**+LINUX**

**FILES & DIRECTORIES**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| **ls** | List files and directories | ls, ls -l, ls -a |
| **pwd** | Print current directory | pwd |
| **cd** | Change directory | cd /home/karan |
| **mkdir** | Create directory | mkdir newdir |
| mkdir -p | Create nested directories | mkdir -p /a/b/c |
| **rmdir** | Remove empty directory | rmdir olddir |
| rm -r | Remove directory with contents | rm -r mydir |
| **touch** | Create an empty file | touch notes.txt |
| **cat** | Display file content | cat file.txt |
| more, less | View file page by page | less bigfile.txt |
| head | View first lines of file | head -n 10 file.txt |
| tail | View last lines of file | tail -n 5 log.txt |
| **cp** | Copy files/directories | cp file1.txt file2.txtcp -r dir1/ dir2/ |
| **mv** | Move or rename files/directories | mv old.txt new.txt |
| **rm** | Remove file | rm file.txt |
| find | Search for files/directories | find /home -name "\*.log" |
| locate | Fast search (needs updatedb) | locate config.yaml |
| stat | View file metadata | stat file.txt |
| file | Identify file type | file image.png |
| basename | Extract file name from path | basename /home/user/doc.txt |
| dirname | Extract directory path | dirname /home/user/doc.txt |
| tree | Show directory tree (install with yum install tree) | tree /var/www |

**USER MANAGEMENT**

|  |  |
| --- | --- |
| **Command** | **Description** |
| whoami | Shows the current logged-in user |
| id | Shows UID, GID, and groups of current user |
| adduser <username> | Adds a new user (Debian-based systems) |
| **useradd <username>** | Adds a new user (general Linux) |
| **passwd <username>** | Sets or changes the user's password |
| **usermod -aG <group> <user>** | Adds a user to a group |
| **userdel <username>** | Deletes a user account |
| su - <username> | Switches to another user |
| sudo | Runs commands as root or another user |

**GROUP MANAGEMENT**

|  |  |
| --- | --- |
| **Command** | **Description** |
| **groupadd <groupname>** | Creates a new group |
| **groupdel <groupname>** | Deletes a group |
| groupmod -n <newname> <oldname> | Renames a group |
| gpasswd -a <user> <group> | Adds a user to a group |
| **gpasswd -M <user1,…,usern> <group>** | Adds multiple user to a group |
| gpasswd -d <user> <group> | Removes a user from a group |
| groups <username> | Lists groups of a user |
| getent group | Lists all groups on the system |

**PERMISSIONS**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| **ls -l** | View file/directory permissions | ls -l myfile.txt |
| **Chmod** | Change file/directory permissions | chmod 755 script.sh |
| chmod u+x | Add execute permission to user | chmod u+x run.sh |
| chmod g-w | Remove write permission from group | chmod g-w report.txt |
| chmod o=r | Set read-only for others | chmod o=r file.txt |
| chmod -R | Change permissions recursively | chmod -R 755 /var/www |
| **Chown** | Change file owner and/or group | chown karan:devops file.txt |
| chown karan | Change only the owner | chown karan test.sh |
| chown :devops | Change only the group | chown :devops test.sh |
| Chgrp | Change group ownership | chgrp devops file.txt |
| **Umask** | Set default permission mask for new files | umask 022 |

**ACL**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| **getfacl <file>** | View ACL of a file or directory | getfacl myfile.txt |
| **setfacl -m u:<user>:<perm> <file>** | Set ACL permission for a user | setfacl -m u:karan:rw myfile.txt |
| **setfacl -m g:<group>:<perm> <file>** | Set ACL permission for a group | setfacl -m g:devs:r-- myfile.txt |
| setfacl -x u:<user> <file> | Remove ACL for a user | setfacl -x u:karan myfile.txt |
| **setfacl -b <file>** | Remove all ACL entries (reset to default permissions) | setfacl -b myfile.txt |
| setfacl -m d:u:<user>:<perm> <dir> | Set default ACL on a directory (applied to new files) | setfacl -m d:u:karan:rw /data |

**HARD LINK & SOFT LINK**

**🔹 1. Create a Soft Link (Symbolic Link)**

**ln -s <target\_file> <link\_name>**

**🧪 Example:**

**ln -s /home/karan/data.txt mysoftlink.txt**

**✅ mysoftlink.txt points to data.txt**

**🔹 2. Create a Hard Link**

**ln <target\_file> <link\_name>**

**🧪 Example:**

**ln /home/karan/data.txt myhardlink.txt**

**✅ myhardlink.txt is another name pointing to the same inode as data.txt.**

**🔎 3. Check Link Type and Inode**

**ls -li**

**PARTITION**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| **Lsblk** | List all block devices | lsblk |
| **fdisk /dev/sdX** | Manage MBR partitions on disk sdX | fdisk /dev/sda |
| parted /dev/sdX | Manage GPT partitions | parted /dev/sdb |
| Cfdisk | Text-based partition manager (interactive) | cfdisk /dev/sdc |
| **mkfs.ext4** | Format partition as ext4 | mkfs.ext4 /dev/sda1 |
| mkfs.xfs | Format partition as xfs | mkfs.xfs /dev/sdb2 |
| Blkid | Display UUID and type of partitions | blkid |
| **Mount** | Temporarily mount a partition | mount /dev/sda1 /mnt |
| **Umount** | Unmount a mounted partition | umount /mnt |
| df -h | Show disk space usage (human-readable) | df -h |
| du -sh /path | Show size of a directory | du -sh /home |
| lsblk -f | Show filesystems & mount points | lsblk -f |

SWAP

**Create Swap Using a Partition**

Useful when you have a dedicated disk/partition (like /dev/sdb1)

**🧱 Step-by-Step:**

**🔹 1. Create partition with fdisk**

fdisk /dev/sdb

* Press n → new partition
* Press t → change type → enter 82 (Linux swap)
* Press w to write and exit

**🔹 2. Format it as swap**

mkswap /dev/sdb1

**🔹 3. Enable the swap**

swapon /dev/sdb1

**🔹 4. Make it permanent (in fstab)**

blkid

Copy the UUID of /dev/sdb1, then:

nano /etc/fstab

Add:

pgsql

UUID=your-uuid-here none swap sw 0 0

**🔹 5. Verify**

swapon --show

free -h

**LVM**

**# Create LVM-compatible partition**

**sudo fdisk /dev/sdb → type 8e for LVM**

**# Initialize Physical Volume**

**sudo pvcreate /dev/sdb1**

**# Create Volume Group**

**sudo vgcreate myvg /dev/sdb1**

**# Create Logical Volume**

**sudo lvcreate -L 1G -n mylv myvg**

**# Format and mount**

**sudo mkfs.ext4 /dev/myvg/mylv**

**sudo mkdir /mnt/lvmdata**

**sudo mount /dev/myvg/mylv /mnt/lvmdata**

**# Add to fstab for auto-mount**

**/dev/myvg/mylv /mnt/lvmdata ext4 defaults 0 0**

**PROCESS**

**ps –** current session

**ps -ef –** show process of entire session

**init :** process start at boot

**ip a :** check the ip (192.xx.xx)

**PUTTY tool:** create new session using the ip address.

|  |  |  |
| --- | --- | --- |
| **Command** | **Purpose** | **Example** |
| ps aux | List all running processes (detailed) | ps aux |
| **ps -ef** | Another format to list all processes | ps -ef |
| `ps aux | grep nginx` | Find specific process |
| **top** | Real-time CPU/memory usage and processes | top |
| htop | Interactive process viewer (better top) | htop *(requires install)* |
| **kill PID** | Gracefully terminate process by PID | kill 1234 |
| **kill -9 PID** | Forcefully kill process | kill -9 1234 |
| pkill name | Kill process by name | pkill firefox |
| pkill -9 name | Force kill process by name | pkill -9 python |
| nice -n N command | Start process with specific priority | nice -n 10 script.sh |
| renice -n N -p PID | Change priority of running process | renice -n 5 -p 1234 |
| command & | Run command in background | sleep 100 & |
| jobs | List background jobs | jobs |
| fg %job | Bring job to foreground | fg %1 |
| bg %job | Resume job in background | bg %1 |
| nohup command & | Run even after logout | nohup long.sh & |
| watch -n N cmd | Repeat a command every N seconds | watch -n 2 ls -l |

**PACKAGE**

Yum 🡪 centos

Apt 🡪 Ubuntu

Rpm

**5 services:**

* httpd
* nfs
* samba
* DNS
* **yum install httpd**
* **systemctl status httpd** – check installed package status
* **systemctl start httpd** – to active the package

**tar:**

* tar -cvf karan.tar tar1 tar2 tar3 🡪 (CREATE .TAR PACKAGE WITHOUT COMPRESSION)
* tar -tf karan.tar 🡪 (DELETE .TAR PACKAGE)
* tar -xvf karan.tar -C /etc 🡪 (UNPACK & MOVE FILES INSIDE THE .TAR TO NEW LOCATION)
* tar -cvzf karan.tar.gz tar1 tar2 tar3 🡪 (GZ COMPRESSION)
* tar -cvjf karan.tar.bz2 tar1 tar2 tar3 🡪 (BZ2 COMPRESSION)

**nmtui (for ip refresh)**

HTTPD:

* yum install httpd
* systemctl start httpd
* systemctl enable httpd
* systemctl status httpd
* firewall-cmd –list-all
* firewall-cmd –add-service=http –permanent
* firewall-cmd –reload
* cd /var/www/html (create new file inside)
* cd /etc/httpd/conf
* vi httpd.conf

<virtualhost ip\_address>

documentroot /var/www/html

servicename host\_name (use hostname to view the name)

</virtualhost>

* httpd -t (check the syntax)
* systemctl restart httpd (restart)

File transfer (from one vm to another vm):

SCP :

* scp /file\_path root@ip\_address:/destination\_path 🡪 (scp -r for directory)

RSYNC :

* rsync /file\_path root@ip\_address:/destination\_path 🡪 (rsync -r for directory)
* (rsync -avz archive,verbose,zip)

NFS (NETWORK FILE SYSTEM):

* systemctl start nfs-server
* systemctl enable nfs-server
* systemctl status nfs-server
* firewall-cmd –add-service=nfs –permanent
* firewall-cmd –reload
* create new directory
* vi /etc/exports

/sharednfs1 192.168.1.100(rw)---

* exportfs -arv
* firewall-cmd –reload

(both client & server side) :

* firewall-cmd --permanent --add-service=mountd --to mount in client system
* firewall-cmd --permanent --add-service=rpc-bind -----to connect client and server machine

**Samba (SMB Server) [SHARE FILES FROM LINUX TO WINDOWS]**

* systemctl start smb
* systemctl enable smb
* systemctl status smb
* firewall-cmd –add-service=samba –permanent
* firewall-cmd –reload
* create new directory
* create new user
* add 🡪 smbpasswd -a <user\_name>
* vi /etc/samba/smb.conf

[FOLDER\_NAME]

Path=/smb\_user

Valid users= smb\_user, smb\_user2

Readable= Yes

Browseable= Yes

* firewall-cmd –add-service=nmb –permanent
* firewall-cmd –reload
* systemctl restart nmb
* sestatus
* setenforce 0
* check on windows 🡪 [\\ip\_address\](file:///\\ip_address\)
* net use \* /delete 🡪 (remove all existing smb sessions)
* net use \\192.168.3.84\smbshare /user:sambauser

**BIND (DNS Server)**

* yum install bind
* systemctl start named
* systemctl enable named
* systemctl status named
* firewall-cmd –add-service=dns –permanent
* firewall-cmd –reload
* vi /etc/named.conf
* cd /var/named 🡪 add forward.example.com & reverse.example.com
* systemctl restart named 🡪 (to check all are working correctly)
* vi /etc/resolv.conf

**DHCP Server**

* systemctl start dhcpd
* systemctl enable dhcpd
* systemctl status dhcpd