**Devops**

**Linux:**

**pwd** - print working directory.

**mkdir**- used to create new folder

**mkdir -v file\_name**- “-v(verbose)” is used to print message.

**mkdir -p file\_name1/filename\_2**

🡪 - p means parent.

🡪 We can use v and p together also.

🡪 It based on your requirement.

**sudo su -** 🡪 It will switch to root user, and it will point to the root user home directory and will load the root user configurations.

**sudo su 🡪** It will switch to root user, and it will not point to the root user home directory and will not load the root user configurations.

**yum install package\_name -y**

🡪 this is the syntax to install any package.

🡪 -y is optional. If you keep -y, then it will install the package directly.

**touch file\_name** 🡪 It is used to create an empty file.

🡪 If the file\_name already exists, then data will not override, updated timestamp will set.

**mkdir -m 700 file\_name 🡪** It will create a file with some permissions.

**ls -lrt**

**🡪** ls means list the directory content

🡪 l means longlisting

🡪 r means reverse order

🡪 t access time

**ls -la** 🡪 It will show all files including hidden files.

**ls -lh** 🡪 human readable.

**Inode**:

Inode is data structure it can store the file/directory information. But it doesn’t contain filename.

**ls -li 🡪** It will show all files with inode number.

🡪 Each file and directory will have a unique inode number.

cd ~ (OR) cd🡪 It will take you to user home directory. User home directory will be **/home/ec2-user**.

cd - 🡪 It will take you to previous directory.

**rmdir** **file\_name** 🡪 It is used to delete an empty directory.

**rm -rf file\_name** 🡪 It will remove non-empty directory. Here r means recursively and f

means forcefully.

**vi file\_name** 🡪 The file will open in command mode. It doesn’t matter whether the file has existed or not. If the file has not existed, then it will create and open in command mode.

**nano ->**

**find 🡪** It is used to search and locate the list of files and directories based on conditions.

**find . -type f -empty 🡪** It will search all the empty files.

**find . -type f \( -name "\*.java" -o -name "\*.txt" \)**

will get all java and txt files

**find . -iname java.txt 🡪** It will search the java.txt file in the current directory and it will ignore the case.

**umask dir file**

**root 0022 755 644**

**user 0022 755 644**

Base permission for dir is : 0777-0022=0755

Base permission for file is: 0666-0022=0644

r ----read 4

w -----write 2

x -----execute 1

**umask** 🡪 This command is used to change the base permissions of file and dir.

**chmod ->** This command is used to change the permissions of a particular file or dirf.

Syntax : chmod 222 file\_name

chmod -R 222 dir\_name -> it will apply for sub directories too.

chown owner\_name file\_name -> it will change the owner of the file if the user is available.

/ 🡪 root directory

/root 🡪 root user home directory

**chgrp group\_name file\_name 🡪**  It will change the group name of that file.

**cat /etc/group 🡪** It will print all the group names available in the server.

**cat\*.txt ->** it will print all text files

**cat -n file\_name**-> it will print with line numbers

**head file\_name** -> It will print first 10 lines

**head file\_name** **-15 file\_name** -> it will print 15 lines

**tail file\_name**-> it will print last 10 lines of the file

**sed -n “100p” filename** -> it will print 100th line only.

**sed -n “100,105p” filename** -> It will print 100 to 105th line

**more file\_name –** It displayscontents of a file one screen at a time. When the screen is filled, it waits for user input to proceed further.

 **Pressing Enter**: Advances by one line.

 **Pressing Space**: Advances by one screen (next page of content).

 **Pressing Ctrl+F**: Also advances by one screen (like Space).

 **Pressing q**: Quits the more command and exits back to the shell.

**LESS**  
**Backward Navigation**: You can scroll backward, which is not possible with more.

**Search Functionality**: Provides both forward and backward search options.

**Efficiency**: Handles large files more efficiently without loading the entire file

**cp file\_name destination\_path 🡪** use -r for folders

**mv file\_name destination\_path**

**wc file\_name 🡪**  It will print the count in the format of lines words characters.

**wc -w \*.txt** 🡪 It will print count of word from all txt files.

**Ln 🡪 Link**

|  |  |
| --- | --- |
| **Hard Link** | **Soft Link** |
| Of and lf will have same properties | Both will not have same properties |
| **ln file\_name** | **ln -s file\_name** |
| Both will contain same inode number. | Both will contain different inode number. |
| **Hard Link** will create only for files | **SoftLink** will create for both files and dir. |
| Even If we delete the original file, we can access hard link. | we cannot access if we delete original file. |

**sed “s/red/blue/g” demo.txt ->** The command replaces the first occurrence of the word **"red"** with **"blue"** on each line of demo.txt and outputs the modified content to the terminal. However:

* It does **not** modify the file itself unless the -i (in-place) flag is used.

**sort file\_name**

It will print text in ascending order

-r will be used for descending order

**sort -r BasicProgram.java | tr 'a-z' 'A-Z'  
-** It will print all the text in descending order with capital letters.

**stickybit 🡪** It is a permission bit

**chmod +t directory\_name** 🡪 It will set sticky bit to the specified directory

**ls -ld directory\_name 🡪** Verifying the sticky bit.

**chmod -t directory\_name 🡪** It will remove the sticky bit.

**chmod 1777 file\_name 🡪** adding sticky bit.

Others will not be able to delete the file if we use sticky bit.

**System Resource Commands**  
**who 🡪** It will print the users who currently logged in.

**-H** It will add headers  
**users 🡪** It will print the users who currently logged in.

**w** 🡪 It will display how CPU is working.

**whoami 🡪** It display current user info who gave this command.

**date🡪** It will display system date and time.

**man timedatectl** We can display list of time zones.

🡪It will give more information about the command.

**Info command\_name**

**command\_name --help**

**df 🡪** report file system disk usage.

**du**(dist usage) file**\_**name**🡪** estimate file space usage

-sh

**hostname 🡪** show or set the system host name.

hostname host\_name

-i 🡪 server ip address

**nproc 🡪 I**t will display no. of CPU cores.

**whereis is file\_name 🡪** It will display the path of the file\_name.

**systemctl list-unit-files 🡪** It will display all the services.

**ps 🡪** it will display process ids

-ef

**zip -r file\_name.zip path**

**unzip file\_name.zip**

**unzip -d /path/ file\_name**

**tar -cvf filename.tar path**

**tar -xvf file\_name.tar**

**useradd user\_name**

**cat /etc/group**

**cat /etc/passwd**

**passwd user\_name**

**cat /etc/shadow**

**chage user\_name**

**groupadd group\_name**

**usermod -g group\_name user\_name 🡪** User will add to specified group.

**lid -g group\_name –>**  It will print usernames related to specified group.

**groups user\_name 🡪** It will print user’s group name.

**usermod -L user\_name 🡪** It is used to lock the username

**usermod -U user\_name** 🡪 It is used to unlock the username.

**su 🡪 switch user**

**Normal user – NU 🡪 password is required**

**Nr – Ru 🡪 password is required**

**Ru -Nu 🡪 password is not required**

**sudo**

**ssh(Secure shell)**

**visudo 🡪** /etc/sudoers file in editor mode

**sudo userdel -r user\_name 🡪** Without r user will delete but directory will not be deleted.

**ssh username@ip\_address**

**vi /etc/ssh/sshd\_config**

**service sshd restart**

**scp(Secure copy) 🡪** It is used copy the files from one server to another server.

**scp file\_name username@ip\_address:/tmp/**

**scp -r directory\_name username@ip\_address:/tmp/**

**free -h**

**dmidecode**

**Crontab: install cronie package to get crontab.**

It is used to scheduling the job.

touch /etc/cron.allow

crontab -l 🡪 to check configured jar files

crontab -e 🡪 update

crontab -r 🡪 remove

**Crontab format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Minute** | **Hour** | **Day Of Month** | **Month** | **Day Of Week** | **Script** |
| 0-59 | 0-23 | 1-31 | 1-12 / Jan-Dec | 0-6 / Sun-Sat | user/bin/find |

> 🡪 redirect the standard output

>> 🡪 Append the standard output

**File Descriptors:**

**0 🡪 std input**

**1 🡪 std output**

**2 🡪 std error**

**\*/1 \* \* \* \* /file\_path > path\_name.log 2>&1**

**🡪** If you are working on crontab, crond.service should be in running mode.

**service crond.service status 🡪** To check the status of crond.service

**service crond.service start**

**wall 🡪** it is used to send messages to all logged in users.

**write user\_name msg 🡪** It is used to send a message to specified user.

**megn n 🡪** It is used to stop the messages from normal users.

**megn y 🡪** enable the messages

**grep name file\_name**

**grep name file\_name | cut -d “=” -f2 🡪** The text will be separated by '=', and we will get the second parameter since we are using f2.

**cal**

**-3**

**Year\_name**

**Month\_name year\_name**

**clear**

**wget link**

**curl -o file\_name link**

**ls > lsoutput.log**

**ls | tee lsoutput.log**

**script**

**ping**

**telnet**

**history**

**-c**

**watch date**

**-n 5**

**restart**

**shutdown**

**SHELL SCRIPT**