**Using Input/Output Redirection and Analyzing Text on RHEL 8**

**1. Generate a server information custom log file and append to it.**

cloud\_user@server1: ~ $ date

Mon May 31 21:28:12 UTC 2021

cloud\_user@server1: ~ $ hostname

server1

cloud\_user@server1: ~ $

cloud\_user@server1: ~ $ date

Mon May 31 21:28:12 UTC 2021

cloud\_user@server1: ~ $ df -hT

Filesystem Type Size Used Avail Use% Mounted on

devtmpfs devtmpfs 1.8G 0 1.8G 0% /dev

tmpfs tmpfs 1.9G 0 1.9G 0% /dev/shm

tmpfs tmpfs 1.9G 17M 1.9G 1% /run

tmpfs tmpfs 1.9G 0 1.9G 0% /sys/fs/cgroup

/dev/xvda2 xfs 20G 13G 7.3G 64% /

tmpfs tmpfs 374M 4.0K 374M 1% /run/user/1001

{ echo " " ; echo "==== date on hostname ====" ; df -hT ; } > server1-health.txt

Change the server health log file to contain a double redirect to ensure that the initial and subsequent outputs are not overwritten:

{ echo " " ; echo "==== date on hostname ====" ; df -hT ; } >> server1-health.txt

2. Generate a log file of a user's owned file in a directory structure, excluding errors. 3. Number the log file for reference and sort it with various options. 4. Generate a listing of the top level directories under /etc with space usage and then sort the listing from smaller to larger space usage.

Using Input/Output Redirection and Analyzing Text on RHEL 8

Introduction

A Linux system administrator's life is filled with text files and data. It's easier to make sense of this mountain of data using the grep utility combined with the power of regular expressions. Input/output redirection is also an important skill for Linux system administrators, who can use it to create log files, quiet error messages, manipulate and sort data, and more. In this hands-on lab, we will explore input/output redirection as well as using grep and regular expressions to analyze and manipulate text.

Solution

Log in to cloud server 1 using the credentials provided:

ssh cloud\_user@<PUBLIC\_IP\_ADDRESS>

Use Input/Output Redirection

Create a server health log file that contains a sequential number of outputs with the hostname, date and time, and a simple header:

{ echo " " ; echo "==== date on hostname ====" ; df -hT ; } > server1-health.txt

After hitting Enter, use cat server1-health.txt to see the output.

Run the command to create the server health log file twice more.

Look at the output by using cat server1-health.txt again to see how many times the command was run.

Change the server health log file to contain a double redirect to ensure that the initial and subsequent outputs are not overwritten:

{ echo " " ; echo "==== date on hostname ====" ; df -hT ; } >> server1-health.txt

Run the command two more times and inspect the output again by using cat server1-health.txt.

Find all the files in the home directory owned by the cloud user by using find -user cloud\_user. You will notice a long listing of file names, along with two errors at the end.

Clear the screen using clear.

Generate clean lines of output without the two errors by using find -user cloud\_user 2> /dev/null.

Find out how many clean lines of output there are by using find -user cloud\_user 2> /dev/null | wc -l.

Save the output into a text file by using find /home -user cloud\_user 2> /dev/null > cloud\_user-files.txt.

Ensure the file names are in the text file by using cat cloud\_user-files.text.

Number the list of file names and save them in another text file by using nl cloud\_user-files.txt > numberedfiles.txt.

Ensure the numbered file names are in the text file by using nl cloud\_user-files.txt > numberedfiles.txt.

Sort the numbered lines by using first sort numberedfiles.txt and note how the lines are sorted.

Do a numeric sort by using sort -n numberedfiles.txt.

Generate a list of directories and file names that includes their space usage:

find /etc -maxdepth 1 -iname "\*.\*" -exec du -sh {} \; > etc-space-usage.txt ; less etc-space-usage.txt

Sort the listing so that the list is sorted from smallest to largest files and directories:

find /etc -maxdepth 1 -iname "\*.\*" -exec du -sh {} \; > etc-space-usage.txt ; sort -h etc-space-usage.txt

Use Grep and Regular Expressions to Analyze Text

Find all the files owned by the cloud\_user in the /home directory that contain the word "file" by using find /home -user cloud\_user.

Find all the files owned by the cloud\_user in the /home directory structure with the word "file" and pipe them to grep by using find /home -user cloud\_user | grep -i file.

Use the -exec feature of find to find the word "file" in the files themselves by using find /home -user cloud\_user -exec grep -i file {} \;.

Find out how many lines of output were found:

find /home -user cloud\_user -exec grep -i file {} \; | wc -l`

Count how many lines were generated without errors:

find /home -user cloud\_user -exec grep -i file {} \; 2> /dev/null | wc -l

Utilize the grep command directly to find the word "file" inside all files in the cloud\_user's home directory by using grep -ir file /usr/share/doc/zip.

Add a total count of the output at the bottom of the list by using grep -ir file /usr/share/doc/harfbuzz ; !! | wc -l.

Run a case-sensitive search for only the word "file" in lowercase:

grep -ir file /usr/share/doc/zip ; !! | wc -l

Create a text file containing all the search results for the word "file":

grep -irn file /usr/share/doc/zip > ~/grepoutput.txt

Check the text file output and note the line numbers added to show where the word "file" was found in each file by using cat grepoutput.txt.