Preparing to Install the OS

1. Download the Ubuntu Server operating system:
   https://ubuntu.com/download/server

2. Download a USB installer to set up your USB stick:
   Unetbootin – https://sourceforge.net/projects/unetbootin/
   Pendrive Linux – https://www.pendrivelinux.com/
   Rufus – https://rufus.ie/

3. Run the USB installer

4. Set the boot order in your servers BIOS so it boots from the USB stick

5. Decide on the IP address for your server
   - When a device (phone, laptop, etc) connects to your network, your router picks an IP address for it. This is called a **dynamic IP address**, as it can change, and for most devices it doesn't matter what IP address they have.
   - For your server though, you want to set a **static IP address** so you can easily connect to it.
   - IP addresses can go from 1 to a maximum of 255, once it is not already being used. **192.168.178.300** for example will not work. This is a feature of network addressing

6. Insert the USB stick into your server and power it on!

First Five Minutes After Installation

1. Add a Coloured Prompt

- When you log in to the command line on a newly installed Ubuntu server, the prompt uses minimal colours.
- To add colours, you need to enable the **force_color_prompt** option in the .bashrc file in your home directory
- When you first log in, you will be in your **home** directory. Type **ls –la** to see all of the files, including hidden files (we go into these in more detail in a later video)
- To edit the **.bashrc** file, type **nano .bashrc** (the nano text editor is also explained in more detail in a later video)
- Remove the **#** from the beginning of the **force_color_prompt=yes** line
- Press **Ctrl + O** to save the file, then **Ctrl + X** to exit the text editor
- Log out and log back in again to see the effect

```
# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
force_color_prompt=yes
```

- Before example:

```
jupiter@homeserver:~$ pwd
/home/jupiter
jupiter@homeserver:~$
jupiter@homeserver:~$ ls -l
total 8
-rw-rw-r-- 1 jupiter jupiter    5 Jun  7 05:06 file01
drwxrwxr-x 2 jupiter jupiter 4096 Jun  7 05:06 folder01
jupiter@homeserver:~$
```

- After example:

```
jupiter@homeserver:~$ pwd
/home/jupiter
jupiter@homeserver:~$
jupiter@homeserver:~$ ls -l
total 8
-rw-rw-r-- 1 jupiter jupiter    5 Jun  7 05:06 file01
drwxrwxr-x 2 jupiter jupiter 4096 Jun  7 05:06 folder01
jupiter@homeserver:~$
```

2. Set the Timezone

- Type **date** to check the system time. If the time is incorrect, you will need to set the timezone
- Type **sudo dpkg-reconfigure tzdata**, and select your continent and closest city
- Type **date** again to confirm the time is now correctly set

3. Sudo Password

- The user account you created during installation is a 'normal' user account, but does have permissions to escalate to admin in order to do administrative tasks
- To run administrative tasks, like setting up new users, changing server settings, etc, you need to use the **sudo** command. This command is put at the beginning of the admin command you want to run. For example, to show the contents of the root (admin) users folder, you would need to run **sudo ls –l /root**
- You will be prompted for your password each time you run the **sudo** command
- To remove this requirement for the password, you need to edit the **sudoers** file. To do this, type **sudo visudo** and add your username (i.e. jupiter) down the bottom of the file:

```
jupiter ALL=(ALL) NOPASSWD:ALL
```

- Press **Ctrl + O** to save the file, then **Ctrl + X** to exit the text editor
- The changes should take place immediately. Try run **sudo ls –l /root** again

4. Install any Updates

- Be sure to update your server soon after installation to install any new security or software patches
- Run **sudo apt update** to refresh your servers local repository of software, and then **sudo apt upgrade –y** to actually install any new patches

5. Shutdown/Reboot the Server

- At this point, your server is setup and ready. Test that you can connect to it from another laptop using PuTTY to ensure you can connect over the network. This is covered in the next video
- If you need to shutdown your server, you can do so by running **sudo shutdown –Ph now**
- To reboot the server, run **sudo reboot**
- Run **man shutdown** and **man reboot** to read the documentation for the shutdown & reboot commands to learn more

PuTTY Settings

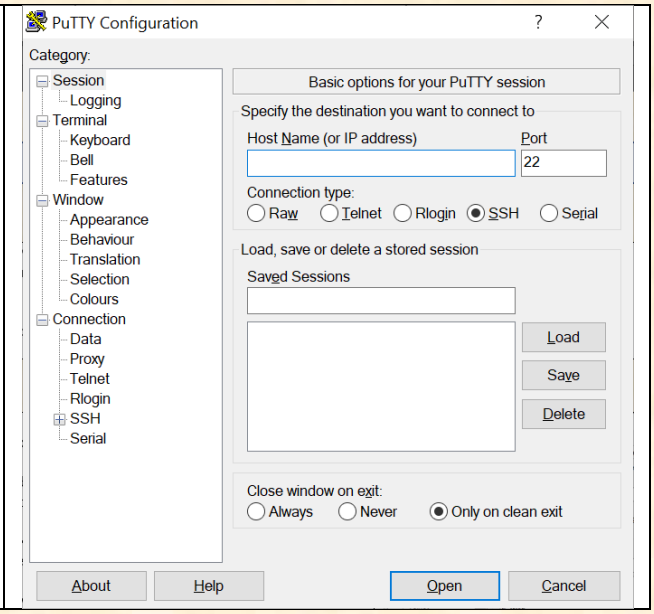
- Download the PuTTY installer package from https://putty.org
- Ctrl + C and Ctrl + V don't work for copying text from or pasting data into PuTTY SSH sessions
- To copy text from a PuTTY session, highlight it using the mouse and press **Shift + Insert**
- To paste text into a session press **Ctrl + Insert** or right click on the mouse

Session:

- Once you make any session setting changes, save these by giving the session a name and clicking Save
- To modify an existing session, click Load, make your changes and click Save again

Session > Logging:

- To save the text in the terminal window, enable logging
- All session output: save all text you will see on the screen during your session



Window:

- Change the number of lines that you can scroll back through (default set to 2000)

Window > Appearance:

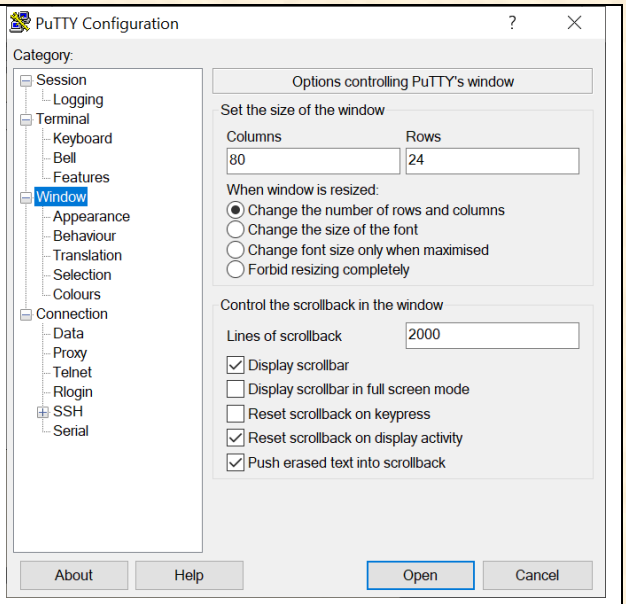
- Change the cursor from a block (default) to an Underline or a Vertical Line, and whether it blinks
- Change the font and font size

Window > Behaviour:

- Disable 'Warn before closing window'. PuTTY will prompt you when you click the X button to close the window. This setting will disable this prompt
- Set the title of the window

Window > Colours:

- Set specific colours for text and background



Connection > Data:

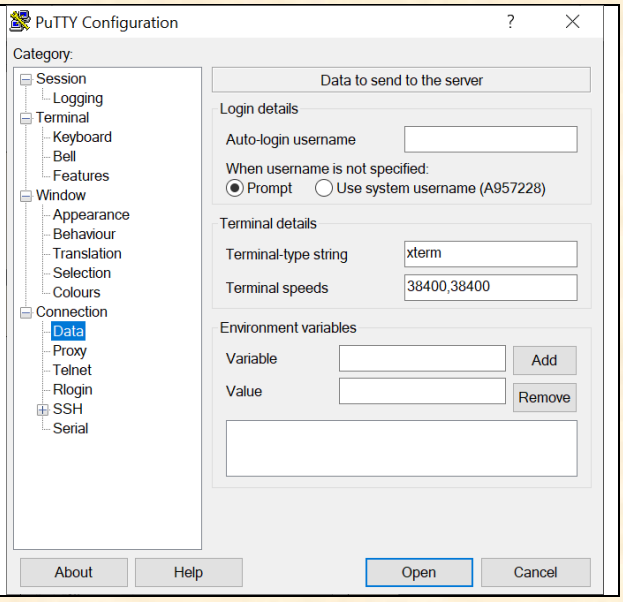
- Set the username of the user you want to automatically log in as, so when you log in using that saved session, the first prompt will be for the password

Connection > SSH:

- Run a command as soon as you login. This will run the command, and then log off again

Connection > SSH > Auth:

- Set a private key file to be used for a password-less logon



Basic Linux Commands

- **Flags**: Options, also called *flags*, can be added to most commands to expand their usage. Flags are separated from the command by a space, and identified using a dash, i.e. `ls -l` to call the `ls` command and use the long format (`l`) flag.

  Flags are typically a single letter, but many (not all) have a longer name also. When using the longer name format, a double dash is used, i.e. `ls -l` can also be called as `ls --long-format`.

  A number of flags can be called together, either as `ls -lh` or `ls --long-format --human-readable`
- **Arguments**: Some commands need to be called with a parameter, also called an argument. The `clear` command can be called by itself, and it clears the screen of text, but when creating a folder (directory), you obviously need to tell Linux what the directory you are creating is to be named: `mkdir <folder_name>`
- **Case**: Unlike Windows, case matters in Linux. `LS` is not the same as `ls`. You can also have files in a folder with the same name, but with different mixes of upper and lower case
- **Hidden Files**:For simple tidiness, when you run `ls` Linux will not show files which begin with a dot (.). These are often config files. To view these, use the `-a` (`--all`) flag, i.e. `ls -al`
- **Piping**: The output from any command can be piped, or redirected, into another command. The cat command for example prints the contents of a file to the screen. You can pipe this into the grep command in order to search that contents for a word or a string of text, e.g. `cat file1 | grep "sandwich"`

Working with File & Directories	
ls	List the contents of the directory you are currently in `ls /etc` lists the contents of the /etc directory Flags: `ls -a` show hidden files `ls -h` show file sizes in human-readable format, i.e. 1Mb instead of 1000000 `ls -l` list in long format. Very useful for seeing info like permissions, file type, etc
mkdir	Make a new directory `mkdir <folder_name>`
rm	Delete (remove) a file Flags: `rm -f` (force) remove files without asking `rm -r` (recursive) remove folder and all contents
rmdir	Delete (remove) an empty directory. To delete a directory with all its contents, use `rm -rf` instead
mv	Move (and/or rename) a file or folder to a new location `mv file1 file2` – rename file1 to file2 `mv file1 /etc` – move file1 to /etc directory `mv file1 /etc/file2` – move file 1 to /etc and rename it to file2
cp	Make a copy of a file: `cp file1 file2` – copies file1 and creates file2
file	When run against a particular file, it shows what type of file it is: `file file1` – shows what type of file file1 is – whether it is a text file, directory, script, link, pipe, tar archive, …
touch	Though its primary, documented, purpose is to update the access & modification times of files, it is most often used to quickly create a new blank file. Useful if you are testing something, like a new script.

ln	Create a link (shortcut) to a file or folder.
	`ln -s <original> <shortcut>`
	`ln -s /etc my_shortcut` – create a shortcut to the /etc directory called my_shortcut. You can then use cd to jump into this directory, i.e. `cd my_shortcut`

Navigating Directories	
pwd	Show the directory you are in currently
cd	Change directory. `cd /etc` moves into the /etc directory `cd ..` moves back up a level `cd` by itself brings you back to your home directory (/home/<username>)
clear	Clears the screen of all text

Downloading Files	
wget	Very useful and configurable tool for downloading files/data from the internet `wget http://example.com/file01.zip`
zip	Package multiple files or folders into a compressed zip file. This reduces the size of them, and makes it easier to send them over networks, i.e. the internet `zip package.zip file1 file2`
unzip	Unzip a zipped file: `unzip package.zip`

tar	tar was traditionally a tool for making tape backups of data (tar = tape archive). Using tar, you can package a number of files into one file, and then compress it to save space. This is still used as a means of sharing data – many linux programs are still available online as 'tarballs'.
	
	To create a tar file:
	`tar cvzf tar_filename.tgz file1 file2 folder1`
	
	To extract a tarball:
	`tar zxvf tar_filename.tgz`
	
	Flags:
	-c – create a tar file
	-v – verbose – print the process on screen as it happens
	-z use
	-f – use the file name that follows this command
	-x – extract (as opposed to create) a tar file

Software Installation & Updates	
apt update	Update your package managers local database of available packages/ software. This does not actually update any software, just the list of available software that your system knows about. Need to be run as sudo
apt upgrade	Upgrade the installed packages on your computer. This does update/upgrade all installed software. Needs to be run as sudo
apt install	Install a package. It is best practice to update your package manager first, so it has the most up-to-date list of available software. To install htop, run: sudo apt install htop
apt remove	Remove a package, but leave any config files, in case you want to reinstall at a later date. Needs to be run as sudo
apt purge	Remove a package and all of its config files
apt autoclean	Clear your system of downloaded package files
apt autoremove	Clear your system of libraries and packages which are no longer needed

System Information	
ip addr	Show your IP address. Run ip -c addr to add colouring
free	Show the server memory (RAM). Add -h to use human readable values, i.e. free -h
df	Show the disk space on the server. Add -h to use human readable values, i.e. df -h

man	Show the man page for a command. This documentation will detail how to use the command and what flags are available: man ls
which	Every time you run a command, you are calling a program saved somewhere on the system. ls, for example, is saved at /usr/sbin/ls. The whipuch command shows the full path to a command. Useful when writing scripts or checking if a program/command is installed
history	All commands you type are saved in a file in your home directory called .bash_history. You can see the history of commands by running the history command.

Viewing Text Files	
cat	Displays all the contents of a text file to the screen. If the file has more lines than your screen can display, you need to use less or more
less	Display contents of a file one screen full at a time. You can scroll forward or back through the text, and perform searches
more	Display the contents of a file one screen at a time. You can only scroll forward
grep	Search for text in a file

Privilege Escalation	
sudo	The sudo command allows a user on the server to perform admin tasks, without having to have access to the admin account. This allows admins to limit which users can do admin tasks, and even which admin tasks they can and cannot do.

Permissions	
chown	Change ownership of a file/folder. Often needs to be used with sudo sudo chown new-owner:new-group file1
chmod	Change permissions on a file/folder. Scripts need to be made executable in order to run them: chmod a+x file1
usermod	Change features of a user account. Commonly used to add a user to a group sudo usermod -aG newgroup username

Editing Text Files	
nano	nano <filename> Enter the vi text editor. If the file you name doesn't exist, it will be created
vi	vi <filename> Enter the vi text editor. If the file you name doesn't exist, it will be created

Running Multiple Commands	
;	Run multiple commands one after the other clear; ls; echo; date
&&	Run multiple commands. The commands will run only if the preceding ones complete successfully clear && ls && echo && date

Filesystem Hierarchy Standard

Linux is based on the early Unix operating systems, developed back in the 1970's, and its file system structure is based on and derived from these roots. On Microsoft Windows computers, you have the C: drive as your main system drive and all of your directories are based off of that.

The Linux structure is officially called the FileSystem Hierarchy Standard (FHS), and in this standard, everything is based off of root, represented by a forward slash, (/). If you type cd /, you are moving to, or changing directory to, the root directory. To confirm the directory you are in, type pwd, i.e. 'present working directory'. You can type ls or ls –l to see all sub-directories under root. Under the FHS, the purpose of each sub-directory is clearly defined, as there have been differences in how various linux distributions organised their file systems.

As per the FHS, the sub-directories under root are used for the following purposes:

/	The root directory. Every file and folder falls comes under root.

/bin	Binaries (executable programs) for the commands run on the server. For example, when you run ls, what is being run is the binary /bin/ls. Other commands, such as cat, nano and mv, live here also

/boot	Boot loader files, used when the server is booting up

/dev	Device files. Device files are interfaces to devices attached to your server. Your hard drive could be represented by **/dev/sda**. There are also special files under **/dev**, which you will sometimes come across: **/dev/null** – an empty file to which you can send any input, and it is discarded. It is often used when running a command which produces output to the screen you don't want to see. For example, if you ran a command to find file01, and didn't want to see any error messages, you would send (redirect) the error messages to **/dev/null**, i.e. find / -name file01 2> /dev/null **/dev/zero** – this device produces a continuous stream of zeroes. Examples of where you would use this would be if you wanted to wipe a device such as a hard disk or USB stick by writing all zeroes to it

/etc	Configuration files for programs and services on your server

/home	The home directories for users on your server. A user called jupiter for example, will have their personal folder under **/home/jupiter**. On desktop Linux versions, the users Desktop, Documents and Downloads folders are all stored here, e.g. **/home/jupiter/Desktop** You will also have some hidden directories, which do not appear when you run ls. Add the -a flag (ls -a) to see these. These are often config or user preference files, used by various programs. Linux treats files or directories whose name begins with a dot (.) as a hidden file/directory

/lib	Code libraries for the binaries in **/bin** and **/sbin**

/lib64	Code libraries for 64 bit binaries in **/bin** and **/sbin**

/media	Mount point for removable media, such as CD-ROM or floppy drives. Historically these would have been put under **/cdrom** or **/mnt**, but under the FHS standard these were aggregated under **/media**. If you put a CD-ROM into your server, its contents should be available under **/media/cdrom**

/mnt	Mount points for other mounted filesystems, i.e. if you added an additional hard drive

/opt	Add-on application software packages often have dependencies outside of the folder created for that application, are placed under **/opt**

/proc	This is a virtual directory. The files within it are not actual files, but rather they are 'calls' to the system for info. If you run cat /proc/cpuinfo, you are asking the server to provide info on the CPU for instance.

/root	Home directory for the root user

/run	A directory used by the system for files needed during the boot process

/sbin	Utilities used for system administration, many of which are root only commands. Examples include fsck for running file system checks, iptables, for firewall configuration, ifconfig and reboot

/srv	A directory, specified under the FHS, for any data which is being served to others by the system

/sys	An old directory structure. The original purpose of this directory has essentially been replaced by **/dev** and **/proc**. Do not delete this directory though as this may break your server

/tmp	A temporary space for putting files, etc, while they are being worked upon. This directory is often cleared upon reboot so don't save any necessary data here

/usr	A very important directory for the Linux system. Some important sub-directories include: **/usr/bin** – the main directory of executable user commands on the server **/usr/include** – as the Linux OS makes heavy use of the C language, this is where general-use files needed for many programs are stored **/usr/lib** – libraries required for system programs **/usr/sbin** – more system programs **/usr/share** - contains directories for word lists, documentation, man pages and timezone information **/usr/src** – source code repositories

/var	A directory for files which are expected to change often, i.e. are variable. Some important sub-directories you commonly visit include: **/var/www** – web server root. Any files/folders stored here are displayed by the Apache web server **/var/log** – a wide variety of system logs

Nano Text Editor Commands

- Commands use either the Control key (^-) or Meta key (M-)
- Meta is commonly mapped to the **Alt** key on keyboards
- Navigate through the file using your keyboard arrow keys or PageUp and PageDown

Starting the Text Editor	
nano	Open nano
nano filename	Open a file using nano. If the file you name doesn't exist, it will be created

Commands	
Control + o	Save file
Control + x	Exit nano
Control + g	See the additional help pages
Control + w	Search for text
Control + _	Jump to a line, and column, i.e. letter in the line. **Control + _ 3,2** jumps to the third line, letter 2

Copying & Pasting	
Control + ^	Mark text for copying/cutting
Meta + k	Copy text
Control + k	Cut text. Cut the whole line if no text has been marked. This is the same as deleting the line
Control + u	Paste text

Vi Text Editor Commands

- **vi** has two modes to be aware of:
  - ❑ **Command Mode** – the default mode, where pressing keys cause actions to be taken on the file
  - ❑ **Insert Mode** – the mode you enter to actually insert/edit text in your file
- In Command Mode, when you press keys on the keyboard, you are performing actions on the file, rather than editing the file itself. These actions can be navigating around the file, copying or pasting text, entering Insert mode or saving the file
- In Insert Mode, the keys you press will be written to the file, i.e. you are editing the text in the file. To escape and go back to Command mode, press the Escape key
- **vi** is case-sensitive, and uppercase and lowercase keys have different actions when in the Command Mode

Fundamental Commands

Starting the Text Editor	
vi filename	Enter the vi text editor. If the file you name doesn't exist, it will be created

Exiting the Text Editor	
:w	Save the file but don't exit
:wq	Save and quit (exit)
:q	Exit if no changes have been made
:q!	Exit and discard any changes that were made

Navigating Through a File	
Arrow keys	Move up, down, left & right in the file. Page Up & Page Down work also
^	(Shift + 6) Jump to start of line
$	(Shift + 4) Jump to end of line
1G	Jump to first line in the file
G	Jump to last line in the file

Editing Text	
i	Enter Insert Mode at the current cursor position
a	Enter Insert Mode just after the cursor position
u	Undo. Pressing this again redoes the last edit, i.e. it toggles the last edit back and forth
Esc	Escape to Command Mode, in order to save and/or exit the file

Deleting Text	
x	Delete a single character (the character the cursor is over)
dd	Delete the current line

Copying & Pasting Text	
yy	Copy (yank) the current line
p	Paste the copied line beneath the current line
P	Paste the copied line above the current line

Additional Commands

Editing Text	
I	Jump to the start of the line and enter Insert Mode
A	Jump to the end of the line and enter Insert Mode
o	Create a new line beneath where your cursor is and enter Insert Mode
O	Create a new line above where your cursor is and enter Insert Mode
r	Overwrite/Replace a single character (no need to press Escape after)
R	Overwrite/Replace multiple characters
D	Delete the rest of the line, everything after the cursor

Line Numbering	
:=	Show the total number of lines at the bottom of the screen
:set number	Turn on line numbering
:set nonumber	Turn off line numbering

Navigating Through a File	
w	Jump ahead to the start of the next word
b	Jump back to the start of the last word
nG	Jump to line number <n>, for example **10G** would jump to line number 10

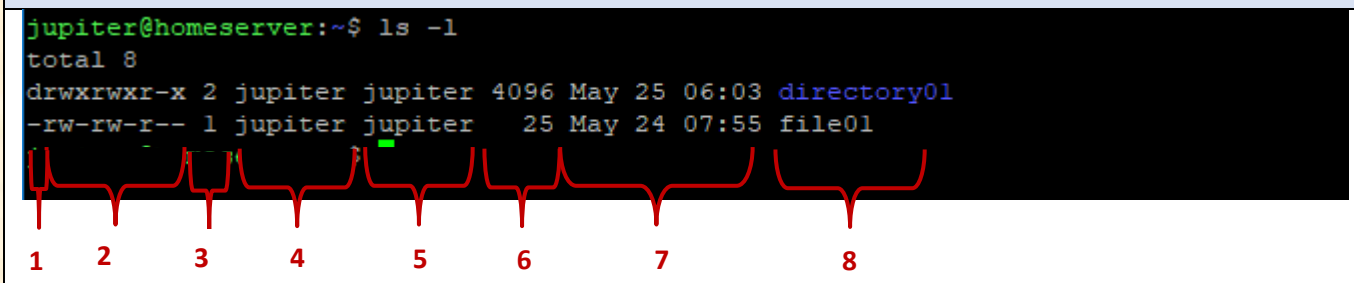
Searching for Text	
/<text>	Search forwards through the file for <text>
?<text>	Search backwards through the file for <text>
n	When results are found, use **n** to jump forward to each result
N	Jump backwards to each result

Run Linux Commands from Within vi	
:!<command>	Run linux commands from within the vi session, for example: :!ls :!pwd :!clear;date :!clear;ls –lhs ../

Linux Permissions

ls –l output

/etc/passwd line format



1. Directory bit	This first character indicates if the object is a directory (d) or a regular file (-). There are other indicators but these are the two you will see most commonly
2. Permissions	These indicate the permissions for the Owner, Group and Others. The file has Read & Write permissions for the Owner & Group, and Read permissions for Others
3. Links	The number of links to a file/directory
4. Owner	This is the name of the user who owns this file
5. Group	This is the name of the user group the file is in
6. Size	The size of the object in bytes. Use the –h flag (ls –lh) to convert this to human readable values, i.e. 4.0K instead of 4096. Directories will always be 4096, as this is the pointer to the directory on the disk, rather than being the actual size of the directory and its contents
7. Last Modification Time	Date & time that this file/directory was last changed
8. Name	Name of the file/directory

Permission	Files	Folders
r	Read the file	List the contents
w	Write to the file	Create & delete files
x	Execute the file	Move into & access files

chmod command

- To change permissions for a file/folder, use **chmod**

Command Syntax				
chmod	u (user) g (group) o (other) a(all)	+ (add permissions) - (remove permissions)	r (read) w (write) x (execute)	file/folder name

Command	**Description**
chmod a+x file01	add the execute flag to user, group & others
chmod o-wx file01	remove write & execute permissions from others
chmod -R go-rwx folder01	remove rwx permissions from folder01 and its contents (recursive)
man chmod	Review the man document for the chown command

```
jupiter@homeserver:~$ ls -l
total 4
-rw-rw-r-- 1 jupiter jupiter 5 May 28 06:04 file01
jupiter@homeserver:~$ chmod a+x file01
jupiter@homeserver:~$ ls -l
total 4
-rwxrwxr-x 1 jupiter jupiter 5 May 28 06:04 file01
jupiter@homeserver:~$
jupiter@homeserver:~$ chmod og-x file01
jupiter@homeserver:~$ ls -l
total 4
-rwxrw-r-- 1 jupiter jupiter 5 May 28 06:04 file01
jupiter@homeserver:~$
```

Set permissions numerically

- Permissions can also be set numerically using three digits, representing the user, group and others:
  - ❑ r = 4
  - ❑ w = 2
  - ❑ x = 1
- These values are added together for each user category:
  - ❑ 1 = execute only
  - ❑ 2 = write only
  - ❑ 3 = write and execute (1+2)
  - ❑ 4 = read only
  - ❑ 5 = read and execute (4+1)
  - ❑ 6 = read and write (4+2)
  - ❑ 7 = read and write and execute (4+2+1)

```
jupiter@homeserver:~$ chmod 640 file01
jupiter@homeserver:~$ chmod 754 file02
jupiter@homeserver:~$ chmod 700 file03
jupiter@homeserver:~$ ls -l
total 12
-rw-r----- 1 jupiter jupiter 5 May 25 06:59 file01
-rwxr-xr-- 1 jupiter jupiter 5 May 25 06:59 file02
-rwx------ 1 jupiter jupiter 5 May 25 06:59 file03
jupiter@homeserver:~$
```

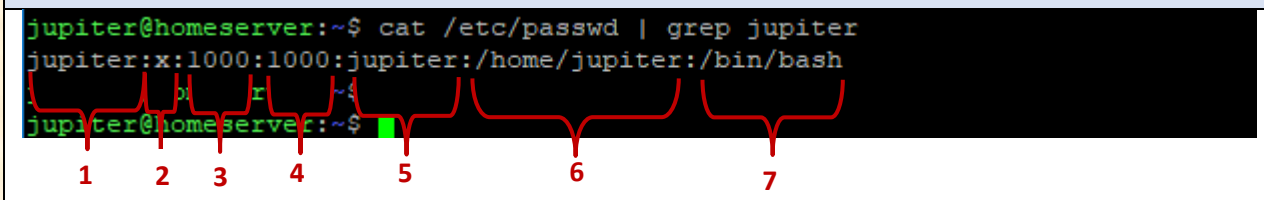
Linux File Ownership

- Commands used in this doc:
  - ❑ cat – this command prints the contents of a file to the screen
  - ❑ pipe (|) – the | character 'pipes' or uses the output from the first command as the input for the second command
  - ❑ grep – this searches for a string (text) within text or a file

/etc/passwd file

- This contains the basic attributes of a user account. Each attribute is separated by a colon (:)
- Passwords are not stored in this file. A second file, /etc/shadow, contains the encrypted password as well as other information such as account or password expiration values, etc. /etc/shadow is accessible only by the root account

/etc/passwd line format

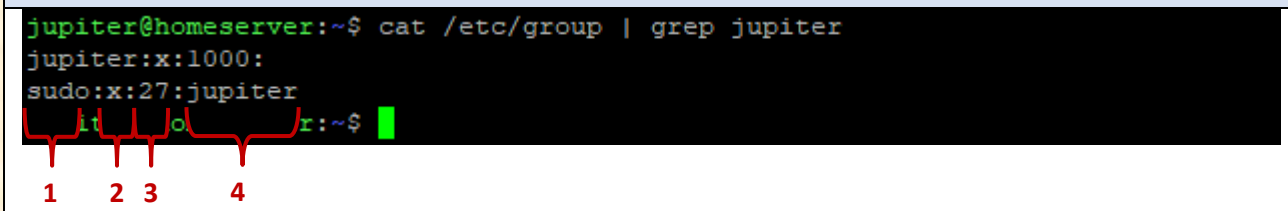


1. Username	The name of the user account. This is what will be used when logging on
2. Password	In older Linux systems, the user's encrypted password was stored in the /etc/passwd file. On most modern systems, this field is set to x, and the user password is stored in the /etc/shadow file.
3. User ID (UID)	The User ID is a number assigned to each user. It is used by the operating system to refer to the user. UID 0 is reserved for the root user, and UIDs 1-999 are reserved for system accounts. Other, 'human', user accounts are given UIDs from 1000 upwards.
4. Group ID (GID)	The Group ID of the users primary group (from /etc/group)
5. Extra User Info	Any additional info about this user, such as their real name, phone number etc. This is usually added when setting up the user account
6. Home Directory	The location, or path, of the users home directory. If none was created, this becomes root (/)
7. Command Shell	The command shell that a user uses when they log in to the command line. On Ubuntu, the Bash shell (/bin/bash) is the default shell. Sometimes this is set to /usr/bin/nologin, or /bin/false. These are not real shells and prevent this user account from being logged in to. This is a security feature.

/etc/group file

- This defines the user groups in use on the system

/etc/group line format



1. Group Name	The name of the group
2. Password	As password can be but is most often not set on the group
3. Group ID	The ID number representing the group
4. Members	The members of this group. In the image above, there are no members of the **jupiter** group and **jupiter** is the only member of the sudo group.

- To check what groups a user is part of, run **groups username**
- To use the **sudo** command, a user must be part of the sudo group
- To add a user to a group, run **sudo usermod -aG newgroup username**

```
jupiter@homeserver:~$ groups jupiter
jupiter : jupiter adm cdrom sudo dip plugdev lxd
jupiter@homeserver:~$
```

chown command

- To change ownership of a file/folder use the **chown** command

Command	Description
sudo chown newuser file01	Change the ownership of a file
sudo chown newuser:newgroup file01	Change the owner and group that a file belongs to
sudo chown :newgroup file01	Change the group that a file belongs to
sudo chown -R newuser:newgroup folder01	Change the owner and group for a folder and all of its contents
man chown	Read the chown man document

```
jupiter@homeserver:~$ ls -l
total 4
-rw-rw-r-- 1 jupiter jupiter 5 May 28 09:03 file01
jupiter@homeserver:~$
jupiter@homeserver:~$ sudo chown root file01
jupiter@homeserver:~$
jupiter@homeserver:~$ ls -l
total 4
-rw-rw-r-- 1 root jupiter 5 May 28 09:03 file01
jupiter@homeserver:~$
```

Samba File Share Settings

- Samba is named after the Server Message Block (SMB) protocol. SMB is a Microsoft protocol for file and printer sharing across a network
- For more details on any of these commands, be sure to read the man pages from the command line for each command, e.g. **man smbpasswd**, **man pdbedit**, etc...
- Samba log files are saved at /var/log/samba. These might be useful to review if you are having any problems
- Samba uses port 445. Be sure to enable this port in your firewall

Sequence:

1. Install samba

```
sudo apt install samba -y
```

This will install the SAMBA file server to your server

2. Create new config file

```
cd /etc/samba
sudo mv smb.conf smb.conf.ORIGINAL
sudo nano smb.conf
```

Samba comes with a lengthy default configuration file (smb.conf), which lists all of its options. Rather than trying to edit this, it is easier to back it up and make a new one. Here we move into the /etc/samba directory and rename the config file to smb.conf.ORIGINAL. We then use the nano text editor to create and open a new file.

You will need to use sudo when creating the new config file. The config lines to be added are listed on **page 2 below**

3. Restart samba

```
sudo service smbd restart
```

To have Samba use the new config file you must restart it.

4. Set permissions on the /srv directory

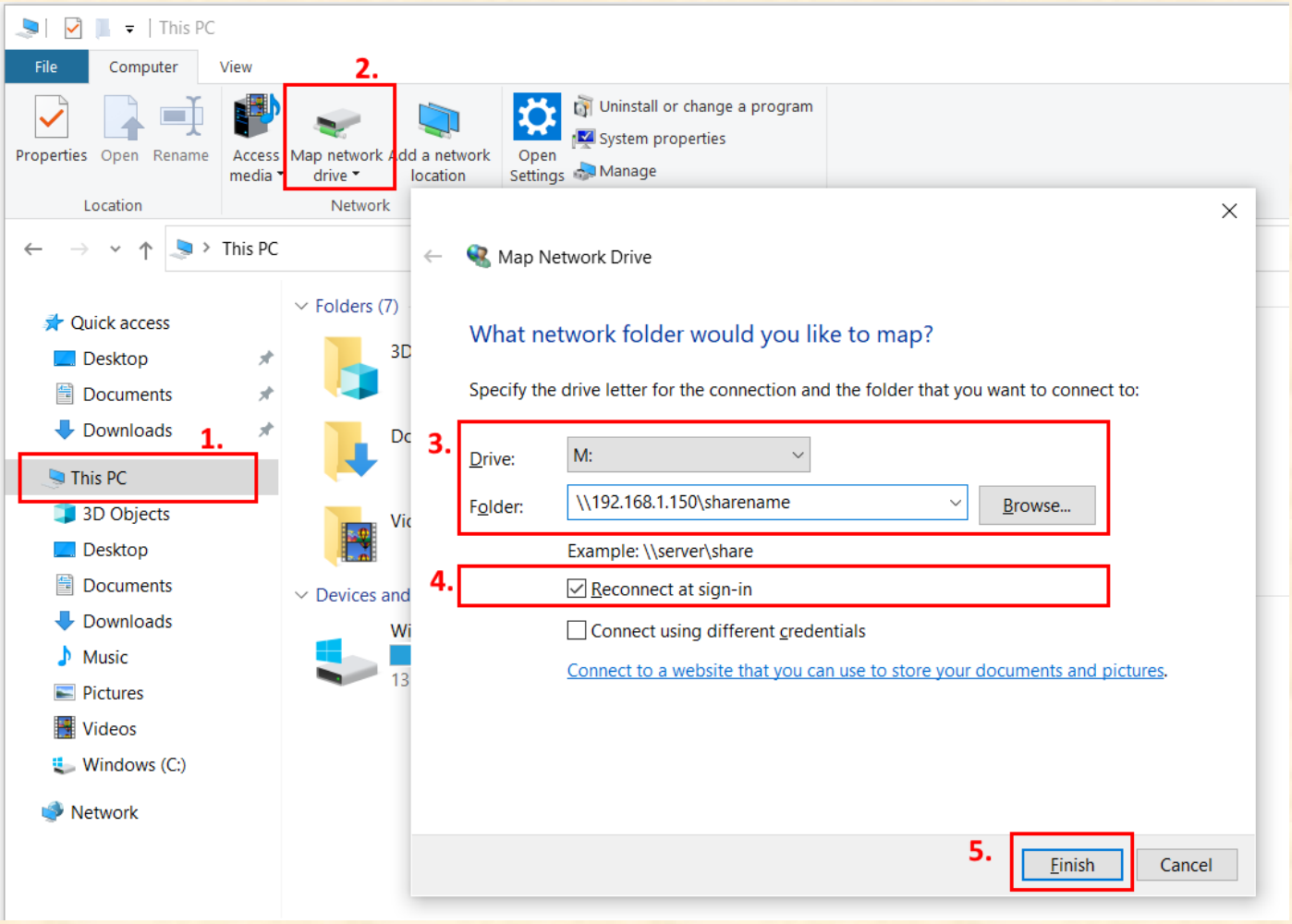
```
sudo chmod 777 /srv
```

In this video, we use /srv as the folder for the file share. By default, this is owned by the root user. To allow the **jupiter** user to create and delete files on it, set the permissions to 777.

5. Connect to the file share from your laptop, desktop or phone

From a Windows 10 laptop:

1. Open File Explorer and click This PC
2. Click Map Network Drive
3. Enter the Drive Letter you want to use and the folder. Be sure to add the \\ before the server IP address
4. Tick the option to reconnect at sign in. This remembers this file share for when you shut down and start up your laptop again
5. Click Finish. The network file share should open straight away for you.



/etc/samba/smb.conf format
[global] map to guest = bad user guest account = jupiter [sharename] path = /srv writable = yes guest ok = yes create mask = 0666 directory mask = 0777 vfs object = recycle recycle:repository = /srv/RecycleBin recycle:keeptree = yes recycle:versions = yes recycle:exclude = *.tmp,*.temp recycle:exclude_dir = RecycleBin recycle:directory_mode = 0777 recycle:subdir = 0777

[global]	These are options that apply to the behavior of the Samba server itself and not to any of the specific shares
map to guest	When set to 'bad user', it allows users to use the file share without having to log in, and they will be assigned the guest account
guest account	This is used to specify the account that guest users should be assigned when connecting to the Samba server. If you remove this entry you can still use the file share, but your files will be assigned a owner account called 'nobody'

[sharename]	Share specific options control the behaviour of the file shares you configure. You can have more than one file share, pointing to different paths for example. You connect to the file share by browsing to \\server_ip_address\sharename, i.e. 192.168.178.150\files
path	The location of the folder on the server being used as the file share.
writable	Required so you can write to the file share, i.e. upload and save files
guest ok	Required so you can access the file share without having to enter a password to log in
create mask	Assigns the permissions to give to files created on or uploaded to the file share
directory mask	Assigns the permissions to folders created on the fileshare
vfs object	Enable the Recycle Bin module
recycle:repository	Set the path/folder for where deleted items will go
recycle:keeptree	If a deleted item is within a folder, that folder will be recreated within the Recycle Bin. This keeps the folder structure, making it easier for you to see where deleted files were stored before deletion
recycle:versions	If two files with the same name are deleted, two versions will be created in the Recycle Bin
recycle:exclude	File types to be excluded. These will not go to the Recycle Bin and will be permanently deleted
recycle:exclude_dir	Folders to be excluded from the Recycle Bin. You need to include your Recycle Bin folder here
recycle:directory_mode	Permissions of the Recycle Bin folder. 777 is required in order to browse into it
recycle:subdir	Permissions of sub-folders created within the Recycle Bin folder

Create New Database Super User

- For security, MySQL & MariaDB do not permit logins to the root user account from web based tools; you can only log in to them from the command line on the server. This prevents attackers from trying to log in to this account by brute force attacking the password
- MySQL/MariaDB commands are not case-sensitive. The commands are typically written in upper-case though, to distinguish them from the options, such as table names or user names, which are also part of a command
- Dont forget the ; at the end of the database commands. If you forget to enter it and press Enter, it will go down to the next line. Simply press ; and hit Enter again

`sudo mysql –u root –p`	From the linux command line, log in to the mysql root account. The database will prompt you for the password
`CREATE USER 'jupiter'@'%' IDENTIFIED BY '2the_Blue_Sky';`	Create a new user (*jupiter*) and give it a password (*2the_Blue_Sky*)

This new user account has no privileges to do anything on the database yet. The next thing is to grant it the necessary privileges. There are several privileges a user account can have:

- **ALL PRIVILEGES** - a full root access to the databases. If no database is specified, it has global access across the system.
- **CREATE** - create new tables or databases
- **DROP** - delete tables or databases
- **DELETE** - delete rows from tables
- **INSERT** - insert rows into tables
- **SELECT** - use the SELECT command to read through databases
- **UPDATE** - update table rows
- **GRANT OPTION** - grant or remove other users' privileges

`GRANT ALL PRIVILEGES ON *.* TO 'jupiter'@'%' WITH GRANT OPTION;`	Grant the user account full privileges to everything on the database, with permission to set up other users' privileges also
`FLUSH PRIVILEGES;`	Finally, reload the privileges for the changes to take effect

- **Note**: this root user will not be used when setting up web applications, it is only for administering the database. Each web application you install gets its own user account with restricted privileges.
- This is a security principle that limits, or contains, the damage an attacker could do if they were to hack your application

---

Other useful commands:

After logging into the database…	
`SELECT user, host FROM mysql.user;`	See the list of users on the database, and what hosts they can log in from: localhost = the servers command line % = any computer, including from across the network
`SHOW GRANTS FOR username;`	Check the privileges for a username

To check if you have MySQL or MariaDB installed (not that it will make a difference – all the commands and how you use it will be the same):

From the Linux command line…	
`sudo mysql --version`	Check the version of the database server
`dpkg -l | grep -e mysql-server -e mariadb-server`	Search the package manager for MySQL or MariaDB
`sudo mysql -u root -p`	Once you log in to the server, the welcome message should say whether it is MySQL or MariaDB

Photo Gallery Installation

1. Download the Gallery Installer

- Go to http://galleryproject.org/ and copy the link for the latest version of Gallery installer

2. Set up folder

- When working from the web directory, your normal user account may not have the permissions to set up the folders, etc.
- If so, start each command with **sudo**, i.e. **sudo wget http://[gallery-installer.zip]**. This will run the command as the root admin user
- Any folders which are created are then owned by root, but this is OK as in the last step you will change the owner to www-data, the web server user, so the web server will be able to upload photos, etc

Command	Description
cd /var/www/html	On the linux command line, change to the web directory
wget http://[gallery-installer.zip]	Use wget to download the gallery installer
unzip gallery-installer.zip	The installer downloads as a zip file. Unzip this to extract the contents.
rm gallery-installer.zip	Once unzipped, remove (delete) the installer zip file as you no longer need it
mv gallery-folder gallery	When extracted from the zip file, the folder of contents will have a version name, i.e. gallery3-3.1.3. Use the move (mv) command to rename this to something which will be easier to type later in your web browser, i.e. gallery
mkdir gallery/var	Make a directory under gallery called var. This is where all of the uploaded photos will be saved
sudo chown –R www-data:www-data gallery	Change the owner of the gallery directory and all of its contents (-R = recursive) to be www-data.

3. Configure PHP

- You need to make some configuration changes to PHP for Gallery to work successfully
- The PHP config file is saved at **/etc/php/7.4/apache2/php.ini**
- The file location depends on the version of PHP installed, e.g. for version 7.3 the location will be **/etc/php/7.3/...**
- The PHP config file and folder are owned by root so you will need to use **sudo** for these commands

1.	**sudo cp php.ini php.ini.BACKUP**	Make a backup copy of the original php config file, called php.ini.BACKUP in case something goes badly wrong
2.	**sudo vi php.ini**	Using the **vi** text editor, open up php.ini for editing. You can also use the **nano** text editor if you prefer Search for and make the following changes: **short_open_tags = On** **post_max_size = 50M** **upload_max_filesize = 50M**
3.	**sudo service apache2 restart**	Restart the Apache web server for the changes to take effect

- short_open_tags: Enable the Short Open Tags feature
- post_max_size: Sets max size of post data allowed. Set it to 50Mb
- upload_max_filesize: The maximum size of an uploaded file. Set it to 50Mb

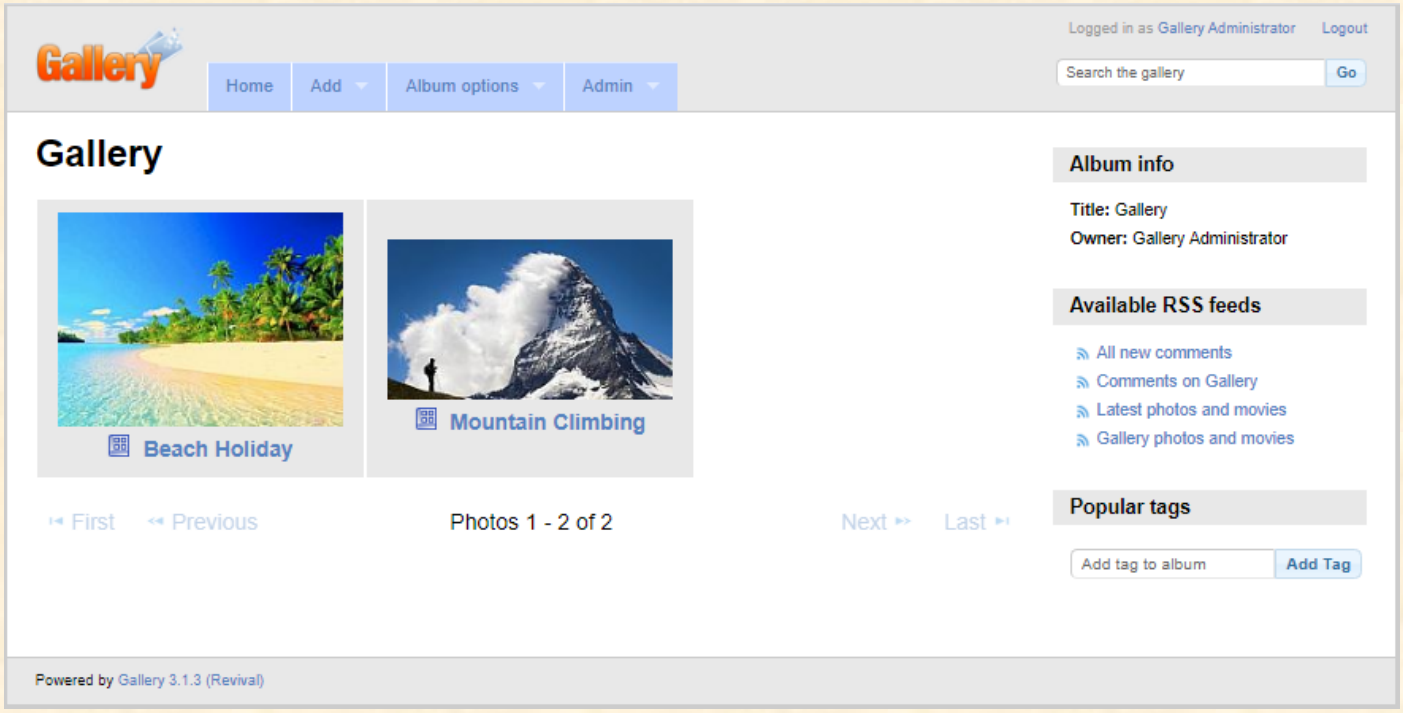
4. Setup the Database

- Using a web browser, go to **http://[your-server-ip]/phpmyadmin,** login and create a new database
- Create a new user account which has access to this database
  **For security reasons, never use the root account when installing a web application, always set up a new user account.

5. Install Gallery

- With your web browser, go to **http://[your-server-ip]/gallery/installer**
- Enter the database name, database user account and password details
- Once the application installs successfully, copy the admin password – you can change this once you log in

Using Gallery to Organise and Manage Your Photos



Creating Photo Albums

- To add albums, photos, etc, you need to be logged in as an admin
- To add an album, go to **Add > Album**
- To add photos to an album, click into the album, and go to **Add > Photos**

Adding Photos from the Server

- If you have photos already saved to your server, these can be added to Gallery automatically
- Go to **Modules** and enable the **Server Add** module
- Then go to **Settings > Server Add** and type in the folder location where the photos are saved
- Go back to the Gallery (click the icon on upper left), and click **Add > Server Add**
- Click the folder name and **Add**

- If you have many photos to add (multiple Gb over many folders), try adding them in batches rather than adding a single overall folder

Adding Movies

- From the Linux command line, install the ffmpeg package: **sudo apt install ffmpeg**
- In the Gallery Admin panel, go to **Settings > Movies** to ensure that it recognises the installed package
- Back in the photo gallery, go into the folder you want to add the movie to, and click **Add > Photos**, and select your movie file

Amazon S3 Backup Script

- Use *nano* or *vi* to create a new empty file called s3_backups.sh using the text in the box below
- Enter the directory to be backed up into the SOURCE variable
- Enter the name of your S3 bucket into the DESTINATION variable

- Save the script and make it executable by running **chmod a+x s3_backups.sh**
- To run the script on a weekly basis, move it to the /etc/cron.weekly directory: **mv s3_backups.sh /etc/cron.weekly**
- Many more options are available in the *s3cmd* documentation. You can view these by running **man s3cmd**

```
#!/bin/bash

#####################################################################
# Variables
#####################################################################

# 1. Server folder to be backed up
SOURCE="/srv/"

# 2. Amazon S3 bucket name
DESTINATION="s3://bucketname"

#####################################################################
# Backup Steps
#####################################################################

#Run the backup
s3cmd sync --delete-removed $SOURCE $DESTINATION
```

s3cmd Commands

- Amazon Web Services (AWS) Simple Storage Solution (S3) stores files in buckets
- s3cmd config file is saved to ~/.s3cfg (~ means the users home directory, e.g. /home/jupiter/.s3cfg)
- s3cmd uses https by default to encrypt the connection & data transfer between you and AWS for backups
- Review man s3cmd to see all available options and commands for the s3cmd command
- Using the synchronise option compares the files in your local server directory and your S3 bucket, based on file name, file size and MD5 checksum (a digital fingerprint) to identify new/deleted files

Flag/ Option	Description
--configure	Run/re-run the s3cmd configuration tool
--dry-run or -n	Only show what should be uploaded or downloaded but don't actually do it.
--recursive or -r	Recursive upload, download or removal
--delete-removed	During syncing, delete the files/objects which exist on the S3 cloud storage bucket but which no longer exist on your local server. Using this ensures your S3 bucket remains a correct mirror of the folder on your server
--human-readable-sizes or -H	Print sizes in human readable form, e.g. 1kB instead of 1234, when using s3cmd ls
--progress	Display progress meter
--stats	Display some additional stats during file transfer
--verbose or -v	Enable verbose output mode

s3cmd command	Description
s3cmd mb s3://BUCKET	Make a new S3 bucket
s3cmd rb s3://BUCKET	Remove an S3 bucket (must be empty)
s3cmd ls [s3://BUCKET[/PREFIX]]	List bucket and objects
S3cmd la	List all object in all buckets
s3cmd put FILE [FILE...] s3://BUCKET[/PREFIX]	Put file into a bucket (upload)
s3cmd get s3://BUCKET/FILE LOCAL_FILE	Get file from a bucket (download)
s3cmd sync LOCAL_DIR s3://BUCKET[/PREFIX] (example: s3cmd sync /srv s3://myfiles)	Synchronise (mirror) a local server folder to your S3 bucket – this will update S3 to match your local folder
s3cmd sync s3://BUCKET[/PREFIX] LOCAL_DIR (example: s3cmd sync s3://myfiles /srv)	Synchronise (mirror) S3 with your server folder - this will update your local folder so it matches what is in your S3 bucket
s3cmd del s3://BUCKET/FILE or s3cmd rm s3://BUCKET/FILE	Delete file from bucket
s3cmd du [s3://BUCKET[/PREFIX]]	Disk usage by buckets
s3cmd info s3://BUCKET[/FILE]	Get various information about buckets or files
s3cmd mv s3://BUCKET1/FILE s3://BUCKET2[/FILE]	Move files/folders between buckets

Uncomplicated Firewall (ufw)

Command	Details
sudo apt install ufw -y	Install the ufw firewall
sudo vi /etc/default/ufw	Disable IPv6, by changing 'IPV6=yes' to 'IPV6=no'
man ufw	Check the UFW man pages for more detailed instructions

1. Add Firewall Rules

- UFW looks to the **/etc/services** file for the corresponding port number for a service name, i.e. http = 80
- You can add a comment to each of your rules by using the comment command after the rule, i.e. **sudo ufw allow http comment 'My web server'**. You will see these comments when you run the sudo ufw status command

Command	Service	Port
sudo ufw allow ssh, or **sudo ufw allow 22**	SSH *Be sure to add this before enabling ufw	22
sudo ufw allow http, or **sudo ufw allow 80**	HTTP Web Server	80
sudo ufw allow https, or **sudo ufw allow 443**	HTTPS Secure Web Server	443
sudo ufw allow 139 & **sudo ufw allow 445**	Samba File Service	139 & 445

2. Check Your Firewalls Status

Command	Details
sudo ufw status	Show the ruleset for your firewall
sudo ufw status verbose	Show the ruleset for your firewall, and also the level of logging. This will also show you the default rules: - deny all incoming traffic unless specifically allowed - allow all traffic going out from your server
sudo ufw status numbered	Show the ruleset for your firewall as a numbered list

3. Enable/Disable the Firewall

Command	Details
sudo ufw enable	Turn on your firewall and begin blocking network traffic ** Be sure to allow SSH traffic (port 22) before doing this, or you will be locked out of your server. You would then need to attach a keyboard and screen to the server to directly administer it
sudo ufw disable	Turn off the firewall. This reduces the security of your server

4. Denying IP Addresses

- You can use the firewall to deny access to specific machines on your network using their IP address

Command	Details
sudo ufw deny from 192.168.178.10	Block traffic from an IP address from connecting to your server
sudo ufw deny 80	Deny any traffic trying to reach port 80. This can also be achieved by deleting any 'allow' rules for port 80, as there is an implicit deny for any traffic which is not explicitly allowed

5. Delete Rules

Command	Details
sudo ufw status numbered	See the numbering of the rules
sudo ufw delete 2	Delete rule number 2

6. Sorting Rules

- UFW lists firewall rules in the order you add them
- Firewalls act upon the first rule which matches or applies to a network connection
- If you have very specific rules (allow traffic on specific port from specific IP), add this above other more general rules

Command	Details
sudo ufw status numbered	See the numbering of the rules
sudo ufw insert 2 allow from 192.168.178.10 to any port 22	Insert a rule into position 2, allowing traffic from 192.168.178.10 over port 22 (SSH traffic) This rule must not exist already in the firewall. If it does, delete it first and re-enter it

7. Firewall Logging

- UFW firewall logs are saved to /var/log/ufw
- These capture a variety of detail of blocked and permitted network connections (depending on logging level), including:
    - The date and time of the event
    - [UFW BLOCK] – the event that was recorded, in this case a blocked connection
    - IN – for incoming connections, this is the network interface (card) which the event occurred on
    - SRC – the IP address of the source of the connection
    - DST – the destination IP which the network packet was being sent to
    - PROTO – the Protocol of the network packet
    - SPT – the source port the network packet came from
    - DPT – the destination port the packet was being sent to

Command	Details
sudo ufw logging on	Turn on firewall logging (default)
sudo ufw logging off	Turn off firewall logging
sudo ufw logging low/medium/high	Set logging level to low (default), medium or high

Update commands

- To update your server often, run sudo apt update && sudo apt upgrade.

sudo apt update	Your computer has a list (like a catalogue) that contains all the available software that the Ubuntu repositories have available. But the available software and versions might change, so this will reach out to the repositories and see what software is available in order to update its local lists. The list of repositories is saved at /etc/apt/sources.list. This will not actually update or install new versions of software itself.
sudo apt upgrade	This will upgrade the installed software on your server, once it knows there is a newer version available. It will not remove any software packages or dependencies which may not be needed any more.
sudo apt dist-upgrade	This will intelligently install or remove software packages and dependencies as needed to upgrade your system.
sudo do-release-upgrade	Upgrade the operating system to the latest version.

- After a while, if you have installed and uninstalled a lot of applications, chances are that your server has a lot of dependant files which you have no use for any longer. These are some commands to get rid of any partial packages, remove unused dependencies and keep your server in good shape overall

sudo apt autoremove	Whenever you install an application using **apt install** the system will also install the software that this application depends on. It is common in Ubuntu/Linux that applications & programs will share the same libraries. When you remove the application/program though the dependency (that the application depended on) will stay on your system. So **apt autoremove** will remove those dependencies that were installed with applications and that are no longer used by anything else on the system.
sudo apt clean	The **clean** command clears out the local repository of downloaded package files. It removes everything except the partials folder and lock file from /var/cache/apt/archives. Use this to free up disk space when necessary, or as part of regularly scheduled maintenance.
sudo apt autoclean	This is also used to clear out the local repository of downloaded package files. The difference though is that **autoclean** only removes package files that can no longer be downloaded from their sources, and are very likely to be useless.

Installing & Removing Software

dpkg --list	List all installed software. Pipe the output into the **less** command in order to browse slowly through the list: dpkg --list | less
sudo apt install [package]	Install a software package
sudo apt purge [package]	Remove the package/program and all of its configuration files
sudo apt remove [package]	Remove a package/program without removing its configuration files. Useful if you plan to reinstall it again in future

System Commands

System	
cat /etc/os-release	Show information about your system release
lsb_release -a	This displays LSB (Linux Standard Base) information about your specific Linux distribution, including version number, release codename, and distributor ID
uptime	Shows how long the system has been up and running for. Also shows the system load averages for the previous 1, 5 and 15 minutes (how busy it has been)
uname -a	This gives you the name of the kernel, the network host name, kernel version number and release level (e.g. 4.15.0-72), kernel release date, CPU type ('x86_64' is a 64bit processor), and operating system name (e.g. GNU/Linux)

CPU	
cat /proc/cpuinfo	Show detailed info about your servers CPU

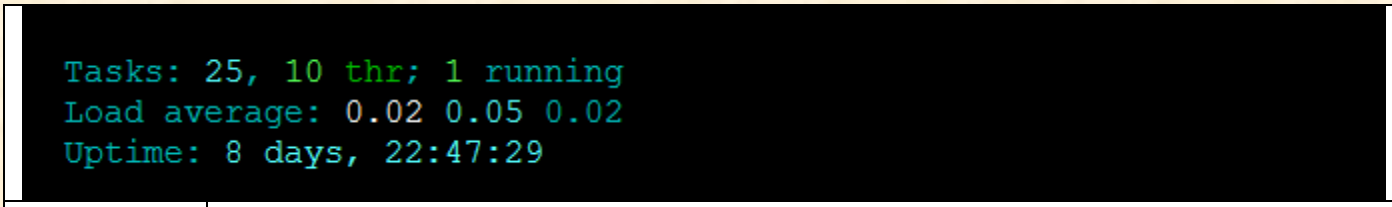
Storage	
sudo fdisk -l	List the partitions on the current disk
mount	This lists the currently mounted filesystems. Use the **sort** command to get a neater output: `mount | sort`
df -h	This displays the amount of available disk space for file systems -h shows the disk sizes in human-readable format, i.e. 1G rather than 1000000

Memory	
cat /proc/meminfo	Show detailed info about your memory (RAM)
free -m	Display the amount of available and used memory in the system Add the -m or -g flags to display figures in Megabytes or Gigabytes, rather than kilobytes (default) **Total**: The total amount of physical RAM installed in your computer. **Used**: This is calculated by Total - (Free+Buffers+Cache). **Free**: The amount of unused memory. Because the Used column contains the Buffers and Cache figures, it's not uncommon for perfectly functioning Linux systems to have very little RAM listed as 'free' **Shared**: Memory that is used by the tmpfs file system. **Buff/cache**: Memory used for buffers and cache. **Available**: This is an estimation of the memory that is available to service memory requests from applications, any other functioning software within your computer, such as your graphical desktop environment and Linux commands.

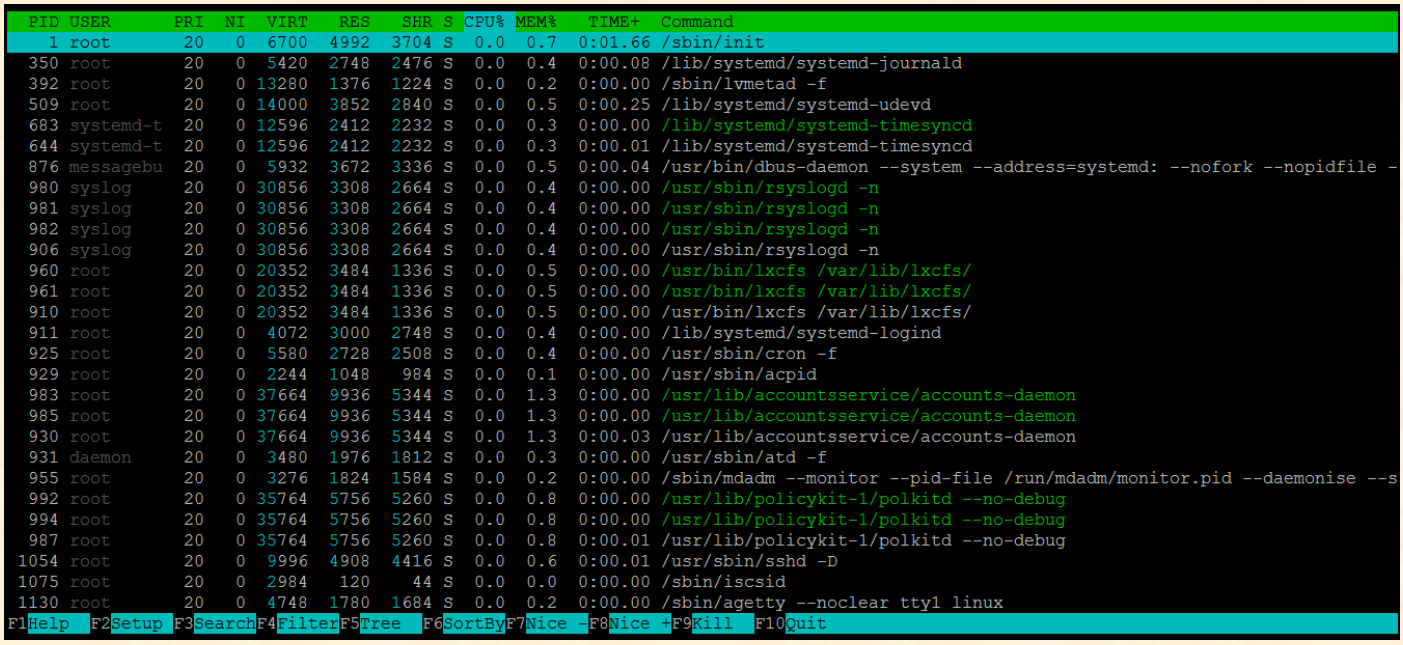
Network	
hostname	View the network hostname for your server
ip addr show	View the IP address of your server. Use *ip -br -c addr show* to get a shortened (i.e. brief) and coloured output

Htop Section Guide

CPU	Indicates how busy your server is, using % of CPU time Blue bars: Low-priority processes Green bars: Normal priority processes Red bars: Kernel (High) priority processes
Memory	Green bars: Memory being used by processes Blue bars: Buffered memory (for data being sent to the CPU) Yellow bars: Cached memory (disk files being cached to memory for faster access)
Swap	Swap memory is space on your hard disk which the system writes tasks to in order to free up running memory, and if the tasks have not been used in some time. Swap memory in use is indicated by a red bar



Tasks	The number of tasks is the number of processes (25 in the image above) and how many are running (1). It also shows how many threads are present (10). Threads are CPU features whereby multiple processes are run at the same time. You can remove the threads from the display by pressing **Shift + H**. Alternatively you can also display kernel threads also by pressing **Shift + K**.
Load Average	These are the average loads on your CPU over the last 1, 5 and 15 minutes. You can get an idea of how hard, and for how long, your processor has been working.
Uptime	The uptime is a count of how long the server has been running for, since it was last booted up.



PID	This is the Process ID, or the unique value the server gives to each process in order to identify and keep track of it. This can be useful when you want to manually kill, or stop, a particular process
User	The user under which the process is running. You can view the processes for specific users by pressing '**u**' and then selecting a user
NI	The 'nice' value of a process, which indicates its priority. Process 'niceness' ranges from -20 to +20, with the lowest nice values having the highest priority, and the highest nice values having the lowest priority. The default priority level is 0. If you enable kernel threads (**Shift + K**) you will see many high priority (-20) processes.
RES	'Resident': how much physical memory (RAM) a process is using, measured in kilobytes
S	The current status of a process. Running processes are marked with an '**R**'. Sleeping processes, that is processes which are not currently using CPU computing resources, are marked with an '**S**'. There other process states, including zombie '**Z**', but these will be the most common
CPU%	The percentage of processor time used by a process
MEM%	The percentage of physical memory (RAM) used by a process
TIME+	How much processor time a process has used
Command	The name of the command that started the process. You can display/hide the path to the command by pressing '**p**'

Function Keys

F1	Open the help screen. You can also get this by pressing '?'
F2	Open the setup screen. You can change the view of the Meters along the top of the htop screen (use the spacebar to select the different display options), or even add additional meters, such as battery, hostname or a clock. You can also change the display and colour scheme of htop, and change the columns that are displayed in the main process window
F3	Search the htop display for a particular process name
F4	Filter the display for a process name. All other processes will be filtered out. Press F4 again and Escape to clear the filter
F5	Show a Tree view of the processes. Very often, processes will spawn 'child' processes as part of their operation. This Tree view shows the Parent process and the Child processes it generated
F6	Choose which column to sort the process display by
F7	Decrease the nice value, i.e. increase the processes priority. This can only be done when htop is run under sudo
F8	Increase the nice value, i.e. decrease its priority
F9	Kill, or stop, a process. You are given a choice of kill signals to use, the default should work fine
F10	Quit htop. Pressing 'q' at any time will also exit htop

Other Htop Tips

Tagging Processes	If the display is moving a lot, or you want to make a particular process stand out, pressing the **spacebar** colours that process in yellow. To tag a parent process and all of its children, press '**c**'. You can use the spacebar to tag and untag as many processes as you want. To untag them all in one go, press **Shift + U**
Quick Sorting	F6 will call up a menu of columns for you to sort the display by. Alternatively there are keys for quick sorting by CPU% (**Shift + C**), MEM% (**Shift + M**) and TIME+ (**Shift + T**). You can also choose to invert any sorted display by pressing **Shift + I**, if you wanted to see the processes with lowest memory usage, for example, rather than the highest