NETWORK ARCHITECTURE – I

Project Report

Part I - GENI/Socket programming Warm-up

Sri Sai Anusha Gandu

16230560

Contents

Introduction of the Project:

In this project, we deploy a simple TCP Client and Server programs on GENI for communication (Part (a)) and file transfer (Part (b)).

In Part (a) of the project, we start from Client message ‘Hello from Client with name’ and Server responses with ‘Hello from Server with name’. The messages are echoed to each other from each side. The program is quit with the message ‘Bye from Client with name’ for which Server responses ‘Bye from Server with name’.

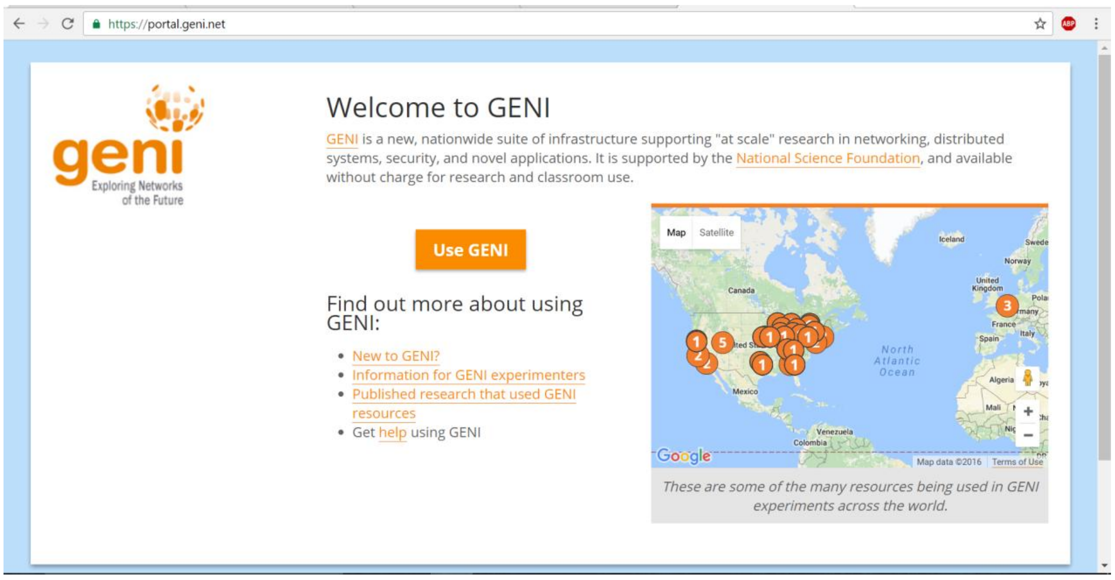
In Part (b) of the project, the Client sends a text file to the Server. The Server prints the file on screen, saves it in a local system, appends an extra line to the file and sends the updated file to the Client. The Client displays the file on screen after it fully receives the file.

Initial Setup:

To continue with the project, we need to first create a GENI account and a slice where we can reserve resources on which to work on.

GENI Account Creation:

1. Login to portal.geni.net



Welcome to GENI

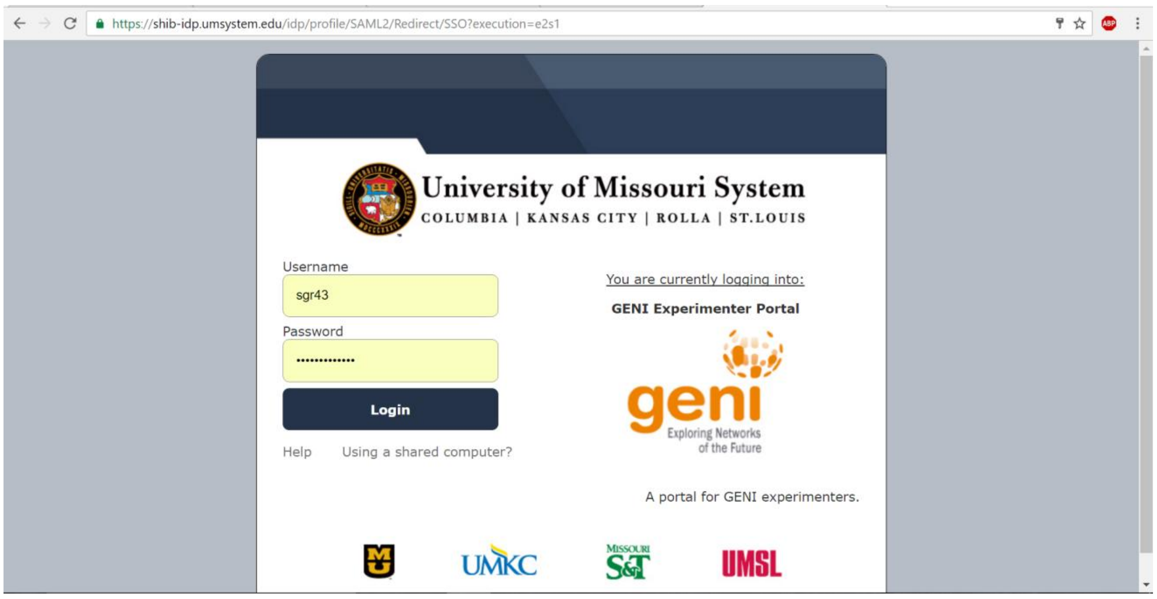
GENI is a new, nationwide suite of infrastructure supporting "at scale" research in networking, distributed systems, security, and novel applications. It is supported by the National Science Foundation, and available without charge for research and classroom use.

Find out more about using GENI:

- New to GENI?
- Information for GENI experimenters
- Published research that used GENI resources
- Get help using GENI

These are some of the many resources being used in GENI experiments across the world.

2. Activate the GENI account with the required credentials.


https://shib-idp.umsystem.edu/idp/profile/SAML2/Redirect/SSO?execution=e2s1

University of Missouri System
COLUMBIA | KANSAS CITY | ROLLA | ST.LOUIS

Username	sgr43
Password	•••••••••••••

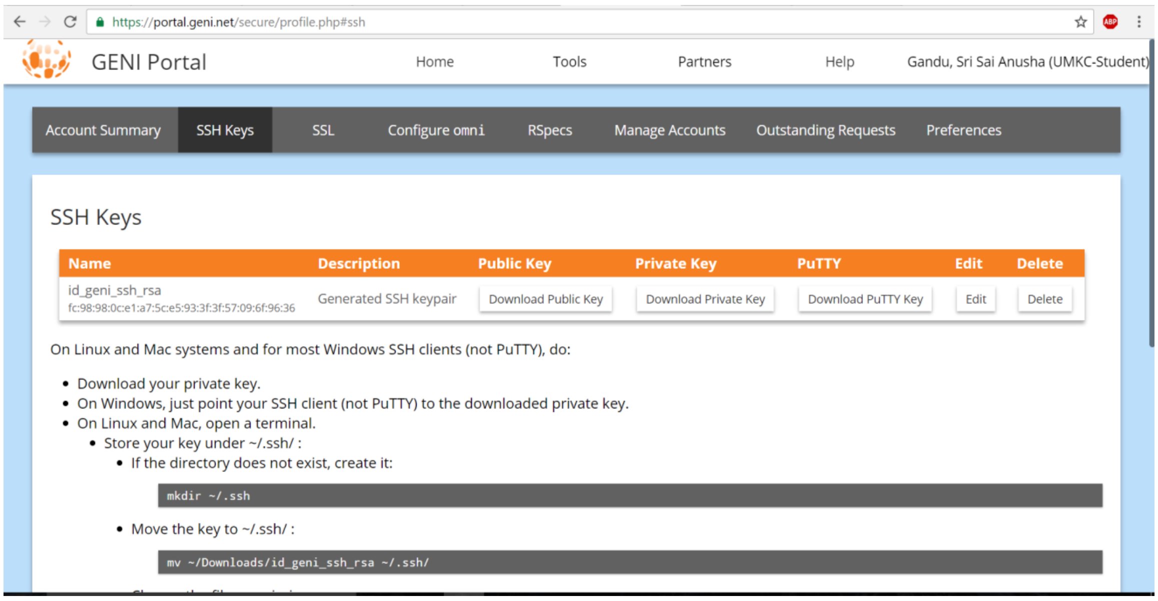
Login

Help   Using a shared computer?

You are currently logging into:
GENI Experimenter Portal

A portal for GENI experimenters.

3. Download the SSH keys (Putty) for authentication process.


https://portal.geni.net/secure/profile.php#ssh

SSH Keys

Name	Description	Public Key	Private Key	PuTTY	Edit	Delete
id_geni_ssh_rsa fc:98:98:0c:e1:a7:5c:e5:93:3f:3f:57:09:6f:96:36	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:

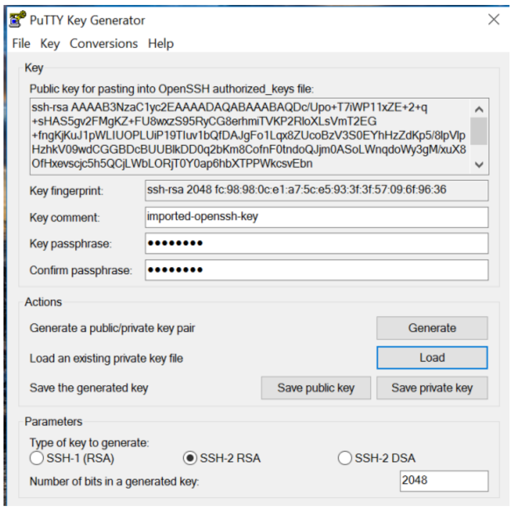
- Download your private key.
- On Windows, just point your SSH client (not PuTTY) to the downloaded private key.
- On Linux and Mac, open a terminal.
  - Store your key under ~/.ssh/ :
    - If the directory does not exist, create it:

```
mkdir ~/.ssh
```

    - Move the key to ~/.ssh/ :

```
mv ~/Downloads/id_geni_ssh_rsa ~/.ssh/
```

4. Using the Putty key generator, generate a private key which will be used to open the Client and Server windows.



PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDc/Upo+T7iWP11xZE+2+q
+sHAS5gv2FMgKZ+FU8wxzS95RyCG8erhmiTVKP2RloXLsVmT2EG
+fngKjKuJ1pWLIUOPLUiP19TIuv1bQfDAJgFo1Lqx8ZUcoBzV3S0EYhHzZdKp5/8lpVlp
HzhkV09wdCGGBDcBUUBlkDD0q2bKm8CofnF0tndoQJjm0ASoLWnqdoWy3gM/xuX8
OfHxevscjc5h5QCjLWbLORjT0Y0ap6hbXTPPWkcsvEbn
```

Key fingerprint: ssh-rsa 2048 fc:98:98:0c:e1:a7:5c:e5:93:3f:3f:57:09:6f:96:36

Key comment: imported-openssh-key

Key passphrase: ••••••••

Confirm passphrase: ••••••••

Actions

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

Parameters

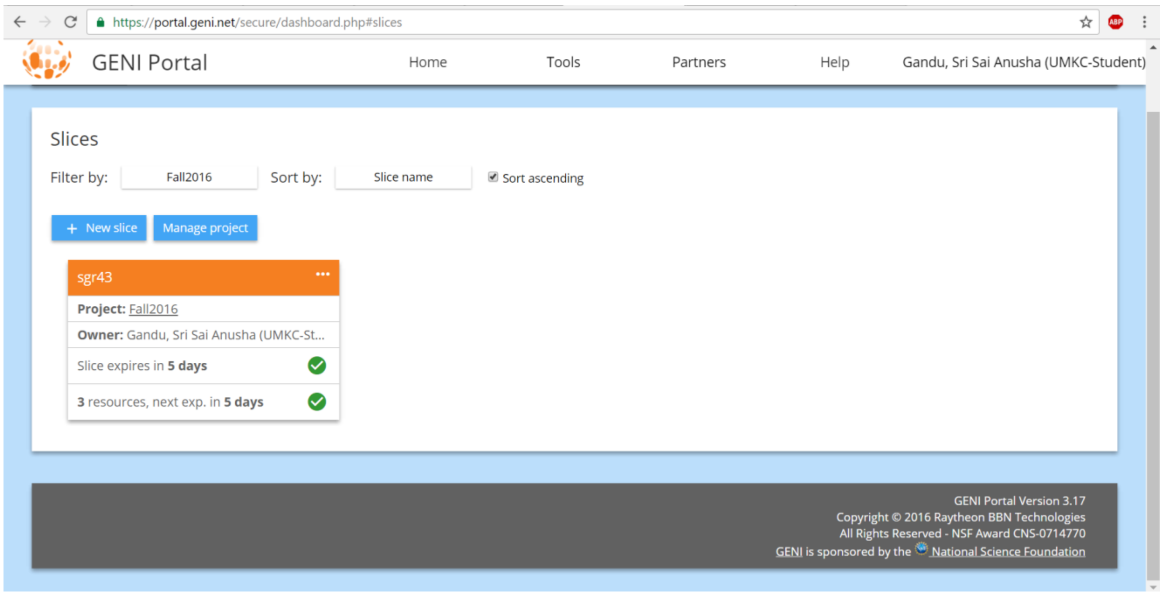
Type of key to generate:

SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 2048

Slice Creation:

5. Create a slice from which the resources can be reserved (sgr43 in this project)



https://portal.geni.net/secure/dashboard.php#slices

GENI Portal — Home — Tools — Partners — Help — Gandu, Sri Sai Anusha (UMKC-Student)

Slices

Filter by: Fall2016 Sort by: Slice name ✓ Sort ascending

+ New slice — Manage project

sgr43

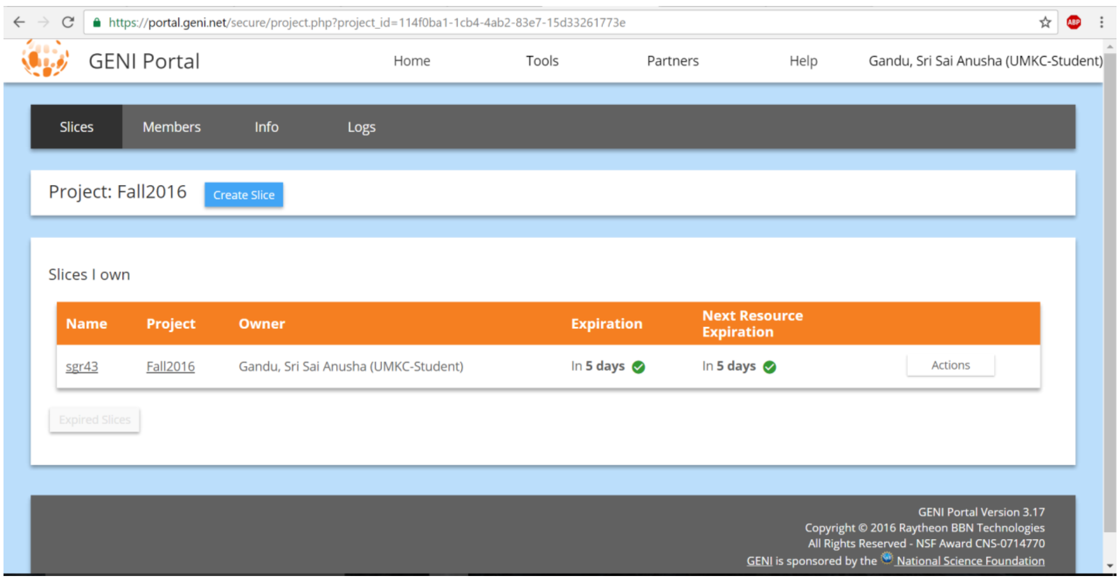
Project: Fall2016

Owner: Gandu, Sri Sai Anusha (UMKC-St...

Slice expires in **5 days** ✓

3 resources, next exp. in **5 days** ✓

GENI Portal Version 3.17
Copyright © 2016 Raytheon BBN Technologies
All Rights Reserved - NSF Award CNS-0714770
GENI is sponsored by the National Science Foundation



https://portal.geni.net/secure/project.php?project_id=114f0ba1-1cb4-4ab2-83e7-15d33261773e

GENI Portal | Home | Tools | Partners | Help | Gandu, Sri Sai Anusha (UMKC-Student)

Slices | Members | Info | Logs

Project: Fall2016 Create Slice

Slices I own

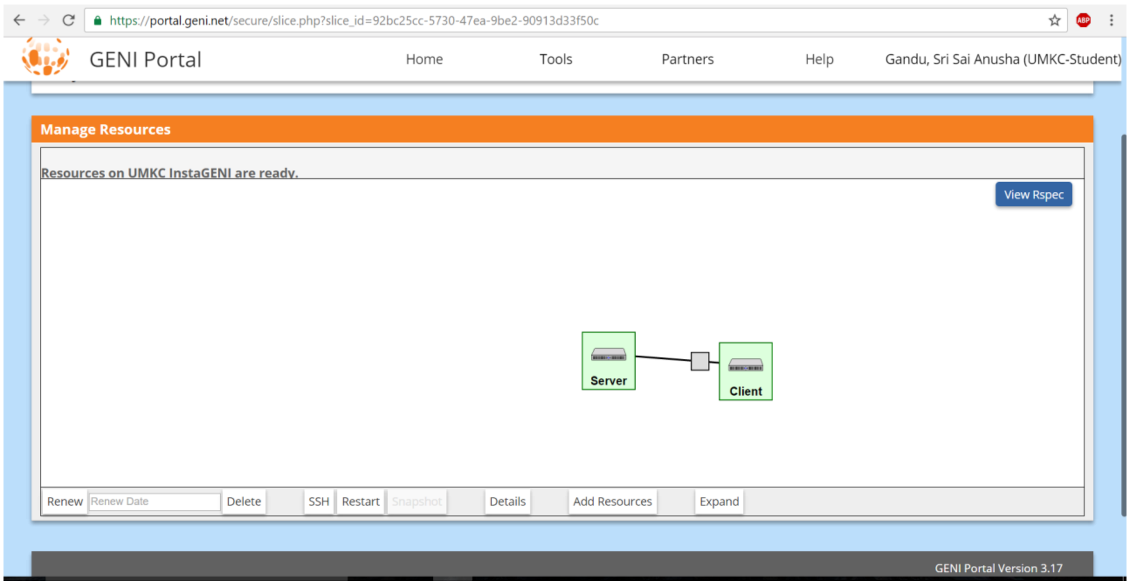
Name	Project	Owner	Expiration	Next Resource Expiration	
sgr43	Fall2016	Gandu, Sri Sai Anusha (UMKC-Student)	In 5 days ✓	In 5 days ✓	Actions

Expired Slices

GENI Portal Version 3.17
Copyright © 2016 Raytheon BBN Technologies
All Rights Reserved - NSF Award CNS-0714770
GENI is sponsored by the National Science Foundation

Resource Reservation:

6. Add two resources, i.e., two VM's and name them as Client and Server and establish connection (link) between them.



https://portal.geni.net/secure/slice.php?slice_id=92bc25cc-5730-47ea-9be2-90913d33f50c

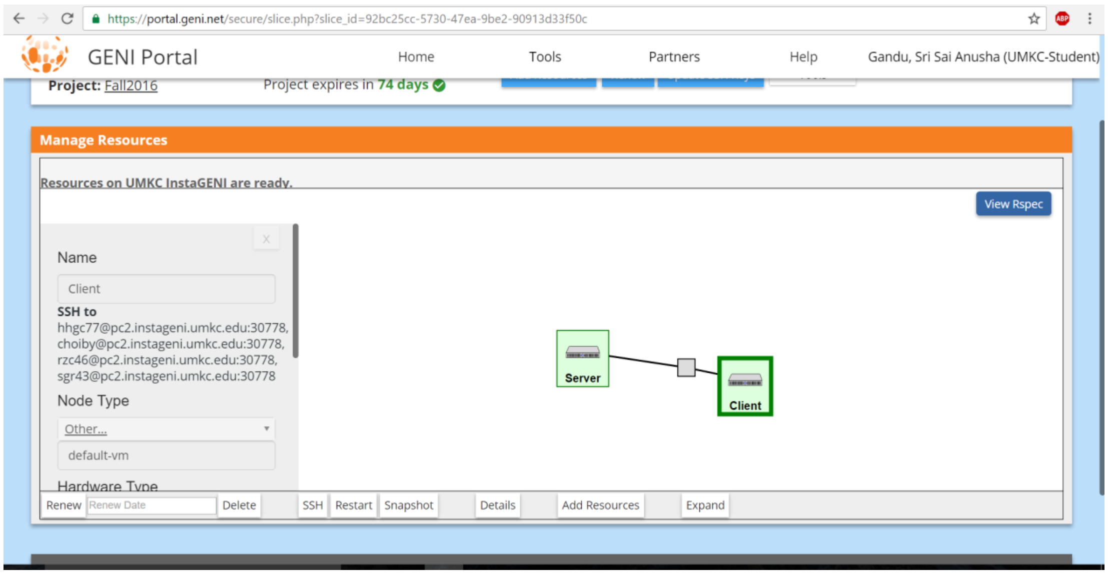
GENI Portal | Home | Tools | Partners | Help | Gandu, Sri Sai Anusha (UMKC-Student)

Manage Resources

Resources on UMKC InstaGENI are ready.

GENI Portal Version 3.17

7. Provide the IP address and subnet mask to the Client and Server along with bandwidth in the link interface.


https://portal.geni.net/secure/slice.php?slice_id=92bc25cc-5730-47ea-9be2-90913d33f50c
GENI Portal
Home
Tools
Partners
Help
Gandu, Sri Sai Anusha (UMKC-Student)
Project: Fall2016
Project expires in 74 days
Manage Resources
Resources on UMKC InstaGENI are ready.
View Rspec
Server
Client
Renew
Renew Date
Delete
SSH
Restart
Snapshot
Details
Add Resources
Expand

Name

Client

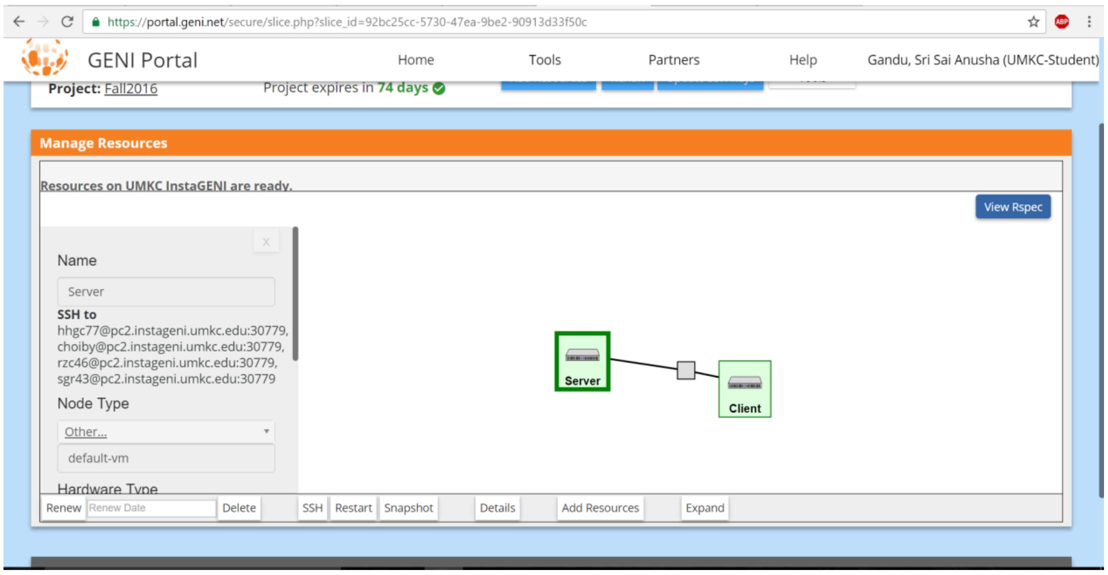
SSH to
hhgc77@pc2.instageni.umkc.edu:30778,
choiby@pc2.instageni.umkc.edu:30778,
rzc46@pc2.instageni.umkc.edu:30778,
sgr43@pc2.instageni.umkc.edu:30778

Node Type

Other...

default-vm

Hardware Type


https://portal.geni.net/secure/slice.php?slice_id=92bc25cc-5730-47ea-9be2-90913d33f50c
GENI Portal
Home
Tools
Partners
Help
Gandu, Sri Sai Anusha (UMKC-Student)
Project: Fall2016
Project expires in 74 days
Manage Resources
Resources on UMKC InstaGENI are ready.
View Rspec
Server
Client
Renew
Renew Date
Delete
SSH
Restart
Snapshot
Details
Add Resources
Expand

Name

Server

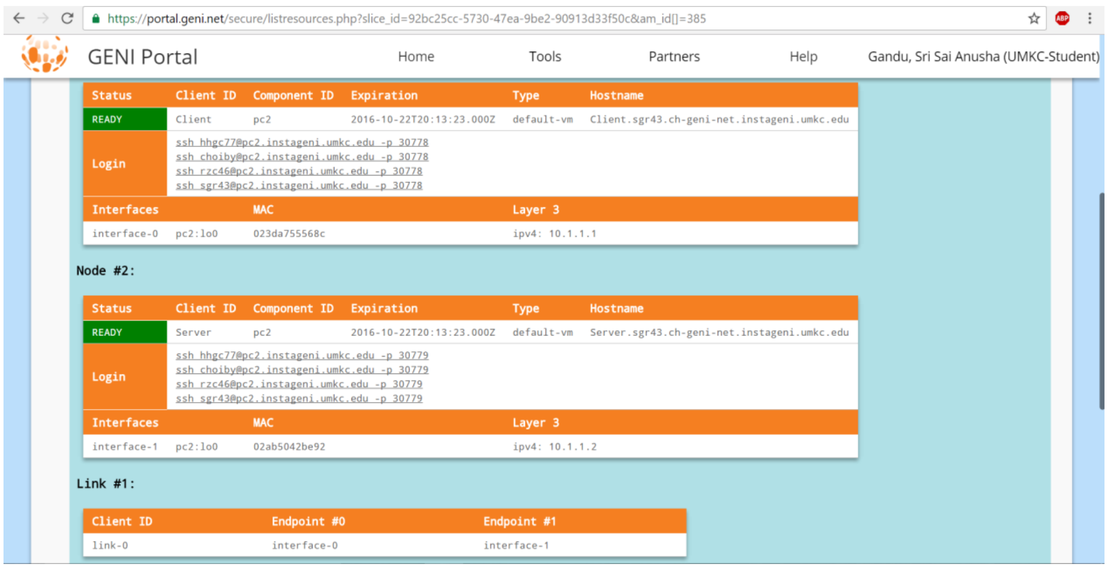
SSH to
hhgc77@pc2.instageni.umkc.edu:30779,
choiby@pc2.instageni.umkc.edu:30779,
rzc46@pc2.instageni.umkc.edu:30779,
sgr43@pc2.instageni.umkc.edu:30779

Node Type

Other...

default-vm

Hardware Type


https://portal.geni.net/secure/listresources.php?slice_id=92bc25cc-5730-47ea-9be2-90913d33f50c&am_id[]=385
GENI Portal
Home
Tools
Partners
Help
Gandu, Sri Sai Anusha (UMKC-Student)

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client	pc2	2016-10-22T20:13:23.000Z	default-vm	Client.sgr43.ch-geni-net.instageni.umkc.edu
Login	ssh hhgc77@pc2.instageni.umkc.edu -p 30778 ssh choiby@pc2.instageni.umkc.edu -p 30778 ssh rzc46@pc2.instageni.umkc.edu -p 30778 ssh sgr43@pc2.instageni.umkc.edu -p 30778				
Interfaces		MAC		Layer 3	
interface-0	pc2:lo0	023da755568c		ipv4: 10.1.1.1	

Node #2:

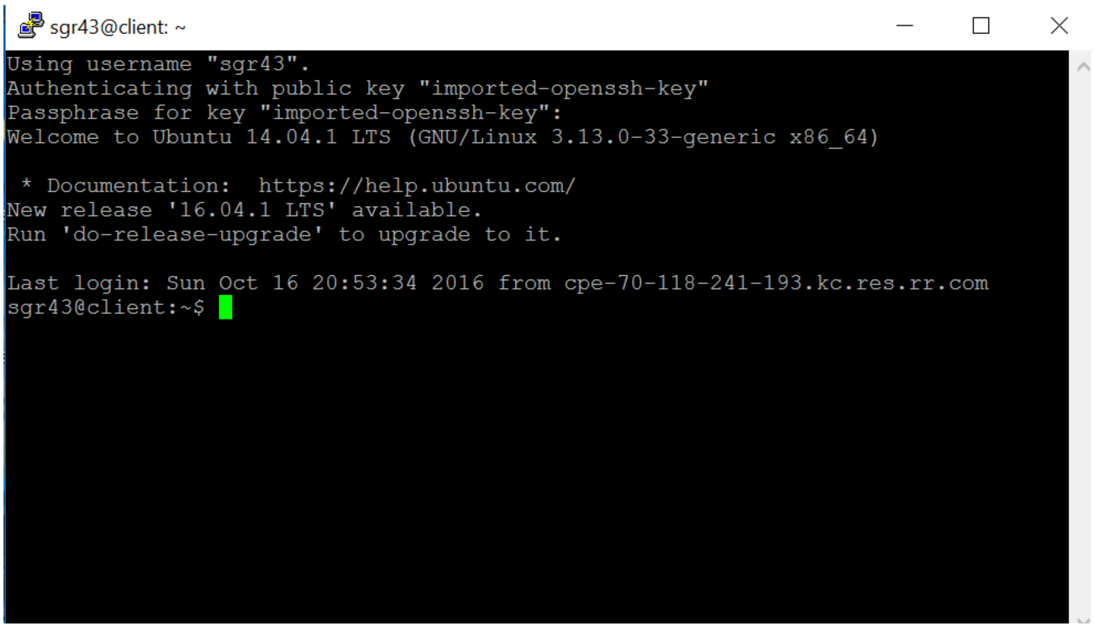
Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Server	pc2	2016-10-22T20:13:23.000Z	default-vm	Server.sgr43.ch-geni-net.instageni.umkc.edu
Login	ssh hhgc77@pc2.instageni.umkc.edu -p 30779 ssh choiby@pc2.instageni.umkc.edu -p 30779 ssh rzc46@pc2.instageni.umkc.edu -p 30779 ssh sgr43@pc2.instageni.umkc.edu -p 30779				
Interfaces		MAC		Layer 3	
interface-1	pc2:lo0	02ab5042be92		ipv4: 10.1.1.2	

Link #1:

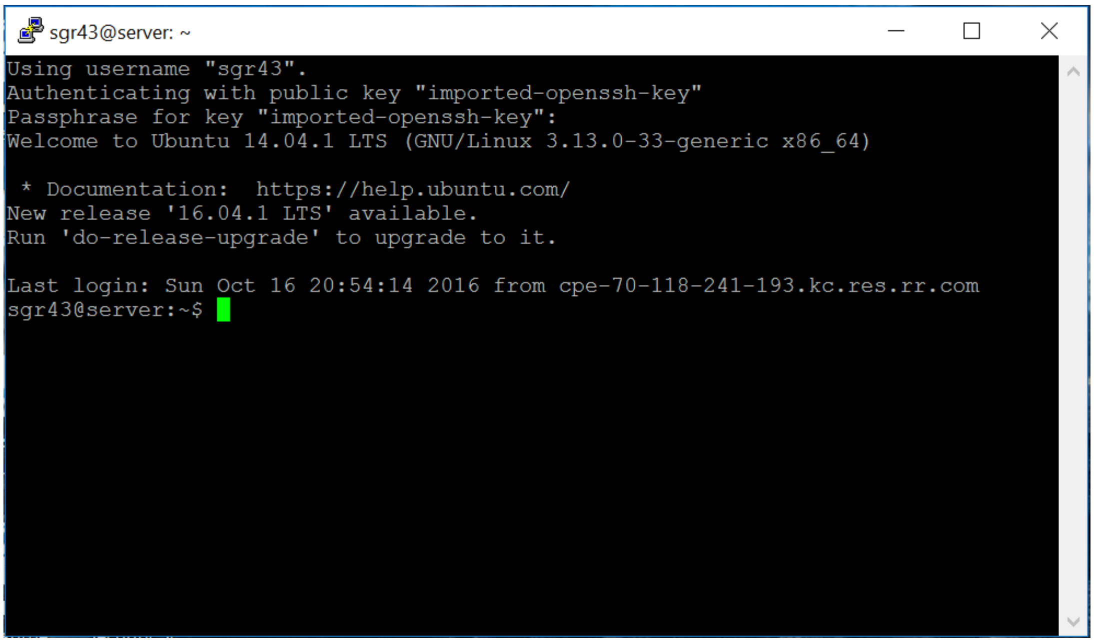
Client ID	Endpoint #0	Endpoint #1
link-0	interface-0	interface-1

Part (a): Client – Server Communication

1. Enable the Client and Server window using Putty with the credentials obtained in the details of the slice created.



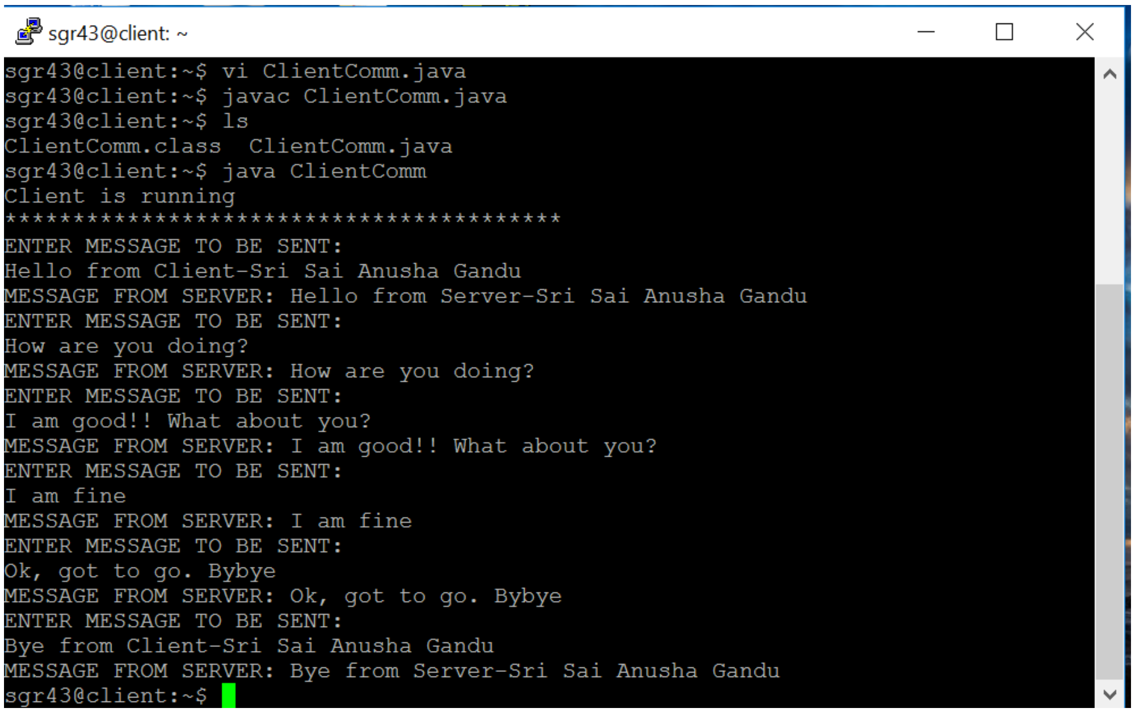
```
Using username "sgr43".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 16 20:53:34 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@client:~$
```



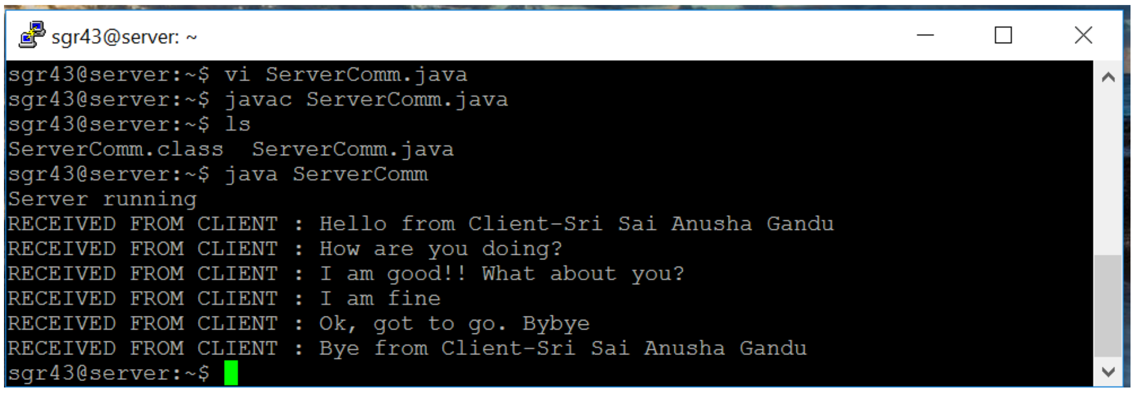
```
Using username "sgr43".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 16 20:54:14 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@server:~$
```

2. Insert the java files containing the program required to enable the communication between the Client and Server using the command **vi filename.java**
3. Here, we have used ClientComm and ServerComm for Client and Server respectively.

4. Compile the programs using the command **javac filename.java**
5. Run the class files using the command **java classfilename**
6. We then continue with the communication between the Client and Server as following.



```
sgr43@client:~$ vi ClientComm.java
sgr43@client:~$ javac ClientComm.java
sgr43@client:~$ ls
ClientComm.class  ClientComm.java
sgr43@client:~$ java ClientComm
Client is running
*****************************************
ENTER MESSAGE TO BE SENT:
Hello from Client-Sri Sai Anusha Gandu
MESSAGE FROM SERVER: Hello from Server-Sri Sai Anusha Gandu
ENTER MESSAGE TO BE SENT:
How are you doing?
MESSAGE FROM SERVER: How are you doing?
ENTER MESSAGE TO BE SENT:
I am good!! What about you?
MESSAGE FROM SERVER: I am good!! What about you?
ENTER MESSAGE TO BE SENT:
I am fine
MESSAGE FROM SERVER: I am fine
ENTER MESSAGE TO BE SENT:
Ok, got to go. Bybye
MESSAGE FROM SERVER: Ok, got to go. Bybye
ENTER MESSAGE TO BE SENT:
Bye from Client-Sri Sai Anusha Gandu
MESSAGE FROM SERVER: Bye from Server-Sri Sai Anusha Gandu
sgr43@client:~$
```

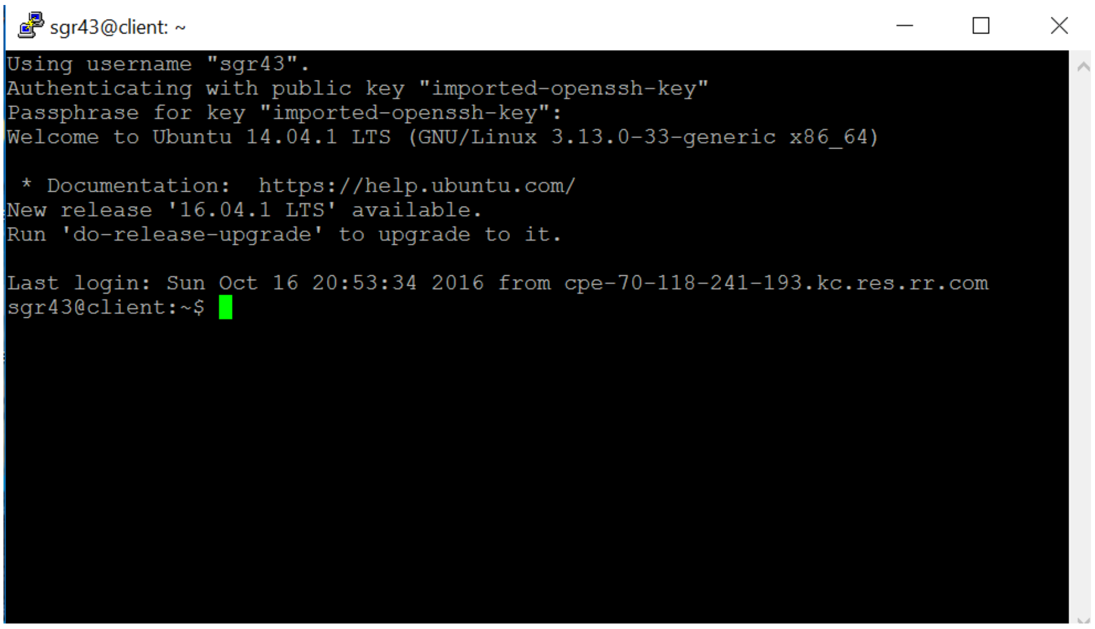


```
sgr43@server:~$ vi ServerComm.java
sgr43@server:~$ javac ServerComm.java
sgr43@server:~$ ls
ServerComm.class  ServerComm.java
sgr43@server:~$ java ServerComm
Server running
RECEIVED FROM CLIENT : Hello from Client-Sri Sai Anusha Gandu
RECEIVED FROM CLIENT : How are you doing?
RECEIVED FROM CLIENT : I am good!! What about you?
RECEIVED FROM CLIENT : I am fine
RECEIVED FROM CLIENT : Ok, got to go. Bybye
RECEIVED FROM CLIENT : Bye from Client-Sri Sai Anusha Gandu
sgr43@server:~$
```

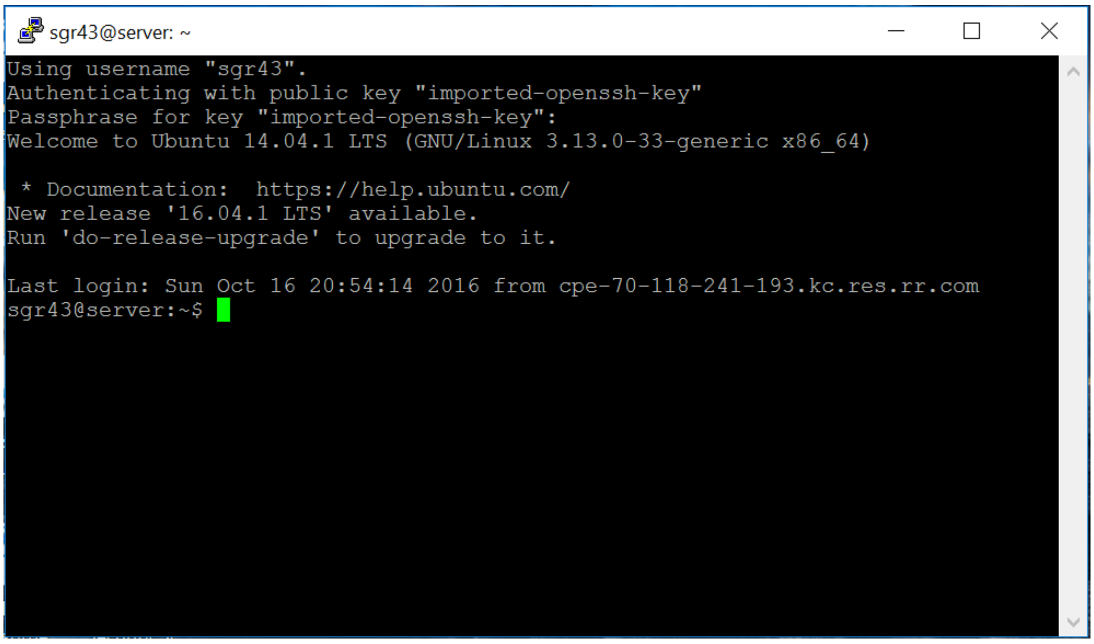
The communication between the Client and Server is successful.

Part (b): Client – Server File Transfer

1. Enable the Client and Server window using Putty with the credentials obtained in the details of the slice created.



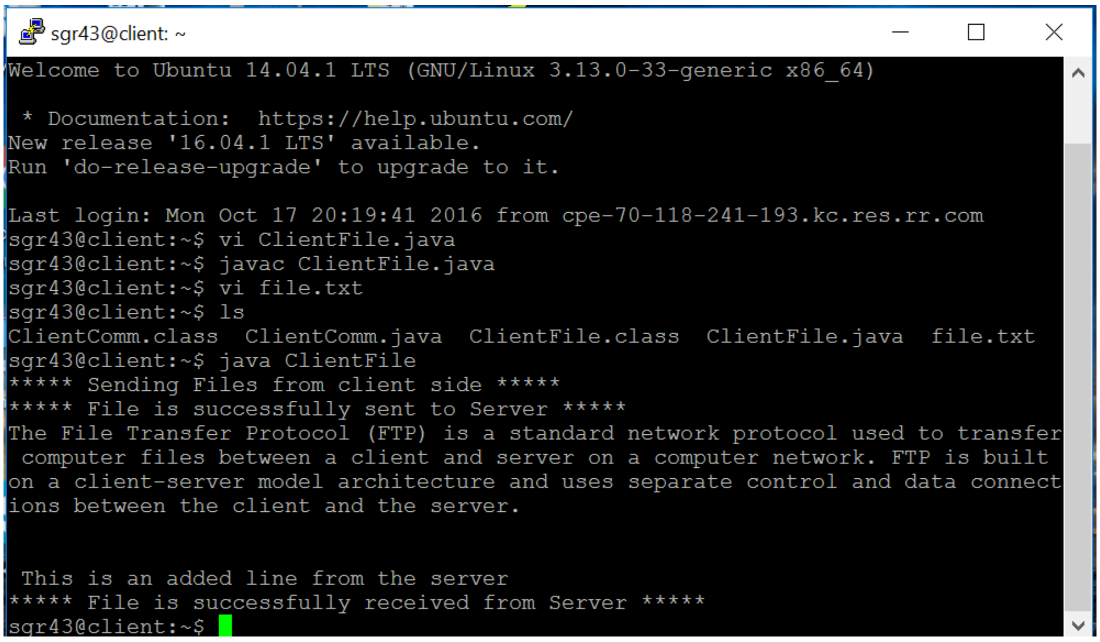
```
Using username "sgr43".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 16 20:53:34 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@client:~$
```



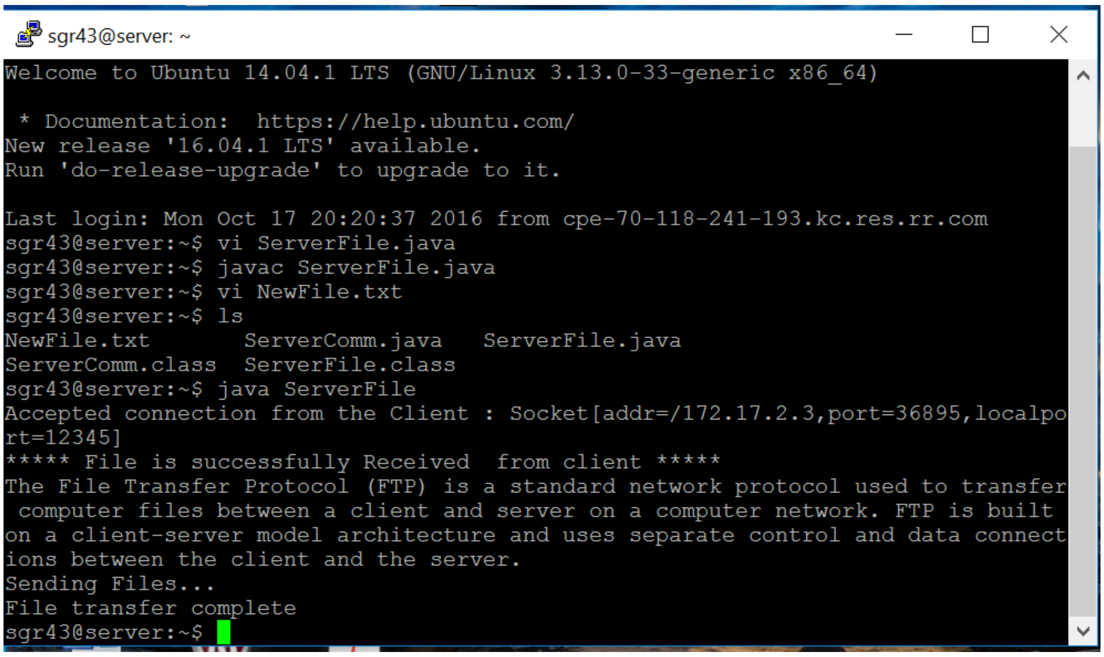
```
Using username "sgr43".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 16 20:54:14 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@server:~$
```

2. Insert the programs required for the transfer of the file from Client to Server using the command **vi filename.java**
3. Here, we have used ClientFile and ServerFile for Client and Server respectively.
4. Compile the programs using the command **javac filename.java**
5. Create a text file in the Client window with content.

6. Create a plain text file in the Server window.
7. Run the class files using the command **java classfilename** in Server first and then in Client.
8. This transfers the file from Client to Server and the Server appends an extra line to the file and sends it to the Client which is displayed on the Client window.



```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 17 20:19:41 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@client:~$ vi ClientFile.java
sgr43@client:~$ javac ClientFile.java
sgr43@client:~$ vi file.txt
sgr43@client:~$ ls
ClientComm.class  ClientComm.java  ClientFile.class  ClientFile.java  file.txt
sgr43@client:~$ java ClientFile
***** Sending Files from client side *****
***** File is successfully sent to Server *****
The File Transfer Protocol (FTP) is a standard network protocol used to transfer
 computer files between a client and server on a computer network. FTP is built
on a client-server model architecture and uses separate control and data connect
ions between the client and the server.


 This is an added line from the server
***** File is successfully received from Server *****
sgr43@client:~$
```



```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 17 20:20:37 2016 from cpe-70-118-241-193.kc.res.rr.com
sgr43@server:~$ vi ServerFile.java
sgr43@server:~$ javac ServerFile.java
sgr43@server:~$ vi NewFile.txt
sgr43@server:~$ ls
NewFile.txt       ServerComm.java   ServerFile.java
ServerComm.class  ServerFile.class
sgr43@server:~$ java ServerFile
Accepted connection from the Client : Socket[addr=/172.17.2.3,port=36895,localpo
rt=12345]
***** File is successfully Received  from client *****
The File Transfer Protocol (FTP) is a standard network protocol used to transfer
 computer files between a client and server on a computer network. FTP is built
on a client-server model architecture and uses separate control and data connect
ions between the client and the server.
Sending Files...
File transfer complete
sgr43@server:~$
```

The Server appended a line and sent the file back to the Client which is displayed on the Client window.