**GREP command**

Grep is a filter command, which is used to search a string/pattern in a given file.

Generally, A Pattern is a string and it represents more than one string.

**Grep command syntax:**

* grep [options] “string/pattern” file/files
* cat file | grep [options] “string/pattern”
* echo “some text” | grep [options] “string/pattern”

# **Basic options: -i, -w, -v, -o, -n, -c, -A, -B, -C, -r, -l, -h**

* grep “string/pattern” file/files
* grep [options] “string/pattern” file/files
  + **-i** : To ignore case for matching/searching
  + **-w** : To match a whole word
  + **-v** : To display the lines which are not having given string or text
  + **-o** : To print/display only matched parts from matched lines
  + **-n** : To display the matched line numbers
  + **-c** : To display matched number of lines
  + **-A** : To display N lines after match (ex: grep –A 3 “string” file)
  + **-B** : To display N lines before match
  + **-C** : To display N lines around match
  + **-r** : To search under current directory and its sub-directory
  + **-l** : To display only file names
  + **-h :** To hide file names

# **Advanced Options: -f, -e, and –E**

**-f :** Takes string/pattern from a file and search in a required file. (only one search pattern per line in a file)

Ex : grep -f 2.txt 1.txt

**-e :** To search multiple strings/patterns

Ex : grep -e 'line' -e 'lines' 1.txt

**-E :** To work with multiple patterns

**Syntax: grep -E[options] “pattern” file/files**

Ex : grep -Ew ‘line’ 1.txt

# **Rules to create patterns:**

* **xy|pq** : Matches for xy or pq

Ex : grep -E 'line|lines' 1.txt

* **^xyz** : Matches for the lines which are starting with “xyz”

Ex : grep - E "^hai" 3.txt

* **xyz$** : Matches for the lines which are ending with “xyz”

Ex : grep -E '!$' 3.txt

* **^$** : Matches for the lines which are empty

Ex: : grep '^$' 3.txt

* **\** : To list the string with special characters. ( Ex: \^ \$)

Ex: : grep -E '\^' 3.txt

* **.** : Matches any one character

Ex : grep -E 'h.i' 3.txt

* **\.** : Matches exactly with .

Ex : grep -E 'h..' 3.txt

* **\b** : Match the empty string at the edge of word

Ex : grep -E 'ai\b' 3.txt

* **?** : The preceding character is optional and will be matched zero and once.

Ex : grep -Ew 'ab?' 3.txt

* **\*** : The preceding character will be matched zero and more times

Ex : grep -Ew 'ab\*' 3.txt

* **+** : The preceding character will be matched ones and more times

Ex : grep -Ew 'ab+' 3.txt

* **[xyz]** : Matches for the lines which are having x or y or z

Ex : grep -E 'hai|please' 3.txt

* **[a-d]** : It is equal to [abcd] and matches for the lines which are having a/b/b/d

Ex : grep -E [a-b] 3.txt

* **[a-ds-z]** It is equal to [abcdstuvwxyz]

Ex : grep -E [a-bd-f]

* **^[abc]** : Matches for the lines which are starting with a/b/c

Ex : grep -E "^[h|p]" 3.txt

* **[^abc]** : Matches for the lines which are not starting with a/b/c

Ex : grep -E "[^h|p]" 3.txt

* **{N}** : The preceding string matched exactly N times

Ex : grep -E 'ab{3}' 3.txt

* **{N,}** : The preceding string matched N or more times

Ex : grep -E 'ab{2,}' 3.txt

* **{N,M}** : The preceding string matched at least N times but not more than M times

Ex : grep -E 'ab{2,4}' 3.txt

* [[:alnum:]] : Alphanumeric characters.
* [[:alpha:]] : Alphabetic characters
* [[:blank:]] : Blank characters: space and tab.
* [[:digit:]] : Digits: ‘0 1 2 3 4 5 6 7 8 9’.
* [[:lower:]] : Lower-case letters: ‘a b c d e f g h ij k l m n o p q r s t u v w x y z’.
* [[:space:]] : Space characters: tab, newline, vertical tab, form feed, carriage return, and space

# **Tasks:**

1.write a pattern to list only directories?

ls -ltr | grep -E 'd'

2. Write a pattern to list only files?

ls -ltr | grep -E '-'

3. Find the server ipv4 from the file

grep -E '\b[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\b' ips.txt