Computer Systems Infrastructure and Management

600099

Virtual Machine Configuration Report

Brian Davis

Student ID: 201707824

Contents

[Changing the Hostname 2](#_Toc38452670)

[Establishing an Internet Connection 2](#_Toc38452671)

[Installing Nano / Updating Pacman 2](#_Toc38452672)

[Securing Root 2](#_Toc38452673)

[Creating User Accounts 3](#_Toc38452674)

[Setting up SSH 4](#_Toc38452675)

[Key Authentication 5](#_Toc38452676)

[Putty 5](#_Toc38452677)

[User authorised keys 5](#_Toc38452678)

[Testing Ashley Admin Remote Login 6](#_Toc38452679)

[Creating and Assigning Groups 6](#_Toc38452680)

[Creating Directories in /srv 7](#_Toc38452681)

[Creating Permissions 7](#_Toc38452682)

[.SSH Folder Permissions 8](#_Toc38452683)

[Logging Permissions for Albert 8](#_Toc38452684)

[STFP 9](#_Toc38452685)

[Critical Evaluation 9](#_Toc38452686)

[Bibliography 10](#_Toc38452687)

[Appendix A – Installing Packages 11](#_Toc38452688)

[Appendix B – DHCPCD Internet Connection 13](#_Toc38452689)

[Appendix C – Securing Root 14](#_Toc38452690)

[Appendix D – User Accounts 15](#_Toc38452691)

[Appendix E - Permissions 17](#_Toc38452692)

[Appendix F – Host Authentication 19](#_Toc38452693)

[Appendix G - SFTP 20](#_Toc38452694)

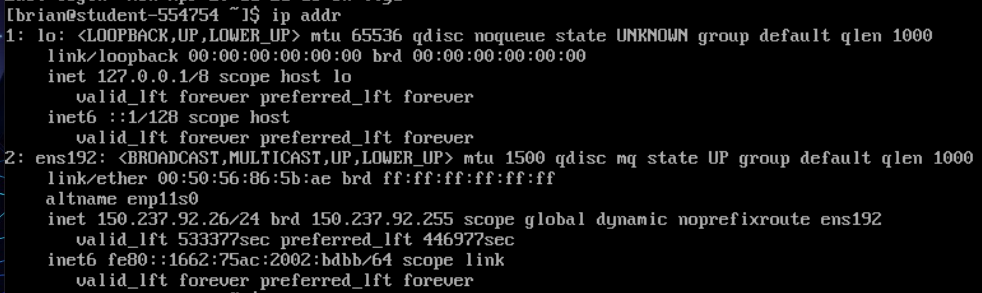
## Changing the Hostname

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. The image below shows the hostname within the /etc/hostname directory through nano.

## 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.

****

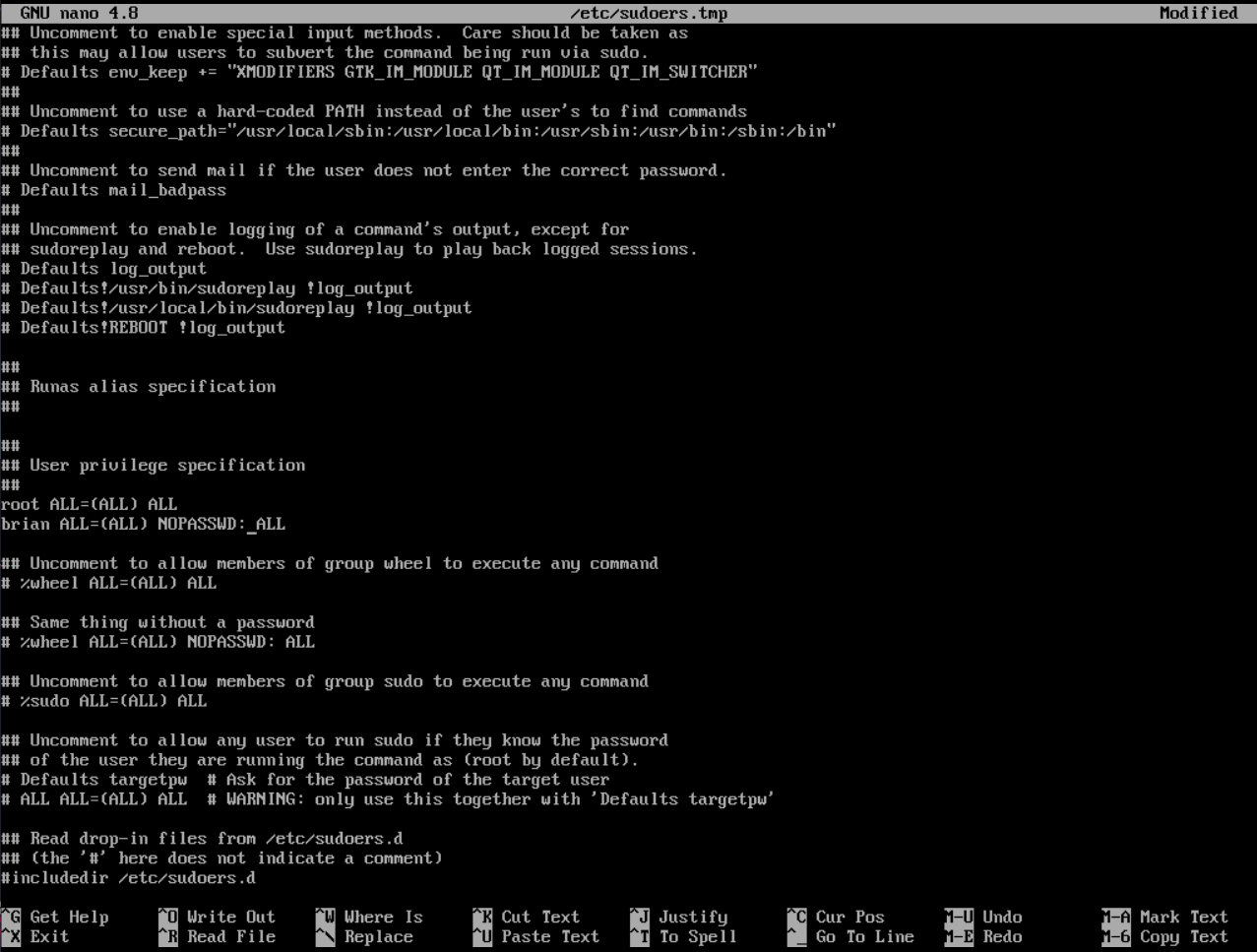
## Installing Nano / Updating Pacman

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.

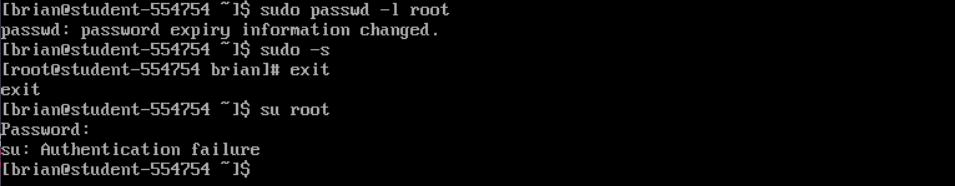
Now that nano is installed, it is worthwhile making sure the package manager Pacman itself is up to date. The command ***‘pacman -Syy’*** is used to force a full update of all the of pacman databases. One of the most important things when setting up a new system is security and having the most up to 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

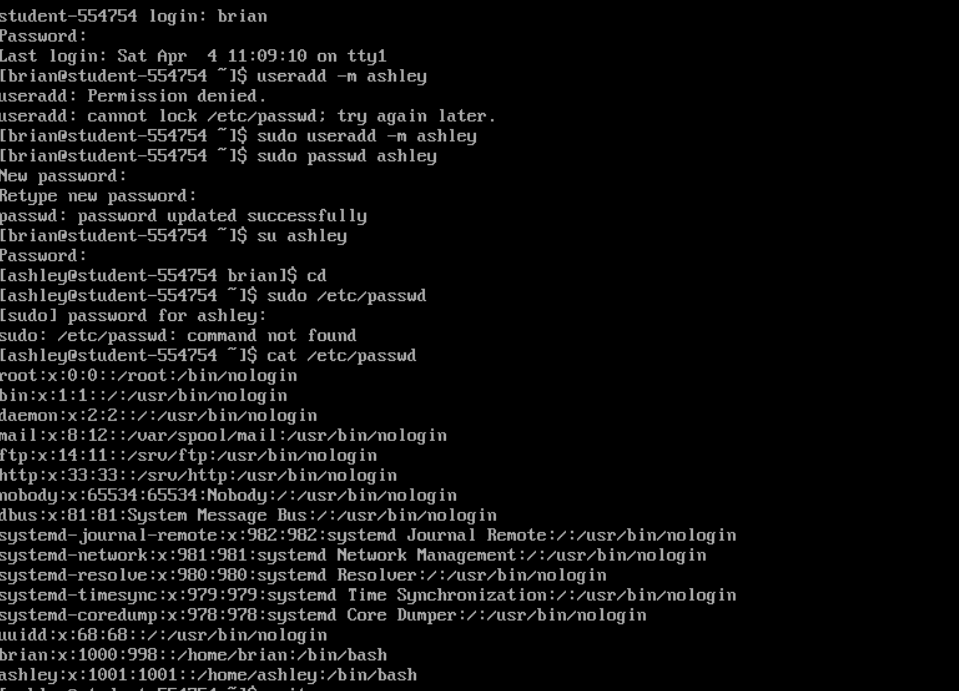
Root is the main account of the system that allows for the ability to customise and essentially do anything you want, but if this account is taken over by someone with malicious intentions it can lead to loss of files and even total system loss *(“How To Disable The Root Account On Linux,” 2018).*

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.

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. The image above 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)*

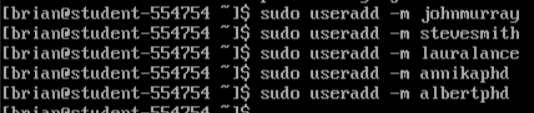
Having admin accounts with their own independent passwords is a lot more secure than having a root account that can be brute force accessed into. To make the system more secure, the root account is locked down. The image below shows how this is done. This locks the root account password authentication, 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.

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.

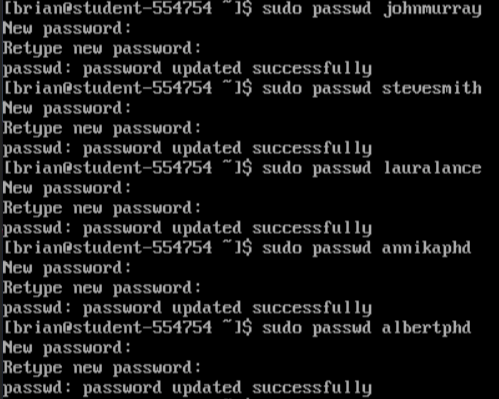
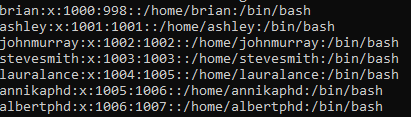


## Creating User Accounts

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.

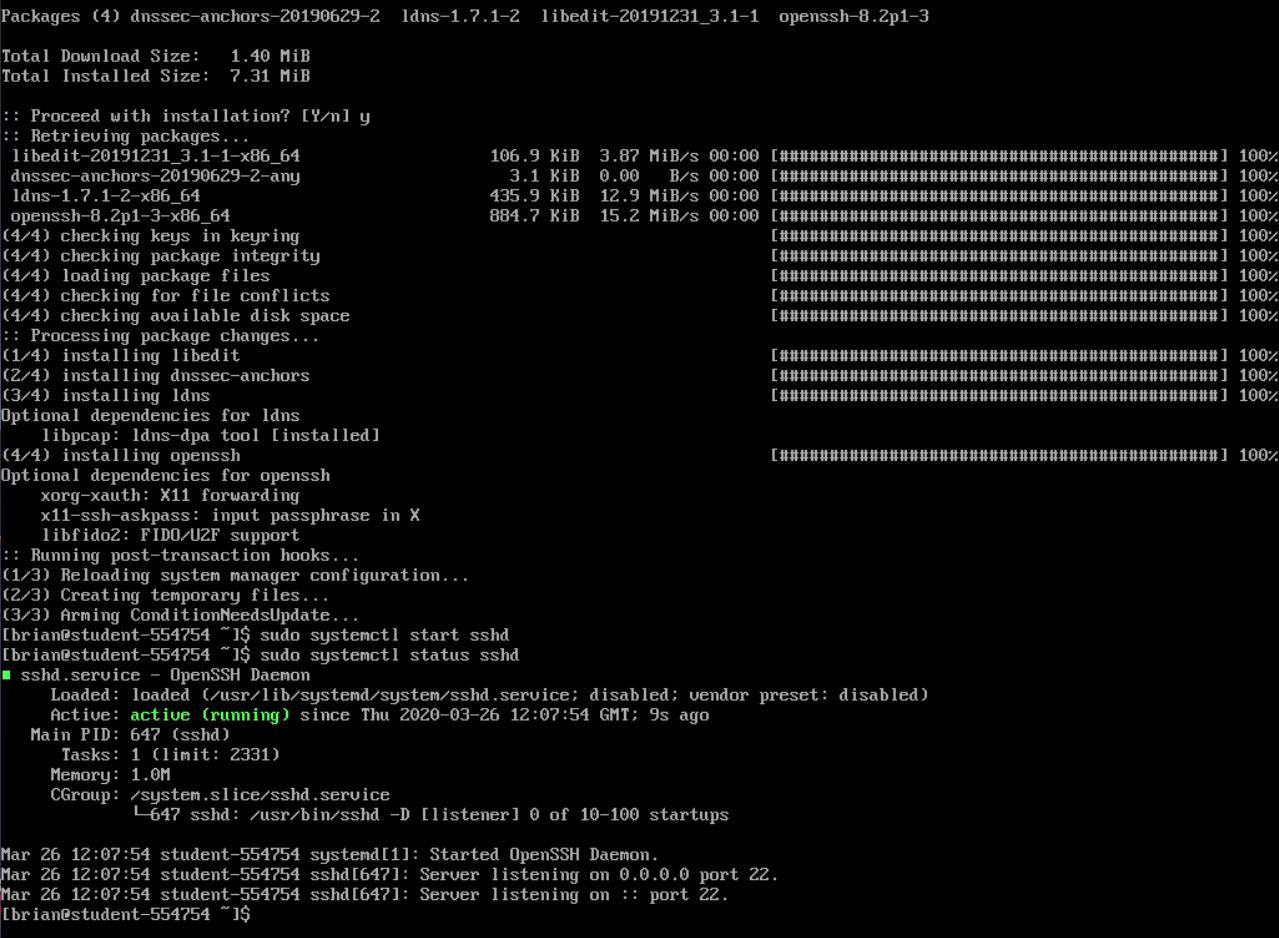


The image above shows the creation of all five additional needed users for the system. **Sudo** needs to be written before each command otherwise there will be a permission denied message. After all users are created, their passwords are created next.

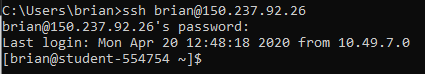
Once the password for each user is written into the terminal and submitted, the password entered is assigned to the user. When this is done for all 5 users, it is possible to check all users on the system by executing ***‘cat /etc/passwd’*** which opens the file to show all users.

## 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.



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. The image below shows this working, ensuring that ssh is set up correctly.

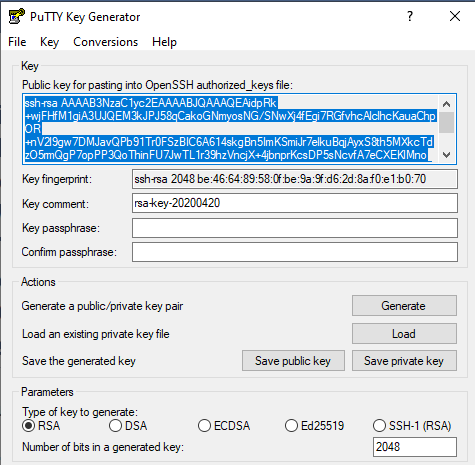


Password access is not very secure, and prone to security issues, so public and private keys will be used instead. These security keys are a way to connect to the system remotely without needing password input, they use the public key found in the user authorized keys file on the system as a different 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 then 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.

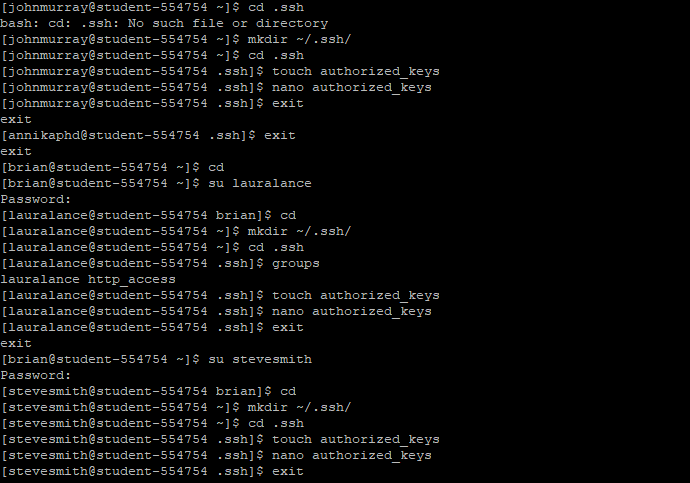
## Putty

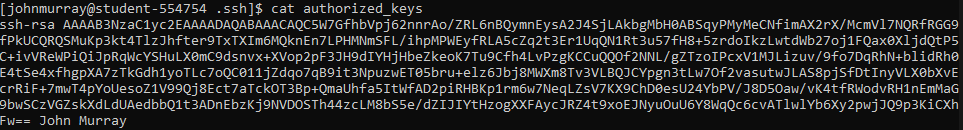
To create the key for the **brian** user, Putty was used as the solution, due to its ease of use and ability to create public and private keys. Putty has a program in its installation called ***‘PuTTYgen’*** which allows the ability to create and then save a public and private key through an interface. 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 file (“How To Configure SSH Keys Authentication With PuTTY And Linux Server In 5 Quick Steps,” n.d.)*.*

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.

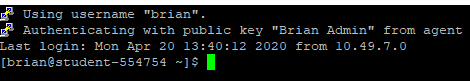
## User authorised keys

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.



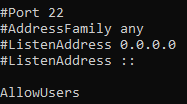
The command of ‘**Touch authorized\_keys’** will create the empty file, **‘nano authorized\_keys’** will edit the file, which allows the public key for each user to be pasted in through Putty. Using the **“cat”** command allows for the ability to see what is in the file.

After the service is restarted, it is possible to test if the authentication is in place, by using [***brian@150.237.92.26***](mailto: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.

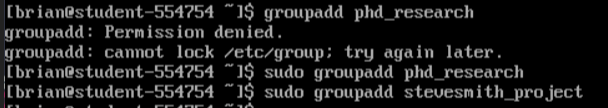


## Testing Ashley Admin Remote Login

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.

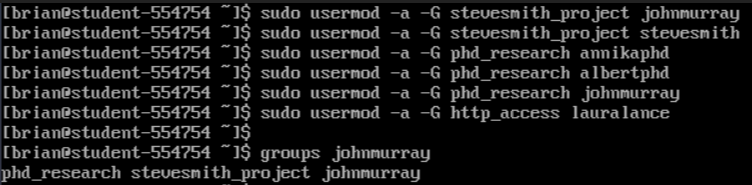
As shown in the image above, the issue is that when **AllowUsers** was written into the **sshd\_config** file beforehand, the user was only specified as **brian**, this meant that no other user could authenticate, and their access was denied. This was fixed by making it so that the **AllowUsers** line in the config file no longer contained a name after it, which meant all users are now able to authenticate remotely successfully.

## Creating and Assigning Groups

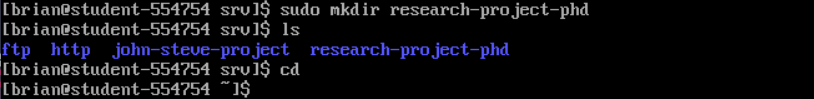
Each user on the system asks for certain requirements be fulfilled for their account. The best way to do this is to create assigned groups that can have permissions available to them, giving each account permissions without the need to give each account admin privileges. For the groups, these are split 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 the following image.

Once these have been created, each user then needs to be assigned to their groups. This is done through the **usermod** command.

The image above 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.



## 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.

The image above 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.

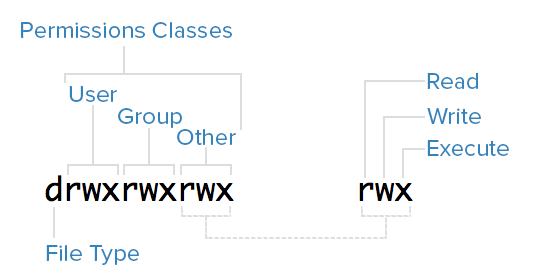
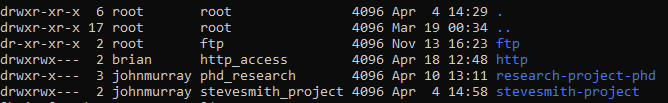
Permissions are set into four categories; R means read permissions are present, W means write permissions are present, X means execute permissions are present and – means there are no permissions present. These can be edited through two modes, absolute (numeric) or symbolic. Editing through the absolute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

Figure - (“An Introduction to Linux Permissions,” n.d.)

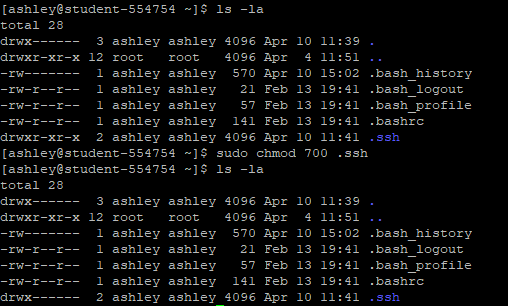
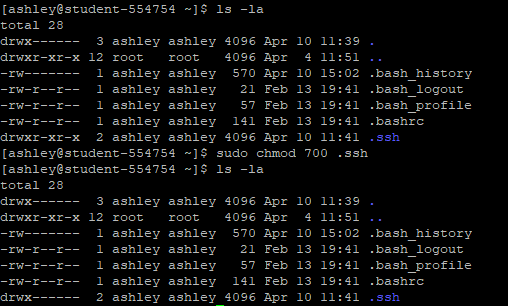
The first thing to do is define each user for the groups. The user is essentially the main owner, whilst the group allows the ability to grant specified permissions to each member of the group, and the other is for all other users not in the first two categories.

As per the 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 the image below. **Mv** is the command that indicates that we want to change the object on the left to be changed to the name on right.

## .SSH Folder Permissions

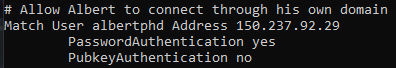
Each user needs their own **.ssh directory** for key authorisation to be possible for their account, but the default permissions given when this directory is created are not suitable.

As shown in the above image on the left side, 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. The image above on the right shows the change made to **---** to remove the permission for a world user to access this folder.

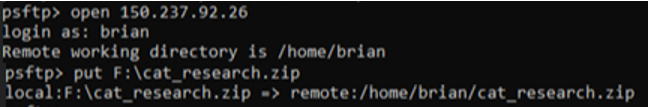
## Logging Permissions for Albert

The requirements state the need for Albert to be able to override the key login from a specified IP address. To be able to do this requires a change in the sshd\_config folder to allow for the ability to override the config settings for authentication. The layout for this is shown below.

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.



## STFP

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.

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.

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 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 of the root account was still available, but I felt it better to find this workaround rather than re-enable the root just for this.

## Bibliography

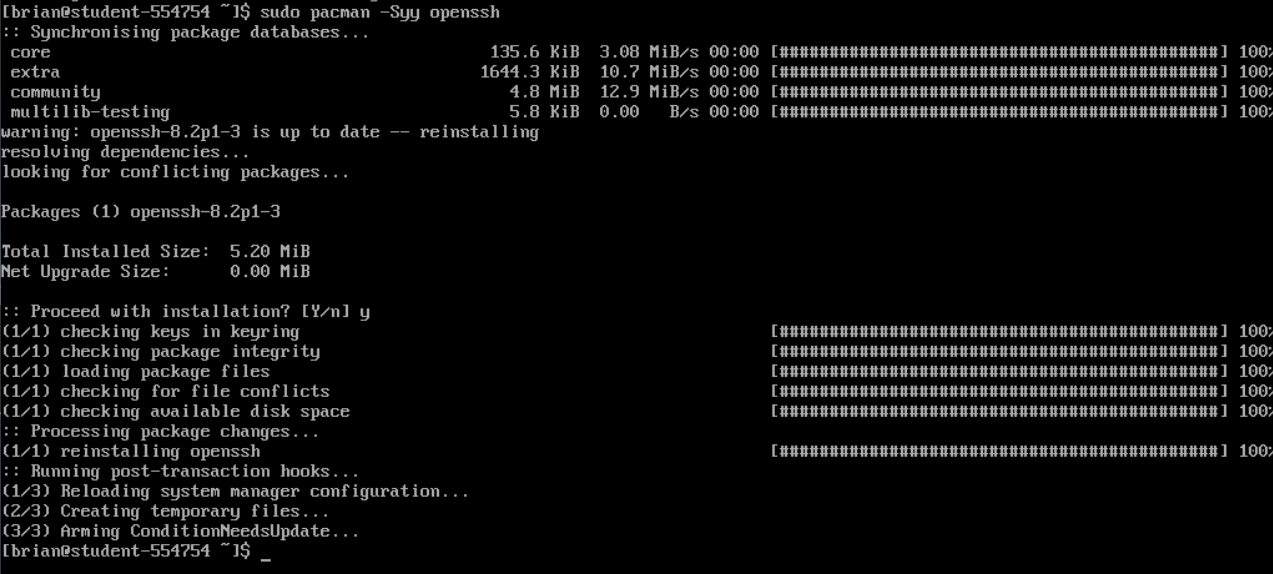
Abdulbasir, I., 2011. Configuring the linux Sudoers file. Linux.com. URL https://www.linux.com/training-tutorials/configuring-linux-sudoers-file/ (accessed 4.20.20).

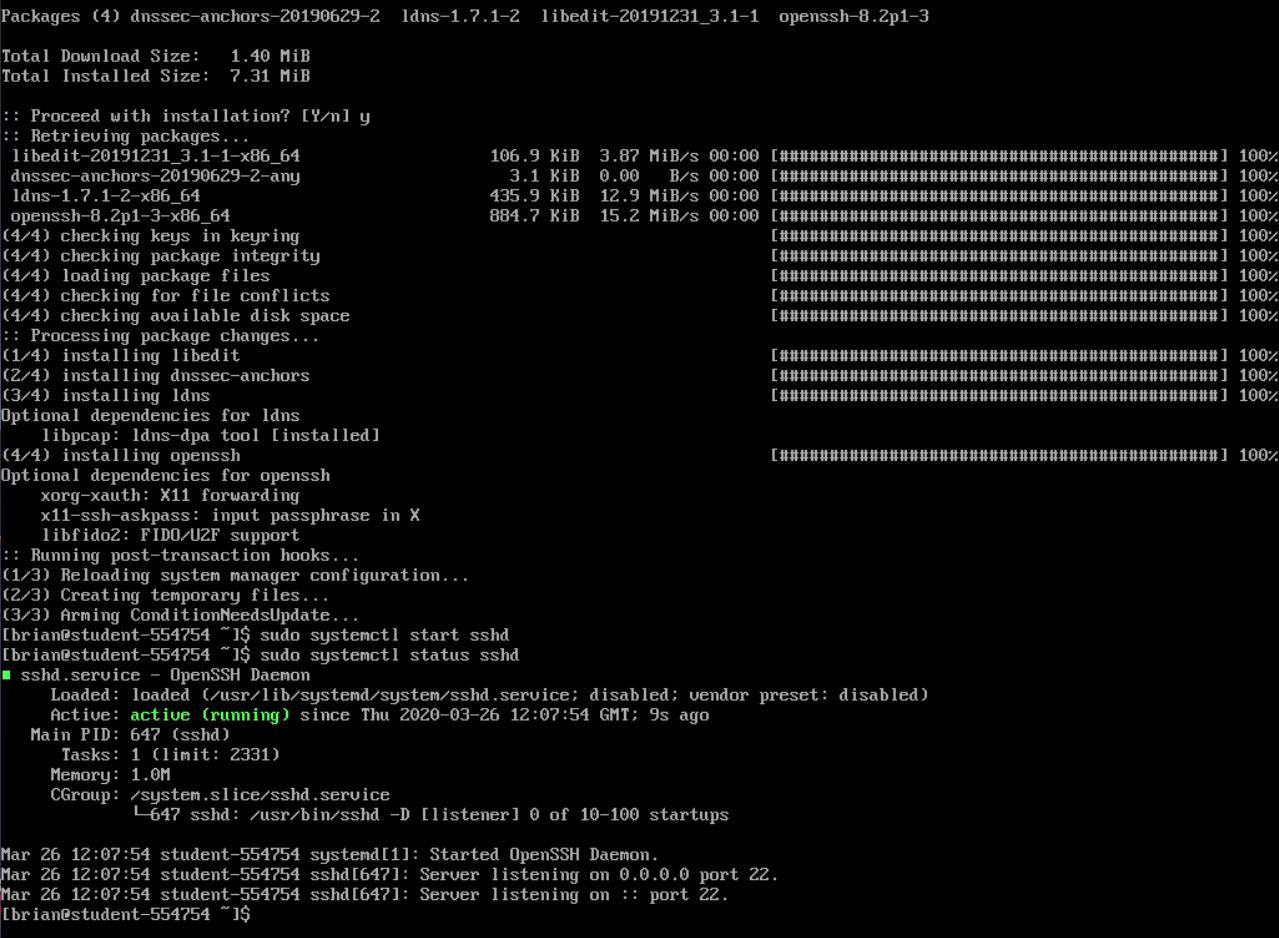
How To Configure SSH Keys Authentication With PuTTY And Linux Server In 5 Quick Steps [WWW Document], n.d. . HowtoForge. URL https://www.howtoforge.com/how-to-configure-ssh-keys-authentication-with-putty-and-linux-server-in-5-quick-steps (accessed 4.20.20).

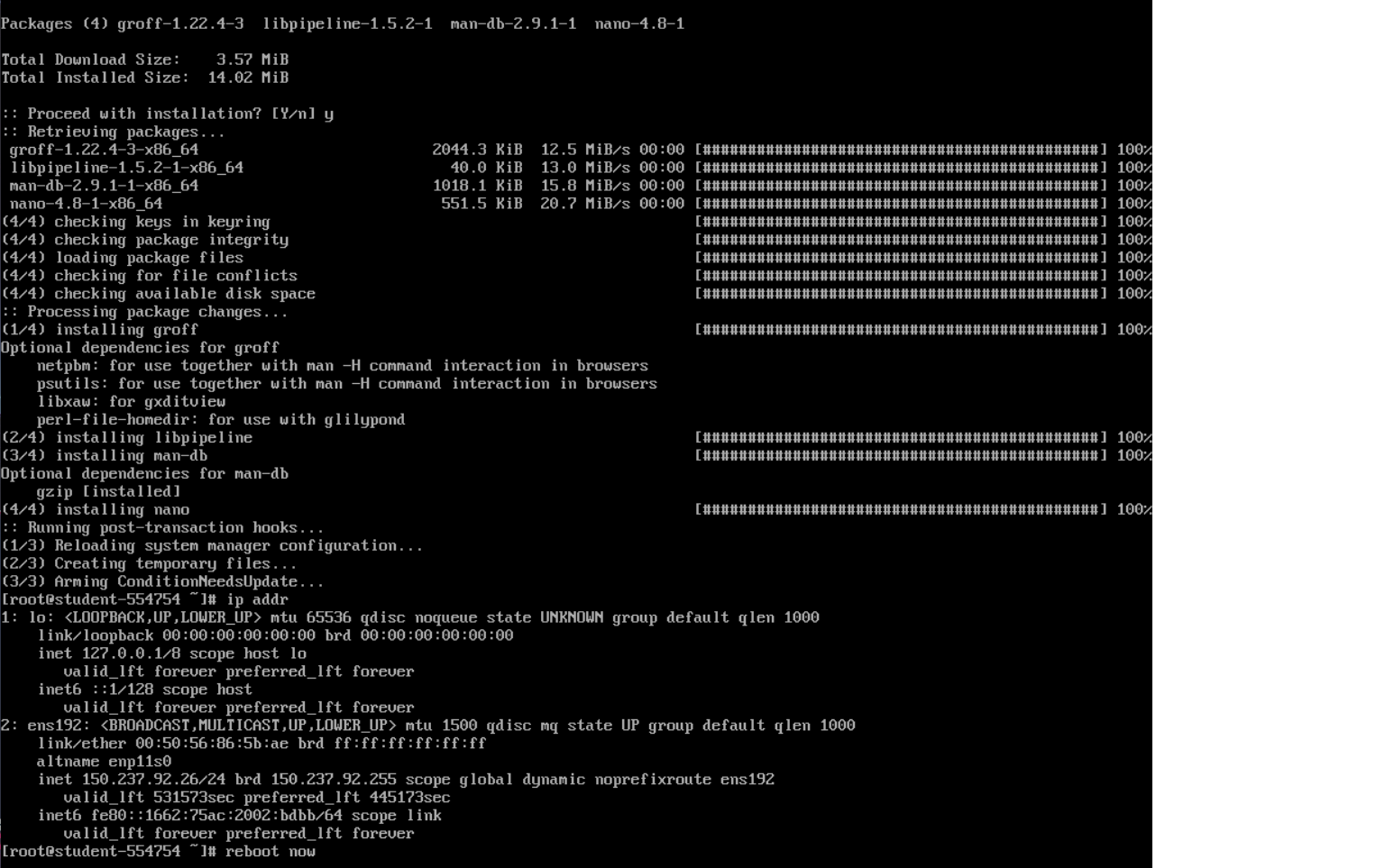
How To Disable The Root Account On Linux, 2018. . AddictiveTips. URL https://www.addictivetips.com/ubuntu-linux-tips/disable-the-root-account-linux/ (accessed 4.20.20).

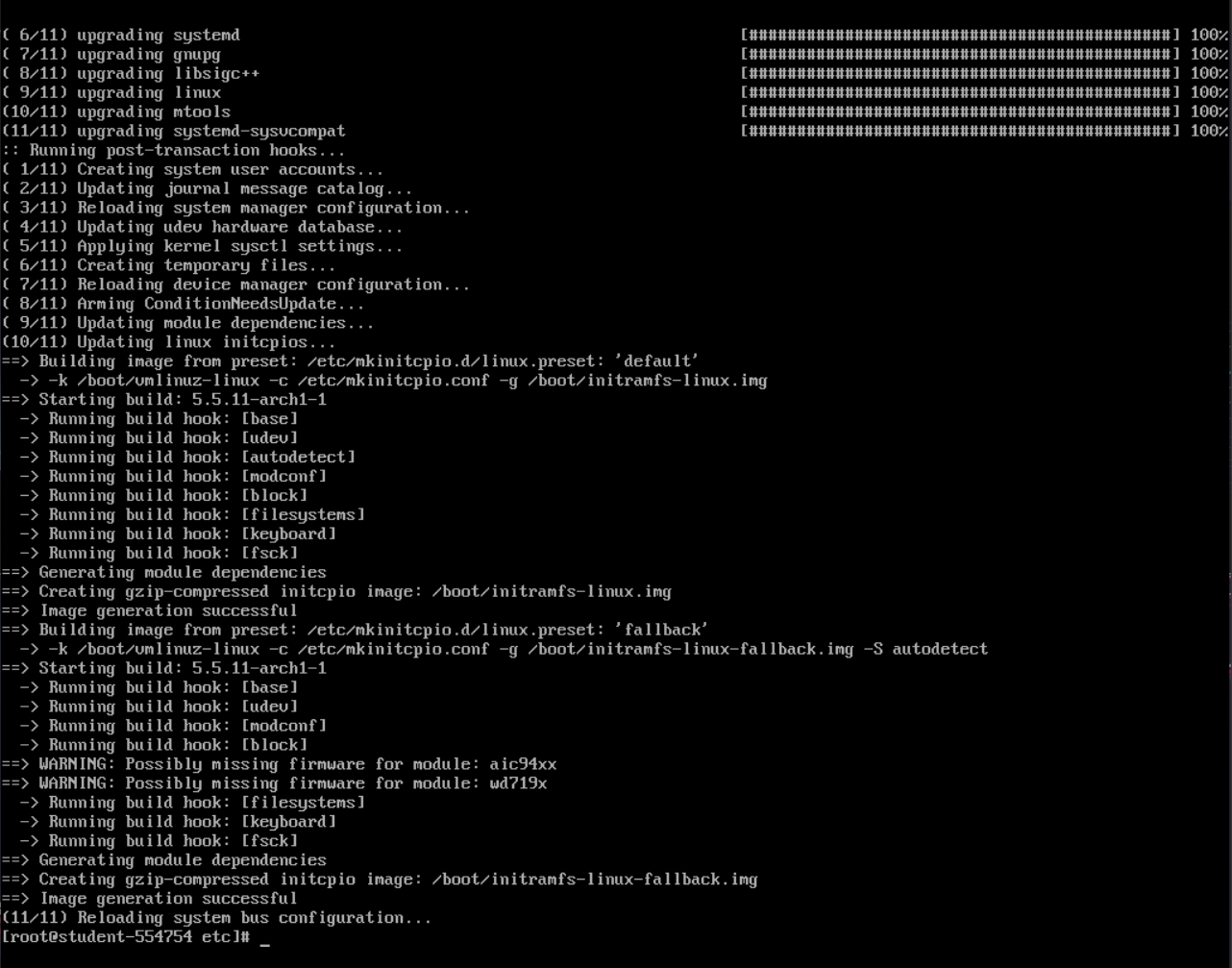
What is SSH Public Key authentication? [WWW Document], n.d. URL https://www.ssh.com/ssh/public-key-authentication (accessed 4.20.20).

## Appendix A – Installing Packages

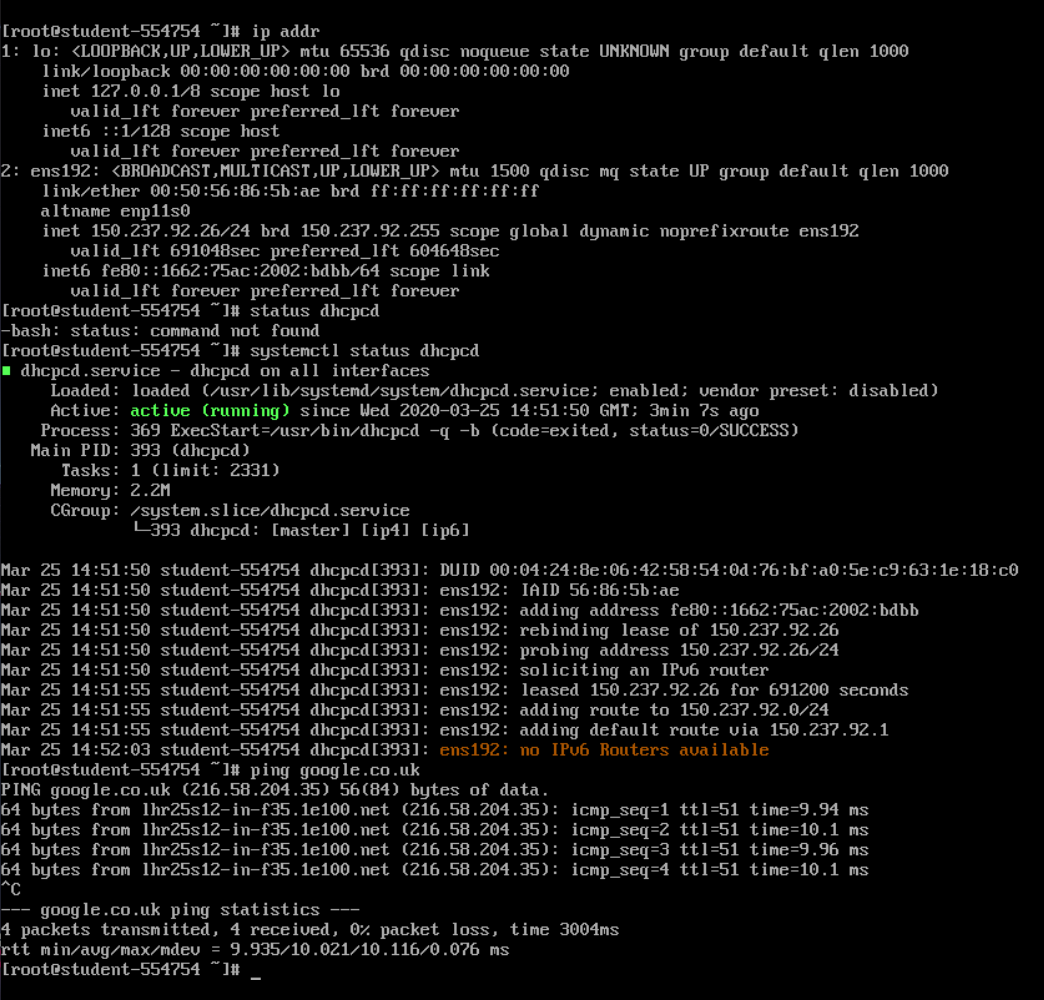
Installing SSH

Installing and running SSHD

Installing Nano

Updating Pacman

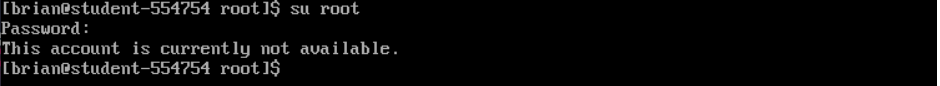
## Appendix B – DHCPCD Internet Connection



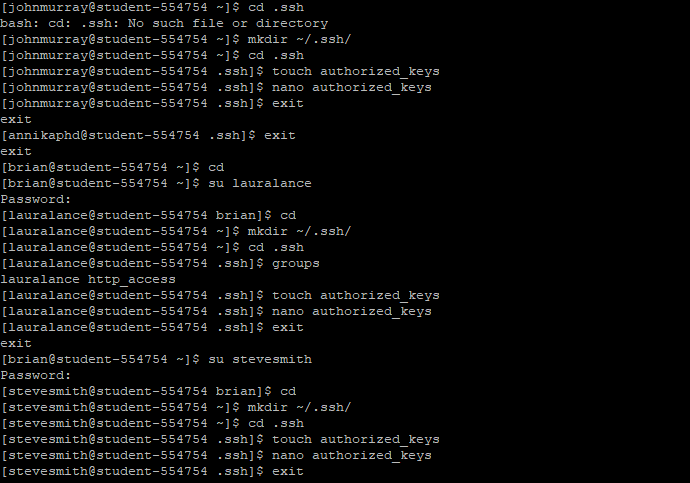
## Appendix C – Securing Root

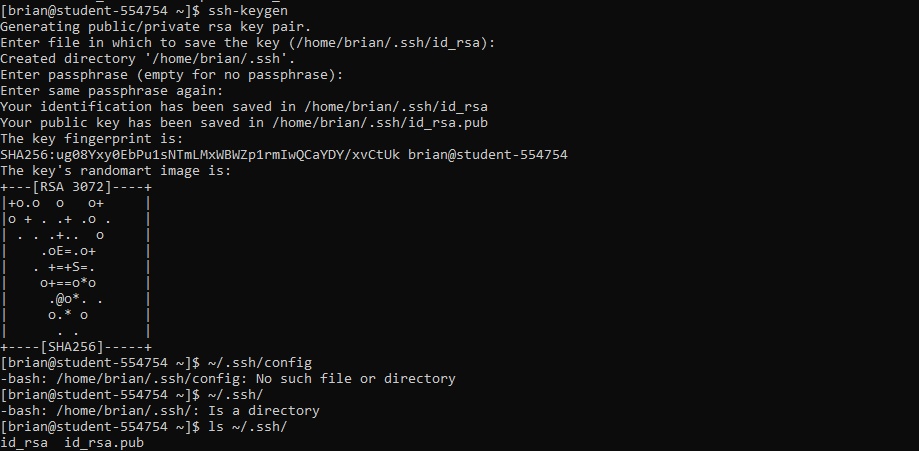


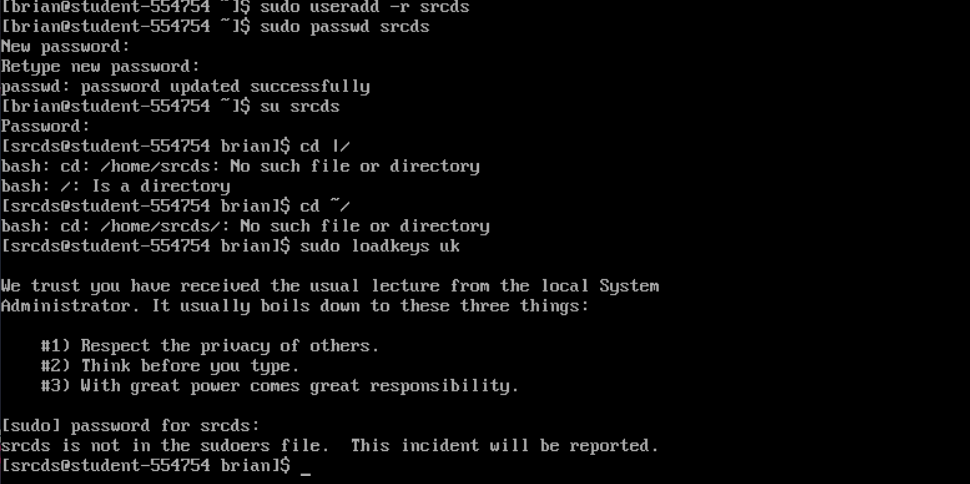
No Root Login

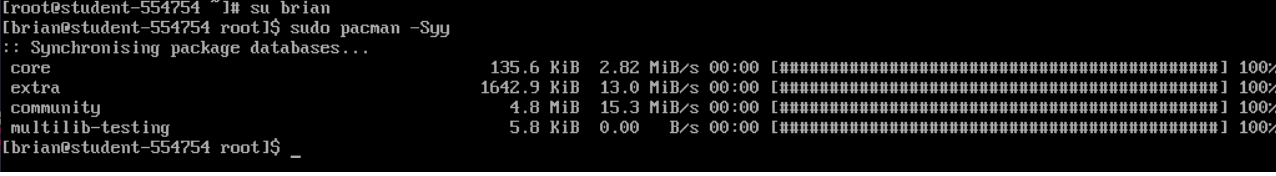
Root Unavailable

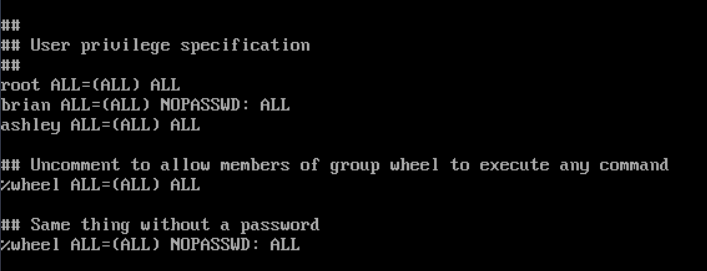
## Appendix D – User Accounts

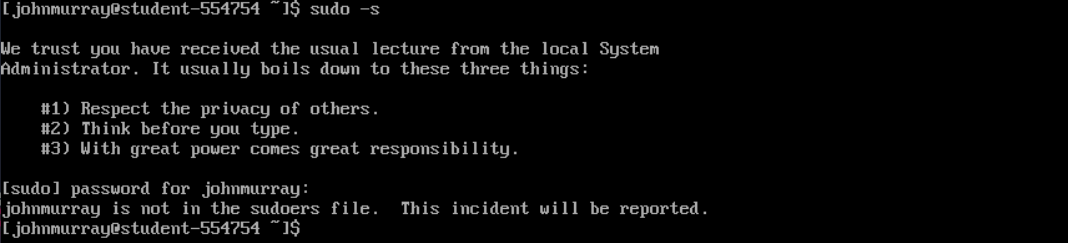
Adding authorized\_keys file for each user

Creating SSH Private Key for brian admin account

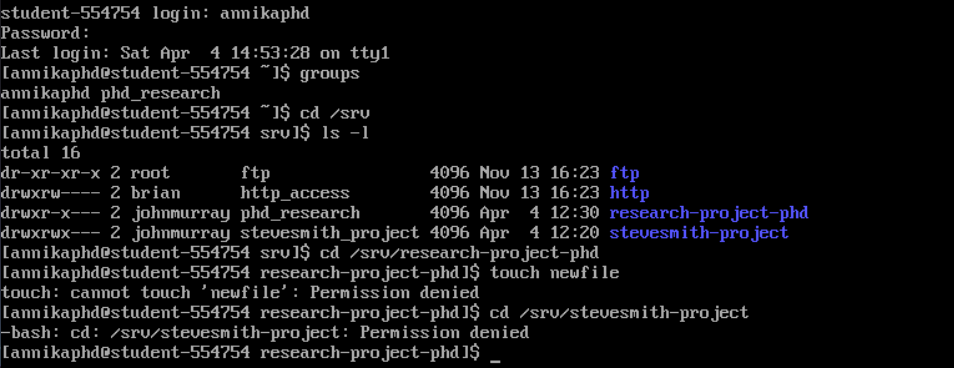
Creating practice admin system account, with error

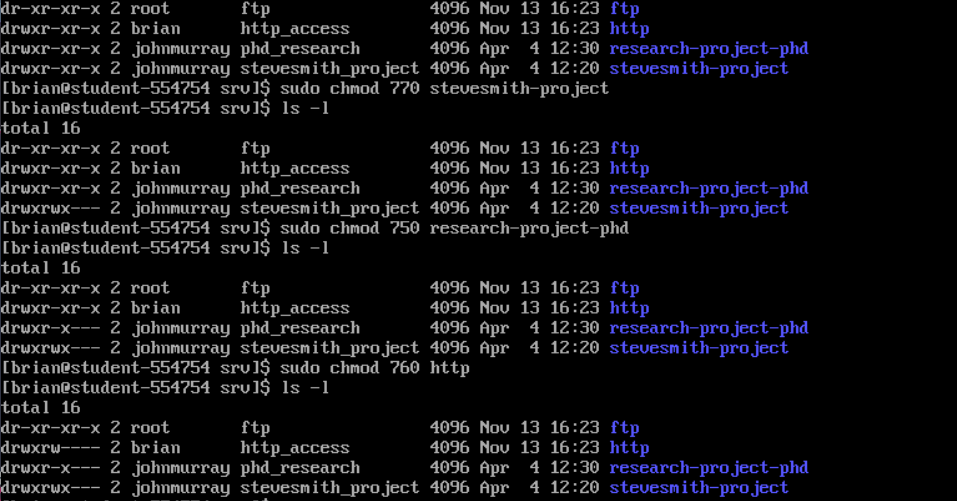
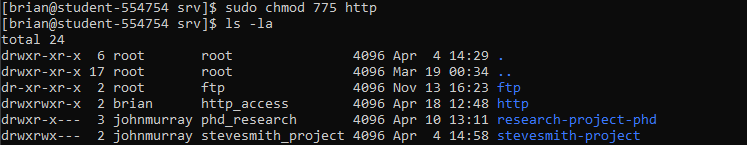
First instance of Elevated user

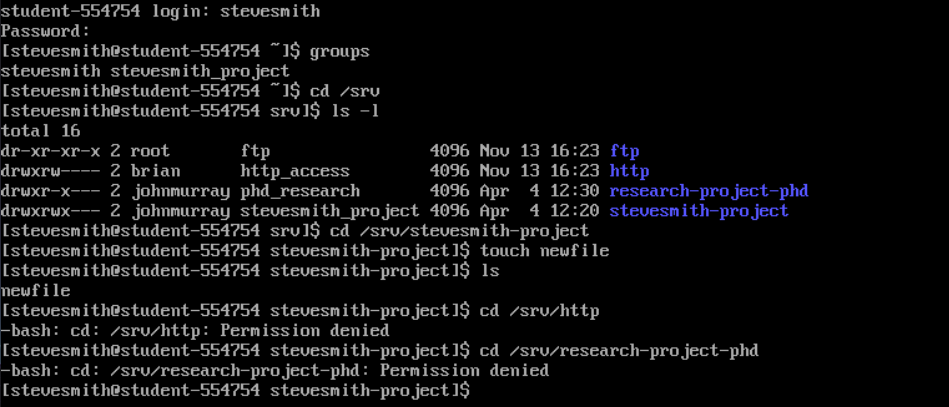
Admin access right for both admin accounts

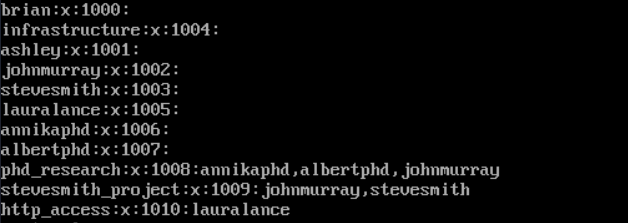
Normal user unable to use sudo, permission check

## Appendix E - Permissions

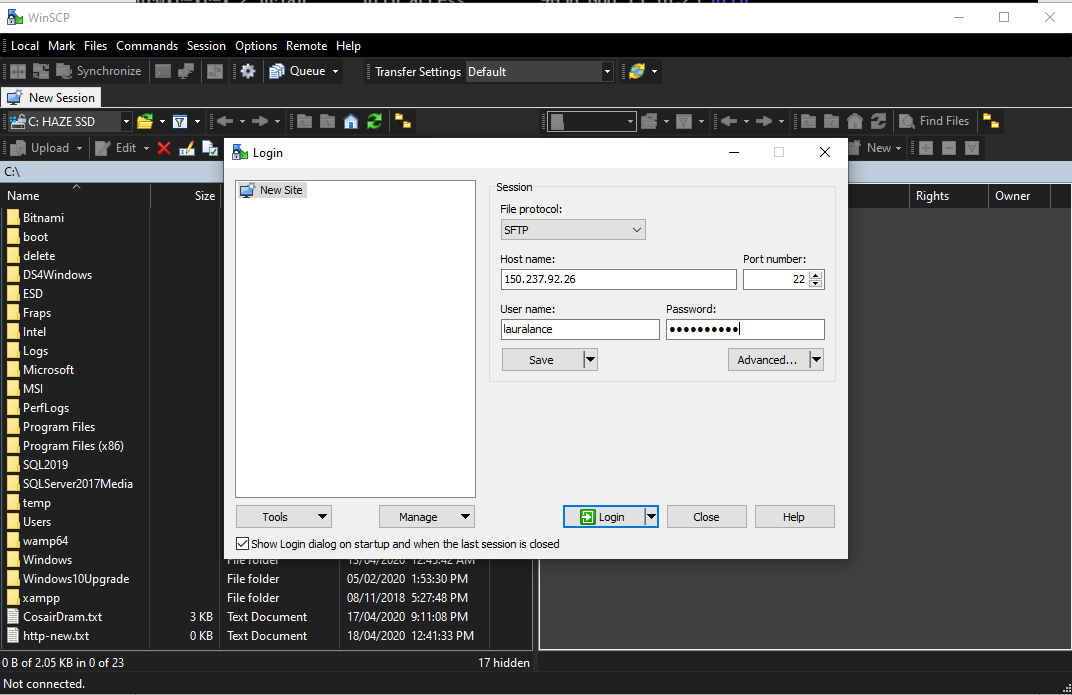
Checking the permissions for the Annika user, unable to write to phd research folder

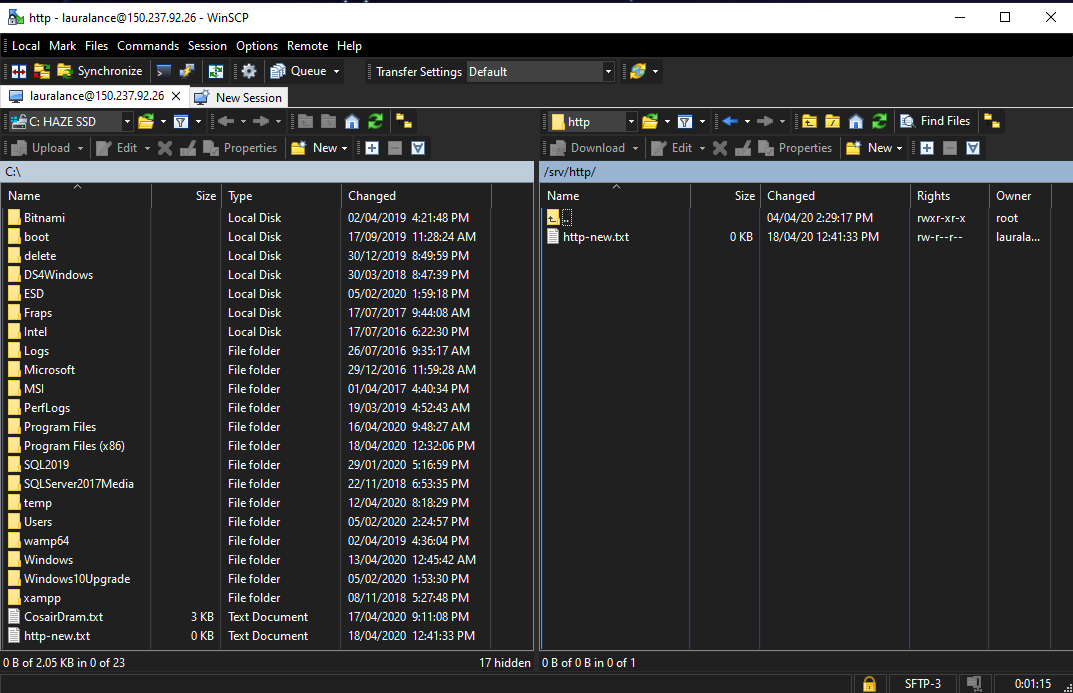
Changing folder permissions using Octal

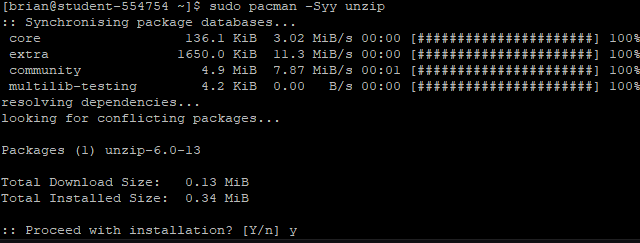
Checking permissions for Steve

Viewing users in their respective groups

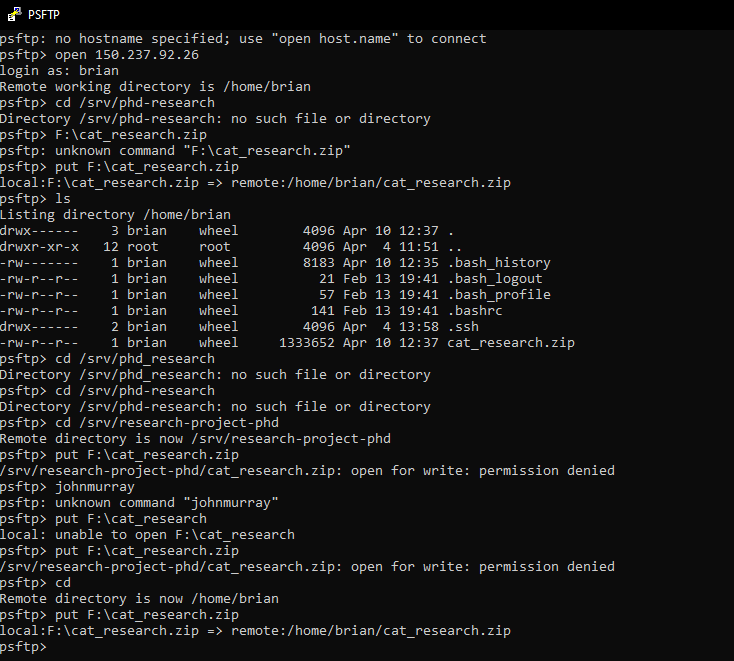
## Appendix F - SFTP

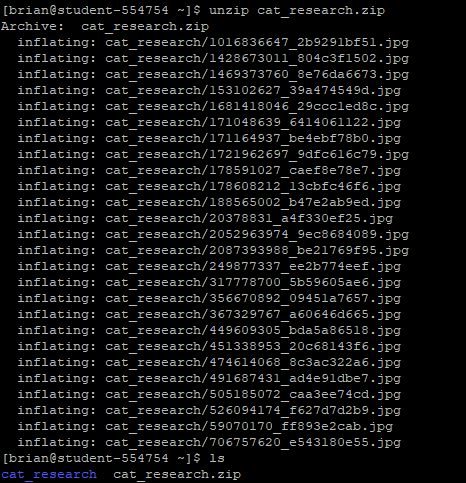
Connecting to SFTP through the WinSCP windows local interface

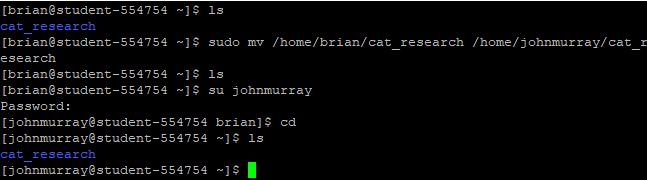
Successfully adding a file to the http folder through WinSCP

Remote SFTP and unzipping process

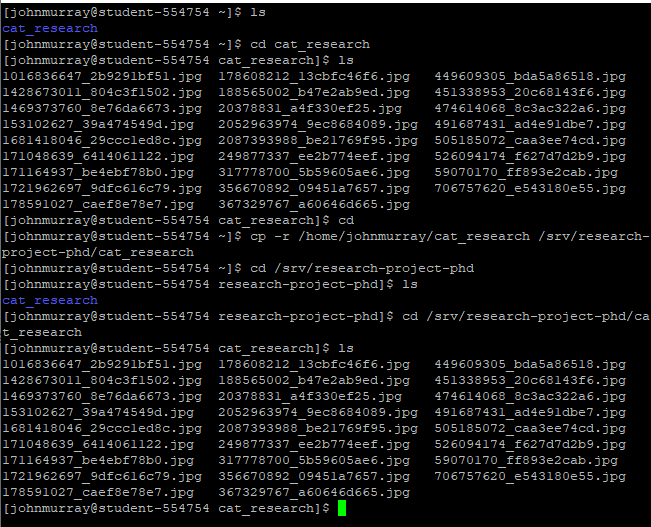
Installing unzip via pacman

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

unzip all the files, which creates a new directory

remove the zip file as it’s no longer needed

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

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