Aux Project 1: Shell Scripting

A shell script is a text file that contains a sequence of commands for a UNIX-based operating system. It is called a shell script because it combines a sequence of commands, that would otherwise have to be typed into the keyboard one at a time, into a single script. The shell is the operating system's command-line interface (CLI) and interpreter for the set of commands that are used to communicate with the system. A shell interfaces between the user and the operating system.

A shell script is usually created for command sequences in which a user has a need to use repeatedly in order to save time. Like other programs, the shell script can contain parameters, comments and subcommands that the shell must follow. Users initiate the sequence of commands in the shell script by simply entering the file name on a command line. Typical operations performed by shell scripts include file manipulation, program execution, and printing text.
In the DOS operating system, a shell script is called a batch file. In IBM's mainframe VM operating systems, it's called an EXEC.

How shell scripting works
The basic steps involved with shell scripting are writing the script, making the script accessible to the shell and giving the shell execute permission.
Shell scripts contain ASCII text and are written using a text editor, word processor or graphical user interface (GUI). The content of the script is a series of commands in a language that can be interpreted by the shell. Functions that shell scripts support include loops, variables, if/then/else statements, arrays and shortcuts. Once complete, the file is saved typically with a .txt or .sh extension and in a location that the shell can access.

Types of shells
In Unix and Linux, the two major types of shell scripts are:

1. Bourne again shells (BASH)- BASH is the default shell for Unix version 7. The character for prompting a bourne again shell is $.

2. C shells- A C shell is run in a text terminal window and is able to easily read file commands. The character for prompting a C shell is %.

CSV File
A Comma Separated Values (CSV) file is a plain text file that contains a list of data. These files are often used for exchanging data between different applications. For example, databases and contact managers often support CSV files. These files may sometimes be called Character Separated Values or Comma Delimited files

It is a delimited text file that uses a comma to separate values. A CSV file stores tabular data in plain text format. Each line of the file is a data record. You can use ‘while shell loop’ to read comma-separated CSV file. IFS variable will set CSV separated to , (comma). The read command will read each line and store data into each field.

They mostly use the comma character to separate (or delimit) data, but sometimes use other characters, like semicolons. The idea is that you can export complex data from one application to a CSV file, and then import the data in that CSV file into another application.

Kernel
A kernel interfaces between hardware and software. The Linux kernel is the main component of a Linux operating system (OS) and is the core interface between a computer's hardware and its processes. It communicates between the 2, managing resources as efficiently as possible.
The kernel is so named because—like a seed inside a hard shell—it exists within the OS and controls all the major functions of the hardware, whether it's a phone, laptop, server, or any other kind of computer.

Functions of the Kernel

1. Memory management: Keep track of how much memory is used to store what, and where
2. Process management: Determine which processes can use the central processing unit (CPU), when, and for how long
3. Device drivers: Act as mediator/interpreter between the hardware and processes
4. System calls and security: Receive requests for service from the processes

The kernel, if implemented properly, is invisible to the user, working in its own little world known as kernel space, where it allocates memory and keeps track of where everything is stored. What the user sees—like web browsers and files—are known as the user space. These applications interact with the kernel through a system call interface (SCI).
The kernel is more like a busy personal assistant for a powerful executive (the hardware). It's the assistant's job to relay messages and requests (processes) from employees and the public (users) to the executive, to remember what is stored where (memory), and to determine who has access to the executive at any given time and for how long.

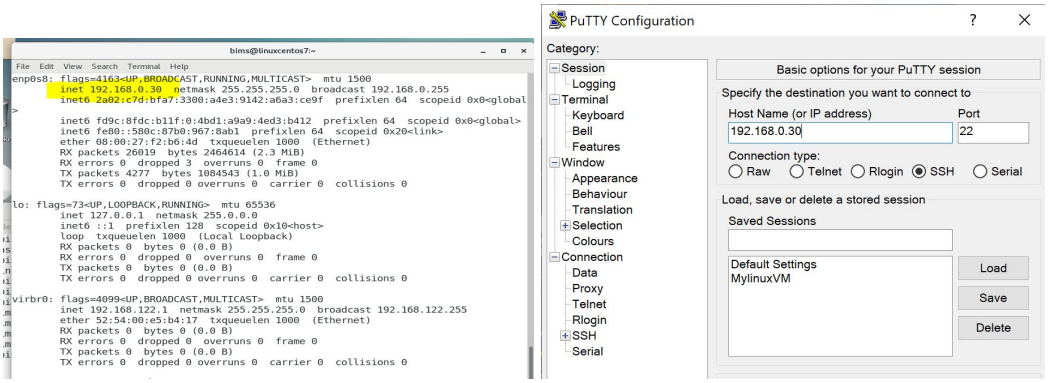
Where the kernel fits within the OS
To put the kernel in context, you can think of a Linux machine as having 3 layers:

1. The hardware: The physical machine—the bottom or base of the system, made up of memory (RAM) and the processor or central processing unit (CPU), as well as input/output (I/O) devices such as storage, networking, and graphics. The CPU performs computations and reads from, and writes to, memory.
2. The Linux kernel: The core of the OS. (See? It's right in the middle.) It's software residing in memory that tells the CPU what to do.
3. User processes: These are the running programs that the kernel manages. User processes are what collectively make up user space. User processes are also known as just processes. The kernel also allows these processes and servers to communicate with each other (known as inter-process communication, or IPC).

In this project, we will onboard 20 new Linux users onto a server. Create a shell script that reads a csv file that contains the first name of the users to be onboarded.

Created Virtual Box
Created virtual machine for Linux
Completed Guest Addition for Linux to allow copy and paste

Locate my IP address in Linux: ifconfig
Input IP address in Putty and press enter in order to open a session in Putty.



Steps taken include:

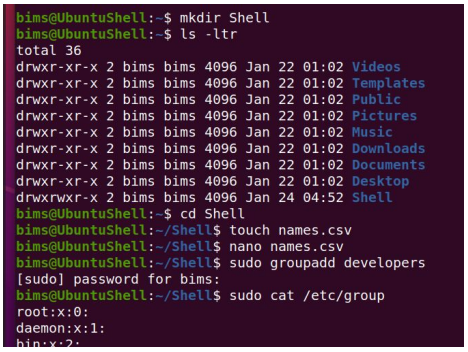
a. Create the project folder called Shell
   mkdir Shell

b. Move into the Shell folder
   cd Shell

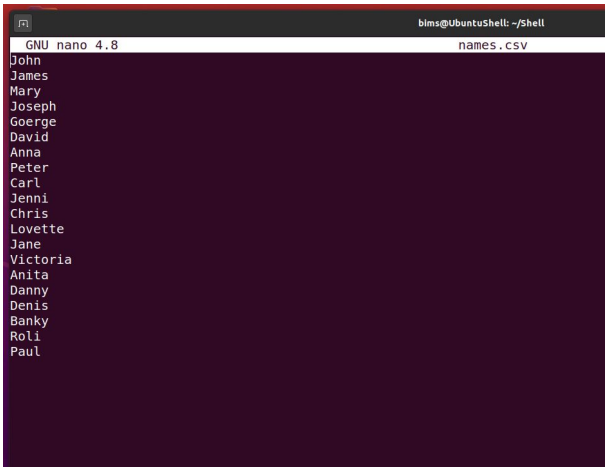
c. Create a csv file name names.csv
   Touch names.csv

d. Open the names.csv file
   nano names.csv

   Create a group called developers: sudo groupadd developers
   Confirm the group has been created: sudo cat /etc/group



e. Insert some random names into it. (One name per line)



Create a script that will read the CSV file, create each user on the server, and add to an existing group called developers (You will need to manually create this group ahead).

Create a file called users: touch users

Create the script and add into the users file:- sudo nano users

```
#!/bin/bash

userfile=$(cat names.csv)

   if [ $(id -u) -eq 0 ]; then

# Reading the CSV file
          for user in $userfile;
          do
         echo $user
      if id "$user" &>/dev/null
      then
         echo "User Exist"
      else

# This will create a new user
      useradd -m -d /home/$user -s /bin/bash -g developers $user
      echo "New User Created"
      echo

# This will create a ssh folder in the user home folder
      su - -c "mkdir ~/.ssh" $user
      echo ".ssh directory created for new user"
      echo

# We need to set the user permission for the ssh dir
       su - -c "chmod 700 ~/.ssh" $user
       echo "user permission for .ssh directory set"
       echo
```

```
# This will create an authorized-key file
        su - -c "touch ~/.ssh/authorized_keys" $user
        echo "Authorized Key File Created"
        echo

# We need to set permission for the key file
        su - -c "chmod 600 ~/.ssh/authorized_keys" $user
        echo "user permission for the Authorized Key File set"
        echo

# We need to create and set public key for users in the server
        cp -R "/home/bims/.ssh/id_rsa.pub" "/home/$user/.ssh/authorized_keys"
        echo "Copyied the Public Key to New User Account on the server"
        echo
        echo

        echo "USER CREATED"
           fi
        done
    else
    echo "Only Admin Can Onboard A User"
    fi
```

Change ysf to bims to allow for connection to my name

Check the users file: sudo cat users

```
ubuntu@ip-172-31-38-31:~/Shell$
ubuntu@ip-172-31-38-31:~/Shell$ touch users
ubuntu@ip-172-31-38-31:~/Shell$ sudo nano users
ubuntu@ip-172-31-38-31:~/Shell$ sudo cat users
#!/bin/bash

userfile=$(cat names.csv)

    if [ $(id -u) -eq 0 ]; then

# Reading the CSV file
        for user in $userfile;
        do
            echo $user
        if id "$user" &>/dev/null
        then
            echo "User Exist"
        else

# This will create a new user
        useradd -m -d /home/$user -s /bin/bash -g developers $user
        echo "New User Created"
        echo

# This will create a ssh folder in the user home folder
        su - -c "mkdir ~/.ssh" $user
        echo ".ssh directory created for new user"
        echo

# We need to set the user permission for the ssh dir
         su - -c "chmod 700 ~/.ssh" $user
         echo "user permission for .ssh directory set"
         echo
```

```
# This will create an authorized-key file
        su - -c "touch ~/.ssh/authorized_keys" $user
        echo "Authorized Key File Created"
        echo

# We need to set permission for the key file
        su - -c "chmod 600 ~/.ssh/authorized_keys" $user
        echo "user permission for the Authorized Key File set"
        echo

# We need to create and set public key for users in the server
        cp -R "/home/bims/.ssh/id_rsa.pub" "/home/$user/.ssh/authorized_keys"
        echo "Copyied the Public Key to New User Account on the server"
        echo
        echo

        echo "USER CREATED"
            fi
        done
    else
    echo "Only Admin Can Onboard A User"
    fi
ubuntu@ip-172-31-38-31:~/Shell$
```

Install SSH: sudo apt-get install openssh-server

SSH folder: sudo nano /etc/ssh/sshd_config

```
ubuntu@ip-172-31-38-31:~/Shell$
ubuntu@ip-172-31-38-31:~/Shell$ cd ..
ubuntu@ip-172-31-38-31:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-38-31:~$
```

Then bashed the files so all the users' files can be edited/ configured at the same time. This will create the users in the users file as a bash

sudo bash users

```
ubuntu@ip-172-31-38-31:~$ cd Shell
ubuntu@ip-172-31-38-31:~/Shell$ sudo bash users
John
New User Created

.ssh directory created for new user

user permission for .ssh directory set

Authorized Key File Created

user permission for the Authorized Key File set

cp: cannot stat '/home/bims/.ssh/id_rsa.pub': No such file or directory
Copyied the Public Key to New User Account on the server


USER CREATED
James
New User Created

.ssh directory created for new user

user permission for .ssh directory set

Authorized Key File Created

user permission for the Authorized Key File set

cp: cannot stat '/home/bims/.ssh/id_rsa.pub': No such file or directory
```

```
ubuntu@ip-172-31-38-31:~$
ubuntu@ip-172-31-38-31:~$ cd Shell
ubuntu@ip-172-31-38-31:~/Shell$ ls -la
total 16
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 10 06:37 .
drwxr-xr-x 6 ubuntu ubuntu 4096 Feb 10 06:34 ..
-rw-rw-r-- 1 ubuntu ubuntu  120 Feb 10 06:34 names.csv
-rw-rw-r-- 1 ubuntu ubuntu 1388 Feb 10 06:37 users
ubuntu@ip-172-31-38-31:~/Shell$ sudo bash users
John
User Exist
James
User Exist
Mary
User Exist
Joseph
User Exist
George
User Exist
David
User Exist
```

Ensure that your script will first check for the existence of the user on the system, before it will attempt to create that in.

Ensure that the user that is being created also has a default home folder
Ensure that each user has a .ssh folder within its HOME folder. If it does not exist, then create it.
For each user's SSH configuration, create an authorized_keys file and add the below public key.

Random user's file to show that the file has been created inside of home directory and confirm the existence of .ssh file.
Go into the home directory of the user: $ cd /home/Paul
Then do ls -la

```
ubuntu@ip-172-31-38-31:~/Shell$ cd /home/Paul
ubuntu@ip-172-31-38-31:/home/Paul$
ubuntu@ip-172-31-38-31:/home/Paul$ ls -la
total 24
drwxr-xr-x  3 Paul developers 4096 Feb 10 06:42 .
drwxr-xr-x 23 root root       4096 Feb 10 06:42 ..
-rw-r--r--  1 Paul developers  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 Paul developers 3771 Feb 25  2020 .bashrc
-rw-r--r--  1 Paul developers  807 Feb 25  2020 .profile
drwx------  2 Paul developers 4096 Feb 10 06:42 .ssh
ubuntu@ip-172-31-38-31:/home/Paul$
```

Next is to generate the authorisation keys both private and public. The public authorisation key will be added to each user's folder in order to permit them access into the server.

Generate public/private rsa key pair for bims to enable the users to connect: ssh-keygen

```
ubuntu@ip-172-31-38-31:/$
ubuntu@ip-172-31-38-31:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:NIDgt8YzIHVCERknCPRQw3svjR1Gpmpimh0NrFNkykc ubuntu@ip-172-31-38-31
The key's randomart image is:
+---[RSA 3072]----+
|+=%Oo..          |
|.o*Eo  +         |
|o=+.o + o        |
|.o+* + + .       |
| o.oO * S        |
|oo.+.= +         |
|o+o.  .          |
|o .              |
|                 |
+----[SHA256]-----+
ubuntu@ip-172-31-38-31:/$
ubuntu@ip-172-31-38-31:/$
```

cd into home and then cd into .ssh file

$ cd

$ cd .ssh

$ ls -la

Go into the private and public authorisation key files, delete and replace with the keys below

$ vi id_rsa

$ vi id_rsa.pub

PUBLIC AUTHORIZATION KEY: vi id_rsa.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCzKZyicHxIkkISrNIxsJyyTrcIdBIt84Z0cQb3R4k0jH53kxkaT5hP8tfWT
e62LXi7vV86fY+SX7TBNM76XGCbw/6vrMGegm6J1x2i1AiLNwq5nqTjOGn0AIwku4IICCLAB7tdfRyVuCarmBlwn
y3lzRyybIUAWXR/D6vpN09MsDILbKdhay+Q/p9OUBMSLPqXdY/QIh/Oe3rVv1IwY3AohNfq7V3tO88zKswfA5iiex
NiSYX1myT0OrX8cBE771j9quoNZhQgaLI1mIMtAvnHQChrn9k2nUaO/BMBCQGol5XzGv1ado7hgoVPoluIUD+F
GNo/pH4zcmDLICH6drXY/C9MESnkMUPLFxBXKO/OitApY71vRao9nAhAwpVMsy6FqiOb5uawhvhoHYIHTV/f4
EtagVagRMP2PxYMYR6jykIV4MPJTkCm+IGhTyMlRu+qRQjdLn8AAtHf4aEV8dIkoGh088DI7eA/4o0wz4OV4upH
5ewSFS+5IHmRECEW5Nc=

PRIVATE AUTHORIZATION KEY: $ vi id_rsa

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAsymconB8SJJJUqzZcbCcsk63CHQSLfOGdHEG90eJNIx+d5MZGk+Y
T/LX1k3uti14u71fOn2Pkl+0wTTO+lxgm8P+r6zBnoJuidcdotQlizcKuZ6k4zhp9ACMJL
uCJQgiwAe7XX0clbgmq5gZcJ8t5c0csmyFAFl0fw+r6TdPTLAyC2ynYWsvkP6fTlATEiz6
l3WP0CIfznt61b9ZcGNwKITX6u1d7TvPMyrMHwOYonsTYkmF9Zsk9Dq1/HARO+9Y/arqDW
YUIGiyNZiDLQL5x0Aoa5/ZNp1GjvwTAQkBqJeV8xr9WnaO4YKFT6JbiFA/hRjaP6R+M3Jg
yyAh+na12PwvTBEp5DFDyxcQVyjvzorQKWO9b0WqPZwIQMKVTLMuhaojm+bmsIb4aB2CB0
1f3+BLWoFWoETD9j8WDGEeo8pCFeDDyU5ApvpRoU8jJUbvqkUI3S5/AALR3+GhFfHSJKBo
dPPAyO3gP+KNMM+DleLqR+XsEhUvuSB5kRAhFuTXAAAFgIuJ0uiLidLoAAAAB3NzaC1yc2
EAAAGBALMpnKJwfEiSSVKs2XGwnLJOtwh0Ei3zhnRxBvdHiTSMfneTGRpPmE/y19ZN7rYt
eLu9Xzp9j5JftME0zvpcYJvD/q+swZ6CbonXHaLUCIs3CrmepOM4afQAjCS7giUIIsAHu1
19HJW4JquYGXCfLeXNHLJshQBZdH8Pq+k3T0ywMgtsp2FrL5D+n05QExIs+pd1j9AiH857
etW/WXBjcCiE1+rtXe07zzMqzB8DmKJ7E2JJhfWbJPQ6tfxwETvvWP2q6g1mFCBosjWYgy
0C+cdAKGuf2TadRo78EwEJAaiXlfMa/Vp2juGChU+iW4hQP4UY2j+kfjNyYMsgIfp2tdj8
L0wRKeQxQ8sXEFco786K0CljvW9Fqj2cCEDClUyzLoWqI5vm5rCG+GgdggdNX9/gS1qBVq

BEw/Y/FgxhHqPKQhXgw8lOQKb6UaFPIyVG76pFCN0ufwAC0d/hoRXx0iSgaHTzwMjt4D/i
jTDPg5Xi6kfl7BIVL7kgeZEQIRbk1wAAAAMBAAEAAAGAPf8KOpOeDibAxKEXZWXt8y2V3J
D9sXTxc92gwXS5n7t2D76REy+zzwaDdZ7mGZhGjQCMsVq9kbMYgzrY3H2W2I/L09J99XHA
+mW71Zp1kmbriSvCdvYQg+SkmhlggZv9GmISjdk7SPu+Nead9wC+CyUc5wjyRRqvW0B7Bm
qjQDBAQP/KM8W5Yf0Z9ylyT/nMhRijOSx1wSeta8WZF3DxYLQHWz3kILFvk48dryW5bZAV
Nw+mEUUsVm7yhnXpIMpDdl7wlDlqAWcuEQKJ7WJ7swuZM/FTQW4rFMmpDO8Q8PgijqOFDQ
jl8XfCPCkOhI9JOFTbmImTxfbRZ/NYYF09cFcqhKyvEi/Egx2oUZq4M81EGpP+EZnWgZtG
/PHqrSqIW166fixe/47eGCSt+AlyeR8SZCA1jjMRf7WB1RjANUHgC59tNTMQiFg+T5c2Yj
ORmPT0PpzEtQ+WMSMI5hGoklmqXuS5iiyJx7HyLOnK7wNloj7oqboz91wcCYnYWCORAAAA
wQDUbuGf0dAtJ4Qr2vdHiIi4dHAlMQMMsw/12CmpuSoqeEIWHVpAEBpqzx67qDZ+AMpBDV
BU9KbXe7IIzzfwUvxI1WCycg/pJM0OMjyigvz4XziuSVmSuy10HNvECvpxI3Qx8iF/HgAP
eyYe369FHEBsNZ5M5KhZ4oHI/XgZB5OGOaxErJd3wXhGASHnsWcmIswIjat7hH9WlAeWAk
/aeMz92iSDnYBOr+gAycsBm/skEDrN7dD45iISvLZ6DQ2hbKAAAADBAOhLy9Tiki1IM2Gg
ma8KkUiLrqqx8IexPd580n7KsL32U2iu6Y88+skC8pkZQmIVG2UQhjiVLpNBgrzKKDJciK
/Iyen21npQjuYaJPUgVUG0sjMtTpgGwbN/IVyHO28KSOogB6MclRBW7Z2SJggSAJaQmO9g
u7kieXbtf+5A7gUSb7icD629OiYCEJMTKTpVS/Pk7NDx/ZXQVzGrkJMKdPFU8nDoOjFLSP
jdbbddYe6zuB/HwabV3Lpaxl38tNG78wAAAMEAxXHS2IRABAvX7+OmZO2JU7+9Gxh/gudJ
eXf76c10kKvUztoe8Mskw79yVq6LtYd0JGOVx0oNgMeZJHmwUc2qVPKaFGEhSG6MuFn3J2
O5+Kt+KfU5M9uAN7tob3+yG18ZJt9FY+5FTK1TV5LmF5OTGBN9XyehT2Miqa8sSu80rwpN
nhe+U/XswAp9KEVYkSIjFeoy/amsOP+qvRke1dKWBsU12IbhnMgjDHVggkYV52l7d9S2bx
kmaSGj362OnCCNAAAACWRhcmVARGFyZQE=
-----END OPENSSH PRIVATE KEY-----

Private Key

```
ubuntu@ip-172-31-38-31:/$
ubuntu@ip-172-31-38-31:/$ cd
ubuntu@ip-172-31-38-31:~$ cd .ssh
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$ ls -la
total 20
drwx------ 2 ubuntu ubuntu 4096 Feb 10 06:50 .
drwxr-xr-x 6 ubuntu ubuntu 4096 Feb 10 06:34 ..
-rw------- 1 ubuntu ubuntu  390 Feb 10 06:31 authorized_keys
-rw------- 1 ubuntu ubuntu 2610 Feb 10 06:50 id_rsa
-rw-r--r-- 1 ubuntu ubuntu  576 Feb 10 06:50 id_rsa.pub
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$ vi id_rsa
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$ sudo cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAsymconB8SJJJUqzZcbCcsk63CHQSLfOGdHEG90eJNIx+d5MZGk+Y
T/LX1k3uti14u71fOn2Pkl+0wTTO+lxgm8P+r6zBnoJuidcdotQIizcKuZ6k4zhp9ACMJL
uCJQgiwAe7XX0clbgmq5gZcJ8t5c0csmyFAFl0fw+r6TdPTLAyC2ynYWsvkP6fTlATEiz6
l3WP0CIfznt61b9ZcGNwKITX6u1d7TvPMyrMHwOYonsTYkmF9Zsk9Dq1/HARO+9Y/arqDW
YUIGiyNZiDLQL5x0Aoa5/ZNp1GjvwTAQkBqJeV8xr9WnaO4YKFT6JbiFA/hRjaP6R+M3Jg
yyAh+na12PwvTBEp5DFDyxcQVyjvzorQKWO9b0WqPZwIQMKVTLMuhaojm+bmsIb4aB2CB0
1f3+BLWoFWoETD9j8WDGEeo8pCFeDDyU5ApvpRoU8jJUbvqkUI3S5/AALR3+GhFfHSJKBo
dPPAyO3gP+KNMM+DleLqR+XsEhUvuSB5kRAhFuTXAAAFgIuJ0uiLidLoAAAAB3NzaC1yc2
EAAAGBALMpnKJwfEiSSVKs2XGwnLJ0twh0Ei3zhnRxBvdHiTSMfneTGRpPmE/y19ZN7rYt
eLu9Xzp9j5JftME0zvpcYJvD/q+swZ6CbonXHaLUCIs3CrmepOM4afQAjCS7giUIIsAHu1
19HJW4JquYGXCfLeXNHLJshQBZdH8Pq+k3T0ywMgtsp2FrL5D+n05QExIs+pd1j9AiH857
etW/WXBjcCiE1+rtXe07zzMqzB8DmKJ7E2JJhfWbJPQ6tfxwETvvWP2q6g1mFCBosjWYgy
```

Public key

```
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$ vi id_rsa.pub
ubuntu@ip-172-31-38-31:~/.ssh$
ubuntu@ip-172-31-38-31:~/.ssh$ sudo cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzKZyicHxIkklSrNlxsJyyTrcIdBIt84Z0cQb3R4k0
jH53kxkaT5hP8tfWTe62LXi7vV86fY+SX7TBNM76XGCbw/6vrMGegm6J1x2i1AiLNwq5nqTjOGn0AIwk
u4IlCCLAB7tdfRyVuCarmBlwny3lzRyybIUAWXR/D6vpN09MsDILbKdhay+Q/p9OUBMSLPqXdY/QIh/O
e3rVv1lwY3AohNfq7V3tO88zKswfA5iiexNiSYX1myT0OrX8cBE771j9quoNZhQgaLI1mIMtAvnHQChr
n9k2nUaO/BMBCQGol5XzGv1ado7hgoVPoluIUD+FGNo/pH4zcmDLICH6drXY/C9MESnkMUPLFxBXKO/O
itApY71vRao9nAhAwpVMsy6FqiOb5uawhvhoHYIHTV/f4EtagVagRMP2PxYMYR6jykIV4MPJTkCm+lGh
TyMlRu+qRQjdLn8AAtHf4aEV8dIkoGh088DI7eA/4o0wz4OV4upH5ewSFS+5IHmRECEW5Nc=
ubuntu@ip-172-31-38-31:~/.ssh$
```

Test a few of the users randomly, and ensure that you are able to connect to the server using the above public key.

Using Paul:

Need to add the Server's public authorization key to Paul's authorisation Key folder to enable access into the server.
Go into Pauls's folder: cd /home/Paul
cd into ssh: cd .ssh
Access denied, then work as root user: sudo su
Then add the server's public key (above) into Paul's authorisation key folder: nano authorized_keys

```
ubuntu@ip-172-31-38-31:~$ sudo su
root@ip-172-31-38-31:/home/ubuntu#
root@ip-172-31-38-31:/home/ubuntu#
root@ip-172-31-38-31:/home/ubuntu#
root@ip-172-31-38-31:/home/ubuntu# cd ..
root@ip-172-31-38-31:/home# cd ..
root@ip-172-31-38-31:/#
root@ip-172-31-38-31:/# cd /home/Paul
root@ip-172-31-38-31:/home/Paul# ls -la
total 24
drwxr-xr-x  3 Paul developers 4096 Feb 10 06:42 .
drwxr-xr-x 23 root root       4096 Feb 10 06:42 ..
-rw-r--r--  1 Paul developers  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 Paul developers 3771 Feb 25  2020 .bashrc
-rw-r--r--  1 Paul developers  807 Feb 25  2020 .profile
drwx------  2 Paul developers 4096 Feb 10 06:42 .ssh
root@ip-172-31-38-31:/home/Paul#
root@ip-172-31-38-31:/home/Paul# cd .ssh
root@ip-172-31-38-31:/home/Paul/.ssh#
root@ip-172-31-38-31:/home/Paul/.ssh#
root@ip-172-31-38-31:/home/Paul/.ssh# ls -la
total 8
drwx------ 2 Paul developers 4096 Feb 10 06:42 .
drwxr-xr-x 3 Paul developers 4096 Feb 10 06:42 ..
-rw------- 1 Paul developers    0 Feb 10 06:42 authorized_keys
root@ip-172-31-38-31:/home/Paul/.ssh#
root@ip-172-31-38-31:/home/Paul/.ssh# nano authorized_keys
root@ip-172-31-38-31:/home/Paul/.ssh#
root@ip-172-31-38-31:/home/Paul/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzKZyicHxIkklSrNlxsJyyTrcIdBIt84Z0cQb3R4k0
jH53kxkaT5hP8tfWTe62LXi7vV86fY+SX7TBNM76XGCbw/6vrMGegm6J1x2i1AiLNwq5nqTjOGn0AIwk
u4IlCCLAB7tdfRyVuCarmBlwny3lzRyybIUAWXR/D6vpN09MsDILbKdhay+Q/p9OUBMSLPqXdY/QIh/O
e3rVv1lwY3AohNfq7V3tO88zKswfA5iiexNiSYX1myTOOrX8cBE771j9quoNZhQgaLI1mIMtAvnHQChr
n9k2nUaO/BMBCQGol5XzGv1ado7hgoVPoluIUD+FGNo/pH4zcmDLICH6drXY/C9MESnkMUPLFxBXKO/O
```

Log out from the root.

Enable Paul access the server: $ ssh Paul@172.31.38.31

```
ubuntu@ip-172-31-38-31:~$
ubuntu@ip-172-31-38-31:~$ ssh Paul@172.31.38.31
The authenticity of host '172.31.38.31 (172.31.38.31)' can't be established.
ECDSA key fingerprint is SHA256:hiE1LqdB2NOBCKBBulyFVLRRrwPK89e2gQePyC1ofW8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.38.31' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1037-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Feb 10 07:08:45 UTC 2021

  System load:  0.0               Processes:             109
  Usage of /:   16.5% of 7.69GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for eth0: 172.31.38.31
  Swap usage:   0%


1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
```

Repeat the same for another user - Peter

```
root@ip-172-31-38-31:/home/ubuntu#
root@ip-172-31-38-31:/home/ubuntu# cd ..
root@ip-172-31-38-31:/home# cd Peter
root@ip-172-31-38-31:/home/Peter# ls -la
total 24
drwxr-xr-x  3 Peter developers 4096 Feb 10 06:42 .
drwxr-xr-x 23 root  root       4096 Feb 10 06:42 ..
-rw-r--r--  1 Peter developers  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 Peter developers 3771 Feb 25  2020 .bashrc
-rw-r--r--  1 Peter developers  807 Feb 25  2020 .profile
drwx------  2 Peter developers 4096 Feb 10 06:42 .ssh
root@ip-172-31-38-31:/home/Peter#
root@ip-172-31-38-31:/home/Peter# cd .ssh
root@ip-172-31-38-31:/home/Peter/.ssh# ls -la
total 8
drwx------ 2 Peter developers 4096 Feb 10 06:42 .
drwxr-xr-x 3 Peter developers 4096 Feb 10 06:42 ..
-rw------- 1 Peter developers    0 Feb 10 06:42 authorized_keys
root@ip-172-31-38-31:/home/Peter/.ssh#
root@ip-172-31-38-31:/home/Peter/.ssh# nano authorized_keys
root@ip-172-31-38-31:/home/Peter/.ssh#
root@ip-172-31-38-31:/home/Peter/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzKZyicHxIkklSrNlxsJyyTrcIdBIt84Z0cQb3R4k0
jH53kxkaT5hP8tfWTe62LXi7vV86fY+SX7TBNM76XGCbw/6vrMGegm6J1x2i1AiLNwq5nqTjOGn0AIwk
u4IlCCLAB7tdfRyVuCarmBlwny3lzRyybIUAWXR/D6vpN09MsDILbKdhay+Q/p9OUBMSLPqXdY/QIh/O
e3rVv1lwY3AohNfq7V3tO88zKswfA5iiexNiSYX1myT0OrX8cBE771j9quoNZhQgaLI1mIMtAvnHQChr
n9k2nUaO/BMBCQGol5XzGv1ado7hgoVPoluIUD+FGNo/pH4zcmDLICH6drXY/C9MESnkMUPLFxBXKO/O
itApY71vRao9nAhAwpVMsy6FqiOb5uawhvhoHYIHTV/f4EtagVagRMP2PxYMYR6jykIV4MPJTkCm+lGh
TyMlRu+qRQjdLn8AAtHf4aEV8dIkoGh088DI7eA/4o0wz4OV4upH5ewSFS+5IHmRECEW5Nc=
root@ip-172-31-38-31:/home/Peter/.ssh#
root@ip-172-31-38-31:/home/Peter/.ssh#
```

Users - Paul and Peter have been able to connect to the server successfully.

```
root@ip-172-31-38-31:/home/Peter/.ssh# exit
exit
ubuntu@ip-172-31-38-31:~$
ubuntu@ip-172-31-38-31:~$
ubuntu@ip-172-31-38-31:~$ ssh Peter@172.31.38.31
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1037-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Feb 10 07:14:17 UTC 2021

  System load:  0.0               Processes:             110
  Usage of /:   16.6% of 7.69GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for eth0: 172.31.38.31
  Swap usage:   0%


1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

Created a Repository on Github to move my project
Name:auxiliary-project
Description: Shell Scripting

Steps:
Opened an account with Github and created a repository
Install git into terminal: git init
Add the folders you want to move: git add .
Check the status: git status

```
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$ git init
Initialized empty Git repository in /home/ubuntu/.git/
ubuntu@ip-172-31-19-243:~$ git add .
ubuntu@ip-172-31-19-243:~$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .Xauthority
        new file:   .bash_logout
        new file:   .bashrc
        new file:   .cache/motd.legal-displayed
        new file:   .profile
        new file:   .ssh/authorized_keys
        new file:   .sudo_as_admin_successful
        new file:   Shell/names.csv
        new file:   Shell/users

ubuntu@ip-172-31-19-243:~$
```

Commit: git commit -m "my initial commit"
Add your name and email for identification purposes:
    git config --global user.name "Bims"
    git config --global user.email gbemisola6@gmail.com

```
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$ git commit -m "my initial commit"
[master (root-commit) b143b1c] my initial commit
 Committer: Ubuntu <ubuntu@ip-172-31-19-243.eu-west-2.compute.interna
l>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 9 files changed, 224 insertions(+)
 create mode 100644 .Xauthority
 create mode 100644 .bash_logout
 create mode 100644 .bashrc
 create mode 100644 .cache/motd.legal-displayed
 create mode 100644 .profile
 create mode 100644 .ssh/authorized_keys
 create mode 100644 .sudo_as_admin_successful
 create mode 100644 Shell/names.csv
 create mode 100644 Shell/users
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$ git config --global user.name "Bims"
ubuntu@ip-172-31-19-243:~$ git config --global user.email gbemisola6@
gmail.com
ubuntu@ip-172-31-19-243:~$
```

Generate ssh key:

$ cd .ssh

$ ssh-keygen

$ ls -la

```
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:poIArkipQvAkAhxJm4Z418cBOI8ugskRK5fxQF5WfDU ubuntu@ip-172-31-1
9-243
The key's randomart image is:
+---[RSA 3072]----+
|+++ o+o...E      |
|+==oo..... .     |
|B+*=.+..o        |
|BB+.o ..         |
|=Bo.    S        |
|Xo...  o         |
|=.... .          |
|.     .          |
|                 |
+----[SHA256]-----+
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$ ls -la
total 20
drwx------ 2 ubuntu ubuntu 4096 Feb 11 06:06 .
drwxr-xr-x 7 ubuntu ubuntu 4096 Feb 11 06:03 ..
-rw------- 1 ubuntu ubuntu  390 Feb 10 06:08 authorized_keys
-rw------- 1 ubuntu ubuntu 2610 Feb 11 06:06 id_rsa
-rw-r--r-- 1 ubuntu ubuntu  577 Feb 11 06:06 id_rsa.pub
ubuntu@ip-172-31-19-243:~/.ssh$
```

$ cat ~/.ssh/id_rsa.pub

Copy the public file: vi ~/.ssh/id_rsa.pub

Exit .ssh folder: cd ..

```
ubuntu@ip-172-31-19-243:~/.ssh$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDT2WzczWeI7j2wjKcSjGOv52KypbCNw
qmJfrZLW5S3LdQP/QKl+VPqbg2fr6fkbdNecz6Jl4SzgNBbhf9qcDvg55zDXCeKMgX2ah
lHp2EU9R42ANsViu2AiQopdLp65fDs0WJO7agFbPLo2bdNJubfrxL0GDn+1CyJEZFOGLD
qnFjaGR/viUF376HM3fzMYw3JAfE96JaQrnOluqh/LLe/QdWamvKa72wPT3f8LG6M50ay
5pxsR2mkKWMJ5Qom6UuhuqiW7YqNPUkOIp1wjEAzk1dmrzaAOcxE6v0Rxqx+reJYym/DZ
VxzN32FcFpxGzxU7Qj7f41i3+mPRqN+r37QWh8twXEvjmGdcCfJkpZkO3e241HIg5yFgB
TyXDayfx71owUr++6//ixJ9nXKb0oZ8tOu1wgqJILrVXZZhD3S/LhmgsiSqc+/ZZcNIZ8
7+XKg6E8hulBsKGZ8zWCnK4hHSRQPLTHhvZG3e8/RwhnihtgsQNsyPcP9uHhsSriMSxk=
 ubuntu@ip-172-31-19-243
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$ vi ~/.ssh/id_rsa.pub
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$
ubuntu@ip-172-31-19-243:~/.ssh$ cd ..
ubuntu@ip-172-31-19-243:~$
```

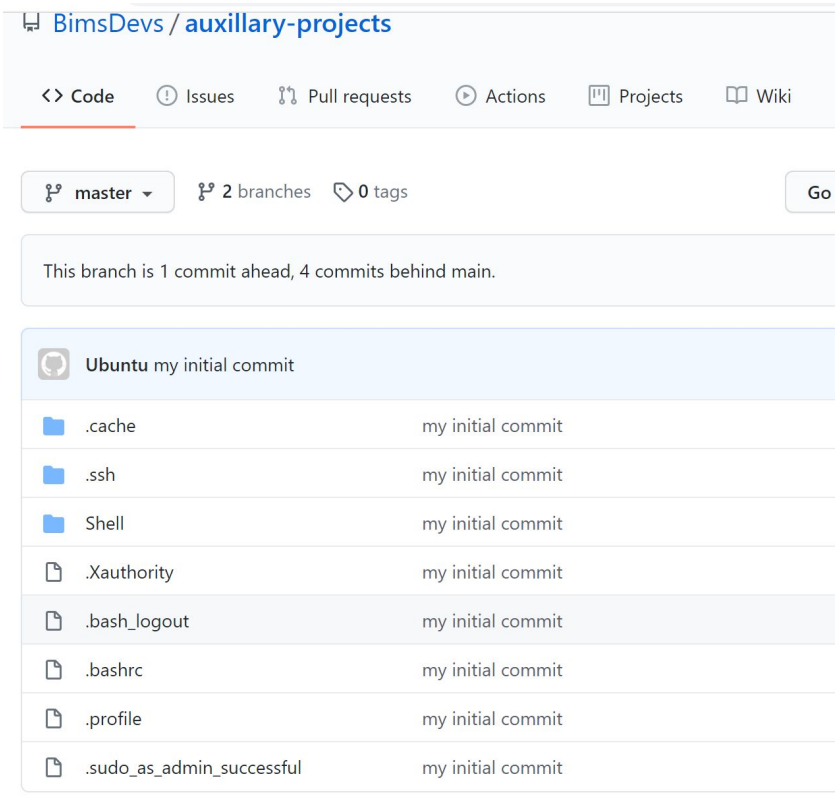
Connect local repository to the remote github
$ git remote add origin git@github.com:BimsDevs/auxillary-projects.git
Check if connected: git remote show

```
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$ git remote add origin git@github.com:BimsD
evs/auxillary-projects.git
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$
ubuntu@ip-172-31-19-243:~$ git remote show
origin
ubuntu@ip-172-31-19-243:~$
```

Push to the remote repository: $ git push -u origin master

```
ubuntu@ip-172-31-19-243:~$ git push -u origin master
The authenticity of host 'github.com (140.82.121.4)' can't be establi
shed.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E
5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
yes
Warning: Permanently added 'github.com,140.82.121.4' (RSA) to the lis
t of known hosts.
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 3.93 KiB | 1.96 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/BimsDevs/auxillary-projects/pull/new/
master
remote:
To github.com:BimsDevs/auxillary-projects.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ubuntu@ip-172-31-19-243:~$
```

Check Github to confirm Push has been done successfully.



Credit:
Shell Script: https://searchdatacenter.techtarget.com/definition/shell-script
Kernel: https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel
https://www.cyberciti.biz/faq/unix-linux-bash-read-comma-separated-cvsfile/

http://www.beginninglinux.com/home/server-administration/openssh-keys-certificates-authentication-pem-pub-crt

https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys

Youtube Channels:
https://www.youtube.com/watch?v=ju9loeXNVW0

https://www.youtube.com/watch?v=CKBv9sn0Dz4

https://www.youtube.com/watch?v=SWYqp7iY_Tc&feature=youtu.be