17:38:f8:10:b3:72:fb:da:de:ae:2d:50	Generated SSH keypair	Download Public Key	Download Private Key	Download PuTTY Key	Edit	Delete

On Linux and Mac systems and for most Windows SSH clients (not PuTTY), do:

- Download your private key.
- On Windows, just point your SSH client (not PuTTY) to the downloaded private key.
- On Linux and Mac, open a terminal.
 - Store your key under ~/.ssh/ :
 - If the directory does not exist, create it:


```
mkdir ~/.ssh
```
 - Move the key to ~/.ssh/ :


```
mv ~/Downloads/id_geni_ssh_rsa ~/.ssh/
```
 - Change the file permissions:


```
chmod 0600 ~/.ssh/id_geni_ssh_rsa
```

Step 2:

Created a slice “Project-2” and added four resources i,e four VM’s. One of the VMs acts as server and the remaining three will act as clients. In the next step, I’ve established connection between these four resources.

GENI Portal

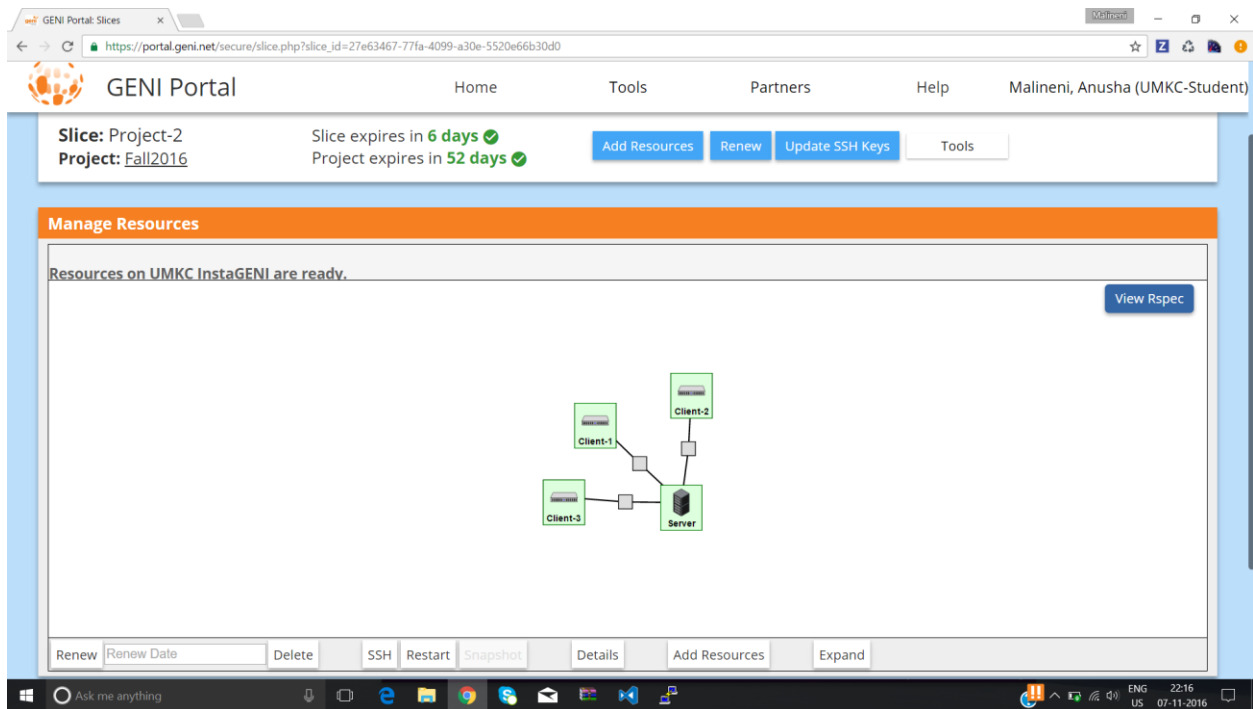
Slices Projects Logs Map Status

Filter by: All slices Sort by: Slice name ☒ Sort ascending

[+ New slice](#)

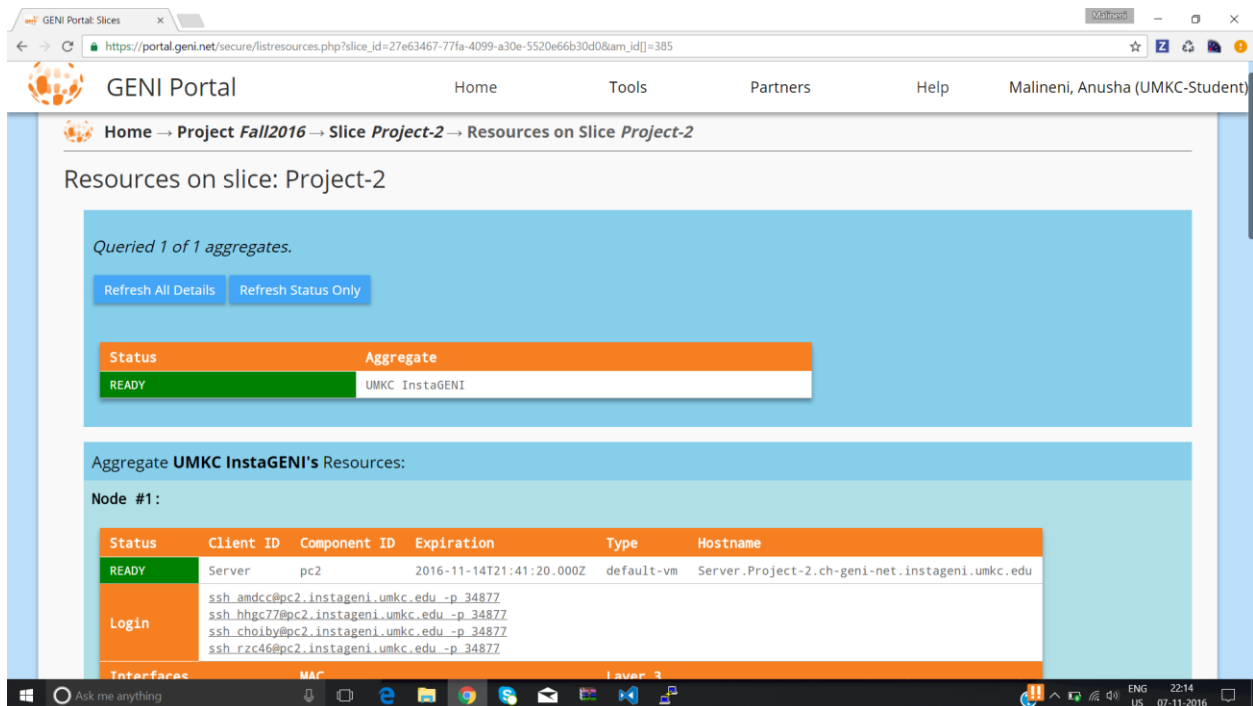
Project-1	Project-2
Project: Fall2016	Project: Fall2016
Owner: Malineni, Anusha (UMKC-Student)	Owner: Malineni, Anusha (UMKC-Student)
Slice expires in 22 days <input checked="" type="checkbox"/>	Slice expires in 6 days <input checked="" type="checkbox"/>
3 resources, next exp. in 22 days <input checked="" type="checkbox"/>	7 resources, next exp. in 6 days <input checked="" type="checkbox"/>

GENI Portal Version 3.17
Copyright © 2016 Raytheon BBN Technologies



Step 3:

After selecting the resources and establishing connection between them, resources will be allocated to the slice and status of these resources will change to 'Ready'.



GENI Portal Slices

https://portal.geni.net/secure/listresources.php?slice_id=27e63467-77fa-4099-a30e-5520e66b30d0&am_id[]=385

GENI Portal Home Tools Partners Help Malineni, Anusha (UMKC-Student)

Node #1:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Server	pc2	2016-11-14T21:41:20.000Z	default-vm	Server.Project-2.ch-genl-net.instageni.umkc.edu

Login

```
ssh_amdccc@pc2.instageni.umkc.edu -p 34877
ssh_hhgc77@pc2.instageni.umkc.edu -p 34877
ssh_choiby@pc2.instageni.umkc.edu -p 34877
ssh_rzc46@pc2.instageni.umkc.edu -p 34877
```

Interfaces	MAC	Layer 3
interface-0	pc2:100 0261ce506646	ipv4: 10.10.1.1
interface-3	pc2:100 0251256de80c	ipv4: 10.10.2.2
interface-5	pc2:100 022ad0b51954	ipv4: 10.10.3.2

Node #2:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client-1	pc2	2016-11-14T21:41:20.000Z	default-vm	Client-1.Project-2.ch-genl-net.instageni.umkc.edu

Login

```
ssh_amdccc@pc2.instageni.umkc.edu -p 34874
ssh_hhgc77@pc2.instageni.umkc.edu -p 34874
ssh_choiby@pc2.instageni.umkc.edu -p 34874
ssh_rzc46@pc2.instageni.umkc.edu -p 34874
```

Interfaces	MAC	Layer 3
interface-1	pc2:100 02a6cb460439	ipv4: 10.10.1.2

GENI Portal Slices

https://portal.geni.net/secure/listresources.php?slice_id=27e63467-77fa-4099-a30e-5520e66b30d0&am_id[]=385

GENI Portal Home Tools Partners Help Malineni, Anusha (UMKC-Student)

interface-1 pc2:100 02a6cb460439 ipv4: 10.10.1.2

Node #3:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client-2	pc2	2016-11-14T21:41:20.000Z	default-vm	Client-2.Project-2.ch-genl-net.instageni.umkc.edu

Login

```
ssh_amdccc@pc2.instageni.umkc.edu -p 34875
ssh_hhgc77@pc2.instageni.umkc.edu -p 34875
ssh_choiby@pc2.instageni.umkc.edu -p 34875
ssh_rzc46@pc2.instageni.umkc.edu -p 34875
```

Interfaces	MAC	Layer 3
interface-2	pc2:100 02160794826e	ipv4: 10.10.2.1

Node #4:

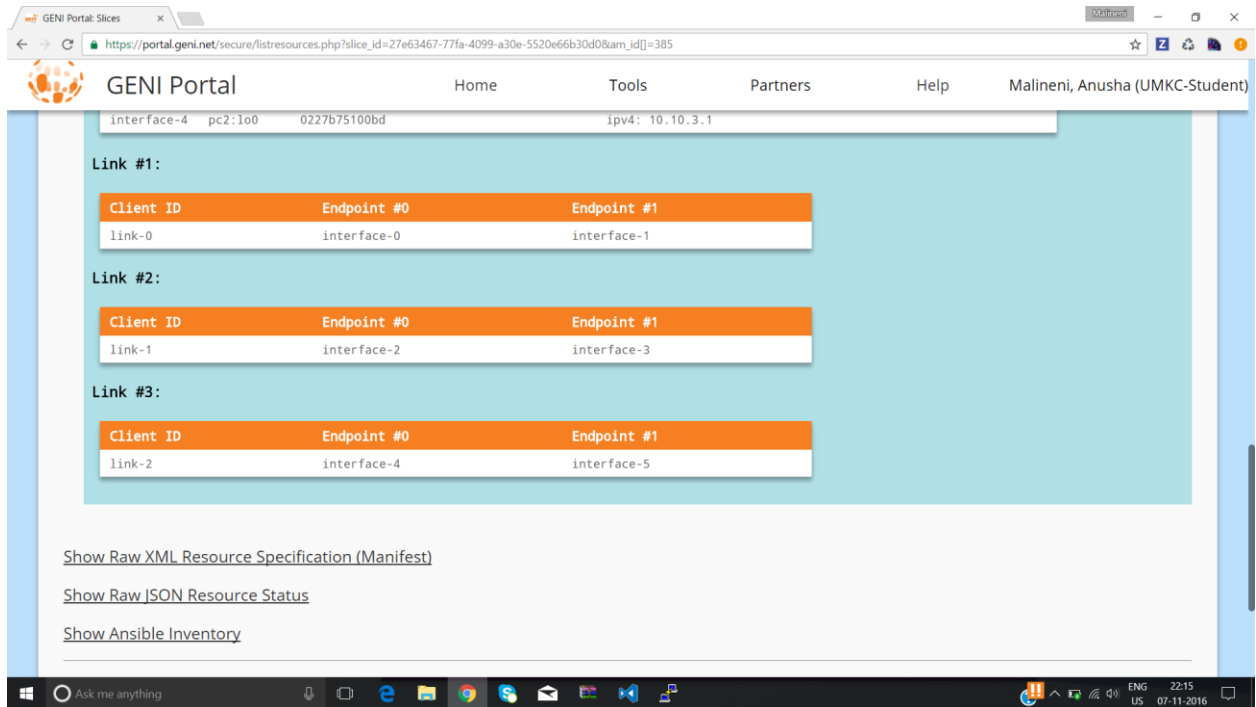
Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client-3	pc2	2016-11-14T21:41:20.000Z	default-vm	Client-3.Project-2.ch-genl-net.instageni.umkc.edu

Login

```
ssh_amdccc@pc2.instageni.umkc.edu -p 34876
ssh_hhgc77@pc2.instageni.umkc.edu -p 34876
ssh_choiby@pc2.instageni.umkc.edu -p 34876
ssh_rzc46@pc2.instageni.umkc.edu -p 34876
```

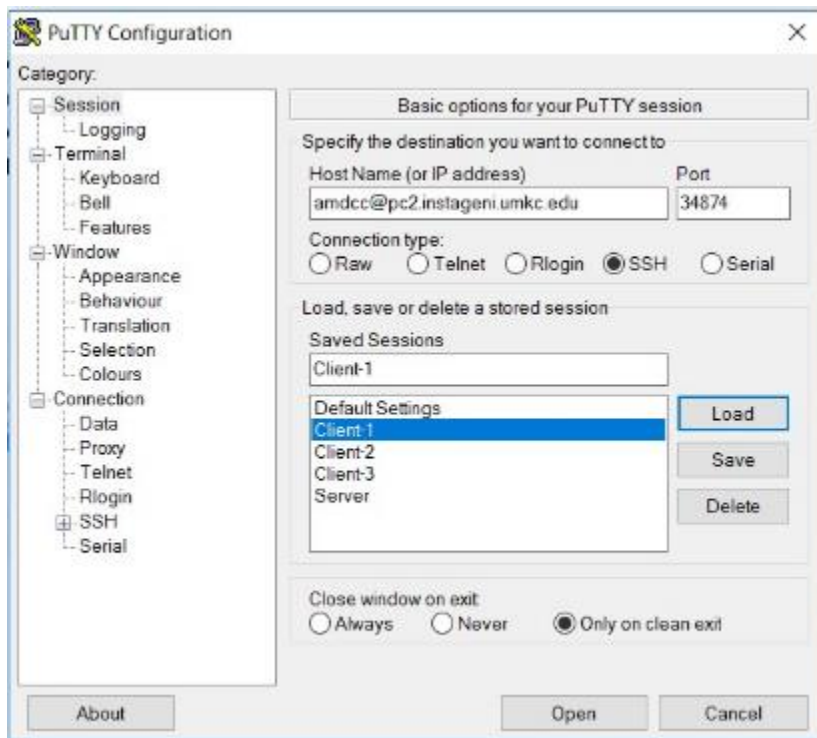
Interfaces	MAC	Layer 3
interface-4	pc2:100 0227b75100bd	ipv4: 10.10.3.1

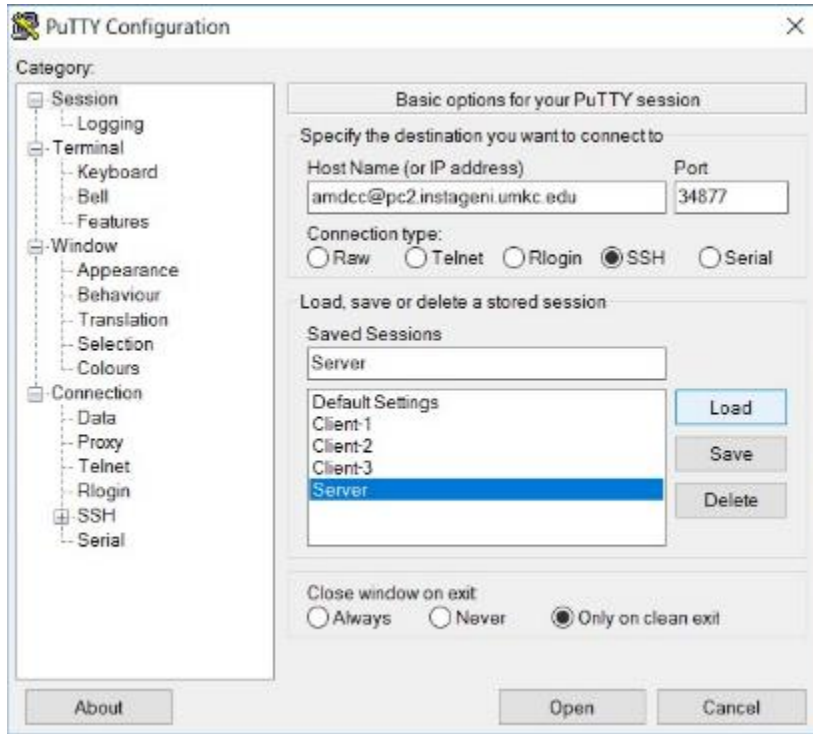
Link #1:



Step 4:

By using putty, we need to load all the clients and servers.





3. Project Description

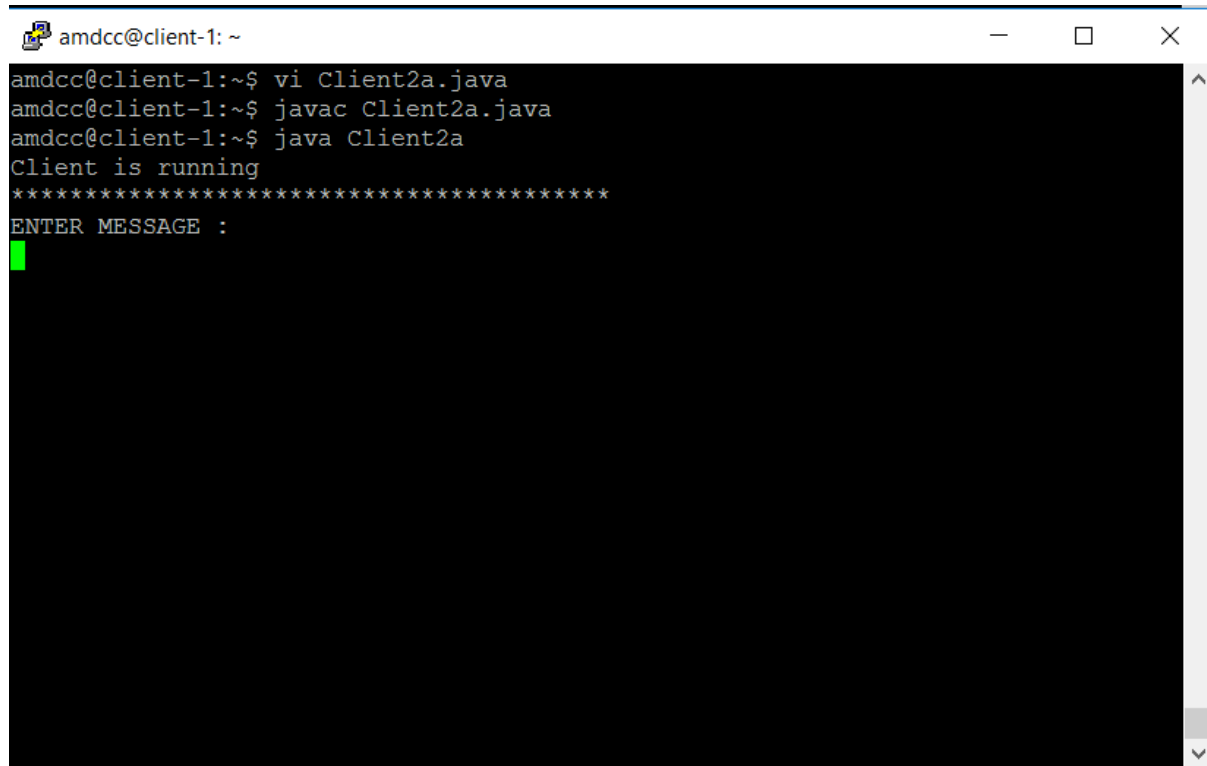
Need to develop a simple chat program (like google hangout and skype chat).

3.1 Question a:

A chat server will accept a single client connection and display everything the client types. If the client user types 'exit', both client and server will end the program.

Solution:

Client and server java programs were loaded to client and server respectively. After compiling and running those programs both client and server will run.

A terminal window titled 'amdcc@client-1: ~' with standard window controls. The terminal shows the following commands and output:

```
amdcc@client-1:~$ vi Client2a.java
amdcc@client-1:~$ javac Client2a.java
amdcc@client-1:~$ java Client2a
Client is running
*****
ENTER MESSAGE :
█
```

Both client and server can communicate with each other through messages. This connection will stay till client sends 'exit'. Once client sends 'exit' connection between client and server will get terminated. Both client and server connections will get terminated.

```
amdcc@client-1: ~  
amdcc@client-1:~$ java Client2a  
Client is running  
*****  
ENTER MESSAGE :  
Hello I am Client  
FROM SERVER: Hello Client, I am Server  
ENTER MESSAGE :  
Hi How are you  
FROM SERVER: I am fine, how about you  
ENTER MESSAGE :  
█
```

```
amdcc@server: ~  
amdcc@server:~$ java Server2a  
Server running  
RECEIVED FROM CLIENT : Hello I am Client  
Hello Client, I am Server  
RECEIVED FROM CLIENT : Hi How are you  
I am fine, how about you  
█
```

```
amdcc@client-1: ~  
amdcc@client-1:~$ java Client2a  
Client is running  
*****  
ENTER MESSAGE :  
Hello I am Client  
FROM SERVER: Hello Client, I am Server  
ENTER MESSAGE :  
Hi How are you  
FROM SERVER: I am fine, how about you  
ENTER MESSAGE :  
exit  
FROM SERVER: exit  
amdcc@client-1:~$
```

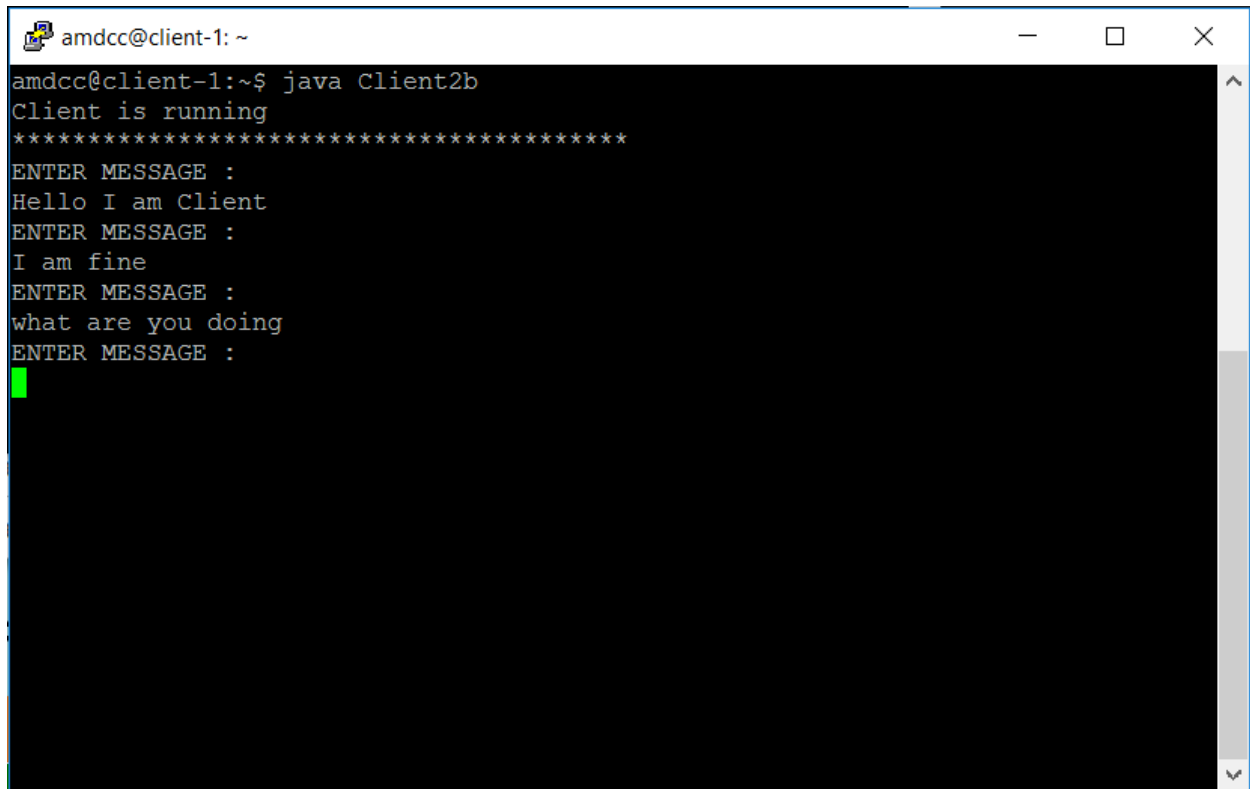
```
amdcc@server: ~  
amdcc@server:~$ java Server2a  
Server running  
RECEIVED FROM CLIENT : Hello I am Client  
Hello Client, I am Server  
RECEIVED FROM CLIENT : Hi How are you  
I am fine, how about you  
RECEIVED FROM CLIENT : exit  
amdcc@server:~$
```

3.2 Question-b:

A server now remains 'open' for additional connection once a client quits. The server can handle at most one connection at a time.

Solution:

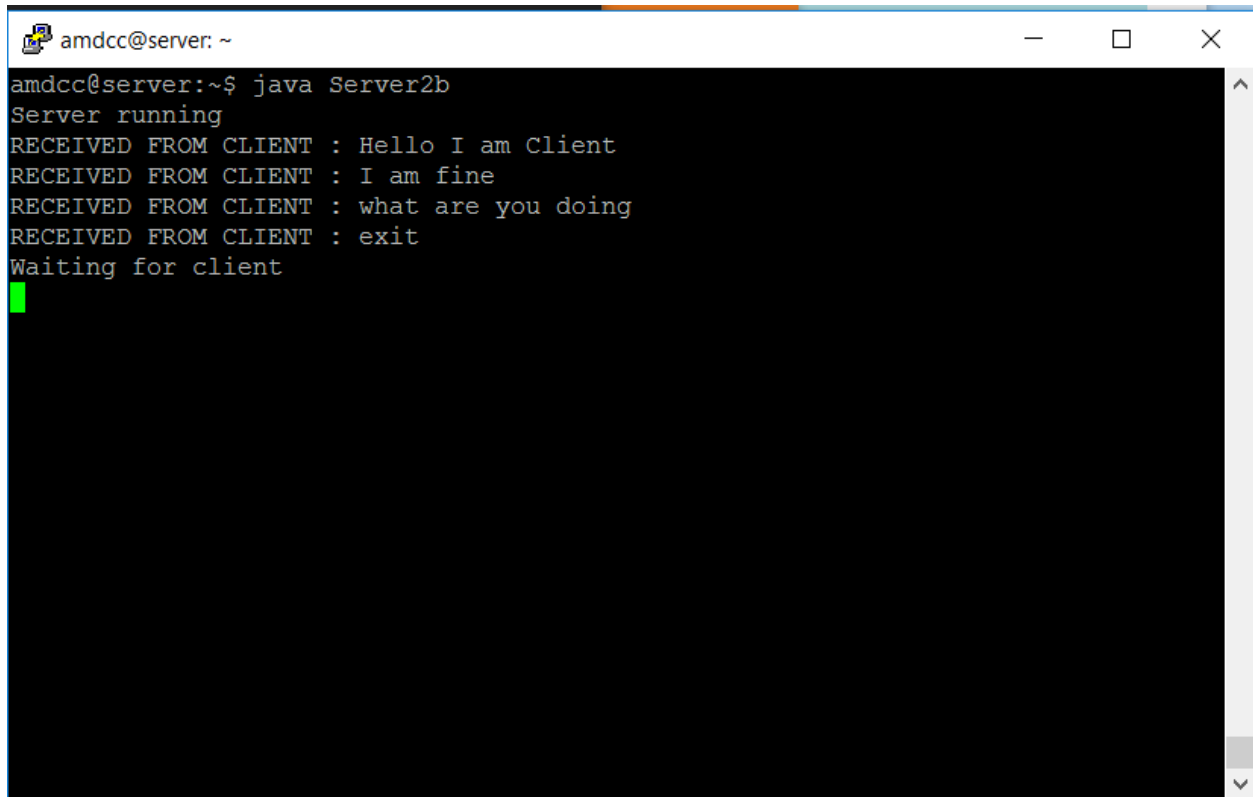
For the question b, client and server java programs were loaded to client and server. Both client and server can communicate with each other through messages. Once client sends 'exit', connection of the client will be terminated and server connection will wait for next client to connect.

A screenshot of a terminal window titled 'amdcc@client-1: ~'. The terminal shows the execution of a Java program 'Client2b'. The output includes 'Client is running', a separator line of asterisks, and a series of prompts 'ENTER MESSAGE :'. The user has entered 'Hello I am Client', 'I am fine', and 'what are you doing'. A green cursor is visible on the line following the last prompt.

```
amdcc@client-1: ~  
amdcc@client-1:~$ java Client2b  
Client is running  
*****  
ENTER MESSAGE :  
Hello I am Client  
ENTER MESSAGE :  
I am fine  
ENTER MESSAGE :  
what are you doing  
ENTER MESSAGE :  
█
```

```
amdcc@server: ~  
amdcc@server:~$ java Server2b  
Server running  
RECEIVED FROM CLIENT : Hello I am Client  
RECEIVED FROM CLIENT : I am fine  
RECEIVED FROM CLIENT : what are you doing  
█
```

```
amdcc@client-1: ~  
amdcc@client-1:~$ java Client2b  
Client is running  
*****  
ENTER MESSAGE :  
Hello I am Client  
ENTER MESSAGE :  
I am fine  
ENTER MESSAGE :  
what are you doing  
ENTER MESSAGE :  
exit  
FROM SERVER: exit  
amdcc@client-1:~$ █
```

A terminal window titled 'amdcc@server: ~' with standard window controls. The terminal shows the execution of 'java Server2b', which outputs 'Server running'. It then displays four lines of received client messages: 'Hello I am Client', 'I am fine', 'what are you doing', and 'exit'. The terminal ends with 'Waiting for client' and a green cursor.

```
amdcc@server: ~  
amdcc@server:~$ java Server2b  
Server running  
RECEIVED FROM CLIENT : Hello I am Client  
RECEIVED FROM CLIENT : I am fine  
RECEIVED FROM CLIENT : what are you doing  
RECEIVED FROM CLIENT : exit  
Waiting for client  
█
```

3.3 Question-c:

A server now can handle multiple clients at the same time. The output from all the connected clients will appear on the server's screen.

Solution:

For the question c, required client and server java programs were loaded to both client and server. Here server handles multiple clients at the same time and it accepts messages from all connected clients and all the messages from the clients were displayed on server.

```
amdcc@client-2: ~  
amdcc@client-2:~$ java Client2b  
Client is running  
*****  
ENTER MESSAGE :  
Hello I am Client-2  
ENTER MESSAGE :  
How are you  
ENTER MESSAGE :  
I am fine  
ENTER MESSAGE :  
█
```

```
amdcc@server: ~  
amdcc@server:~$ java Server2b  
Server running  
RECEIVED FROM CLIENT : Hello I am Client  
RECEIVED FROM CLIENT : I am fine  
RECEIVED FROM CLIENT : what are you doing  
RECEIVED FROM CLIENT : exit  
Waiting for client  
RECEIVED FROM CLIENT : Hello I am Client-2  
RECEIVED FROM CLIENT : How are you  
RECEIVED FROM CLIENT : I am fine  
█
```

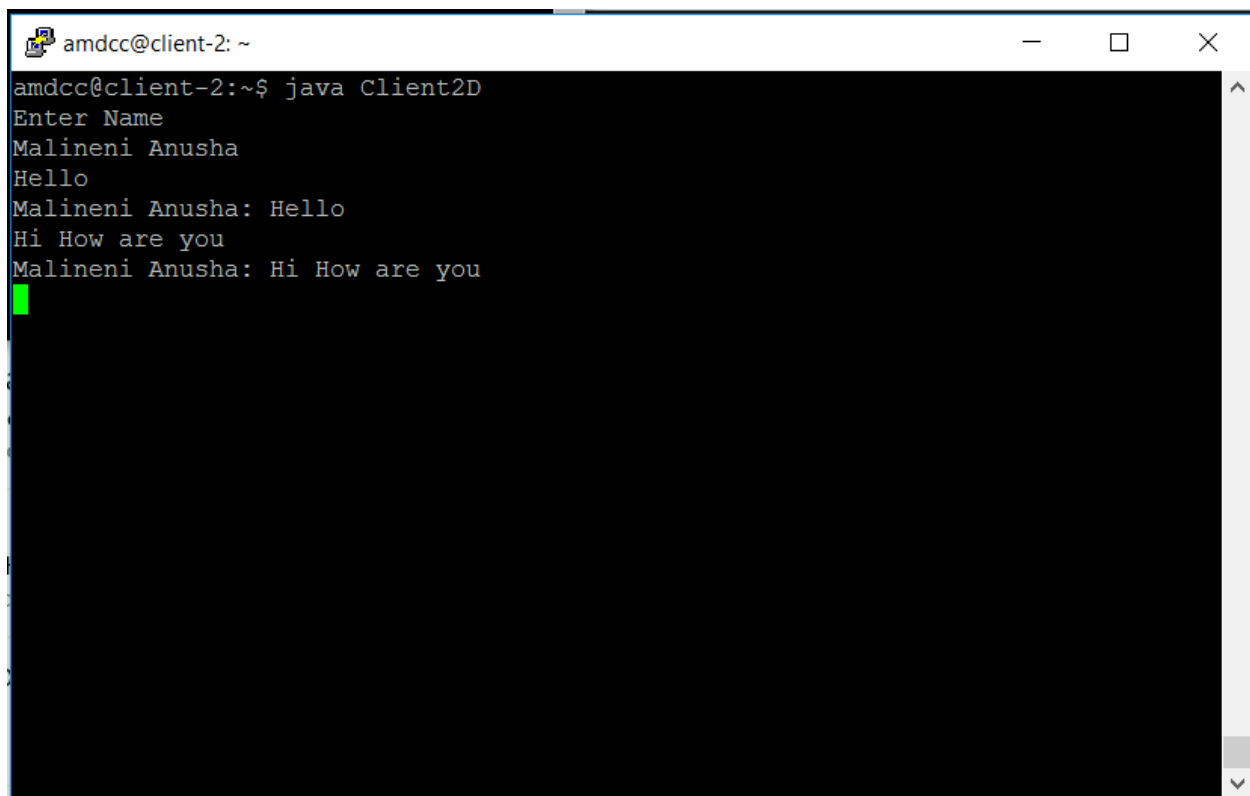
3.4 Question-d:

A server now echoes all the text received from any of the connected clients to all.

Solution:

Java files are created for each of the Clients and Server through which messages are received and sent back. In this query, we show that when any of the Clients sends its message to the Server, it broadcasts the message to the remaining Clients with the name of the Client specified.

Here we are connecting to 4 Clients, whenever new Client is started to Chat Application the intimation is given to all other Clients. Also, when Client is left from Chat the exit message is sent to Server. The communication between client and server is echoed to all other servers. For example, Client-1 sends message 'hi', this message will be displayed on Client-2, Client-3, Client-4 along with Server.



```
amdcc@client-2: ~  
amdcc@client-2:~$ java Client2D  
Enter Name  
Malineni Anusha  
Hello  
Malineni Anusha: Hello  
Hi How are you  
Malineni Anusha: Hi How are you  
█
```



```
amdcc@server: ~  
amdcc@server:~$ java Server2D  
Server Started  
Malineni Anusha:Hello  
Malineni Anusha:Hi How are you  
█
```

```
amdcc@server: ~  amdcc@client-3: ~  
amdcc@server:~$ java Server2D  
Server Started  
Malineni Anusha:Hello  
Malineni Anusha:Hi How are you  
Ram Gopal:Hi All  
Ram Gopal:Howz the day  
Hi Ram, its good how about you  
█  
amdcc@client-3:~$ vi Client2D.java  
amdcc@client-3:~$ javac Client2D.java  
amdcc@client-3:~$ java Client2D  
Enter Name  
Ram Gopal  
Hi All  
Ram Gopal: Hi All  
Howz the day  
Ram Gopal: Howz the day  
█  
amdcc@client-2: ~  
amdcc@client-2:~$ java Client2D  
Enter Name  
Malineni Anusha  
Hello  
Malineni Anusha: Hello  
Hi How are you  
Malineni Anusha: Hi How are you  
Ram Gopal: Hi All  
Ram Gopal: Howz the day  
█
```

```
amdcc@server:~$ java Server2D
Server Started
Malineni Anusha:Hello
Malineni Anusha:Hi How are you
Ram Gopal:Hi All
Ram Gopal:Howz the day
Hi Ram, its good how about you
Sai Nath:Hi Friends
Sai Nath:Good Evening
Ram Gopal:Hi SaiNath How are you
Sai Nath:Hi I am fine
]

amdcc@client-1:~$ java Client2D
Enter Name
Sai Nath
Hi Friends
Sai Nath: Hi Friends
Good Evening
Sai Nath: Good Evening
Ram Gopal: Hi SaiNath How are you
Hi I am fine
Sai Nath: Hi I am fine
]

amdcc@client-2:~$ java Client2D
Enter Name
Malineni Anusha
Hello
Malineni Anusha: Hello
Hi How are you
Malineni Anusha: Hi How are you
Ram Gopal: Hi All
Ram Gopal: Howz the day
Sai Nath: Hi Friends
Sai Nath: Good Evening
Ram Gopal: Hi SaiNath How are you
Sai Nath: Hi I am fine
]

amdcc@client-3:~$ vi Client2D.java
amdcc@client-3:~$ javac Client2D.java
amdcc@client-3:~$ java Client2D
Enter Name
Ram Gopal
Hi All
Ram Gopal: Hi All
Howz the day
Ram Gopal: Howz the day
Sai Nath: Hi Friends
Sai Nath: Good Evening
Hi SaiNath How are you
Ram Gopal: Hi SaiNath How are you
Sai Nath: Hi I am fine
]
```

```
amdcc@server:~$ java Server2D
Server Started
Malineni Anusha:Hello
Malineni Anusha:Hi How are you
Ram Gopal:Hi All
Ram Gopal:Howz the day
Hi Ram, its good how about you
Sai Nath:Hi Friends
Sai Nath:Good Evening
Ram Gopal:Hi SaiNath How are you
Sai Nath:Hi I am fine
Ram Gopal:exit
Ram Gopal:left
Ram Gopal:left
]

amdcc@client-1:~$ java Client2D
Enter Name
Sai Nath
Hi Friends
Sai Nath: Hi Friends
Good Evening
Sai Nath: Good Evening
Ram Gopal: Hi SaiNath How are you
Hi I am fine
Sai Nath: Hi I am fine
]

amdcc@client-2:~$ java Client2D
Enter Name
Malineni Anusha
Hello
Malineni Anusha: Hello
Hi How are you
Malineni Anusha: Hi How are you
Ram Gopal: Hi All
Ram Gopal: Howz the day
Sai Nath: Hi Friends
Sai Nath: Good Evening
Ram Gopal: Hi SaiNath How are you
Sai Nath: Hi I am fine
]

amdcc@client-3:~$ vi Client2D.java
amdcc@client-3:~$ javac Client2D.java
amdcc@client-3:~$ java Client2D
Enter Name
Ram Gopal
Hi All
Ram Gopal: Hi All
Howz the day
Ram Gopal: Howz the day
Sai Nath: Hi Friends
Sai Nath: Good Evening
Hi SaiNath How are you
Ram Gopal: Hi SaiNath How are you
Sai Nath: Hi I am fine
exit
Ram Gopal:left
]
```

4. References

- a. <http://portal.geni.net/secure/dashboard.php>
- b. <http://stackoverflow.com/>
- c. <http://www.putty.org/>