GETTING STARTED
WITH
ADVANCED FILTERS

SED:

- SED stands for **stream editor** and it can perform lots of functions on file like searching, find and replace, insertion or deletion.
- Most commonly used for substitution or for find and replace.
- It is a multipurpose filter command.

SYNTAX: **$sed [options] <file-name>**

→ To replace a string in a first occurrence of sample file:
$sed "s/linux/aws/" sample

→ To replace a string globally in a file:
$sed "s/linux/aws/g" sample

→ To ignore case sensitive in a file:
$sed "s/linux/linux/gi

→ Replacing the 2nd occurrence of a pattern in a line:
$sed "s/linux/aws/2" sample

→ Replacing from 2nd occurrence to all occurrences in a line including case:
$sed "s/linux/aws/3gi" sample

→ Printing only the replaced lines:
$sed -n "s/linux/aws/p" sample

→ Replacing two words at a time in a file:
$sed -e "s/linux/aws/g" -e "s/unix/azure/g" sample

→ Printing only a third line of a file:
$sed -n "3p" sample

→ Printing only 3rd to 5th lines of a file:
$sed -n "3,5p" sample

→ Printing only 3rd and 5th line of a file:

$sed -n "3p
>5p" sample

→ Deleting a third line of a file:

$sed "3d" sample

→ Deleting a last line of a file:

$sed "$d" sample

→ To delete lines from range:

$sed "3,6d" sample

→ To delete pattern matching lines:

$sed "/linux/d" sample

GREP:

- GREP is used to **print lines matching a regular expression.**
- Use GREP to search for lines of text that match one or many regular expressions, and outputs only the matching lines.
- grep, egrep, fgrep - print lines that match patterns

SYNTAX: **$grep [options] [pattern] [files]**

→ To print a "cloud" pattern in a file:

$grep cloud filename

→ To print a "cloud" pattern in multiple files:

$grep cloud file1 file2 file3

→ To print a "cloud" pattern in all files in a current directory:

$grep cloud *

→ To print a "cloud" pattern in all files including directories:

$grep -R cloud *

→ To print "cloud" pattern lines including line numbers:

$grep -n cloud filename

→ To print non pattern matching lines:

$grep -v cloud filename

→ To print "cloud" pattern lines, ignore case sensitive:

$grep -i cloud filename

→ To print lines starting with "cloud" pattern lines:

$grep ^cloud filename

→ To print lines ending with "cloud" pattern lines:

$grep cloud$ filename

→ To print empty lines:

$grep "^$" filename

PIPING (|)

- A pipe is a form of redirection that is used for filtering**.**
- It is used to combined two or more commands. Here the **standard output** of one command to the **standard input** of another command.

SYNTAX: **$command_1 | command_2 | command_3……| command_n**

→ Counting number of files and directories:

$ls -l | wc -l

→ Counting top ten lines in a file:

$cat aws| head |wc -l

→ Print only "cloud" pattern lines:

$cat aws | grep cloud

→ Replace a word in a file:

$cat sample | sed "s/unix/cloud/g"

TEE:

- It reads the **standard input** and **writes** it to **both the standard output and one** or more files.

SYNTAX: **$tee [options] <filename>**

→ Read input file data and writes to multiple files and on screen also:

$cat sample | tee sample1 sample2
$cat sample1
$cat sample2

→ Count number of lines from input file:

$wc -l aws | tee aws1 aws2
$cat aws1
$cat aws2

→ Write a file and append output to a file:

$echo "This is a Raju…" | tee aws3
$cat aws3

LOCATING FILES & DIRECTORIES:

- Two popular commands for locating files on Linux are **LOCATE** and **FIND.**
- Depending on the size of your file system and the depth of your search, the find command can sometime take a long time to scan all the data.

LOCATE:

- **Searching for a file or directory** can be easier with the locate command.
- The locate command uses a database (**updatedb)** to check for files and directories.
- Opposite to find, the locate command doesn't search the entire filesystem, but looks through a regularly updated file database in the system.

SYNTAX: **$locate [OPTION]... PATTERN...**

→ To update a database:

$updatedb

→ To find a file with name sample:

$locate sample

→ To display the just 5 results from our queries:

$locate -n 5 "*.txt"

→ To ignore case sensitive:

$locate -i *sample.txt*

FIND:

- It can be used to **find files and directories** and perform **subsequent operations** on them.
- The find command is used in various ways, such as file, directory, name, size, user…. etc.

SYNTAX: **$find [options] [paths] [expression]**

→ Search for a sample file with a specific name under root:

$find / -name sample

→ Search for a sample file in a /usr location:

$find /usr -name sample

→ Search all files with extension ".cfg" files:

$find / -name "*.cfg"

→ Search for empty files and directories:

$find / -empty

→ To search only empty files:

$find / -type f -empty

→ To search only files name with "cloud"

$find / -type f -name cloud

→ To search only directories name with "cloud"

$find / -type d -name cloud

→ To search more than 20MB files:
$find / -size +20M

→ To search less than 20MB files:
$find / -size -20M

→ To search between +20MB and -30MB files:
$find / -size +20M -size -30M

→ Search 644 files and directories:
$find /root -perm 644

→ Search 751 files and directories:
$find / -size 751

→ To find last 3 days access files:
$find /home -atime +3

→ To find before last 3 days access files:
$find /home -atime -3

→ To find exactly last 3rd day access files:
$find /home -atime 3

→ Search last 3 days modification files:
$find / -mtime +3

→ To find last 3 days creation files:
$find /root -ctime +3

→ Seach last 3 minutes modification files:
$find /root -mmin +3

→ Search before last 3 minutes access files:
$find . -amin -3

→ Search files what are created by user raju:

$find / -user raju

→ Search files what are created by group Developers:

$find / -group developers

→ Find and delete a file with confirmation.

$find /root -name sample -exec rm -i {} \;

→ Search for text within multiple files.

$find / -type f -name "*.md" -exec grep 'foo' {} \;

→ To find and operate on files:

$find / -type f -name bar -exec chmod 777 {} \;

→ Find files by content:

$find / -name “*.txt” -exec grep -Hi cloud {} \;