Computer Systems
Infrastructure and Management
600099
Virtual Machine Configuration Report

Brian Davis

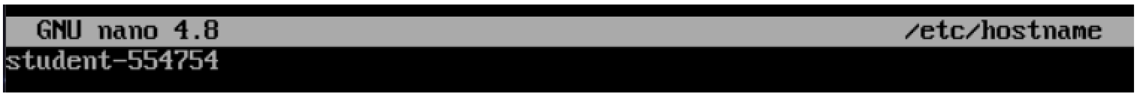
Student ID: 201707824

Contents

Changing the Hostname

The first step towards preparation of the Virtual Machine system (VM) is the editing of the default hostname. This needs to be changed before an internet connection can be established.

Modification of the hostname is possible by using the command ***'vim /etc/hostname'***, as at this point nano is not available yet. Once this is done, the change is confirmed by saving the file by typing ***':wq'*** which writes the file and then quits. A reboot of the VM through the ***'reboot now'*** command refreshes the VM with the correct hostname. Figure 1 shows the hostname within the /etc/hostname directory.

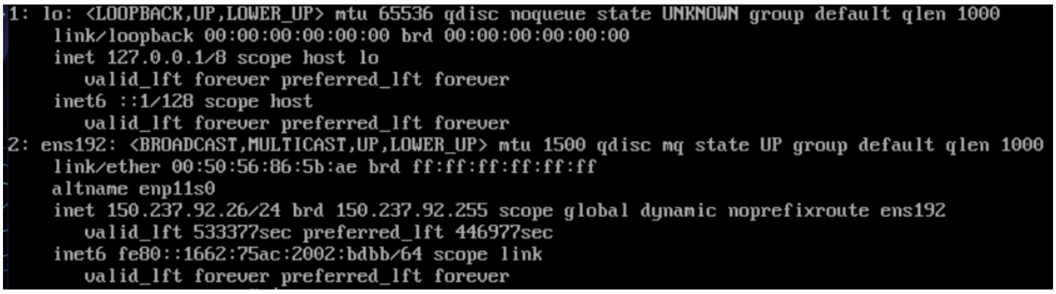


```
  GNU nano 4.8                                        /etc/hostname
student-554754
```

Figure 1 – changing hostname

Establishing an Internet Connection

An internet connection needs to be established before it is possible to install any packages. The DHCP Client Daemon allows for the ability to handle internet connections for the server. Writing the command ***'systemctl enable dhcpcd'*** then ***'systemctl start dhcpcd'*** will enable and then start the DHCP server which will connect us and get an IP address from within the network. Entering the command here ***'ip addr'*** allows the ability to see the IP address that has been assigned. This can be tested by pinging an external network to check to make sure a response is returned, confirming internet connection is available and running.



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:86:5b:ae brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 150.237.92.26/24 brd 150.237.92.255 scope global dynamic noprefixroute ens192
       valid_lft 533377sec preferred_lft 446977sec
    inet6 fe80::1662:75ac:2002:bdbb/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 2 – checking for internet connection

Installing Nano / Updating Pacman

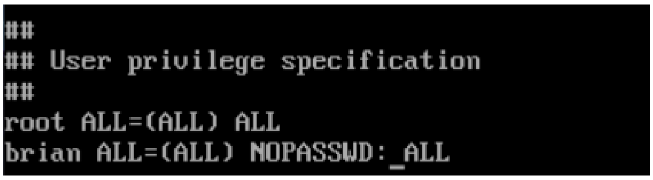
Nano is a text editor for Linux that is not installed by default. It is possible to install this using the package installer called pacman. Pacman is the main package installer for ArchLinux. To install nano onto the system, **'*pacman -Sy man nano'*** is used. This is installed because of the ease of use of nano compared to that of vim.

Now that nano is installed, it is worthwhile making sure the package manager Pacman itself is up to date. The command ***'pacman -Syy'*** is used to force a full update of all the of pacman databases. One of the most important things when setting up a new system is security and having the most up to date packages is one of the first things worth doing, as system vulnerabilities are normally resolved by something as simple as an up to date system. The installation of nano can be found in Appendix section A.

Securing Root

Root is the main account of the system that allows for the ability to customise and essentially do anything you want, but if this account is taken over by someone with malicious intentions it can lead to loss of files and even total system loss *("How To Disable The Root Account On Linux," 2018).*

An administrator account is created using the ***'useradd -m'*** command followed by the name **brian**, to indicate the account name. **-m** defines that this new user will have their own home directory, where they can keep their own files. To give a user admin privileges on Linux, there is need to edit the sudoers file. To elevate a user account, the command ***'sudo'*** must be written before the command you want to execute. At this point the system checks the sudoers file to see if the current user is present and has the necessary privileges, if they don't then the system will present an error and log the interaction.

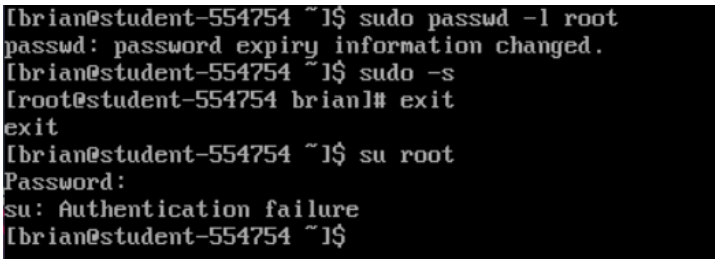


```
##
## User privilege specification
##
root ALL=(ALL) ALL
brian ALL=(ALL) NOPASSWD:_ALL
```

Figure 3 – user privilege specification

This file is editable by typing in the command ***'visudo´*** whilst logged into the root account. This causes errors due to there being no editor found. To correct this, typing in the command ***'export EDITOR=nano'*** makes the nano text editor the default and allows ability to therefore access the sudoers file. Figure 3 shows the user privileges for the user account brian. The use of **all** means that when the brian user tries to sudo an action, the command will execute without asking for a password each time. *(Abdulbasir, 2011)*

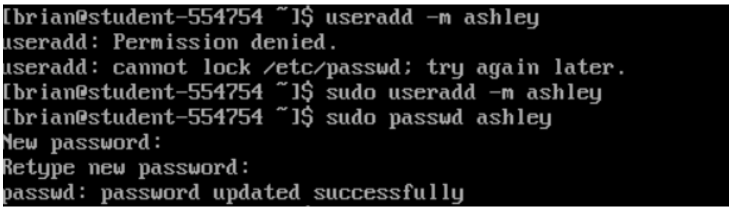
Having admin accounts with their own independent passwords is a lot more secure than having a root account that can be brute force accessed into. To make the system more secure, the root account is locked down. Figure 4 shows how this is done. This locks the root account password authentication, greeting a password entry with the authentication failure error message. Attempts to change the shell of root will still lead to no ability to log in as root.



```
[brian@student-554754 ~]$ sudo passwd -l root
passwd: password expiry information changed.
[brian@student-554754 ~]$ sudo -s
[root@student-554754 brian]# exit
exit
[brian@student-554754 ~]$ su root
Password:
su: Authentication failure
[brian@student-554754 ~]$
```

Figure 4 – unable to access root

The disabling of root removes a direct line into the system, but the addition of admin users with elevated privileges means that it is still possible to configure the system. Once this has been completed, a second admin account is created for the academic member of staff to have access to admin privileges, this is done to allow more than one person the ability to access the features of the root account.

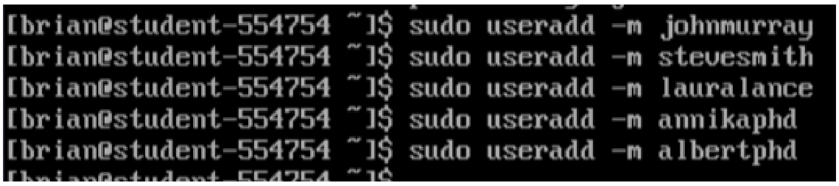


```
[brian@student-554754 ~]$ useradd -m ashley
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
[brian@student-554754 ~]$ sudo useradd -m ashley
[brian@student-554754 ~]$ sudo passwd ashley
New password:
Retype new password:
passwd: password updated successfully
```

Figure 5 – creating Ashley admin user account

Creating User Accounts

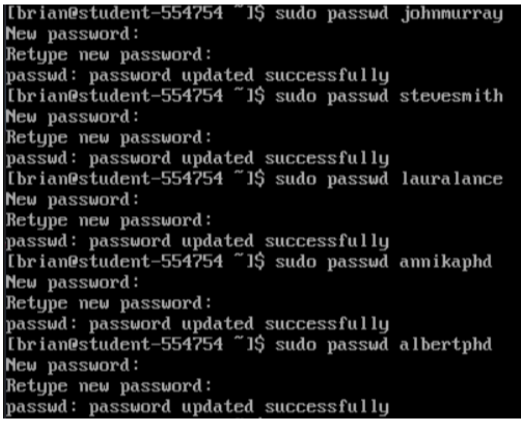
The accounts for the five additional users for the system are created before SSH is configured. To create users, the process is much the same as creating an admin account, except there are no extra permissions granted. They will not receive sudo access.



```
[brian@student-554754 ~]$ sudo useradd -m johnmurray
[brian@student-554754 ~]$ sudo useradd -m stevesmith
[brian@student-554754 ~]$ sudo useradd -m lauralance
[brian@student-554754 ~]$ sudo useradd -m annikaphd
[brian@student-554754 ~]$ sudo useradd -m albertphd
[brian@student-554754 ~]$
```

Figure 6 – creating additional users

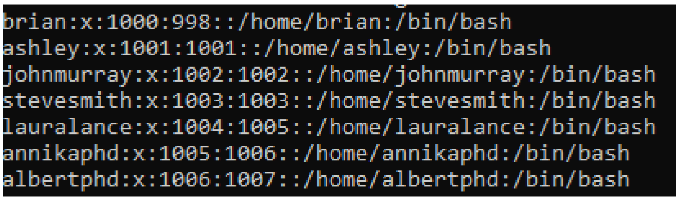
Figure 6 shows the creation of all five additional needed users for the system. **Sudo** needs to be written before each command otherwise there will be a permission denied message. After all users are created, their passwords are created next.



```
[brian@student-554754 ~]$ sudo passwd johnmurray
New password:
Retype new password:
passwd: password updated successfully
[brian@student-554754 ~]$ sudo passwd stevesmith
New password:
Retype new password:
passwd: password updated successfully
[brian@student-554754 ~]$ sudo passwd lauralance
New password:
Retype new password:
passwd: password updated successfully
[brian@student-554754 ~]$ sudo passwd annikaphd
New password:
Retype new password:
passwd: password updated successfully
[brian@student-554754 ~]$ sudo passwd albertphd
New password:
Retype new password:
passwd: password updated successfully
```

Figure 8 – creating user passwords

Once the password for each user is written into the terminal and submitted, the password entered is assigned to the user. When this is done for all 5 users, it is possible to check all users on the system by executing ***'cat /etc/passwd'*** which opens the file to show all users.

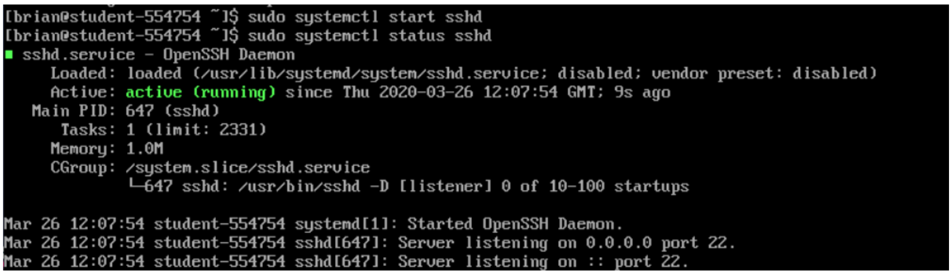


```
brian:x:1000:998::/home/brian:/bin/bash
ashley:x:1001:1001::/home/ashley:/bin/bash
johnmurray:x:1002:1002::/home/johnmurray:/bin/bash
stevesmith:x:1003:1003::/home/stevesmith:/bin/bash
lauralance:x:1004:1005::/home/lauralance:/bin/bash
annikaphd:x:1005:1006::/home/annikaphd:/bin/bash
albertphd:x:1006:1007::/home/albertphd:/bin/bash
```

Figure 7 – visual look at all users present on the VM

Setting up SSH

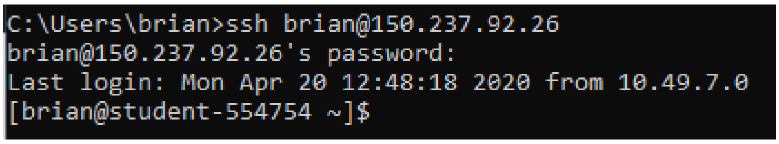
To set up the SSH capabilities of the server, the first thing needed is to install the openssh package through pacman. Without sudo, an error will be flagged due to insufficient privileges. Once this is installed, it needs to then be activated. The service can be started through systemctl, which allows the ability to see that the service is now working.



```
[brian@student-554754 ~]$ sudo systemctl start sshd
[brian@student-554754 ~]$ sudo systemctl status sshd
■ sshd.service - OpenSSH Daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset: disabled)
     Active: active (running) since Thu 2020-03-26 12:07:54 GMT; 9s ago
   Main PID: 647 (sshd)
      Tasks: 1 (limit: 2331)
     Memory: 1.0M
     CGroup: /system.slice/sshd.service
             └─647 sshd: /usr/bin/sshd -D [listener] 0 of 10-100 startups

Mar 26 12:07:54 student-554754 systemd[1]: Started OpenSSH Daemon.
Mar 26 12:07:54 student-554754 sshd[647]: Server listening on 0.0.0.0 port 22.
Mar 26 12:07:54 student-554754 sshd[647]: Server listening on :: port 22.
```

Figure 9 – starting the sshd service

With the server activated and running, the last thing to do is add ***'AllowUsers brian'*** into the sshd_config file found in ***'/etc/ssh/sshd_config'***. With this set up, it is now possible to connect remotely using a user password for the user brian. This can be tested through the command line interface of a windows pc using ***'ssh user@hostname'*** which establishes a connection and asks for the password of the user account. Figure 10 shows this working, ensuring that ssh is set up correctly.



```
C:\Users\brian>ssh brian@150.237.92.26
brian@150.237.92.26's password:
Last login: Mon Apr 20 12:48:18 2020 from 10.49.7.0
[brian@student-554754 ~]$
```

Figure 10 – logging in

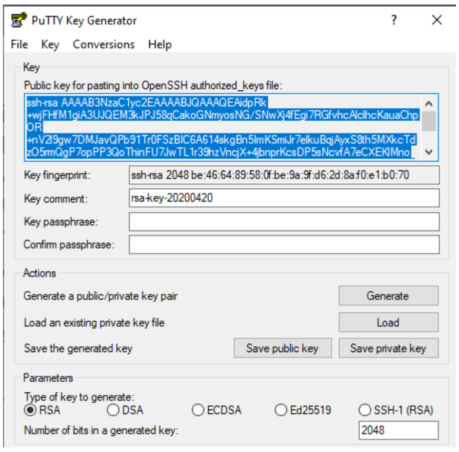
Password access is not very secure, and prone to security issues, so public and private keys will be used instead. These security keys are a way to connect to the system remotely without needing password input, they use the public key found in the user authorized keys file on the system as a different method of authentication *("What is SSH Public Key authentication?," n.d.)*. Whilst the public key can be shared, the private key needs to be safely encrypted and secured. A public key is shareable and is copied to the ssh server, whilst the private key remains with the original user to identify themselves when connecting with the use of a key.

Key Authentication

With the public key set up, the next step is to remove password authentication to increase overall security . There are three things that need to be changed. Firstly, the **PermitRootLogin** is disabled, **PasswordAuthentication** is set to no and **PubkeyAuthentication** is set to yes. The reason the public key and private key are useful in ensuring a secure tunnel for authentication is due to their encryption. Both keys can encrypt and decrypt. If a user wants to send a message to another user, having the public key means that the message sent will be encrypted and can only be decrypted by the paired private key, as only the associated private key can decrypt the data that is encrypted by the public key it is paired to. The benefit of the private key is authenticity, you know where the data came from if it uses a specific private key.

Putty

To create the key for the **brian** user, Putty was used as the solution, due to its ease of use and ability to create public and private keys. Putty has a program in its installation called ***'PuTTYgen'*** which allows the ability to create and then save a public and private key through an interface. With these keys saved, it is then possible to insert the private key into the authorized keys file of the user through Putty SSH. SSH through Putty can paste text into the Linux terminal, which allows for an easy solution to get the generated keys into the authorized_keys file ("How To Configure SSH Keys Authentication With PuTTY And Linux Server In 5 Quick Steps," n.d.).



PuTTY Key Generator

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAidpRk
+wjFHfM1giA3UJQEM3kJPJ58qCakoGNmyosNG/SNwXj4fEgi7RGfvhcAlclhcKauaChp
OR
+nV2l9gw7DMJavQPb91Tr0FSzBIC6A614skgBn5lmKSmiJr7elkuBqjAyxS8th5MXkcTd
zO5mQgP7opPP3QoThinFU7JwTL1r39hzVncjX+4jbnprKcsDP5sNcvfA7eCXEKIMno
```

Key fingerprint: ssh-rsa 2048 be:46:64:89:58:0f:be:9a:9f:d6:2d:8a:f0:e1:b0:70

Key comment: rsa-key-20200420

Key passphrase:

Confirm passphrase:

Type of key to generate: ● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

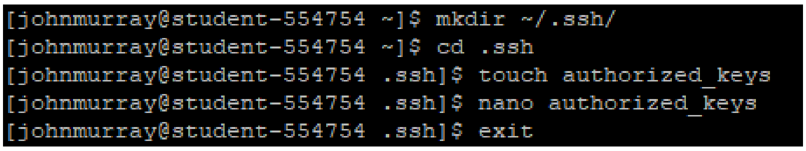
? X
File Key Conversions Help
Key
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters

Figure 11 – putty key generator

Putty also has an SSH authentication agent called ***'pageant'*** that is used to identify the user to establish and unencrypt the public key if the private key matches.

User authorised keys

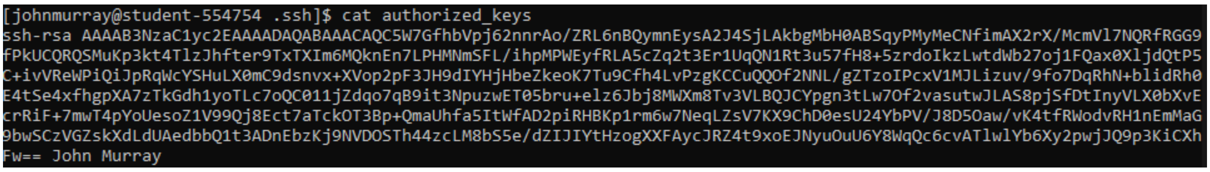
With all users created, each user needs a **.ssh directory** which will hold their **authorized_keys** file where their public key will be. This is so that they can be authenticated for SSH login. This is accomplished through Putty, logging in through SSH via the brian admin user and creating the **.ssh directory** per user through their user account alongside an empty authorized_keys file.



```
[johnmurray@student-554754 ~]$ mkdir ~/.ssh/
[johnmurray@student-554754 ~]$ cd .ssh
[johnmurray@student-554754 .ssh]$ touch authorized_keys
[johnmurray@student-554754 .ssh]$ nano authorized_keys
[johnmurray@student-554754 .ssh]$ exit
```

Figure 12 – creating the .ssh directory for johnmurray

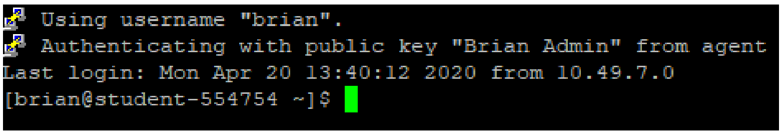
The command of '**Touch authorized_keys'** will create the empty file, **'nano authorized_keys'** will edit the file, which allows the public key for each user to be pasted in through Putty. Using the **"cat"** command allows for the ability to see what is in the file.



```
[johnmurray@student-554754 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC5W7GfhbVpj62nnrAo/ZRL6nBQymnEysA2J4SjLAkbgMbH0ABSqyPMyMeCNfimAX2rX/McmVl7NQRfRGG9
fPkUCQRQSMuKp3kt4TlzJhfter9TxTXIm6MQknEn7LPHMNmSFL/ihpMPWEyfRLA5cZq2t3Er1UqQN1Rt3u57fH8+5zrdoIkzLwtdWb27oj1FQax0XljdQtP5
C+ivVReWPiQiJpRqWcYSHuLX0mC9dsnvx+XVop2pF3JH9dIYHjHbeZkeoK7Tu9Cfh4LvPzgKCCuQQOf2NNL/gZTzoIPcxV1MJLizuv/9fo7DqRhN+blidRh0
E4tSe4xfhgpXA7zTkGdh1yoTLc7oQC011jZdqo7qB9it3NpuzwET05bru+elz6Jbj8MWXm8Tv3VLBQJCYpgn3tLw7Of2vasutwJLAS8pjSfDtInyVLX0bXvE
crRiF+7mwT4pYoUesoZ1V99Qj8Ect7aTckOT3Bp+QmaUhfa5ItWfAD2piRHBKp1rm6w7NeqLZsV7KX9ChD0esU24YbPV/J8D5Oaw/vK4tfRWodvRH1nEmMaG
9bwSCzVGZskXdLdUAedbbQ1t3ADnEbzKj9NVDOSTh44zcLM8bS5e/dZIJIYtHzogXXFAycJRZ4t9xoEJNyuOuU6Y8WqQc6cvATlwlYb6Xy2pwjJQ9p3KiCXh
Fw== John Murray
```

Figure 13 – key added for the johnmurray user

After the service is restarted, it is possible to test if the authentication is in place, by using ***brian@150.237.92.26*** to login as the brian user, which results in a successful login with a message stating the key was used.

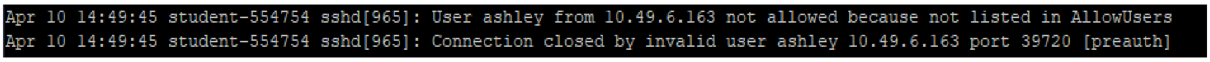


```
 Using username "brian".
 Authenticating with public key "Brian Admin" from agent
Last login: Mon Apr 20 13:40:12 2020 from 10.49.7.0
[brian@student-554754 ~]$
```

Figure 14 – logging in with a public key

Testing Ashley Admin Remote Login

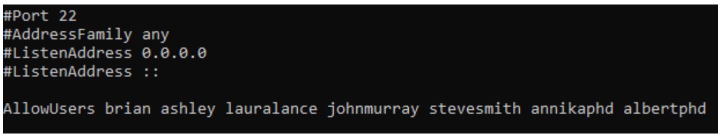
To test the Ashley admin user account through remote log in, there was communication with the intended user of the account. There were initial issues with this, the use of ***journalctl*** was used to discover what the problem was.



```
Apr 10 14:49:45 student-554754 sshd[965]: User ashley from 10.49.6.163 not allowed because not listed in AllowUsers
Apr 10 14:49:45 student-554754 sshd[965]: Connection closed by invalid user ashley 10.49.6.163 port 39720 [preauth]
```

Figure 15 – unable to log in remotely due to permission issues

As shown in figure 15, the issue is that when **AllowUsers** was written into the **sshd_config** file beforehand, the user was only specified as **brian**, this meant that no other user could authenticate, and their access was denied. This was fixed by adding all created users after the **AllowUsers** line in the config file which means all users are now able to authenticate remotely successfully.

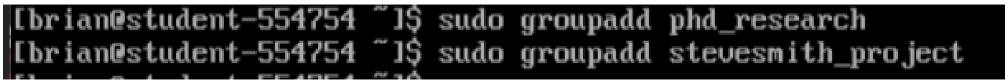


```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

AllowUsers brian ashley lauralance johnmurray stevesmith annikaphd albertphd
```

Figure 16 – AllowUsers showing all users present

Creating and Assigning Groups

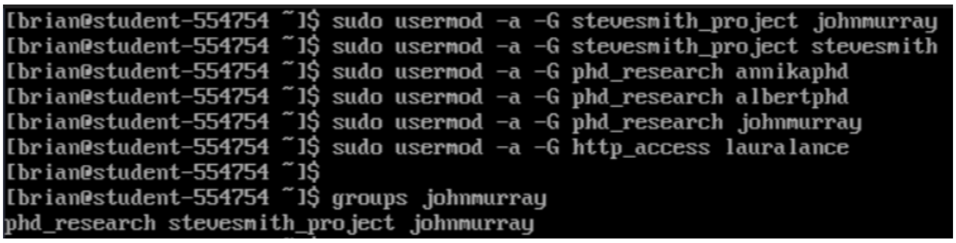
Each user on the system asks for certain requirements be fulfilled for their account. The best way to do this is to create assigned groups that can have permissions available to them, giving each account permissions in the form of group permissions. For the groups, these are split based on user needs. Steve is doing his honours project with John as his supervisor, so a group called stevesmith_project is made for this. For the PhD research group, a group called phd_research is created, and to cover the access to http for Laura, a final group called http_access is created. Groups creation is shown in figure 17.



```
[brian@student-554754 ~]$ sudo groupadd phd_research
[brian@student-554754 ~]$ sudo groupadd stevesmith_project
```

Figure 17 – creating groups

Once these have been created, each user then needs to be assigned to their groups. This is done through the **usermod** command. Figure 18 shows the process of adding each user to their respective groups. The -a means that the user will be added and the -G means that the assignment needs to be in the group specified next. It is possible to add a single user to many groups at once using a single execution, but this was not known at the time. To speed up adding users to groups this could be taken advantage of in the future.

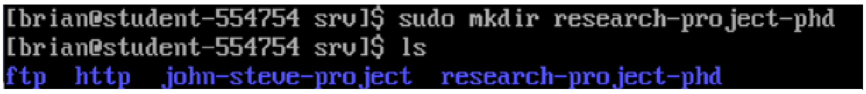


```
[brian@student-554754 ~]$ sudo usermod -a -G stevesmith_project johnmurray
[brian@student-554754 ~]$ sudo usermod -a -G stevesmith_project stevesmith
[brian@student-554754 ~]$ sudo usermod -a -G phd_research annikaphd
[brian@student-554754 ~]$ sudo usermod -a -G phd_research albertphd
[brian@student-554754 ~]$ sudo usermod -a -G phd_research johnmurray
[brian@student-554754 ~]$ sudo usermod -a -G http_access lauralance
[brian@student-554754 ~]$
[brian@student-554754 ~]$ groups johnmurray
phd_research stevesmith_project johnmurray
```

Figure 18 – assigning users to groups

Creating Directories in /srv

A directory acts as a location where files are located on the system, and each user has their own location. In the requirements for johnmurray, he asks that he has a location in the /srv/ directory, to add files to a project directory for Steve. It is also requested that the PhD students have a directory where sensitive data will be kept in the /srv/ location.



```
[brian@student-554754 srv]$ sudo mkdir research-project-phd
[brian@student-554754 srv]$ ls
ftp  http  john-steve-project  research-project-phd
```

Figure 19 – creating the research directory

Figure 19 shows the creation of the **research-project-phd** directory within the /srv location, with a **ls** command used to show all directories now found within the location. The working directory must be **/srv** for these to be created here, accessed through the ***"cd /srv"*** command.

Creating Permissions

Every file and directory in the Linux system has permissions that indicate what each user can do. There are three elements to this. The User, the Group, and the World or All, which indicates the permissions of users who are not the User owner or part of the Group owner. ***'ls-l'*** in the terminal whilst in the /srv/ directory shows all permissions of the directories within.

Permissions are set into four categories; R means read permissions are present, W means write permissions are present, X means execute permissions are present and – means there are no permissions present. These can be edited through two modes, absolute (numeric) or symbolic. Editing through the absolute mode requires knowledge of the octal form through which permissions are granted to each sub-group.

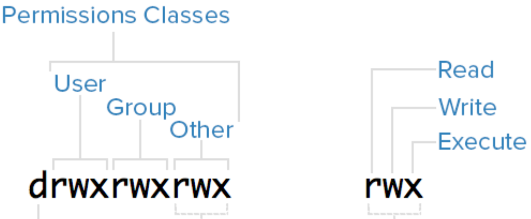
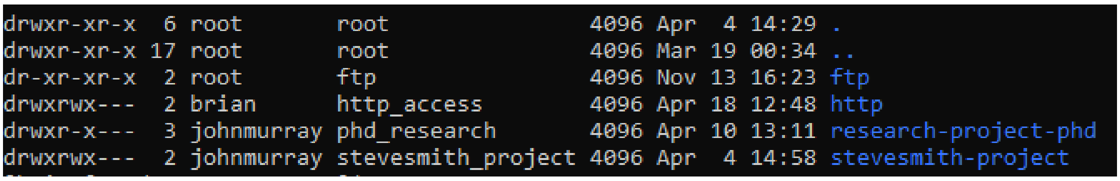


Figure 20 - ("An Introduction to Linux Permissions," n.d.)

As per requirements, johnmurray is set as the user owner for both newly created directories, whilst the groups that match the directory name are assigned as their respective group owner. When it comes to the group permissions for the john-steve-project, it was decided to go with **r-x** but this was later adjusted to **rwx** to allow steve to edit and even add files within his own project folder found on the server. John was given all the permissions for the research-project-phd directory, but the group for this directory does not have write permissions, as the ability to write is stated to only be for John in the requirements, and is set as **r-x**.

The last group is http_access. The permissions for this group are set as **rwx,** there was consideration towards making this **rw-** as Laura is not expected to use a Command Line Interface (CLI) but without the execute permission, it was found to be impossible to add files to the folder. For the Other for each of these directories, the permissions are set to **---**, which means that nobody on the server who is not in the group or the user owner, can interact with these directories. This is a harsh set of permissions, but I believe them to be justified in this case. If a user needs permissions, they can be added to the relevant group.

The directories are now configured, but the name of the directory **john-steve-project** is not suitable so this can be changed. This can be seen in figure 21. **Mv** is the command that indicates that we want to change the object on the left to be changed to the name on right.

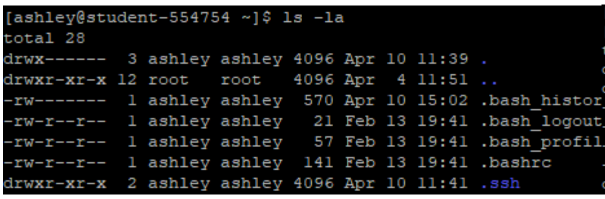


```
drwxr-xr-x  6 root       root                4096 Apr  4 14:29 .
drwxr-xr-x 17 root       root                4096 Mar 19 00:34 ..
dr-xr-xr-x  2 root       ftp                 4096 Nov 13 16:23 ftp
drwxrwx---  2 brian      http_access         4096 Apr 18 12:48 http
drwxr-x---  3 johnmurray phd_research        4096 Apr 10 13:11 research-project-phd
drwxrwx---  2 johnmurray stevesmith_project  4096 Apr  4 14:58 stevesmith-project
```

Figure 21 – permissions for the directories

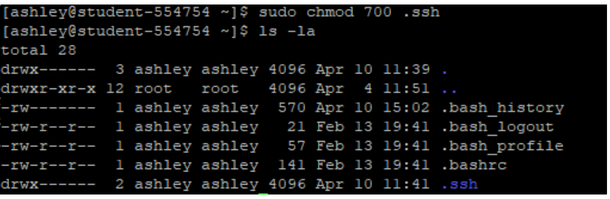
.SSH Folder Permissions

Each user needs their own **.ssh directory** for key authorisation to be possible for their account, but the default permissions given when this directory is created are not suitable.



```
[ashley@student-554754 ~]$ ls -la
total 28
drwx------  3 ashley ashley 4096 Apr 10 11:39 .
drwxr-xr-x 12 root   root   4096 Apr  4 11:51 ..
-rw-------  1 ashley ashley  570 Apr 10 15:02 .bash_histor
-rw-r--r--  1 ashley ashley   21 Feb 13 19:41 .bash_logout
-rw-r--r--  1 ashley ashley   57 Feb 13 19:41 .bash_profil
-rw-r--r--  1 ashley ashley  141 Feb 13 19:41 .bashrc
drwxr-xr-x  2 ashley ashley 4096 Apr 10 11:41 .ssh
```

Figure 22 - .ssh permissions before change



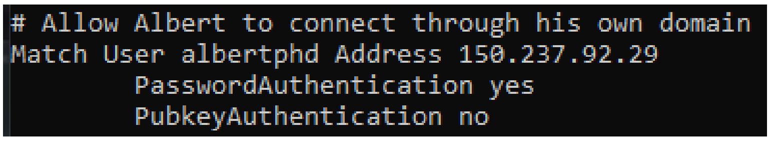
```
[ashley@student-554754 ~]$ sudo chmod 700 .ssh
[ashley@student-554754 ~]$ ls -la
total 28
drwx------  3 ashley ashley 4096 Apr 10 11:39 .
drwxr-xr-x 12 root   root   4096 Apr  4 11:51 ..
-rw-------  1 ashley ashley  570 Apr 10 15:02 .bash_history
-rw-r--r--  1 ashley ashley   21 Feb 13 19:41 .bash_logout
-rw-r--r--  1 ashley ashley   57 Feb 13 19:41 .bash_profile
-rw-r--r--  1 ashley ashley  141 Feb 13 19:41 .bashrc
drwx------  2 ashley ashley 4096 Apr 10 11:41 .ssh
```

Figure 23 - .ssh permissions after change

As shown in figure 22, the default permissions of **r-x** allow a world user the ability to read and execute the .ssh directory for the Ashley user. This means that a world user can read the keys present within this folder. This needs to be changed for every user, so that access to this directory is only possible for the intended user. Figure 23 shows the change made to **---** to remove the permission for a world user to access this folder.

Logging Permissions for Albert

The requirements state the need for Albert to be able to override the key login from a specified IP address. This is done within the sshd_config folder to allow for the ability to override the config settings for authentication. The layout for this is shown in figure 24. To allow authentication via a password, first the **Match User** needs to be specified as **albertphd**, and then the address must be specified to allow for setting override from a specific host. This host address needs to be entered as an IP address, but the requirements only show a domain name. It is possible to ping the hostname to get the IP address of the domain. With the IP address, it is possible to then set the special permissions to allow albert to access the server through his domain via password rather than through key authentication.

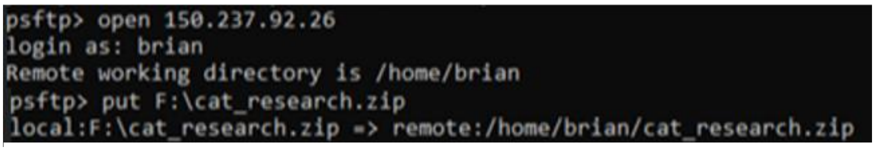


```
# Allow Albert to connect through his own domain
Match User albertphd Address 150.237.92.29
        PasswordAuthentication yes
        PubkeyAuthentication no
```

Figure 24 – logging permissions for albertphd user

SFTP and Jailing a User

To transfer files to the server, the use of Secure File Transfer Protocol is used. Putty is the chosen way of doing this through PSFTP. The way this is done is not the most optimal, but it leads to the successful transfer of files.



```
psftp> open 150.237.92.26
login as: brian
Remote working directory is /home/brian
psftp> put F:\cat_research.zip
local:F:\cat_research.zip => remote:/home/brian/cat_research.zip
```

Figure 25 – using sftp to move the research zip folder

As the user brian does not have write permissions, and the only user who does is John, then once the zipped folder is transferred through PSFTP, this is then moved to the home directory of john, so that the files can be successfully added into the phd research folder. The process of moving files, unzipping them, and then adding them to the phd research folder are found within Appendix section F.

The user Laura is not expected to use a command line interface, so the process of jailing the user can be done to set Laura to access the /srv/ directory by default, and then it's possible to force the Laura user to only be able to use internal-sftp, as this user doesn't need traditional ssh access. To allow Laura to add files to the http folder without the use of the CLI, a local windows application called **WinSCP** is used for this. As this program has an easy to use interface, it allows the ability to simply drag and drop files into the http folder. A screenshot of the interface and the process of jailing the Laura user, with testing to make sure this is implemented correctly, can be found in Appendix section F, which shows successful file transfer of a new text file into the http folder on the server.

Evaluation

The creation of the server can be deemed as successful, with all features added as specified in the user requirements. The process of creating the server was at times, a bit challenging as some elements of a full command line interface with no UI caused issues at times when it came to navigation. Once the navigation features such as ls and cd were memorised, the process of creating the features was a lot easier to realise. Some of the issues with permissions were easy to implement and they went through checks to make sure that the user could do what was needed with the permissions they were given. If anything was not possible for them, then these permissions were reviewed and adjusted to make sure that this was no longer the case.

Overall, I enjoyed the challenge of the servers creation from beginning to end and enjoyed the journey to get there. The hardest part for me was the SFTP solution, which had some small problems with the inability at the time to use the required account to transfer the files to the server, which meant a workaround was found which involved moving files from the admin account to the user that could add the files where they need to go. This process would have been easier if the root account were still available, but I felt it better to find this workaround rather than re-enable the root just for this.

Bibliography

Abdulbasir, I., 2011. Configuring the linux Sudoers file. Linux.com. URL https://www.linux.com/training-tutorials/configuring-linux-sudoers-file/ (accessed 4.20.20).

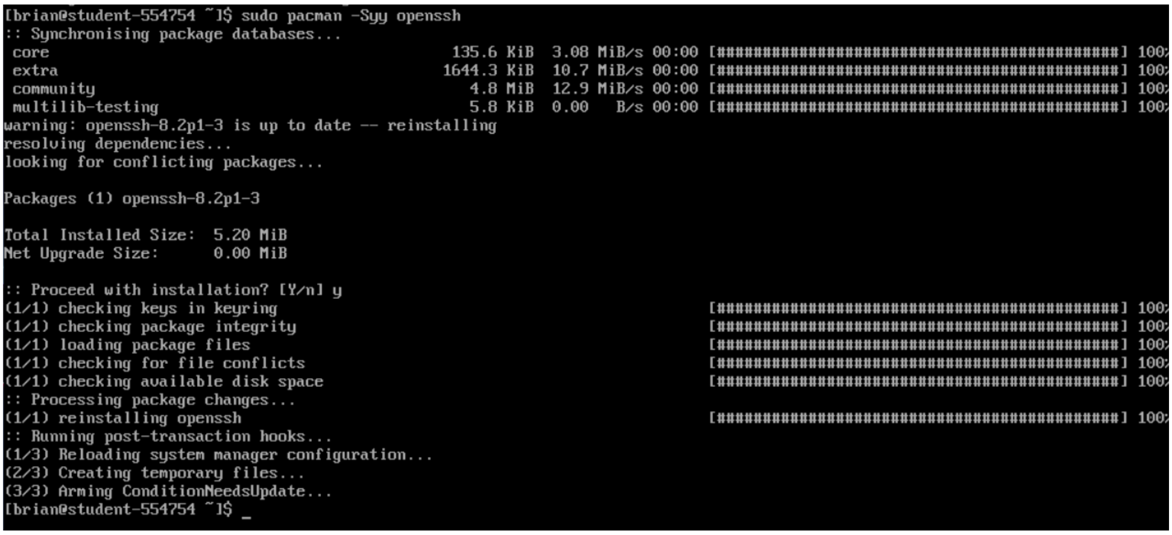
How To Configure SSH Keys Authentication With PuTTY And Linux Server In 5 Quick Steps [WWW Document], n.d. . HowtoForge. URL https://www.howtoforge.com/how-to-configure-ssh-keys-authentication-with-putty-and-linux-server-in-5-quick-steps (accessed 4.20.20).

How To Disable The Root Account On Linux, 2018. . AddictiveTips. URL https://www.addictivetips.com/ubuntu-linux-tips/disable-the-root-account-linux/ (accessed 4.20.20).

What is SSH Public Key authentication? [WWW Document], n.d. URL https://www.ssh.com/ssh/public-key-authentication (accessed 4.20.20).

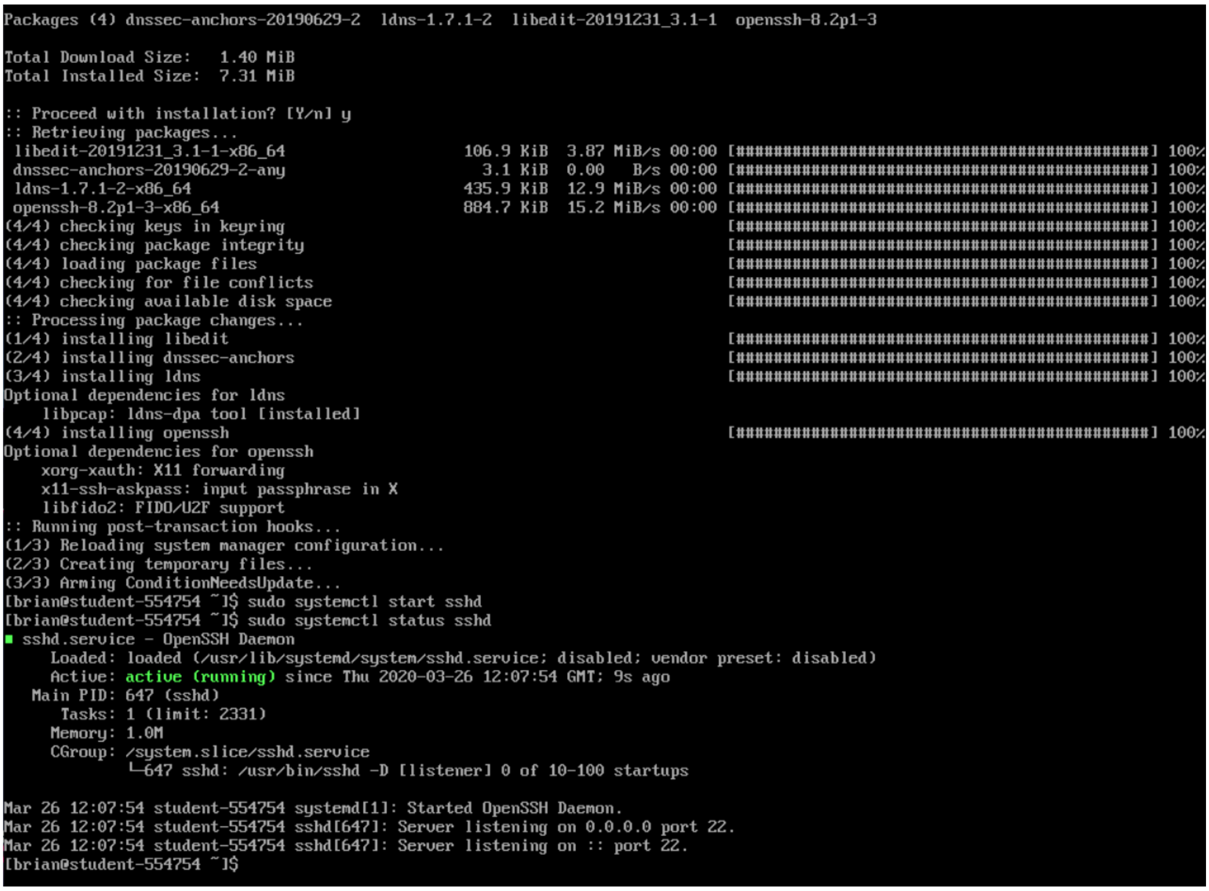
Appendix A – Installing Packages

Installing SSH



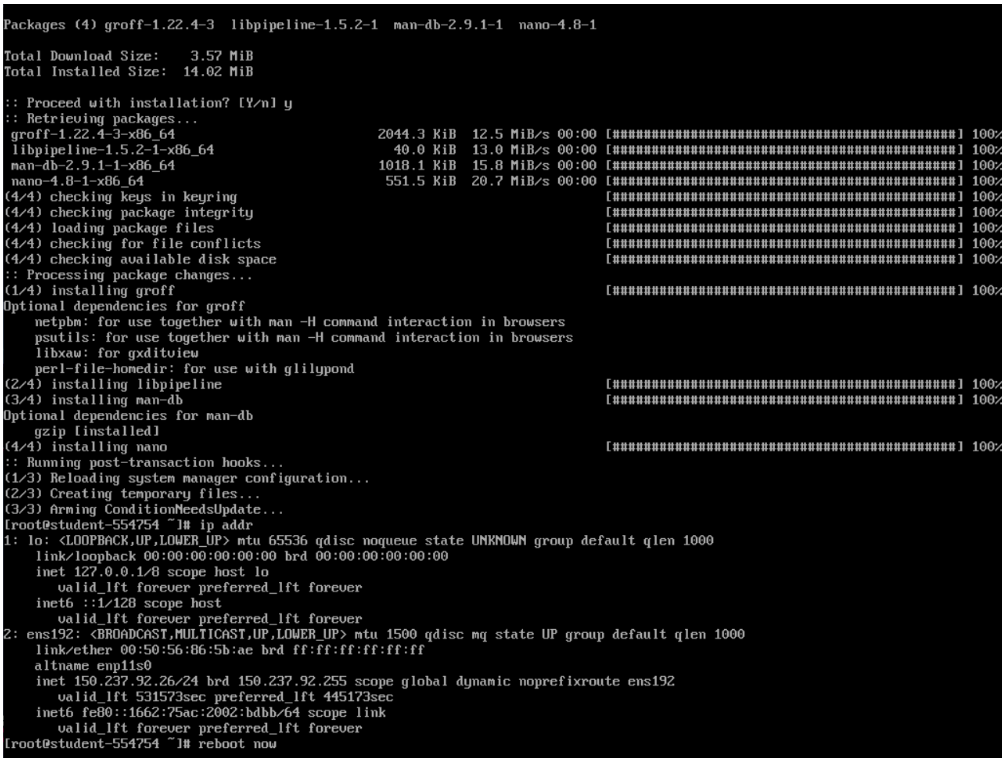
```
[brian@student-554754 ~]$ sudo pacman -Syy openssh
:: Synchronising package databases...
 core                                  135.6 KiB  3.08 MiB/s 00:00 [#############################################] 100%
 extra                                1644.3 KiB  10.7 MiB/s 00:00 [#############################################] 100%
 community                               4.8 MiB  12.9 MiB/s 00:00 [#############################################] 100%
 multilib-testing                        5.8 KiB  0.00   B/s 00:00 [#############################################] 100%
warning: openssh-8.2p1-3 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) openssh-8.2p1-3

Total Installed Size:  5.20 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n] y
(1/1) checking keys in keyring                                     [#############################################] 100%
(1/1) checking package integrity                                   [#############################################] 100%
(1/1) loading package files                                        [#############################################] 100%
(1/1) checking for file conflicts                                  [#############################################] 100%
(1/1) checking available disk space                                [#############################################] 100%
:: Processing package changes...
(1/1) reinstalling openssh                                         [#############################################] 100%
:: Running post-transaction hooks...
(1/3) Reloading system manager configuration...
(2/3) Creating temporary files...
(3/3) Arming ConditionNeedsUpdate...
[brian@student-554754 ~]$ _
```

Installing and running SSHD



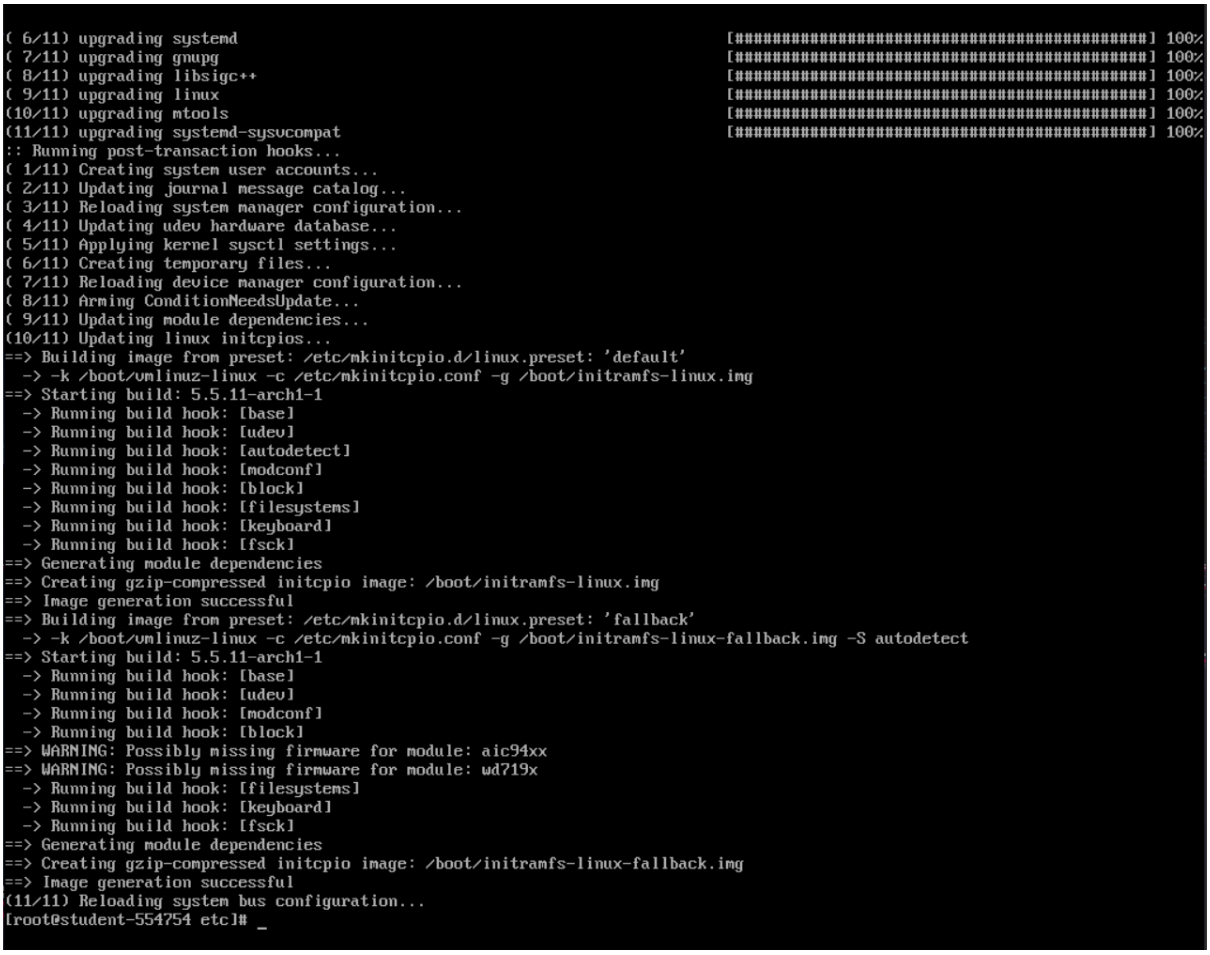
```
Packages (4) dnssec-anchors-20190629-2  ldns-1.7.1-2  libedit-20191231_3.1-1  openssh-8.2p1-3

Total Download Size:   1.40 MiB
Total Installed Size:  7.31 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
 libedit-20191231_3.1-1-x86_64             106.9 KiB  3.87 MiB/s 00:00 [#############################################] 100%
 dnssec-anchors-20190629-2-any               3.1 KiB  0.00   B/s 00:00 [#############################################] 100%
 ldns-1.7.1-2-x86_64                       435.9 KiB  12.9 MiB/s 00:00 [#############################################] 100%
 openssh-8.2p1-3-x86_64                    884.7 KiB  15.2 MiB/s 00:00 [#############################################] 100%
(4/4) checking keys in keyring                                         [#############################################] 100%
(4/4) checking package integrity                                       [#############################################] 100%
(4/4) loading package files                                            [#############################################] 100%
(4/4) checking for file conflicts                                      [#############################################] 100%
(4/4) checking available disk space                                    [#############################################] 100%
:: Processing package changes...
(1/4) installing libedit                                               [#############################################] 100%
(2/4) installing dnssec-anchors                                        [#############################################] 100%
(3/4) installing ldns                                                  [#############################################] 100%
Optional dependencies for ldns
    libpcap: ldns-dpa tool [installed]
(4/4) installing openssh                                               [#############################################] 100%
Optional dependencies for openssh
    xorg-xauth: X11 forwarding
    x11-ssh-askpass: input passphrase in X
    libfido2: FIDO/U2F support
:: Running post-transaction hooks...
(1/3) Reloading system manager configuration...
(2/3) Creating temporary files...
(3/3) Arming ConditionNeedsUpdate...
[brian@student-554754 ~]$ sudo systemctl start sshd
[brian@student-554754 ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH Daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset: disabled)
     Active: active (running) since Thu 2020-03-26 12:07:54 GMT; 9s ago
   Main PID: 647 (sshd)
      Tasks: 1 (limit: 2331)
     Memory: 1.0M
     CGroup: /system.slice/sshd.service
             └─647 sshd: /usr/bin/sshd -D [listener] 0 of 10-100 startups

Mar 26 12:07:54 student-554754 systemd[1]: Started OpenSSH Daemon.
Mar 26 12:07:54 student-554754 sshd[647]: Server listening on 0.0.0.0 port 22.
Mar 26 12:07:54 student-554754 sshd[647]: Server listening on :: port 22.
[brian@student-554754 ~]$
```

Installing Nano



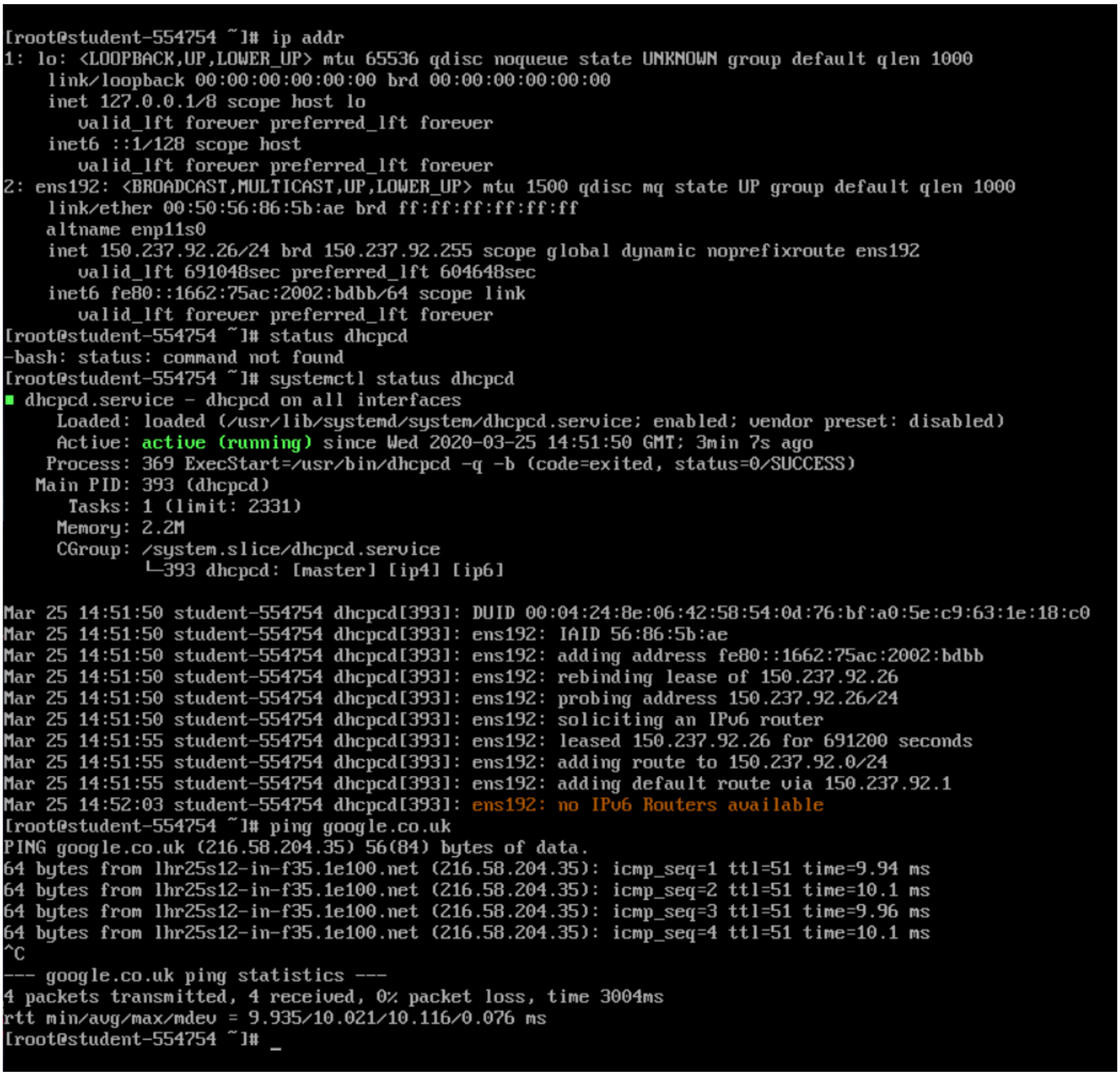
```
Packages (4) groff-1.22.4-3  libpipeline-1.5.2-1  man-db-2.9.1-1  nano-4.8-1

Total Download Size:    3.57 MiB
Total Installed Size:  14.02 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
 groff-1.22.4-3-x86_64                    2044.3 KiB  12.5 MiB/s 00:00 [#############################################] 100%
 libpipeline-1.5.2-1-x86_64                 40.0 KiB  13.0 MiB/s 00:00 [#############################################] 100%
 man-db-2.9.1-1-x86_64                    1018.1 KiB  15.8 MiB/s 00:00 [#############################################] 100%
 nano-4.8-1-x86_64                         551.5 KiB  20.7 MiB/s 00:00 [#############################################] 100%
(4/4) checking keys in keyring                                         [#############################################] 100%
(4/4) checking package integrity                                       [#############################################] 100%
(4/4) loading package files                                            [#############################################] 100%
(4/4) checking for file conflicts                                      [#############################################] 100%
(4/4) checking available disk space                                    [#############################################] 100%
:: Processing package changes...
(1/4) installing groff                                                 [#############################################] 100%
Optional dependencies for groff
    netpbm: for use together with man -H command interaction in browsers
    psutils: for use together with man -H command interaction in browsers
    libxaw: for gxditview
    perl-file-homedir: for use with glilypond
(2/4) installing libpipeline                                           [#############################################] 100%
(3/4) installing man-db                                                [#############################################] 100%
Optional dependencies for man-db
    gzip [installed]
(4/4) installing nano                                                  [#############################################] 100%
:: Running post-transaction hooks...
(1/3) Reloading system manager configuration...
(2/3) Creating temporary files...
(3/3) Arming ConditionNeedsUpdate...
[root@student-554754 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:86:5b:ae brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 150.237.92.26/24 brd 150.237.92.255 scope global dynamic noprefixroute ens192
       valid_lft 531573sec preferred_lft 445173sec
    inet6 fe80::1662:75ac:2002:bdbb/64 scope link
       valid_lft forever preferred_lft forever
[root@student-554754 ~]# reboot now
```

Updating Pacman



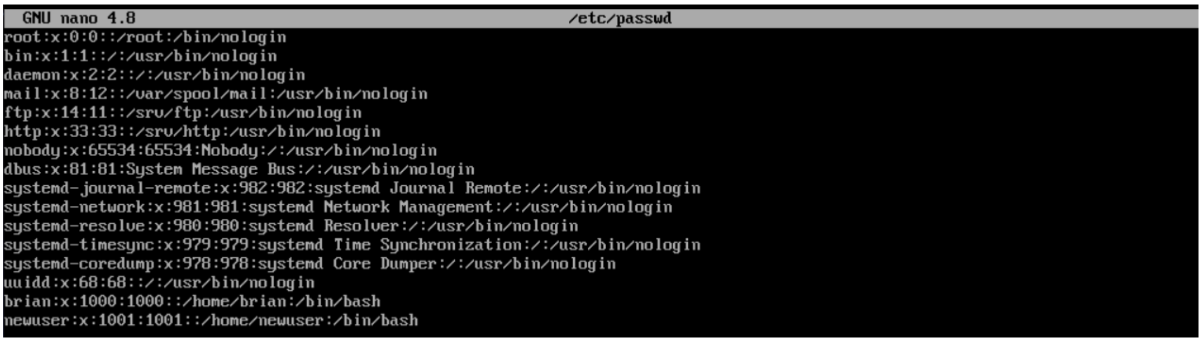
```
( 6/11) upgrading systemd                                                                 [##############################################] 100%
( 7/11) upgrading gnupg                                                                   [##############################################] 100%
( 8/11) upgrading libsigc++                                                               [##############################################] 100%
( 9/11) upgrading linux                                                                   [##############################################] 100%
(10/11) upgrading mtools                                                                  [##############################################] 100%
(11/11) upgrading systemd-sysvcompat                                                      [##############################################] 100%
:: Running post-transaction hooks...
( 1/11) Creating system user accounts...
( 2/11) Updating journal message catalog...
( 3/11) Reloading system manager configuration...
( 4/11) Updating udev hardware database...
( 5/11) Applying kernel sysctl settings...
( 6/11) Creating temporary files...
( 7/11) Reloading device manager configuration...
( 8/11) Arming ConditionNeedsUpdate...
( 9/11) Updating module dependencies...
(10/11) Updating linux initcpios...
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'default'
  -> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 5.5.11-arch1-1
  -> Running build hook: [base]
  -> Running build hook: [udev]
  -> Running build hook: [autodetect]
  -> Running build hook: [modconf]
  -> Running build hook: [block]
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
  -> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 5.5.11-arch1-1
  -> Running build hook: [base]
  -> Running build hook: [udev]
  -> Running build hook: [modconf]
  -> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: wd719x
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
(11/11) Reloading system bus configuration...
[root@student-554754 etc]# _
```

Appendix B – DHCPCD Internet Connection



```
[root@student-554754 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:86:5b:ae brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 150.237.92.26/24 brd 150.237.92.255 scope global dynamic noprefixroute ens192
       valid_lft 691048sec preferred_lft 604648sec
    inet6 fe80::1662:75ac:2002:bdbb/64 scope link
       valid_lft forever preferred_lft forever
[root@student-554754 ~]# status dhcpcd
-bash: status: command not found
[root@student-554754 ~]# systemctl status dhcpcd
● dhcpcd.service - dhcpcd on all interfaces
     Loaded: loaded (/usr/lib/systemd/system/dhcpcd.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2020-03-25 14:51:50 GMT; 3min 7s ago
    Process: 369 ExecStart=/usr/bin/dhcpcd -q -b (code=exited, status=0/SUCCESS)
   Main PID: 393 (dhcpcd)
      Tasks: 1 (limit: 2331)
     Memory: 2.2M
     CGroup: /system.slice/dhcpcd.service
             └─393 dhcpcd: [master] [ip4] [ip6]

Mar 25 14:51:50 student-554754 dhcpcd[393]: DUID 00:04:24:8e:06:42:58:54:0d:76:bf:a0:5e:c9:63:1e:18:c0
Mar 25 14:51:50 student-554754 dhcpcd[393]: ens192: IAID 56:86:5b:ae
Mar 25 14:51:50 student-554754 dhcpcd[393]: ens192: adding address fe80::1662:75ac:2002:bdbb
Mar 25 14:51:50 student-554754 dhcpcd[393]: ens192: rebinding lease of 150.237.92.26
Mar 25 14:51:50 student-554754 dhcpcd[393]: ens192: probing address 150.237.92.26/24
Mar 25 14:51:50 student-554754 dhcpcd[393]: ens192: soliciting an IPv6 router
Mar 25 14:51:55 student-554754 dhcpcd[393]: ens192: leased 150.237.92.26 for 691200 seconds
Mar 25 14:51:55 student-554754 dhcpcd[393]: ens192: adding route to 150.237.92.0/24
Mar 25 14:51:55 student-554754 dhcpcd[393]: ens192: adding default route via 150.237.92.1
Mar 25 14:52:03 student-554754 dhcpcd[393]: ens192: no IPv6 Routers available
[root@student-554754 ~]# ping google.co.uk
PING google.co.uk (216.58.204.35) 56(84) bytes of data.
64 bytes from lhr25s12-in-f35.1e100.net (216.58.204.35): icmp_seq=1 ttl=51 time=9.94 ms
64 bytes from lhr25s12-in-f35.1e100.net (216.58.204.35): icmp_seq=2 ttl=51 time=10.1 ms
64 bytes from lhr25s12-in-f35.1e100.net (216.58.204.35): icmp_seq=3 ttl=51 time=9.96 ms
64 bytes from lhr25s12-in-f35.1e100.net (216.58.204.35): icmp_seq=4 ttl=51 time=10.1 ms
^C
--- google.co.uk ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 9.935/10.021/10.116/0.076 ms
[root@student-554754 ~]# _
```

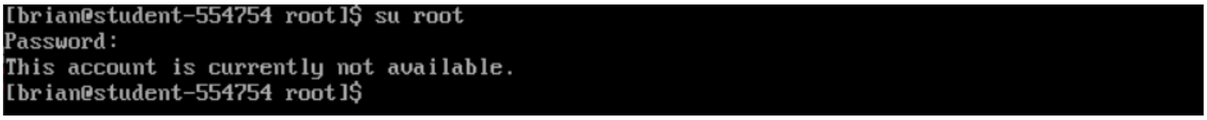
Appendix C – Securing Root



```
  GNU nano 4.8                                              /etc/passwd
root:x:0:0::/root:/bin/nologin
bin:x:1:1::/:/usr/bin/nologin
daemon:x:2:2::/:/usr/bin/nologin
mail:x:8:12::/var/spool/mail:/usr/bin/nologin
ftp:x:14:11::/srv/ftp:/usr/bin/nologin
http:x:33:33::/srv/http:/usr/bin/nologin
nobody:x:65534:65534:Nobody:/:/usr/bin/nologin
dbus:x:81:81:System Message Bus:/:/usr/bin/nologin
systemd-journal-remote:x:982:982:systemd Journal Remote:/:/usr/bin/nologin
systemd-network:x:981:981:systemd Network Management:/:/usr/bin/nologin
systemd-resolve:x:980:980:systemd Resolver:/:/usr/bin/nologin
systemd-timesync:x:979:979:systemd Time Synchronization:/:/usr/bin/nologin
systemd-coredump:x:978:978:systemd Core Dumper:/:/usr/bin/nologin
uuidd:x:68:68::/:/usr/bin/nologin
brian:x:1000:1000::/home/brian:/bin/bash
newuser:x:1001:1001::/home/newuser:/bin/bash
```

No Root Login

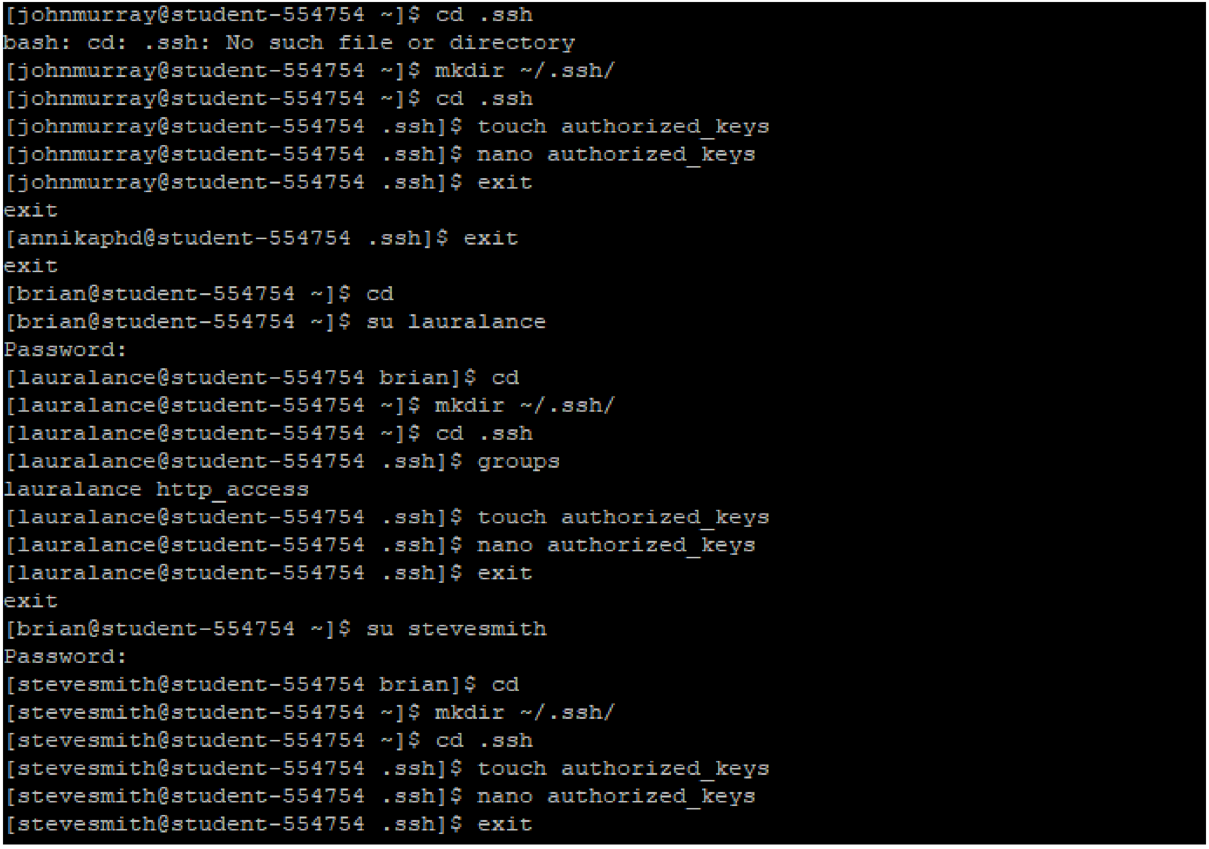
Root Unavailable



```
[brian@student-554754 root]$ su root
Password:
This account is currently not available.
[brian@student-554754 root]$
```

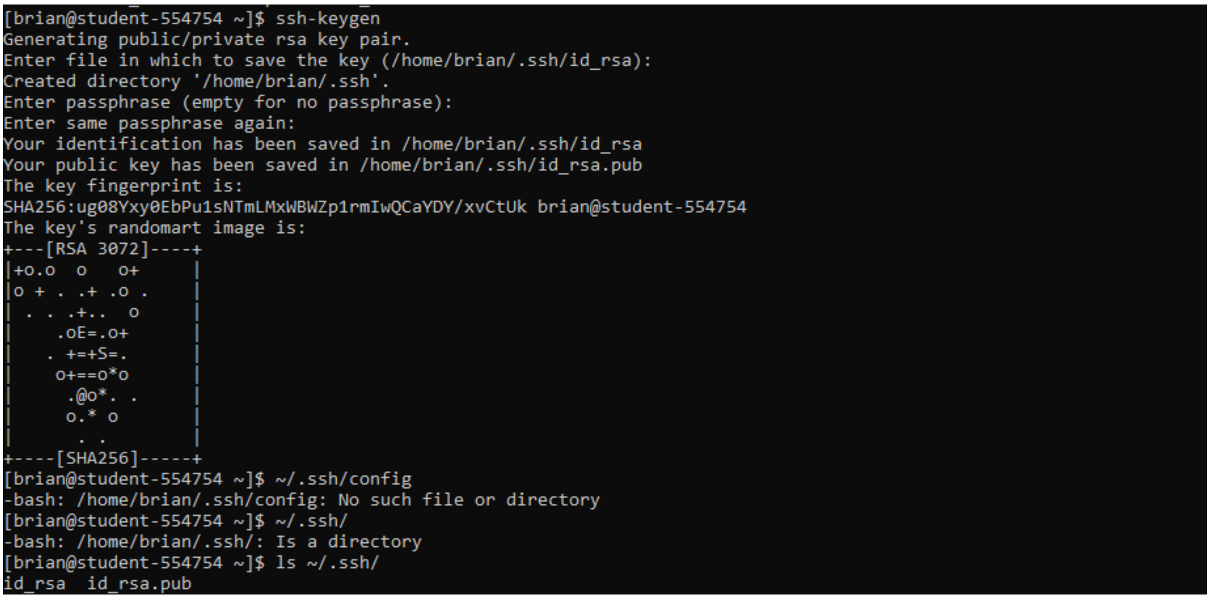
Appendix D – User Accounts

Adding authorized_keys file for each user



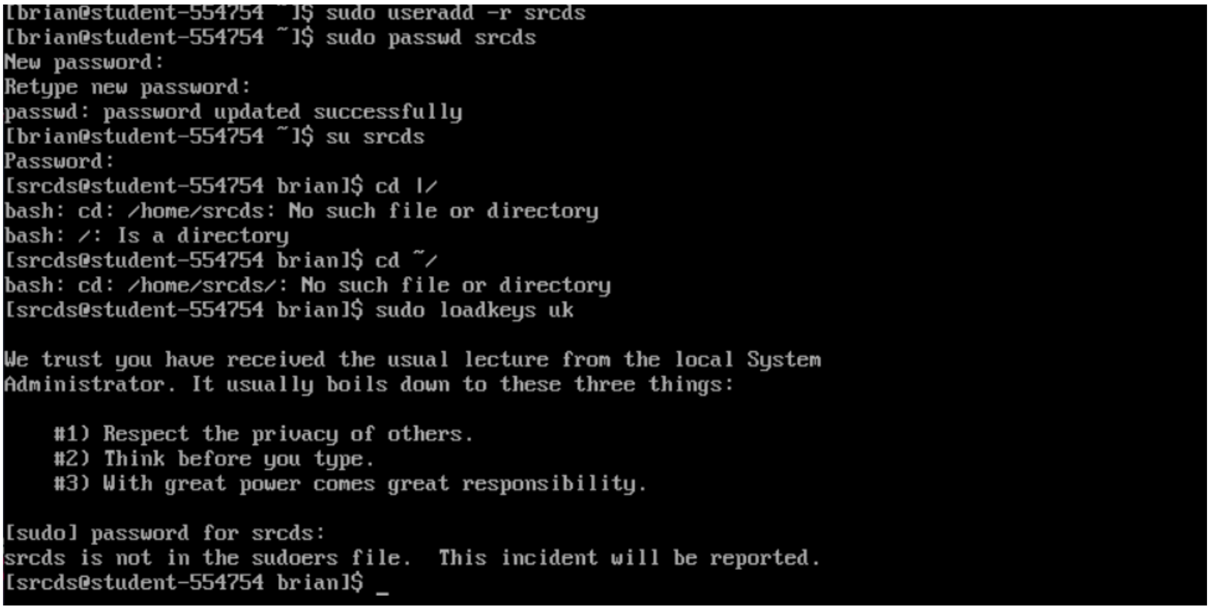
```
[johnmurray@student-554754 ~]$ cd .ssh
bash: cd: .ssh: No such file or directory
[johnmurray@student-554754 ~]$ mkdir ~/.ssh/
[johnmurray@student-554754 ~]$ cd .ssh
[johnmurray@student-554754 .ssh]$ touch authorized_keys
[johnmurray@student-554754 .ssh]$ nano authorized_keys
[johnmurray@student-554754 .ssh]$ exit
exit
[annikaphd@student-554754 .ssh]$ exit
exit
[brian@student-554754 ~]$ cd
[brian@student-554754 ~]$ su lauralance
Password:
[lauralance@student-554754 brian]$ cd
[lauralance@student-554754 ~]$ mkdir ~/.ssh/
[lauralance@student-554754 ~]$ cd .ssh
[lauralance@student-554754 .ssh]$ groups
lauralance http_access
[lauralance@student-554754 .ssh]$ touch authorized_keys
[lauralance@student-554754 .ssh]$ nano authorized_keys
[lauralance@student-554754 .ssh]$ exit
exit
[brian@student-554754 ~]$ su stevesmith
Password:
[stevesmith@student-554754 brian]$ cd
[stevesmith@student-554754 ~]$ mkdir ~/.ssh/
[stevesmith@student-554754 ~]$ cd .ssh
[stevesmith@student-554754 .ssh]$ touch authorized_keys
[stevesmith@student-554754 .ssh]$ nano authorized_keys
[stevesmith@student-554754 .ssh]$ exit
```

Creating SSH Private Key for brian admin account



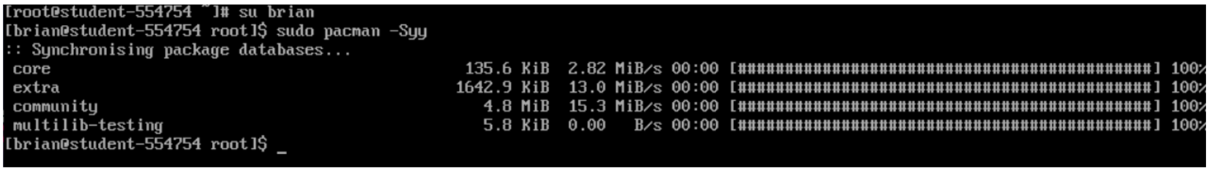
```
[brian@student-554754 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/brian/.ssh/id_rsa):
Created directory '/home/brian/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/brian/.ssh/id_rsa
Your public key has been saved in /home/brian/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ug08Yxy0EbPu1sNTmLMxWBWZp1rmIwQCaYDY/xvCtUk brian@student-554754
The key's randomart image is:
+---[RSA 3072]----+
|+o.o  o   o+     |
|o + . .+ .o .    |
| . . .+..  o     |
|    .oE=.o+      |
|   . +=+S=.      |
|    o+==o*o      |
|     .@o*. .     |
|     o.* o       |
|      . .        |
+----[SHA256]-----+
[brian@student-554754 ~]$ ~/.ssh/config
-bash: /home/brian/.ssh/config: No such file or directory
[brian@student-554754 ~]$ ~/.ssh/
-bash: /home/brian/.ssh/: Is a directory
[brian@student-554754 ~]$ ls ~/.ssh/
id_rsa  id_rsa.pub
```

Creating practice admin system account, with error



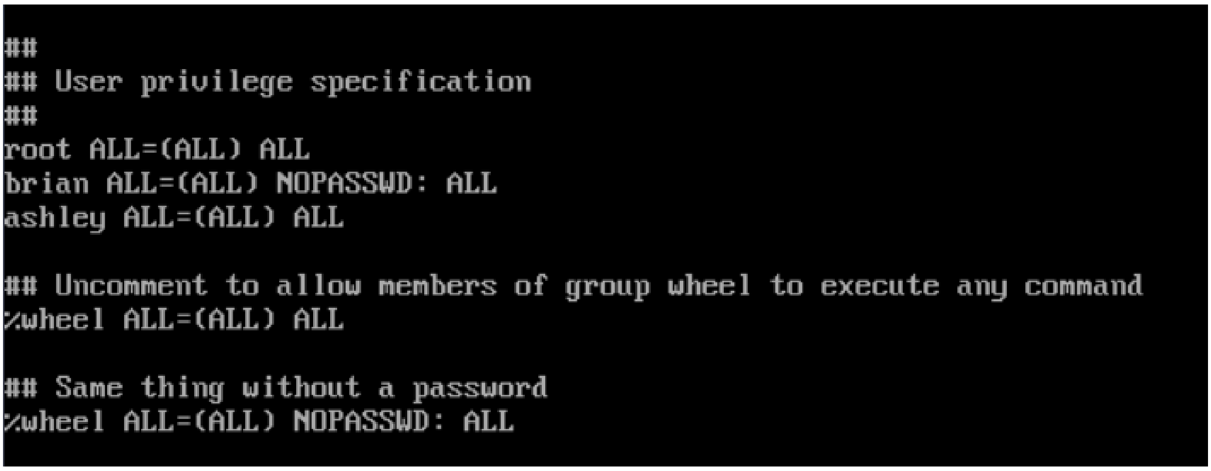
```
[brian@student-554754 ~]$ sudo useradd -r srcds
[brian@student-554754 ~]$ sudo passwd srcds
New password:
Retype new password:
passwd: password updated successfully
[brian@student-554754 ~]$ su srcds
Password:
[srcds@student-554754 brian]$ cd |/
bash: cd: /home/srcds: No such file or directory
bash: /: Is a directory
[srcds@student-554754 brian]$ cd ~/
bash: cd: /home/srcds/: No such file or directory
[srcds@student-554754 brian]$ sudo loadkeys uk

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for srcds:
srcds is not in the sudoers file.  This incident will be reported.
[srcds@student-554754 brian]$ _
```

First instance of Elevated user



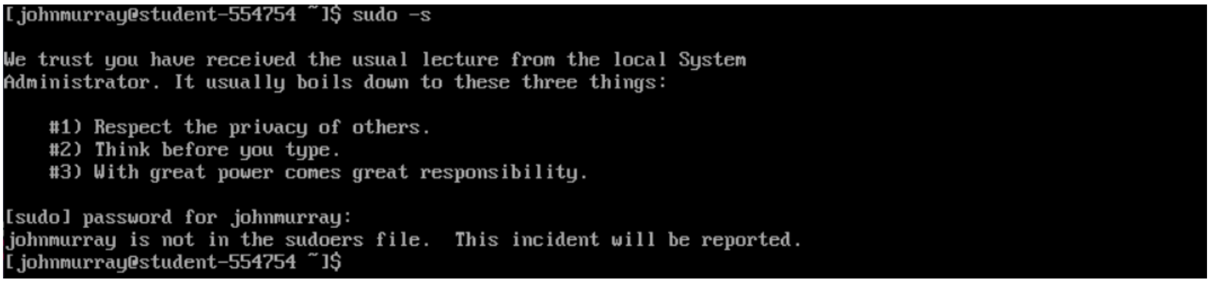
```
[root@student-554754 ~]# su brian
[brian@student-554754 root]$ sudo pacman -Syy
:: Synchronising package databases...
 core                                       135.6 KiB  2.82 MiB/s 00:00 [#############################################] 100%
 extra                                     1642.9 KiB  13.0 MiB/s 00:00 [#############################################] 100%
 community                                    4.8 MiB  15.3 MiB/s 00:00 [#############################################] 100%
 multilib-testing                             5.8 KiB  0.00   B/s 00:00 [#############################################] 100%
[brian@student-554754 root]$ _
```

Admin access right for both admin accounts



```
##
## User privilege specification
##
root ALL=(ALL) ALL
brian ALL=(ALL) NOPASSWD: ALL
ashley ALL=(ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL) ALL

## Same thing without a password
%wheel ALL=(ALL) NOPASSWD: ALL
```

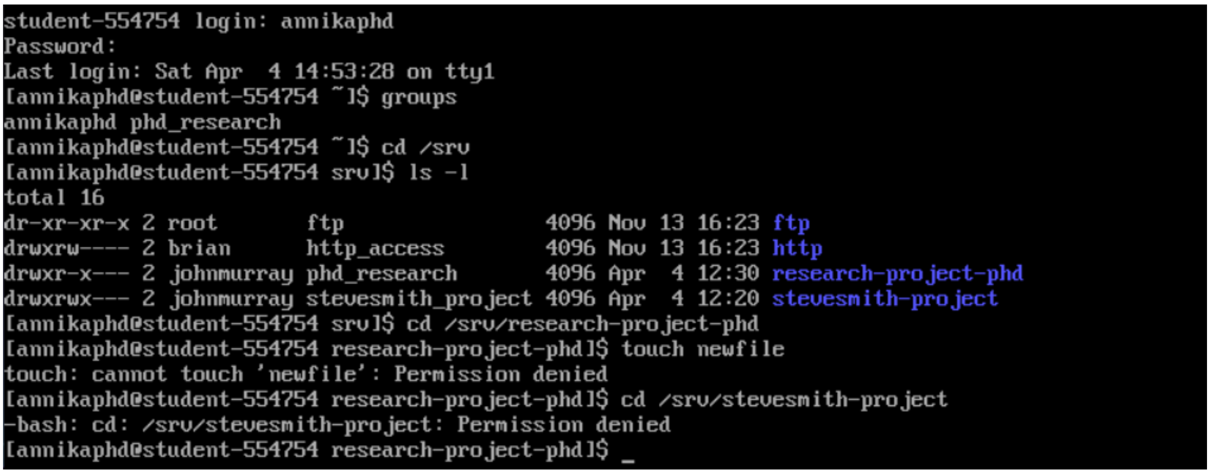
Normal user unable to use sudo, permission check



```
[johnmurray@student-554754 ~]$ sudo -s

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for johnmurray:
johnmurray is not in the sudoers file.  This incident will be reported.
[johnmurray@student-554754 ~]$
```

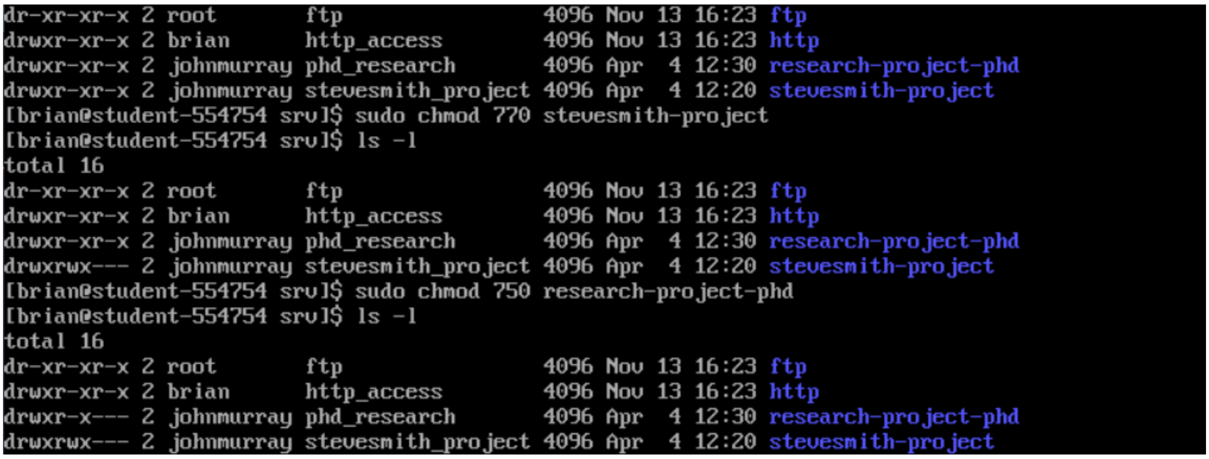
Appendix E - Permissions

Checking the permissions for the Annika user, unable to write to phd research folder

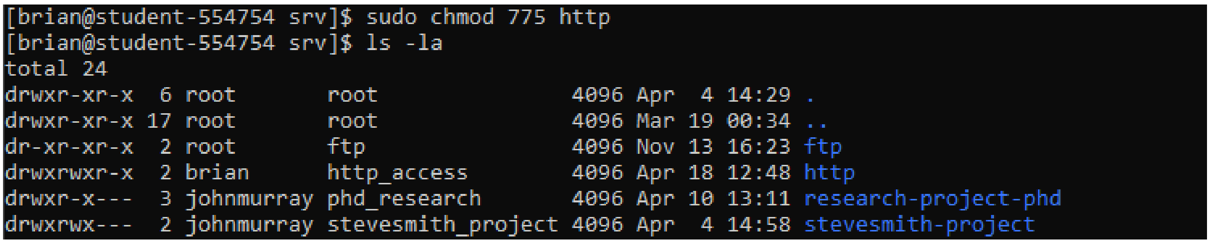


```
student-554754 login: annikaphd
Password:
Last login: Sat Apr  4 14:53:28 on tty1
[annikaphd@student-554754 ~]$ groups
annikaphd phd_research
[annikaphd@student-554754 ~]$ cd /srv
[annikaphd@student-554754 srv]$ ls -l
total 16
dr-xr-xr-x 2 root       ftp                4096 Nov 13 16:23 ftp
drwxrw---- 2 brian      http_access        4096 Nov 13 16:23 http
drwxr-x--- 2 johnmurray phd_research       4096 Apr  4 12:30 research-project-phd
drwxrwx--- 2 johnmurray stevesmith_project 4096 Apr  4 12:20 stevesmith-project
[annikaphd@student-554754 srv]$ cd /srv/research-project-phd
[annikaphd@student-554754 research-project-phd]$ touch newfile
touch: cannot touch 'newfile': Permission denied
[annikaphd@student-554754 research-project-phd]$ cd /srv/stevesmith-project
-bash: cd: /srv/stevesmith-project: Permission denied
[annikaphd@student-554754 research-project-phd]$ _
```

Changing folder permissions using Octal

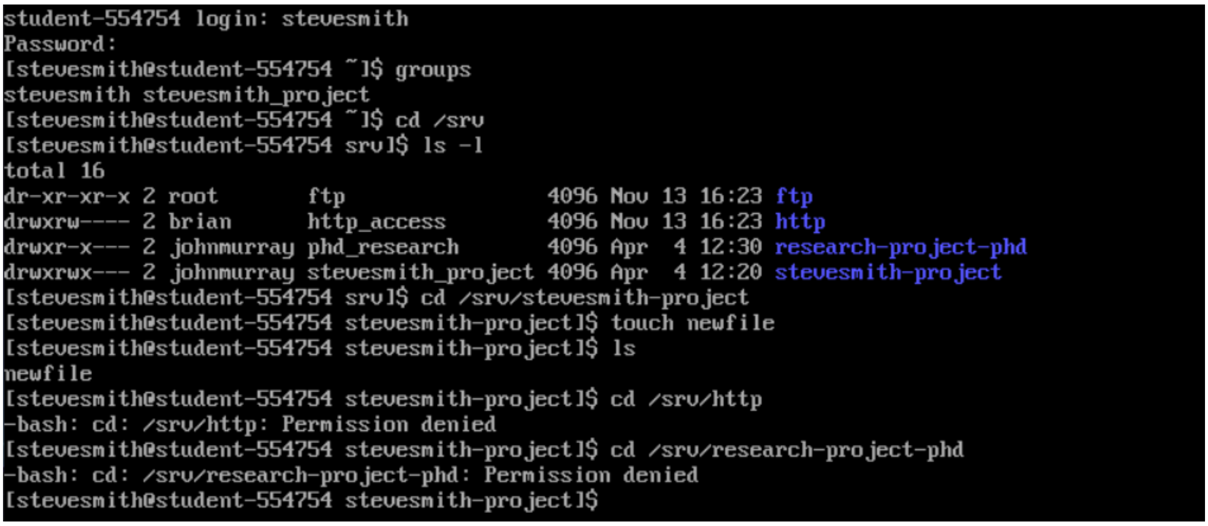


```
dr-xr-xr-x 2 root       ftp                4096 Nov 13 16:23 ftp
drwxr-xr-x 2 brian      http_access        4096 Nov 13 16:23 http
drwxr-xr-x 2 johnmurray phd_research       4096 Apr  4 12:30 research-project-phd
drwxr-xr-x 2 johnmurray stevesmith_project 4096 Apr  4 12:20 stevesmith-project
[brian@student-554754 srv]$ sudo chmod 770 stevesmith-project
[brian@student-554754 srv]$ ls -l
total 16
dr-xr-xr-x 2 root       ftp                4096 Nov 13 16:23 ftp
drwxr-xr-x 2 brian      http_access        4096 Nov 13 16:23 http
drwxr-xr-x 2 johnmurray phd_research       4096 Apr  4 12:30 research-project-phd
drwxrwx--- 2 johnmurray stevesmith_project 4096 Apr  4 12:20 stevesmith-project
[brian@student-554754 srv]$ sudo chmod 750 research-project-phd
[brian@student-554754 srv]$ ls -l
total 16
dr-xr-xr-x 2 root       ftp                4096 Nov 13 16:23 ftp
drwxr-xr-x 2 brian      http_access        4096 Nov 13 16:23 http
drwxr-x--- 2 johnmurray phd_research       4096 Apr  4 12:30 research-project-phd
drwxrwx--- 2 johnmurray stevesmith_project 4096 Apr  4 12:20 stevesmith-project
```



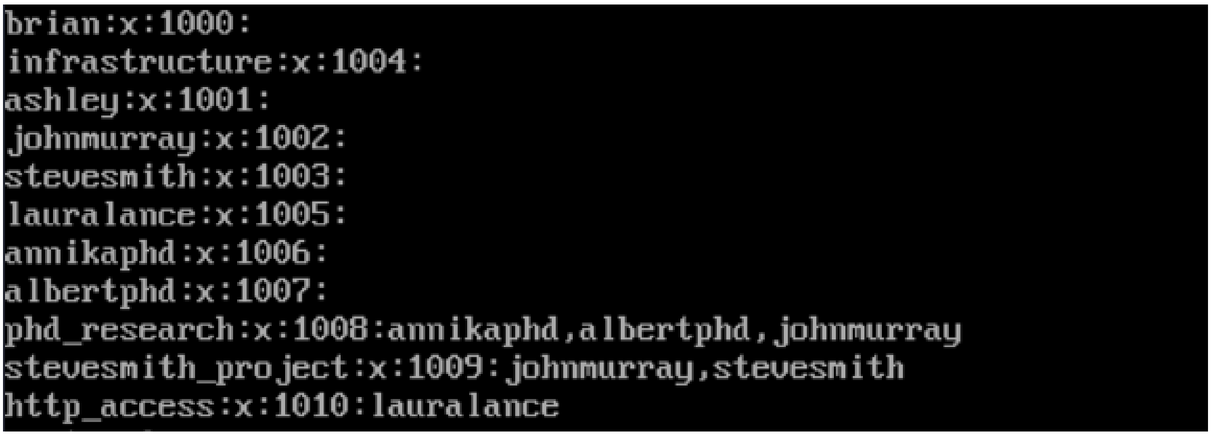
```
[brian@student-554754 srv]$ sudo chmod 775 http
[brian@student-554754 srv]$ ls -la
total 24
drwxr-xr-x  6 root       root               4096 Apr  4 14:29 .
drwxr-xr-x 17 root       root               4096 Mar 19 00:34 ..
dr-xr-xr-x  2 root       ftp                4096 Nov 13 16:23 ftp
drwxrwxr-x  2 brian      http_access        4096 Apr 18 12:48 http
drwxr-x---  3 johnmurray phd_research       4096 Apr 10 13:11 research-project-phd
drwxrwx---  2 johnmurray stevesmith_project 4096 Apr  4 14:58 stevesmith-project
```

Checking permissions for Steve



```
student-554754 login: stevesmith
Password:
[stevesmith@student-554754 ~]$ groups
stevesmith stevesmith_project
[stevesmith@student-554754 ~]$ cd /srv
[stevesmith@student-554754 srv]$ ls -l
total 16
dr-xr-xr-x 2 root       ftp                4096 Nov 13 16:23 ftp
drwxrw---- 2 brian      http_access        4096 Nov 13 16:23 http
drwxr-x--- 2 johnmurray phd_research       4096 Apr  4 12:30 research-project-phd
drwxrwx--- 2 johnmurray stevesmith_project 4096 Apr  4 12:20 stevesmith-project
[stevesmith@student-554754 srv]$ cd /srv/stevesmith-project
[stevesmith@student-554754 stevesmith-project]$ touch newfile
[stevesmith@student-554754 stevesmith-project]$ ls
newfile
[stevesmith@student-554754 stevesmith-project]$ cd /srv/http
-bash: cd: /srv/http: Permission denied
[stevesmith@student-554754 stevesmith-project]$ cd /srv/research-project-phd
-bash: cd: /srv/research-project-phd: Permission denied
[stevesmith@student-554754 stevesmith-project]$
```

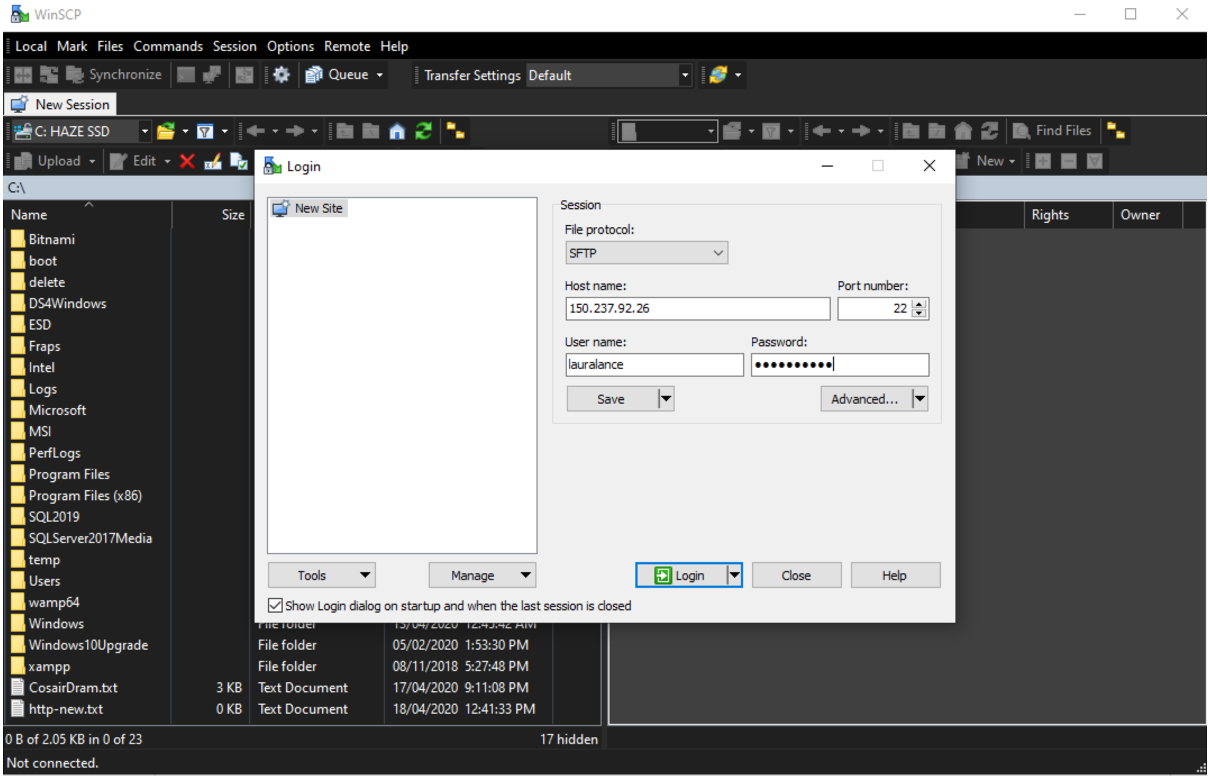
Viewing users in their respective groups



```
brian:x:1000:
infrastructure:x:1004:
ashley:x:1001:
johnmurray:x:1002:
stevesmith:x:1003:
lauralance:x:1005:
annikaphd:x:1006:
albertphd:x:1007:
phd_research:x:1008:annikaphd,albertphd,johnmurray
stevesmith_project:x:1009:johnmurray,stevesmith
http_access:x:1010:lauralance
```

Appendix F - SFTP

Connecting to SFTP through the WinSCP windows local interface



C:\

Session

File protocol: SFTP

Host name: 150.237.92.26

Port number: 22

User name: lauralance

Password: ••••••••••

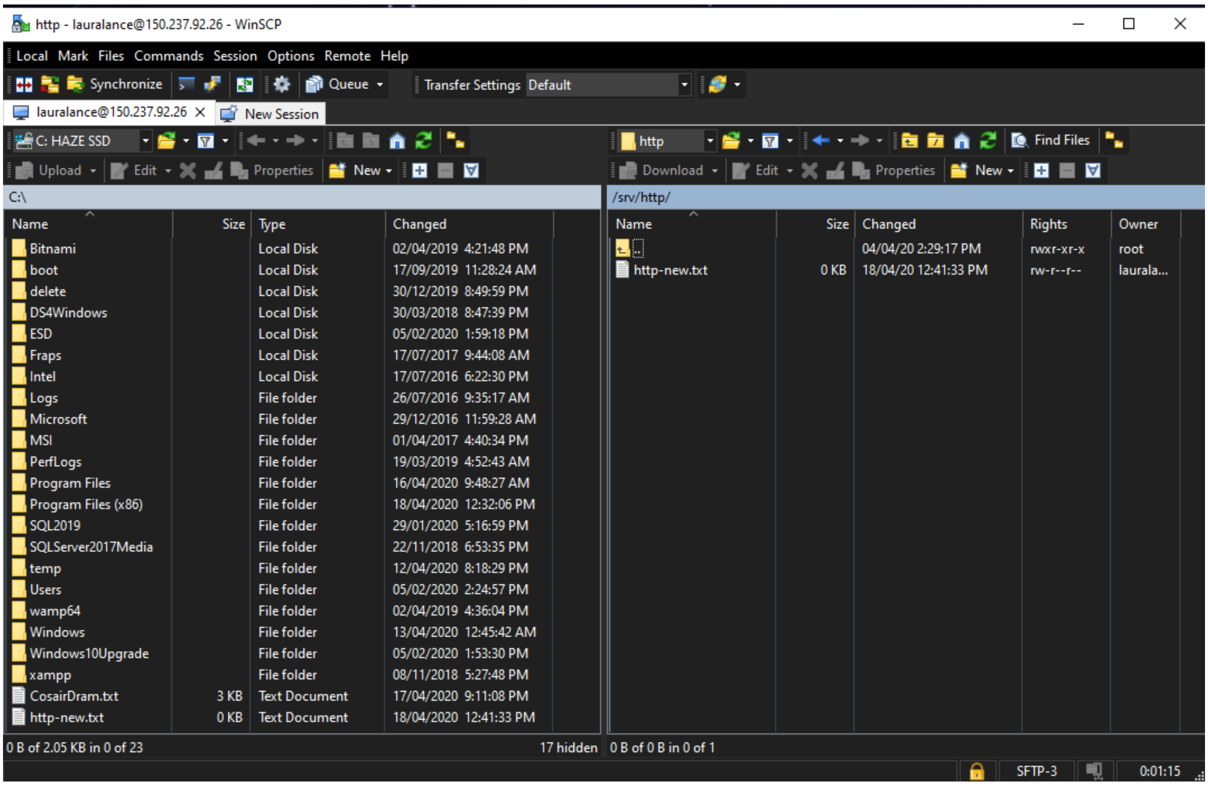
☑ Show Login dialog on startup and when the last session is closed

Name	Size		
Bitnami			
boot			
delete			
DS4Windows			
ESD			
Fraps			
Intel			
Logs			
Microsoft			
MSI			
PerfLogs			
Program Files			
Program Files (x86)			
SQL2019			
SQLServer2017Media			
temp			
Users			
wamp64			
Windows		File folder	13/04/2020 12:45:42 AM
Windows10Upgrade		File folder	05/02/2020 1:53:30 PM
xampp		File folder	08/11/2018 5:27:48 PM
CosairDram.txt	3 KB	Text Document	17/04/2020 9:11:08 PM
http-new.txt	0 KB	Text Document	18/04/2020 12:41:33 PM

0 B of 2.05 KB in 0 of 23 17 hidden

Not connected.

Successfully adding a file to the http folder through WinSCP



C:\

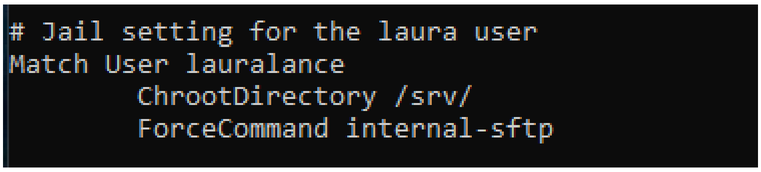
Name	Size	Type	Changed
Bitnami		Local Disk	02/04/2019 4:21:48 PM
boot		Local Disk	17/09/2019 11:28:24 AM
delete		Local Disk	30/12/2019 8:49:59 PM
DS4Windows		Local Disk	30/03/2018 8:47:39 PM
ESD		Local Disk	05/02/2020 1:59:18 PM
Fraps		Local Disk	17/07/2017 9:44:08 AM
Intel		Local Disk	17/07/2016 6:22:30 PM
Logs		File folder	26/07/2016 9:35:17 AM
Microsoft		File folder	29/12/2016 11:59:28 AM
MSI		File folder	01/04/2017 4:40:34 PM
PerfLogs		File folder	19/03/2019 4:52:43 AM
Program Files		File folder	16/04/2020 9:48:27 AM
Program Files (x86)		File folder	18/04/2020 12:32:06 PM
SQL2019		File folder	29/01/2020 5:16:59 PM
SQLServer2017Media		File folder	22/11/2018 6:53:35 PM
temp		File folder	12/04/2020 8:18:29 PM
Users		File folder	05/02/2020 2:24:57 PM
wamp64		File folder	02/04/2019 4:36:04 PM
Windows		File folder	13/04/2020 12:45:42 AM
Windows10Upgrade		File folder	05/02/2020 1:53:30 PM
xampp		File folder	08/11/2018 5:27:48 PM
CosairDram.txt	3 KB	Text Document	17/04/2020 9:11:08 PM
http-new.txt	0 KB	Text Document	18/04/2020 12:41:33 PM

/srv/http/

Name	Size	Changed	Rights	Owner
..		04/04/20 2:29:17 PM	rwxr-xr-x	root
http-new.txt	0 KB	18/04/20 12:41:33 PM	rw-r--r--	laurala...

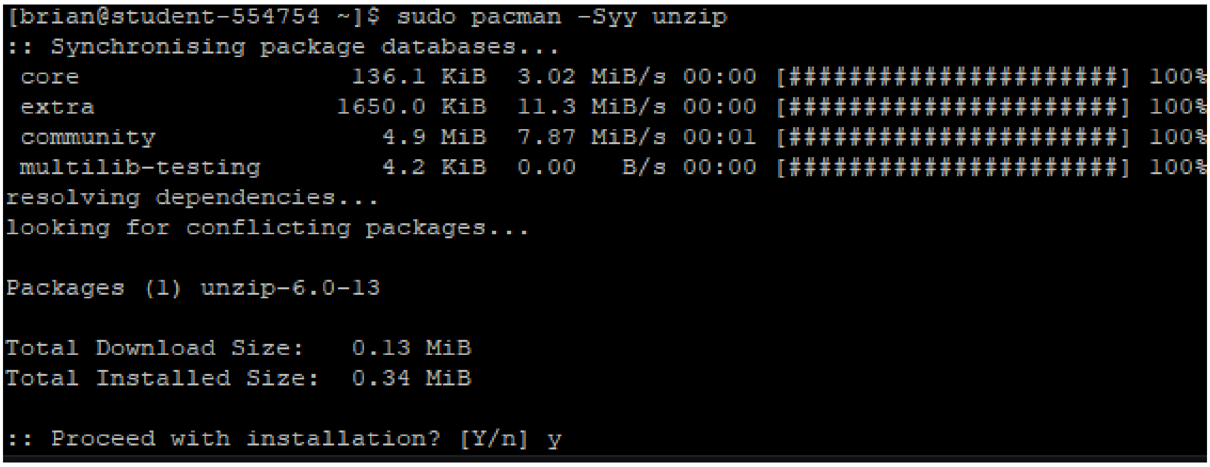
0 B of 2.05 KB in 0 of 23 17 hidden 0 B of 0 B in 0 of 1

Jail settings for the Laura user



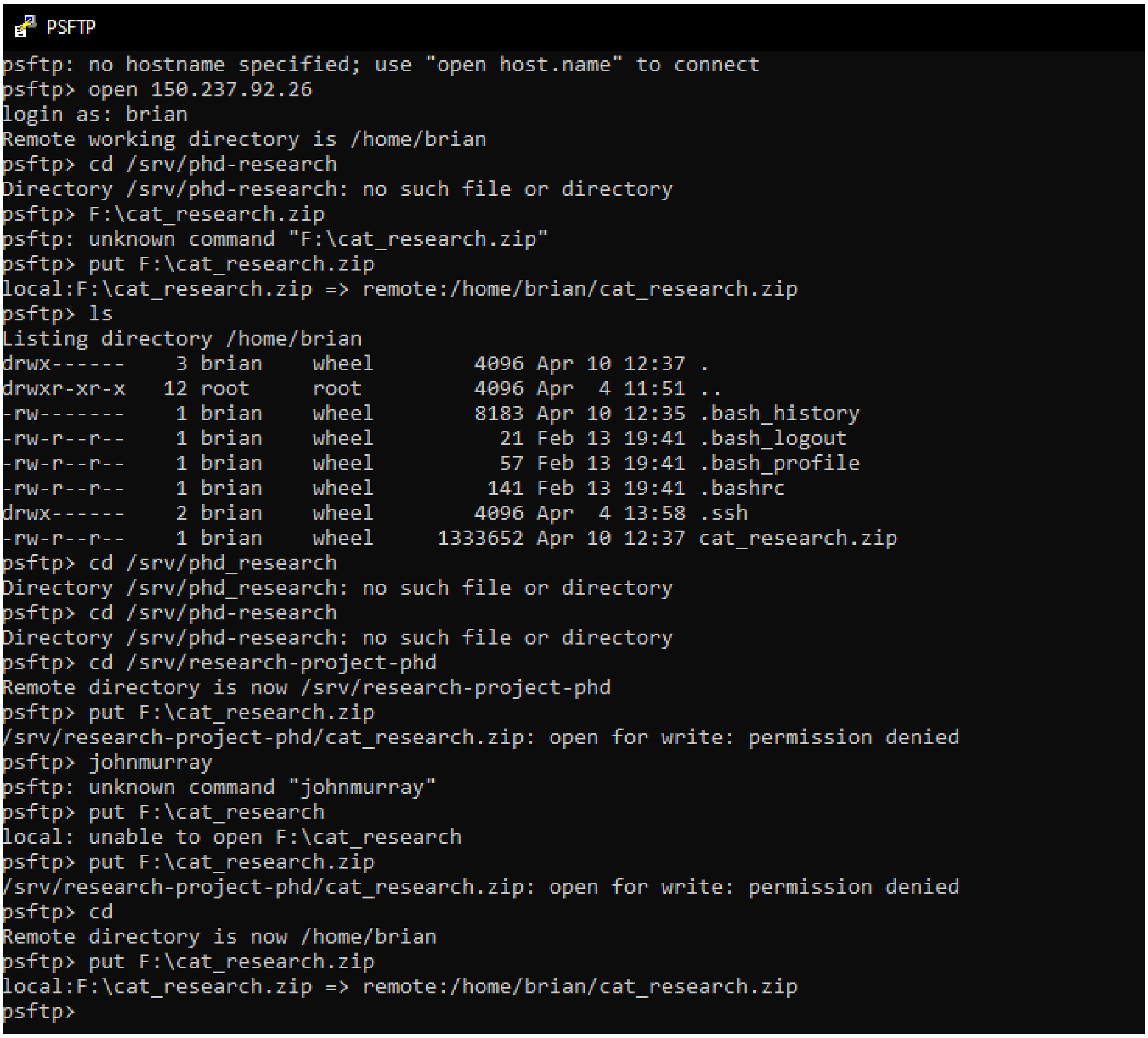
```
# Jail setting for the laura user
Match User lauralance
        ChrootDirectory /srv/
        ForceCommand internal-sftp
```

Remote SFTP and unzipping process



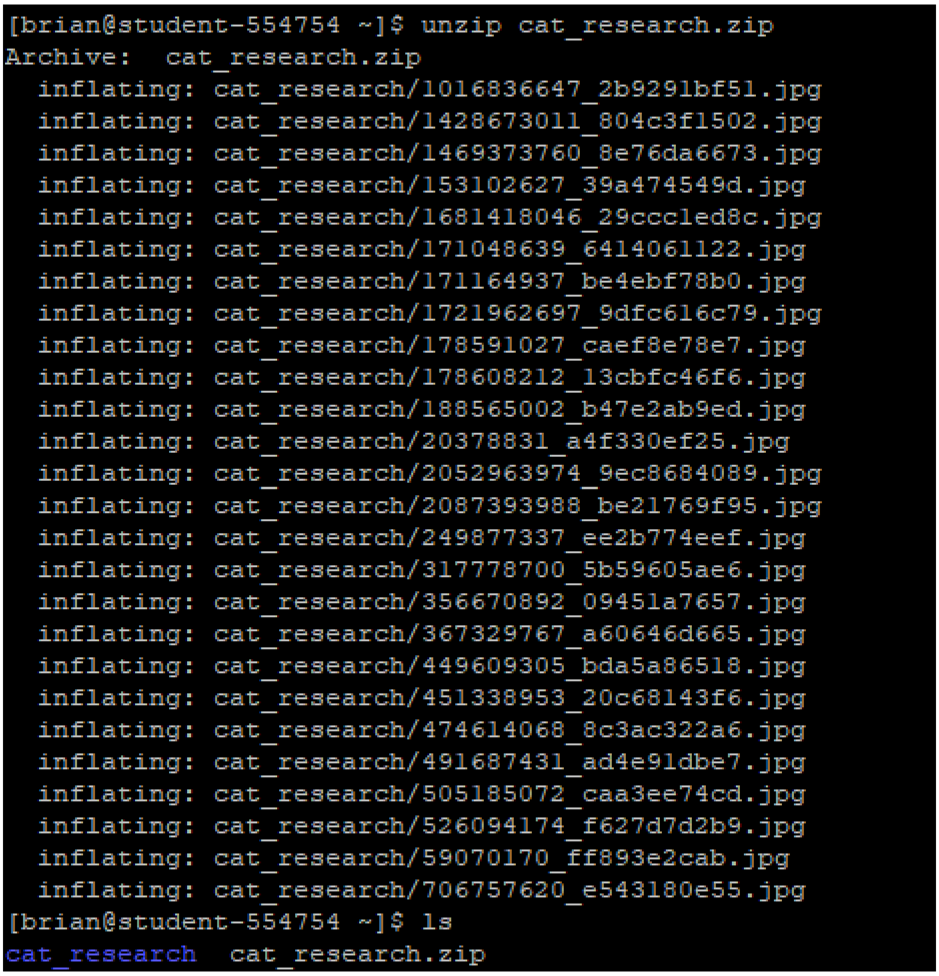
```
[brian@student-554754 ~]$ sudo pacman -Syy unzip
:: Synchronising package databases...
 core                    136.1 KiB  3.02 MiB/s 00:00 [######################] 100%
 extra                  1650.0 KiB  11.3 MiB/s 00:00 [######################] 100%
 community                 4.9 MiB  7.87 MiB/s 00:01 [######################] 100%
 multilib-testing          4.2 KiB  0.00   B/s 00:00 [######################] 100%
resolving dependencies...
looking for conflicting packages...

Packages (1) unzip-6.0-13

Total Download Size:   0.13 MiB
Total Installed Size:  0.34 MiB

:: Proceed with installation? [Y/n] y
```

Installing unzip via pacman



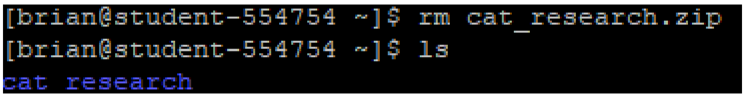
```
psftp: no hostname specified; use "open host.name" to connect
psftp> open 150.237.92.26
login as: brian
Remote working directory is /home/brian
psftp> cd /srv/phd-research
Directory /srv/phd-research: no such file or directory
psftp> F:\cat_research.zip
psftp: unknown command "F:\cat_research.zip"
psftp> put F:\cat_research.zip
local:F:\cat_research.zip => remote:/home/brian/cat_research.zip
psftp> ls
Listing directory /home/brian
drwx------    3 brian    wheel        4096 Apr 10 12:37 .
drwxr-xr-x   12 root     root         4096 Apr  4 11:51 ..
-rw-------    1 brian    wheel        8183 Apr 10 12:35 .bash_history
-rw-r--r--    1 brian    wheel          21 Feb 13 19:41 .bash_logout
-rw-r--r--    1 brian    wheel          57 Feb 13 19:41 .bash_profile
-rw-r--r--    1 brian    wheel         141 Feb 13 19:41 .bashrc
drwx------    2 brian    wheel        4096 Apr  4 13:58 .ssh
-rw-r--r--    1 brian    wheel     1333652 Apr 10 12:37 cat_research.zip
psftp> cd /srv/phd_research
Directory /srv/phd_research: no such file or directory
psftp> cd /srv/phd-research
Directory /srv/phd-research: no such file or directory
psftp> cd /srv/research-project-phd
Remote directory is now /srv/research-project-phd
psftp> put F:\cat_research.zip
/srv/research-project-phd/cat_research.zip: open for write: permission denied
psftp> johnmurray
psftp: unknown command "johnmurray"
psftp> put F:\cat_research
local: unable to open F:\cat_research
psftp> put F:\cat_research.zip
/srv/research-project-phd/cat_research.zip: open for write: permission denied
psftp> cd
Remote directory is now /home/brian
psftp> put F:\cat_research.zip
local:F:\cat_research.zip => remote:/home/brian/cat_research.zip
psftp>
```

using putty sftp to add the files via zip file to the home directory of the brian admin user



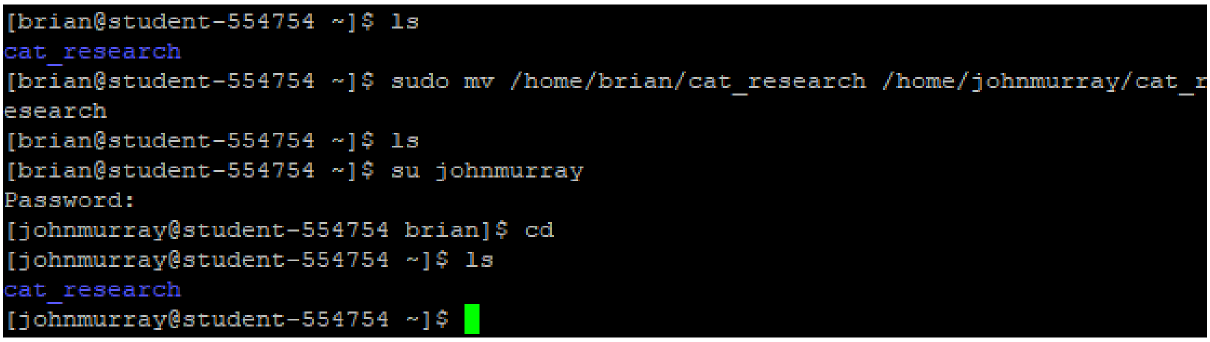
```
[brian@student-554754 ~]$ unzip cat_research.zip
Archive:  cat_research.zip
  inflating: cat_research/1016836647_2b9291bf51.jpg
  inflating: cat_research/1428673011_804c3f1502.jpg
  inflating: cat_research/1469373760_8e76da6673.jpg
  inflating: cat_research/153102627_39a474549d.jpg
  inflating: cat_research/1681418046_29cccled8c.jpg
  inflating: cat_research/171048639_6414061122.jpg
  inflating: cat_research/171164937_be4ebf78b0.jpg
  inflating: cat_research/1721962697_9dfc616c79.jpg
  inflating: cat_research/178591027_caef8e78e7.jpg
  inflating: cat_research/178608212_13cbfc46f6.jpg
  inflating: cat_research/188565002_b47e2ab9ed.jpg
  inflating: cat_research/20378831_a4f330ef25.jpg
  inflating: cat_research/2052963974_9ec8684089.jpg
  inflating: cat_research/2087393988_be21769f95.jpg
  inflating: cat_research/249877337_ee2b774eef.jpg
  inflating: cat_research/317778700_5b59605ae6.jpg
  inflating: cat_research/356670892_09451a7657.jpg
  inflating: cat_research/367329767_a60646d665.jpg
  inflating: cat_research/449609305_bda5a86518.jpg
  inflating: cat_research/451338953_20c68143f6.jpg
  inflating: cat_research/474614068_8c3ac322a6.jpg
  inflating: cat_research/491687431_ad4e91dbe7.jpg
  inflating: cat_research/505185072_caa3ee74cd.jpg
  inflating: cat_research/526094174_f627d7d2b9.jpg
  inflating: cat_research/59070170_ff893e2cab.jpg
  inflating: cat_research/706757620_e543180e55.jpg
[brian@student-554754 ~]$ ls
cat_research  cat_research.zip
```

unzip all the files, which creates a new directory



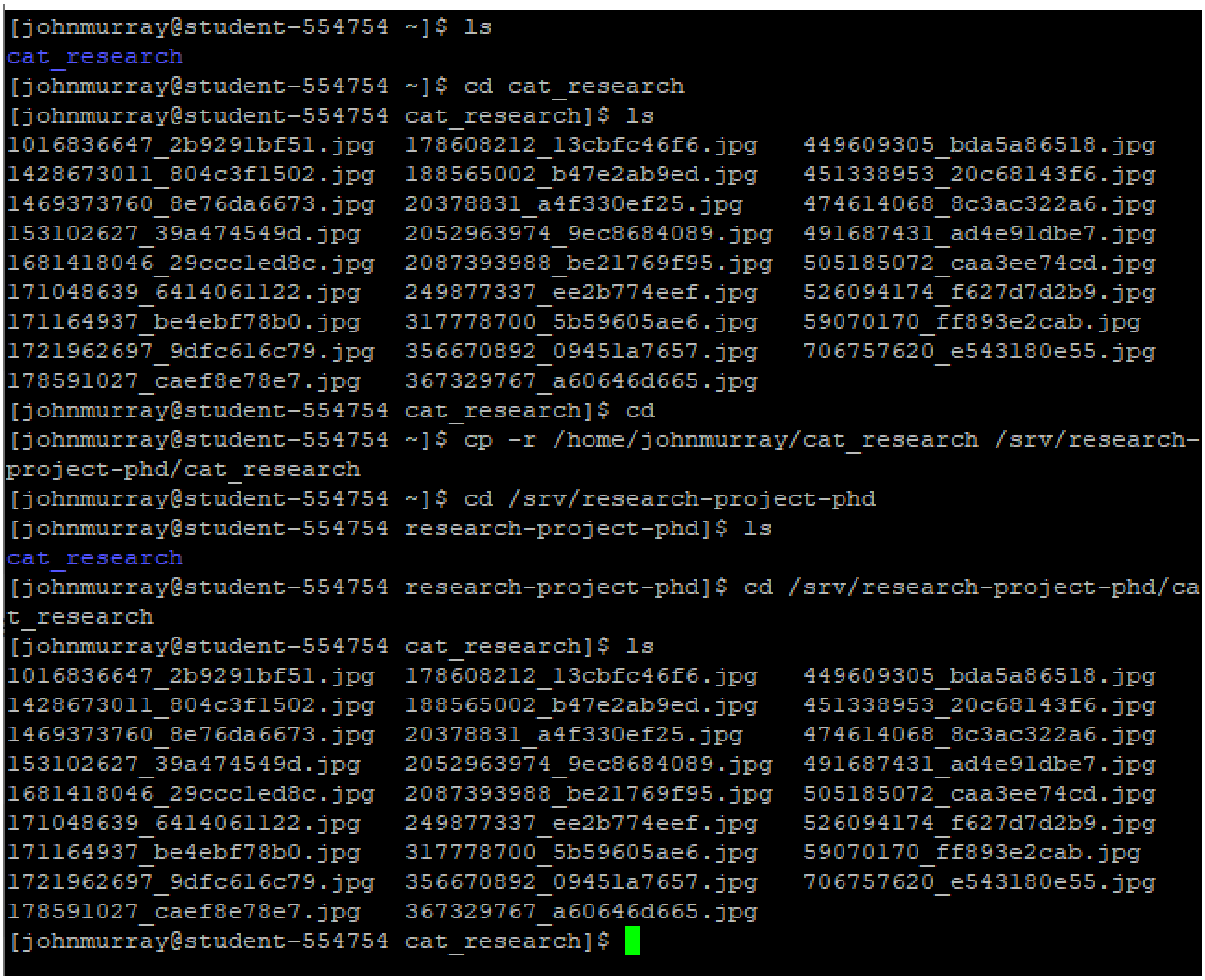
```
[brian@student-554754 ~]$ rm cat_research.zip
[brian@student-554754 ~]$ ls
cat_research
```

remove the zip file as it's no longer needed



```
[brian@student-554754 ~]$ ls
cat_research
[brian@student-554754 ~]$ sudo mv /home/brian/cat_research /home/johnmurray/cat_r
esearch
[brian@student-554754 ~]$ ls
[brian@student-554754 ~]$ su johnmurray
Password:
[johnmurray@student-554754 brian]$ cd
[johnmurray@student-554754 ~]$ ls
cat_research
[johnmurray@student-554754 ~]$ 
```

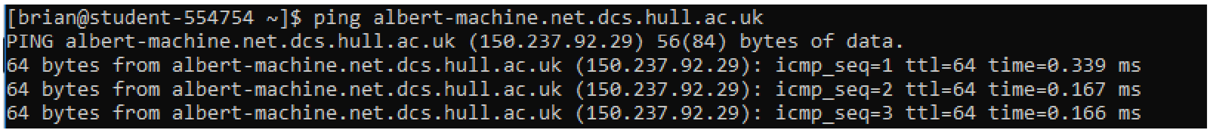
moving files to johnmurray home directory as he has the write permissions for the research project folder



```
[johnmurray@student-554754 ~]$ ls
cat_research
[johnmurray@student-554754 ~]$ cd cat_research
[johnmurray@student-554754 cat_research]$ ls
1016836647_2b9291bf51.jpg  178608212_13cbfc46f6.jpg   449609305_bda5a86518.jpg
1428673011_804c3f1502.jpg  188565002_b47e2ab9ed.jpg   451338953_20c68143f6.jpg
1469373760_8e76da6673.jpg  20378831_a4f330ef25.jpg    474614068_8c3ac322a6.jpg
153102627_39a474549d.jpg   2052963974_9ec8684089.jpg  491687431_ad4e91dbe7.jpg
1681418046_29ccc1ed8c.jpg  2087393988_be21769f95.jpg  505185072_caa3ee74cd.jpg
171048639_6414061122.jpg   249877337_ee2b774eef.jpg   526094174_f627d7d2b9.jpg
171164937_be4ebf78b0.jpg   317778700_5b59605ae6.jpg   59070170_ff893e2cab.jpg
1721962697_9dfc616c79.jpg  356670892_09451a7657.jpg   706757620_e543180e55.jpg
178591027_caef8e78e7.jpg   367329767_a60646d665.jpg
[johnmurray@student-554754 cat_research]$ cd
[johnmurray@student-554754 ~]$ cp -r /home/johnmurray/cat_research /srv/research-
project-phd/cat_research
[johnmurray@student-554754 ~]$ cd /srv/research-project-phd
[johnmurray@student-554754 research-project-phd]$ ls
cat_research
[johnmurray@student-554754 research-project-phd]$ cd /srv/research-project-phd/ca
t_research
[johnmurray@student-554754 cat_research]$ ls
1016836647_2b9291bf51.jpg  178608212_13cbfc46f6.jpg   449609305_bda5a86518.jpg
1428673011_804c3f1502.jpg  188565002_b47e2ab9ed.jpg   451338953_20c68143f6.jpg
1469373760_8e76da6673.jpg  20378831_a4f330ef25.jpg    474614068_8c3ac322a6.jpg
153102627_39a474549d.jpg   2052963974_9ec8684089.jpg  491687431_ad4e91dbe7.jpg
1681418046_29ccc1ed8c.jpg  2087393988_be21769f95.jpg  505185072_caa3ee74cd.jpg
171048639_6414061122.jpg   249877337_ee2b774eef.jpg   526094174_f627d7d2b9.jpg
171164937_be4ebf78b0.jpg   317778700_5b59605ae6.jpg   59070170_ff893e2cab.jpg
1721962697_9dfc616c79.jpg  356670892_09451a7657.jpg   706757620_e543180e55.jpg
178591027_caef8e78e7.jpg   367329767_a60646d665.jpg
[johnmurray@student-554754 cat_research]$
```

copying files from john murray home directory to the folder for the research project

Pinging the domain to get the IP address for Albert



```
[brian@student-554754 ~]$ ping albert-machine.net.dcs.hull.ac.uk
PING albert-machine.net.dcs.hull.ac.uk (150.237.92.29) 56(84) bytes of data.
64 bytes from albert-machine.net.dcs.hull.ac.uk (150.237.92.29): icmp_seq=1 ttl=64 time=0.339 ms
64 bytes from albert-machine.net.dcs.hull.ac.uk (150.237.92.29): icmp_seq=2 ttl=64 time=0.167 ms
64 bytes from albert-machine.net.dcs.hull.ac.uk (150.237.92.29): icmp_seq=3 ttl=64 time=0.166 ms
```

Appendix G – User Passwords

User passwords have been added here in case they are needed instead of the key access.

Brian – 018590

Ashley – ashley

John Murray – johnmurray

Steve Smith – stevesmith

Laura Lance (cannot SSH, only SFTP) – lauralance

Annika PhD – annikaphd

Albert PhD - albertphd