

log parsing cheat sheet

LOG PARSING CHEAT SHEET

 GREP	GREP allows you to search patterns in files. ZGREP for GZIP files. \$grep <pattern> filelog	-n: Number of lines that matches -i: Case insensitive -v: Invert matches -E: Extended regex -C: Count number of matches -l: Find filenames that matches the pattern
 NGREP	NGREP is used for analyzing network packets. \$ngrep -I filepcap	-d: Specify network interface -i: Case insensitive -X: Print in alternate hexdump -t: Print timestamp -I: Read pcap file
 CUT	The CUT command is used to parse fields from delimited logs. \$cut -d '*' -f 2 filelog	-d: Use the field delimiter -f: The field numbers -C: Specifies characters position
 SED	SED (Stream Editor) is used to replace strings in a file. \$sed s/regex/replace/g	s: Search -e: Execute command g: Replace -n: Suppress output d: Delete W: Append to file
 SORT	SORT is used to sort a file. \$sort footxt	-o: Output to file -C: Check if ordered -r: Reverse order -u: Sort and remove -n: Numerical sort -f: Ignore case -k: Sort by column -h: Human sort
 UNIQ	UNIQ is used to extract uniq occurrences. \$uniq footxt	-c: Count the number of duplicates -d: Print duplicates -i: Case insensitive
 DIFF	DIFF is used to display differences in files by comparing line by line. \$diff foolog barlog	How to read output? a: Add #: Line numbers c: Change <: File 1 d: Delete >: File 2
 AWK	AWK is a programming language use to manipulate data. \$awk {print \$2} foolog	Print first column with separator '*' \$awk -F '{print \$1}' /etc/passwd Extract uniq value from two files: awk FNR==NR {a[\$0]++; next} !(\$0 in a) {f1.txt f2.txt}

From:

<https://matebcn.eu/> - miguel angel torres egea

Permanent link:

<https://matebcn.eu/linux:log:log-parsing>

Last update: 18/04/2023 12:23

