Part 1: Connect to BC's Web Lab Linux Server using SSH.

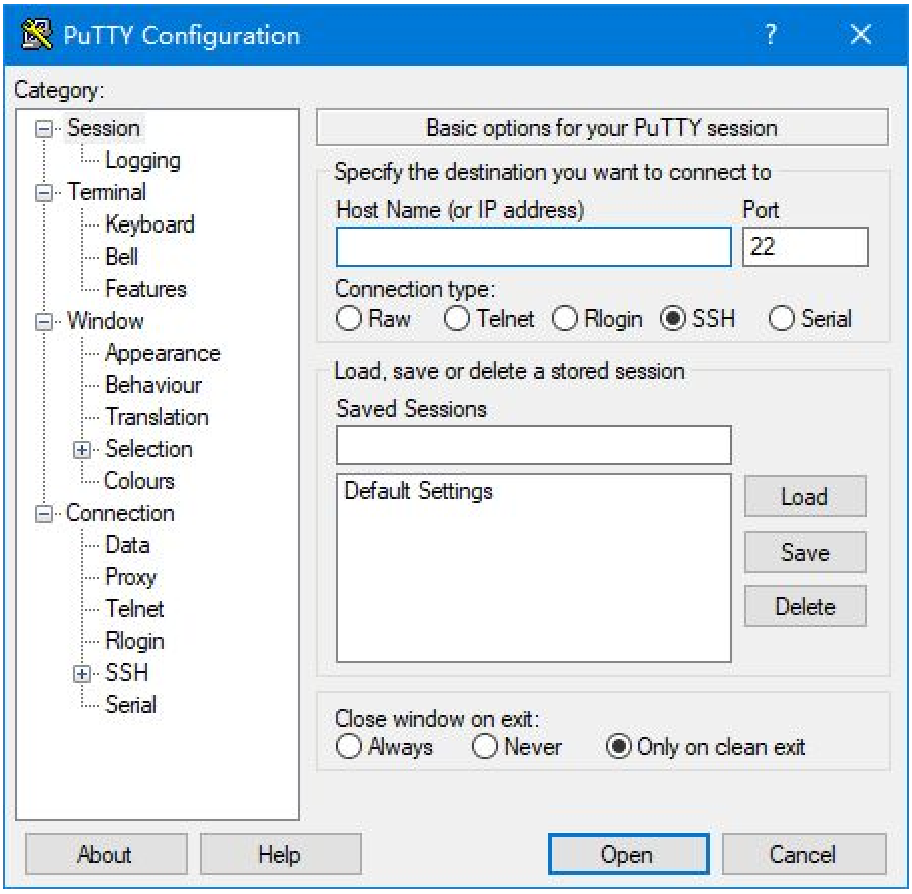
Secure Shell (SSH) is a UNIX-based command interface and protocol that is used by nowadays people for securely getting access to a remote computer. SSH is actually a suite of three utilities - slogin, ssh, and scp. slogin is short for secure login and is a Linux and Unix secure shell SSH (secure shell) remote login program. For slogin to work, you must log in to a machine that supports a secure login and that in most cases, a computer or Internet service provider has multiple login addresses. While scp is short for secure copy. It is a command-line utility that allows you to securely copy files and directories between two locations. With the usage of scp, you can copy a file or directory in three ways: from your local system to a remote system, from a remote system to your local system, and between two remote systems from your local system. When transferring data with scp, both the files and password are encrypted so that anyone snooping on the traffic doesn't get anything sensitive information. SSH commands are encrypted and secure in several ways. Both ends of the client and the server connection are authenticated using a digital certificate, and passwords are protected by being encrypted. In short, SSH allows you to connect to your server securely to perform Linux command-line operations. And although you will be running Linux commands, SSH can still be performed from a Windows, Mac, or Linux computer.

Today I would be using PuTTY from Window to make an SSH connection to the BC's Web Lab Linux server. PuTTY is a free implementation of SSH for Windows and Unix platforms, along with a terminal emulator.

Before starting, there would be three things that we need to make an SSH connection to a server, that is:

- Server IP Address
- SSH Username
- SSH Password

Then download PuTTY and launch the program. It would lead you to this page:



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

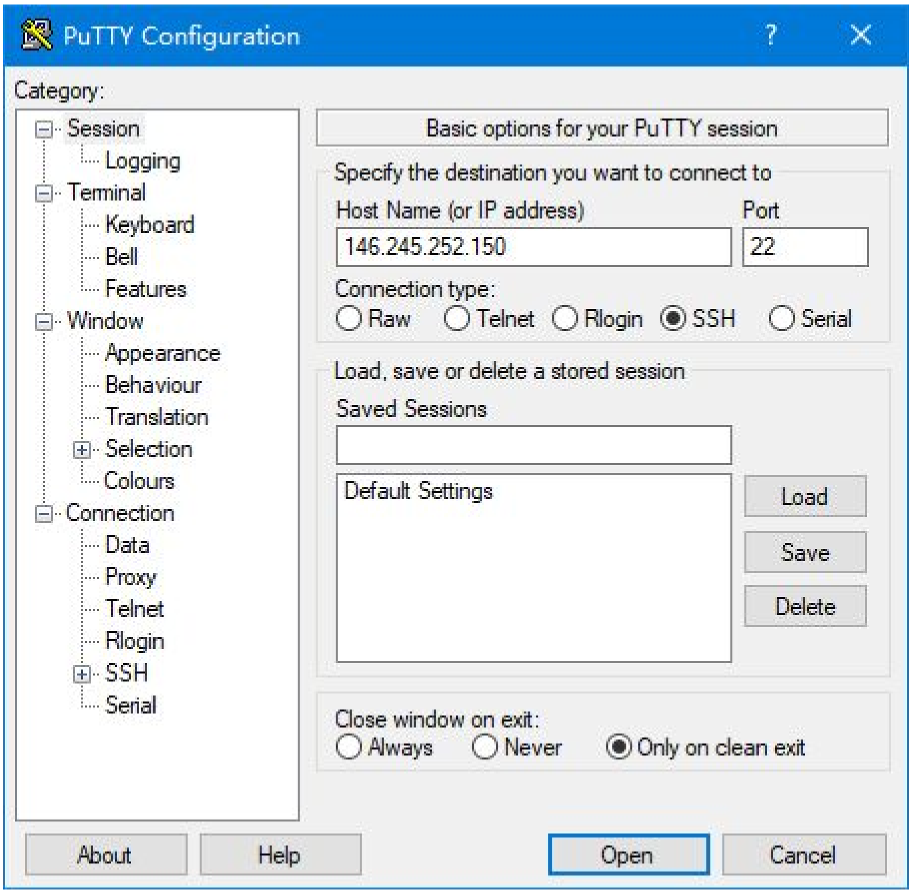
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

This is the main page of the PuTTY program. The information we needed on this page would be the Host Name/IP address of the server that we are connecting to. Also, we have to make sure that the connection type is SSH. The IP address that we can use to connect to the Brooklyn College WEB Lab Linux server are:

- 146.245.252.25
- 146.245.252.26
- 146.245.252.28
- 146.245.252.29
- 146.245.252.30
- 146.245.252.31
- 146.245.252.150

Since either one would be fine, I just chose the last IP address and input it into the program. The port would be set as default to 22 and also the connection type would be set to the default of SSH.



Specify the destination you want to connect to

Host Name (or IP address): 146.245.252.150

Port: 22

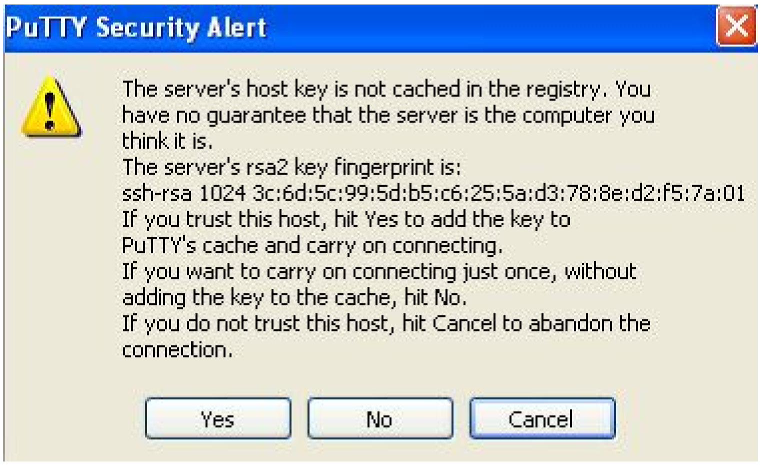
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions

Default Settings

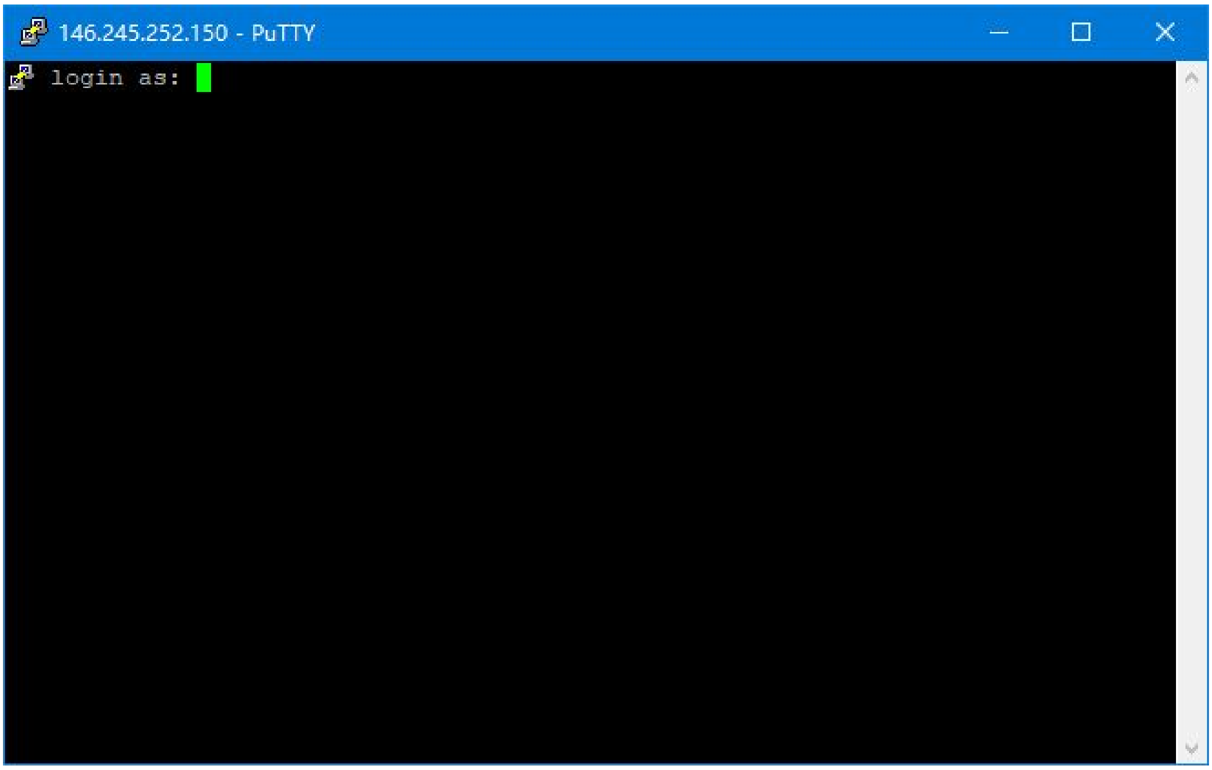
Close window on exit: ○ Always ○ Never ● Only on clean exit

Click open to start the SSH session. If this is your first time connecting to the server from this computer, you will see the following output. Just simply accept the connection by clicking Yes.



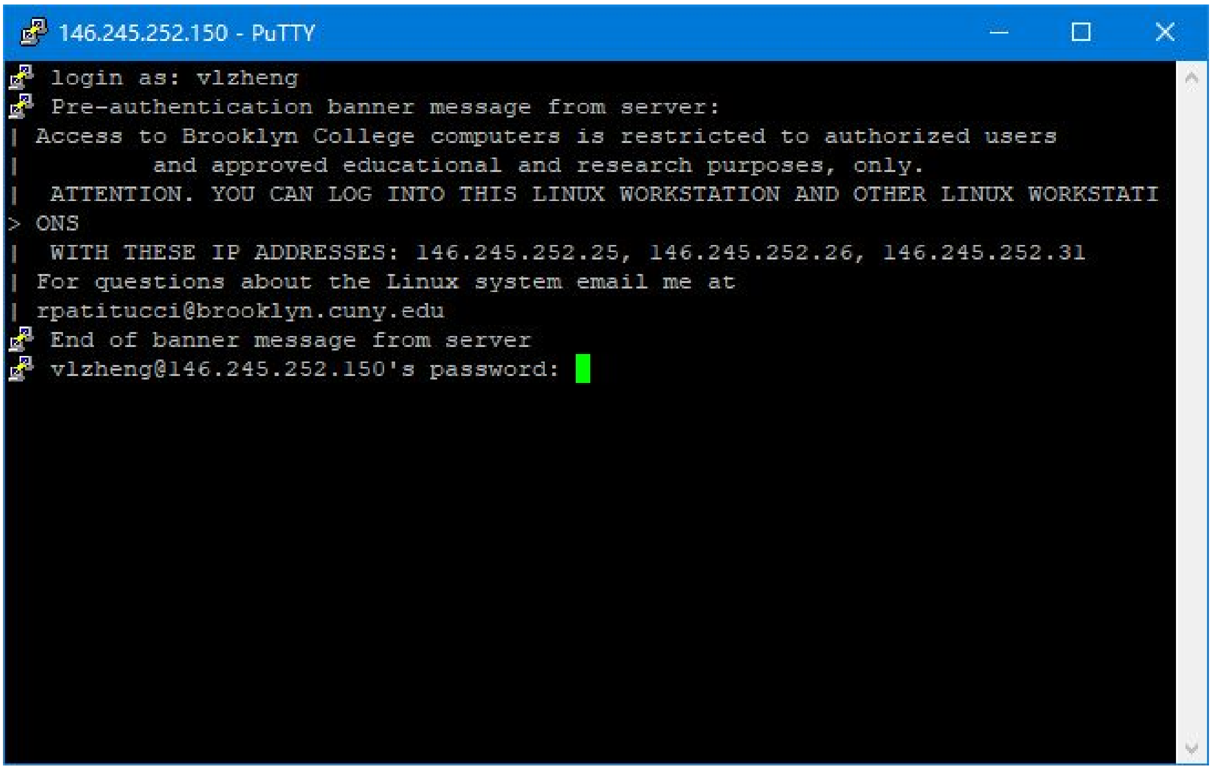
The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 1024 3c:6d:5c:99:5d:b5:c6:25:5a:d3:78:8e:d2:f5:7a:01
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Once the SSH Connection is open, you should see a terminal prompt asking for your username and would look like this:

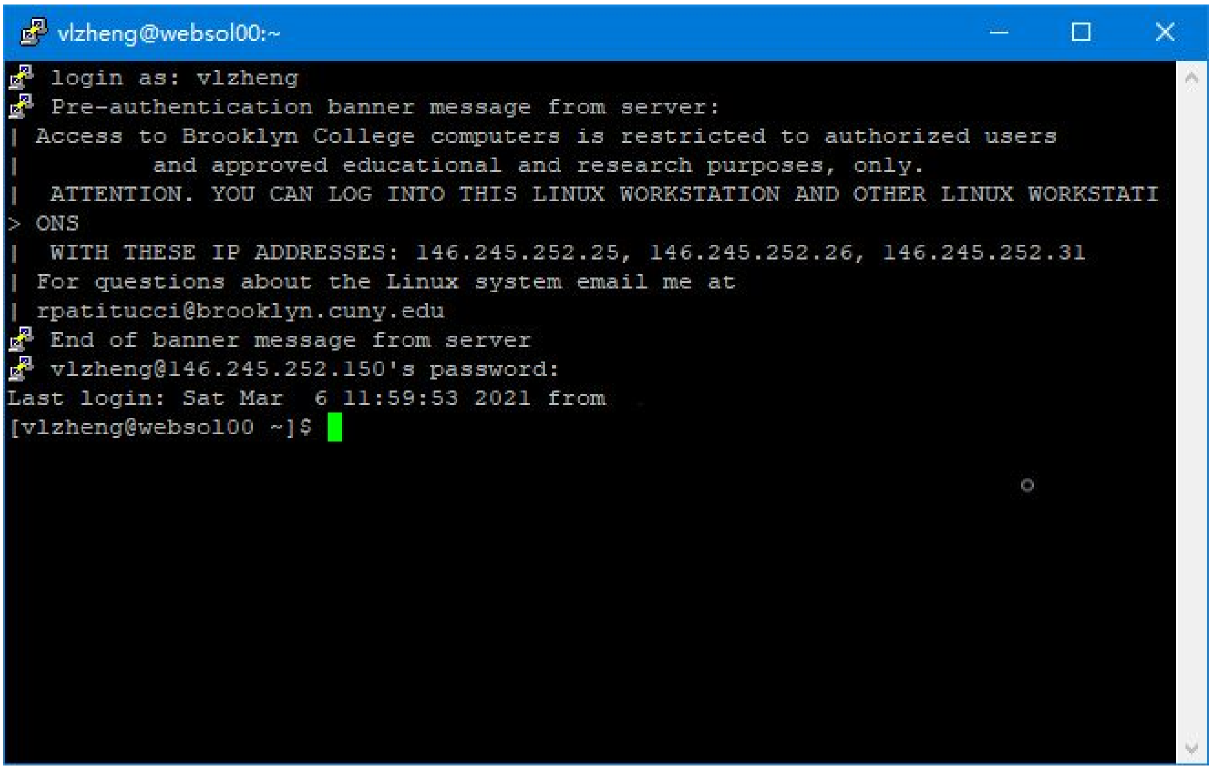


```
login as: 
```

In this case, type your login username. The login credentials are provided by the school. After inputting your username and click enter key, it would prompt and ask for your password. Enter the password and click enter key. Different than inputting the username, when inputting the password, the user is not able to see the password as it may not be displayed as the user type.



```
 login as: vlzheng
 Pre-authentication banner message from server:
| Access to Brooklyn College computers is restricted to authorized users
|         and approved educational and research purposes, only.
|  ATTENTION. YOU CAN LOG INTO THIS LINUX WORKSTATION AND OTHER LINUX WORKSTATI
> ONS
|  WITH THESE IP ADDRESSES: 146.245.252.25, 146.245.252.26, 146.245.252.31
| For questions about the Linux system email me at
| rpatitucci@brooklyn.cuny.edu
 End of banner message from server
 vlzheng@146.245.252.150's password: 
```



```
login as: vlzheng
Pre-authentication banner message from server:
| Access to Brooklyn College computers is restricted to authorized users
|         and approved educational and research purposes, only.
|  ATTENTION. YOU CAN LOG INTO THIS LINUX WORKSTATION AND OTHER LINUX WORKSTATI
> ONS
|  WITH THESE IP ADDRESSES: 146.245.252.25, 146.245.252.26, 146.245.252.31
| For questions about the Linux system email me at
| rpatitucci@brooklyn.cuny.edu
End of banner message from server
vlzheng@146.245.252.150's password:
Last login: Sat Mar  6 11:59:53 2021 from
[vlzheng@websol00 ~]$ 
```

This is the page showing that you were successfully connected to the Linux Server by using SSH.

Part 2: Lookup information for Linux programs and summarize the information.

Examples of commands used in Linux:

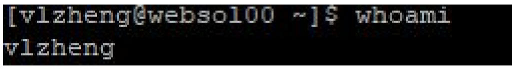
- **curl** is a command-line tool to transfer data to or from a server.
  - Location: curl: /usr/bin/curl /usr/share/man/man1/curl.1.gz
  - Syntax: curl [options] [URL...]
  - Example: curl google.com
- **head** is the complementary of Tail command. It prints the top N number of data of the given input.
  - Location: head: /usr/bin/head /usr/share/man/man1/head.1.gz /usr/share/man/man1p/head.1p.gz
  - Syntax: head [OPTION]... [FILE]...
  - Example: head example.txt
- **tail** is the complementary of head command. The tail command, as the name implies, prints the last N number of data of the given input.

    - Location: tail: /usr/bin/tail /usr/share/man/man1/tail.1.gz /usr/share/man/man1p/tail.1p.gz
    - Syntax: tail [OPTION]... [FILE]...
    - Example: tail example.txt
- **wget** is a command-line utility for downloading files from the web. With wget, you can download files using HTTP, HTTPS, and FTP protocols.
    - Location: wget: /usr/bin/wget /usr/share/man/man1/wget.1.gz
    - Syntax: wget [options] [url]
    - Example: wget http://apache.cs.utah.edu/tomcat/tomcat-9/v9.0.20/bin/apache-tomcat-9.0.20.tar.gz

Other useful commands:

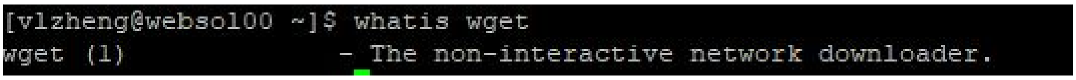
- **ls** - Lists all files and directories in the present working directory.
- **cd or cd ~** - Navigate to HOME directory.
- **cat > [filename]** - Creates a new file.
- **cat [filename]** - Displays the file content.
- **sudo** - Allows regular users to run programs with the security privileges of the superuser or root.
- **rm [filename]** - Deletes a file.
- **history** - Gives a list of all past commands typed in the current terminal session.
- **clear** - Clears the terminal.
- **mkdir** [directoryname] - Creates a new directory in the present working directory or a at the specified path.
- **rmdir** - Deletes a directory.
- **mv** - Renames a directory.
- **finger** - Gives information on all logged-in users.
- **dir** - Display files in the current directory of a remote computer.
- **put file -** upload 'file' from local to the remote computer.
- **get file** - Download 'file' from remote to the local computer.

whoami - To see which user you are using.



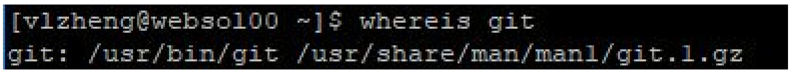
```
[vlzheng@websol00 ~]$ whoami
vlzheng
```

whatis - To get a one-line manual page description.



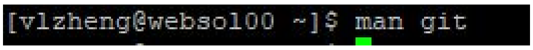
```
[vlzheng@websol00 ~]$ whatis wget
wget (1)             - The non-interactive network downloader.
```

whereis - To find the location of source/binary file of a command and manuals sections for a specified file in Linux system.

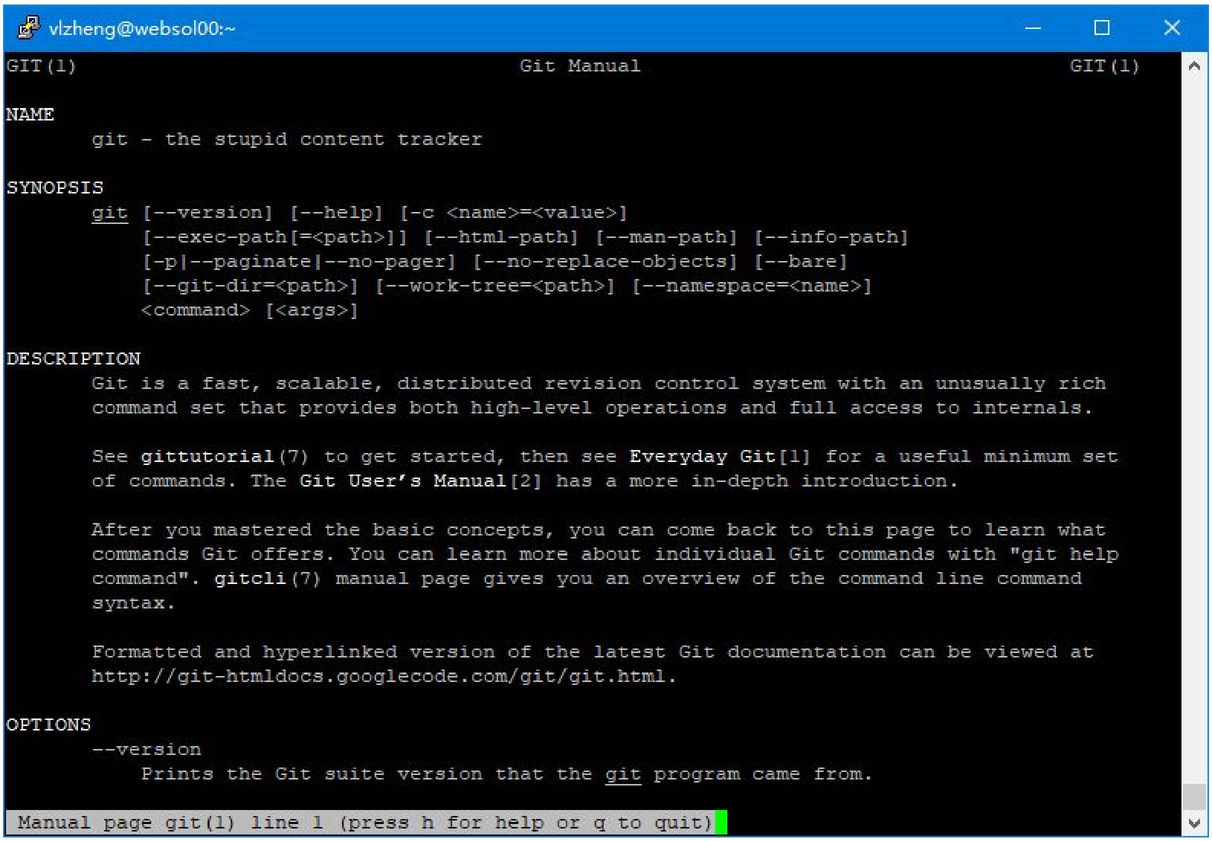


```
[vlzheng@websol00 ~]$ whereis git
git: /usr/bin/git /usr/share/man/man1/git.1.gz
```

man - To view a system's reference manuals

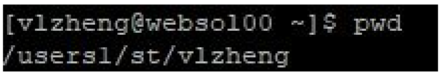


```
[vlzheng@websol00 ~]$ man git
```



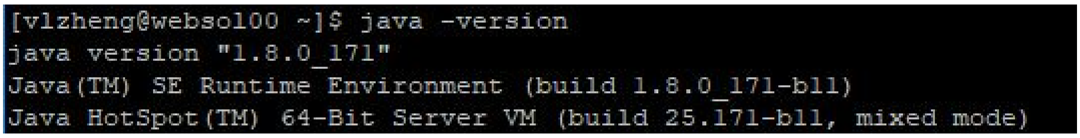
```
GIT(1)                            Git Manual                            GIT(1)

NAME
       git - the stupid content tracker

SYNOPSIS
       git [--version] [--help] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

DESCRIPTION
       Git is a fast, scalable, distributed revision control system with an unusually rich
       command set that provides both high-level operations and full access to internals.

       See gittutorial(7) to get started, then see Everyday Git[1] for a useful minimum set
       of commands. The Git User's Manual[2] has a more in-depth introduction.

       After you mastered the basic concepts, you can come back to this page to learn what
       commands Git offers. You can learn more about individual Git commands with "git help
       command". gitcli(7) manual page gives you an overview of the command line command
       syntax.

       Formatted and hyperlinked version of the latest Git documentation can be viewed at
       http://git-htmldocs.googlecode.com/git/git.html.

OPTIONS
       --version
           Prints the Git suite version that the git program came from.

 Manual page git(1) line 1 (press h for help or q to quit)
```

pwd - Show current directory



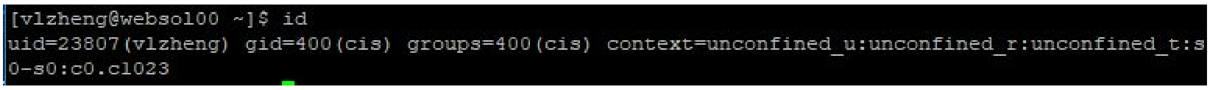
```
[vlzheng@websol00 ~]$ pwd
/users1/st/vlzheng
```

application_name **-version** - To check the version of the program



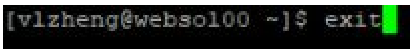
```
[vlzheng@websol00 ~]$ java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

id - Displays the details of the active user e.g. uid, gid, and groups



```
[vlzheng@websol00 ~]$ id
uid=23807(vlzheng) gid=400(cis) groups=400(cis) context=unconfined_u:unconfined_r:unconfined_t:s
0-s0:c0.c1023
```

In order to terminate the connection to the Linux server and exit the terminal, you can just simply type **exit**.



```
[vlzheng@websol00 ~]$ exit
```

Part 3: Text Editing in Terminal

Shortcuts:

- Type nano [fileName] to open an existing file, or create a new file.
- Arrow keys are used to move the cursor up, down, left, and right.

Ctrl+Y	Move page up
Ctrl+V	Move page down
Ctrl+_	Move the cursor to a certain location
Alt+6	Copy the selected text
Ctrl+K	Cut the selected text
Ctrl+U	Paste the selected text
Ctrl+6	Cancel the selection
Alt+U	Undo last action
Alt+E	Redo last action
Ctrl+W	Search for text, move to next match
Ctrl+\	Search and replace
Ctrl+O	Save the modification
Ctrl+X	Exit the editor