Linux and Bash Command Cheat Sheet: The Basics

Getting information

# return your user name  
whoami

# return your user and group id  
id

# return operating system name, username, and other info  
uname -a

# display reference manual for a command  
man top

# get help on a command  
curl --help

# return the current date and time  
date

Monitoring performance and status

# list selection of or all running processes and their PIDs  
ps  
ps -e

# display resource usage  
top

# list mounted file systems and usage  
df

Working with files

# copy a file  
cp file.txt new\_path/new\_name.txt

# change file name or path  
mv this\_file.txt that\_path/that\_file.txt

# remove a file verbosely  
rm this\_old\_file.txt -v

# create an empty file, or update existing file's timestamp  
touch a\_new\_file.txt

# change/modify file permissions to 'execute' for all users  
chmod +x my\_script.sh

# get count of lines, words, or characters in file  
wc -l table\_of\_data.csv  
wc -w my\_essay.txt  
wc -m some\_document.txt

# return lines matching a pattern from files matching a filename pattern - case insensitive and whole words only  
grep -iw hello \\*.txt

# return file names with lines matching the pattern 'hello' from files matching a filename pattern  
grep -l hello \\*.txt

Navigating and working with directories

# list files and directories by date, newest last  
ls -lrt

# find files in directory tree with suffix 'sh'  
find -name '\\*.sh'

# return present working directory  
pwd

# make a new directory  
mkdir new\_folder

# change the current directory: up one level, home, or some other path  
cd ../  
cd ~ or cd  
cd another\_directory

# remove directory, verbosely  
rmdir temp\_directory -v

Printing file and string contents

# print file contents  
cat my\_shell\_script.sh

# print file contents page-by-page  
more ReadMe.txt

# print first N lines of file  
head -10 data\_table.csv

# print last N lines of file  
tail -10 data\_table.csv

# print string or variable value  
echo "I am not a robot"  
echo "I am $USERNAME"

Compression and archiving

# archive a set of files  
tar -cvf my\_archive.tar.gz file1 file2 file3

# compress a set of files  
zip my\_zipped\_files.zip file1 file2  
zip my\_zipped\_folders.zip directory1 directory2

# extract files from a compressed zip archive  
unzip my\_zipped\_file.zip  
unzip my\_zipped\_file.zip -d extract\_to\_this\_direcory

Performing network operations

# print hostname  
hostname

# send packets to URL and print response  
ping www.google.com

# display or configure system network interfaces  
ifconfig  
ip

# display contents of file at a URL  
curl <url>

# download file from a URL  
wget <url>

Bash shebang

#!/bin/bash

Pipes and Filters

# chain filter commands using the pipe operator  
ls | sort -r

# pipe the output of manual page for ls to head to display the first 20 lines  
man ls | head -20

Shell and Environment Variables

# list all shell variables  
set

# define a shell variable called my\_planet and assign value Earth to it  
my\_planet=Earth

# display shell variable  
echo $my\_planet

# list all environment variables  
env

# environment vars: define/extend variable scope to child processes  
export my\_planet  
export my\_galaxy='Milky Way'

Metacharacters

# comments  
# The shell will not respond to this message

# command separator  
echo 'here are some files and folders'; ls

# file name expansion wildcard  
ls \*.json

# single character wildcard  
ls file\_2021-06-??.json

Quoting

# single quotes - interpret literally  
echo 'My home directory can be accessed by entering: echo $HOME'

# double quotes - interpret literally, but evaluate metacharacters  
echo "My home directory is $HOME"

# backslash - escape metacharacter interpretation  
echo "This dollar sign should render: \$"

I/O Redirection

# redirect output to file  
echo 'Write this text to file x' > x

# append output to file  
echo 'Add this line to file x' >> x

# redirect standard error to file  
bad\_command\_1 2> error.log

# append standard error to file  
bad\_command\_2 2>> error.log

# redirect file contents to standard input  
$ tr “[a-z]” “[A-Z]” < a\_text\_file.txt

# the input redirection above is equivalent to  
$cat a\_text\_file.txt | tr “[a-z]” “[A-Z]”

Command Substitution

# capture output of a command and echo its value  
THE\_PRESENT=$(date)  
echo "There is no time like $THE\_PRESENT"

Command line arguments

./My\_Bash\_Script.sh arg1 arg2 arg3

Batch vs. concurrent modes

# run commands sequentially  
start=$(date); ./MyBigScript.sh ; end=$(date)

# run commands in parallel  
./ETL\_chunk\_one\_on\_these\_nodes.sh & ./ETL\_chunk\_two\_on\_those\_nodes.sh

Scheduling jobs with Cron

# open crontab editor  
crontab -e

# job scheduling syntax  
m h dom mon dow command  
*minute, hour, day of month, month, day of week*  
\* means any

# append the date/time to file every Sunday at 6:15 pm  
15 18 \* \* 0 date >> sundays.txt

# run a shell script on the first minute of the first day of each month  
1 0 1 \* \* ./My\_Shell\_Script.sh

# back up your home directory every Monday at 3 am  
0 3 \* \* 1 tar -cvf my\_backup\_path\my\_archive.tar.gz $HOME\

# deploy your cron job  
*Close the crontab editor and save the file*

# list all cron jobs  
crontab -l

This line declares a variable called toBackup, which is an array.  
An array contains a list of values, and items can be appended to arrays using the syntax:

1. 1
2. myArray+=($myVariable)

Copied!

When you print (or echo) an array you will see its string representation, which is simply all of its values separated by spaces:

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. $ declare -a myArray
8. $ myArray+=("Linux")
9. $ myArray+=("is")
10. $ myArray+=("cool!")
11. $ echo ${myArray[@]}
12. Linux is cool!

Practice exercises

Before you begin, ensure you're in your home directory:

cd ~

pwd

1. Problem:

Display your username

Click here for Hint

Use the whoami command

Click here for Solution

whoami

2. Problem:

View the kernel version

1/25/23, 4:22 PM Skills Network Labs

https://labs.cognitiveclass.ai/v2/tools/cloud-ide?ulid=ulid-ea6bcfc56cef40bdf235c6630ce9252a708d3c48 7/8

Click here for Hint

Click here for solution

uname -r

3. Problem:

Display the number of lines in the /etc/passwd file.

Click here for Hint

use the wc command with right option.

Click here for Solution

wc -l /etc/passwd

4. Problem:

Display the lines that contain the string 'not installed' in /var/log/bootstrap.log.

Click here for Hint

use the grep command.

Click here for Solution

grep "not installed" /var/log/bootstrap.log

5. Problem:

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250ENSkillsNetwork/labs/Bash%20Scripting/top-sites.txt contains most popular websites. Find out all the websites that have the word

org in them.

Click here for Hint

use the wget command to download the file.

use the grep command to search

Click here for Solution

wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt

grep org top-sites.txt

Alternative Solution

6. Problem:

Print the first 7 lines of top-sites.txt

Click here for Hint

use the head command with the correct arguments

Click here for Solution

head -n 7 top-sites.txt

7. Problem:

Print the last 7 lines of top-sites.txt

Click here for Hint

use the tail command with the correct arguments

Click here for Solution

tail -n 7 top-sites.txt

8. Problem:

Zip the file top-sites.txt into a file called top-sites.zip

1/25/23, 4:22 PM Skills Network Labs

https://labs.cognitiveclass.ai/v2/tools/cloud-ide?ulid=ulid-ea6bcfc56cef40bdf235c6630ce9252a708d3c48 8/8

Previous

Click here for Hint

Click here for Solution

zip top-sites.zip top-sites.txt

9. Problem:

Print the first three characters of each line from top-sites.txt

Click here for Hint

Click here for Solution

cut -c -3 top-sites.txt

10. Problem:

Print details of the eth0 internet adapter

Click here for Hint

Click here for Solution

ifconfig eth0

A shell is an interactive user interface

You can use shell commands for navigating and working with files and directories, and to zip and unzip files

You can use the “curl” and “wget” commands to display and download files from URLs

The “echo” command prints string or variable values

The “cat” and “tail” commands display file contents

You can get user information with the "whoami" and "id" commands

You can check system disk usage using the "df" command

The “ls” command lists all files and directories contained within a specified directory tree

The “cd” command allows you to navigate directories

The “touch” command allows you to create a file or update its last–modified timestamp

The “mkdir” command creates directories and “rmdir” deletes empty directories

You can determine line, word, and character counts with “wc”

You can use “grep” to get the lines of a file matching your desired criteria

The “tar” command decompresses and unpacks a “tar.gz” archive

You can view network configuration with “hostname” and “ifconfig”