Exercises

Data Engineering Diploma

Content developed by: WeCloudData Academy

Linux Commands All Data Engineers Should Know

Objective: Gain hands on working knowledge of various Linux command line use cases

1. **Go through and perform each exercise. If you get stuck, don't be afraid to use the internet for help.**
2. **Some of the questions have a *Challenge* exercise after the main. Try to complete these questions too for an extra challenge.**

Linux Class Exercises

1: Navigating the File System

Navigate to the `/var/log` directory and list all files.

Challenge: Try to do this with one line of commands.

2: Listing Hidden Files

In your home directory, find all hidden files in *long* form. (Hidden files start with a dot).

3: Listing files in time modified order

1. In your home directory, list all files in descending order by time (most recently modified first).
2. Now list all files in ascending order (most recently modified last).
    - **Hint**: Be careful on this one. I suggest using the `man ls` command to try to figure it out if you have trouble.

4: Installing and Uninstalling Packages on Ubuntu

1. Install the package `tree`, then use it to display the directory structure of your home directory.
2. Uninstall `tree`.
3. Install the package `tldr`, then try to use it similar to how you use `man`.
4. Run `man ls`. Read it a bit and then exit by pressing `q`
5. Now run `tldr ls`. Read it. See the difference?
    - Some people find `tldr` to be easier to understand than `man`, thought it is not as complete. So keep this installed and use it if you want.

5: Understanding and Changing Permissions

1. Create a file named `test.sh` using the `touch` command. Can this file be executed? How can you tell?
2. Change its permissions so that only the user can read and execute, but not write.
3. Verify the changes with an appropriate command.

6: Creating and Using Environment Variables

1. Create an environment variable named `MY_NAME` with your name as its value.
2. Print its value to standard output(the terminal).

7: Using Nano/Vim

Open a new file named `diary.txt` with nano or vim. Write about your day, save the file, and exit the editor.

Challenge: Vim is much harder to use and you may not even know how to close it. If you want the added challenge try this in Vim. Look up some basic Vim commands online or see the hints for the basic commands you will need

8: Execution Permissions

1. Create a new script called `runme.sh` that when run simply prints `This is a script` to the terminal.
2. Change its permissions to make it executable and run it.

Challenge: There are two ways to change permissions, using numbers and using letters. Try to complete this exercise using both methods.

9: Executing Files

Write a simple bash script named `greet.sh` that prints "Hello, !" to the terminal Execute the script.

You set the value for MY_NAME in question 4. If you are not getting the results you should, then try to debug. Look at the hints if you need to

Challenge: Try to do this exercise in the terminal without opening and writing directly into the file

10: Using Wget

Use `wget` to download the contents of `http://example.com` and save it to a file named `example.html`.

11: Finding Working Directory and Printing Variables

Write a script that prints the current working directory and the value of the `PATH` environment variable.

12: Making and Deleting a Directory Tree with One Command Each

1. Use a single command to create the following directory tree: `~/Documents/LinuxClass/Project1`.
2. Then, delete the entire `Project1` directory without affecting the `LinuxClass` directory.

13: Determining Running Processes in Terminal

1. Display all currently running processes in the current terminal.

2. Now open two new terminal windows and display the running processes again. What's the difference?
3. Direct the output to a file named `processes.txt` in your home directory.
4. Now close the two new terminal windows you just opened, and then open 2 more again.
5. Again, display the running processes and direct the output to the same file **without** overwriting the previous contents.

14: Identifying Processes Using a Specific Port

Pretend you are going to run an application that will require an open port, preferably port 7000. Use `lsof`.

1. First, check to see if port 7000 is open. Try to check status of port 7000 only and no other ports.
2. Next, what is the status of **all** ports that are currently open.
3. Run the command to answer this and direct the output to a file `openports.txt`.

15: Safely Killing a Process

1. Start a background process, like a simple `ping` command to `example.com`.
2. Then, identify its process ID (PID) and safely terminate it using its PID.

16: Using Top and Ps

1. Run the `top` command.
2. Now quit the running command.
3. Then, use the `ps aux` command to display all running processes and redirect the output to a file named `all_processes.txt`.
4. What is the difference between `ps` and `ps aux` and `top`?

17: Monitoring System Resources

Monitor the system resources for 2 minutes and identify the top three processes consuming the most CPU.

18: Persisting Environmental Variables

Recreate your `MY_NAME` variable as you did before, but this time create it in such a way that even if you close your terminal or shut down your computer, the value will still persist.

Next, create a file called `my_info.txt` which will print the following information on separate lines when run.

1. Your name, using the `MY_NAME` variable.
2. Your computer name.
3. Your username.

Note that none of this should be hard coded. You should use variable references for each of these lines

Solutions to "Linux Commands All Data Engineers Should Know"

Try to answer on your own first. If you need help, use the internet and/or the hints file. If you are still stuck here are the solutions

1: Navigating the File System

```
cd /var/log
ls -la
```

In one line:

```
cd /var/log && ls -la
```

2: Listing Hidden Files

```
ls -la
```

3: Listing files in time modified order

`ls -lt` or `ls -lta`
bonus : `ls -ltr(a)` note: must use the -t along with the -r. Using -r alone will not accomplish the task.

4: Understanding and Changing Permissions

```
touch test.txt
chmod 600 test.txt
ls -l test.txt
```

5: Creating and Using Environment Variables

```
export MY_NAME="YourName"
echo $MY_NAME
```

6: Using Nano/Vim

For Nano:
To open `nano diary.txt`
To close Ctrl+O, Enter, Ctrl+X
For Vim:
To open `vim diary.txt`
To close :wq

7: Execution Permissions

```
echo 'echo This is a script' > runme.sh
```

Using letters:

```
chmod +x runme.sh
```

Using numbers:

```
chmod 755 runme.sh
./runme.sh
```

8: Executing Files

`vim` or `nano greet.sh`

```
chmod +x greet.sh
./greet.sh
```

***For an added challenge, try to do this exercise in the terminal without opening and writing directly into the file*

```
echo "echo Hello, \$MY_NAME!" > greet.sh
```

9: Using Wget

```
wget -O example.html http://example.com
```

10: Installing and Uninstalling Packages on Ubuntu

```
sudo apt-get install tree
tree ~
sudo apt-get remove tree
```

11: Finding Working Directory and Printing Variables

```
echo "Current Directory: $(pwd)"
echo "Path Variable: $PATH"
```

12: Making and Deleting a Directory Tree with One Command Each

```
mkdir -p ~/Documents/LinuxClass/Project1
rm -r ~/Documents/LinuxClass/Project1
```

13: Determining Running Processes in Terminal

```
ps
```

Open new terminals and repeat the command

The difference is that it shows additional running processes because of the additional terminals

Open two more terminals and repeat the command

```
ps >> ~/processes.txt
```

If the students use a single > then it will overwrite the previous contents

14: Identifying Processes Using a Specific Port

```
lsof -i :7000
lsof -i -P | grep -i "listen" > openports.txt
```

15: Safely Killing a Process

```
ping example.com
```

In another terminal:

```
ps
kill <PID for PING>
```

16: Using Top and Ps

```
top
q

ps aux > all_processes.txt
```

17: Monitoring System Resources

```
top
```

The top three processes appearing in top are consuming the most CPU. the `top` command lists processes in order of CPU usage

18: Persisting Environmental Variables

Edit ~/.bashrc or ~/.profile and add `export MY_NAME="YourName"` to the bottom of the file.
Source the file or restart the terminal.

```
echo "$MY_NAME" > my_info.txt
echo "$HOSTNAME" >> my_info.txt
echo "$USER" >> my_info.txt
```

Chapter
Week 03 - Client Project
>
Chapter
Week 04 - Linux, AWS and Docker
>
Chapter overview
Sunday - Linux
[Lecture Material] Linux
[Lab] Exercise: Bash Commands
[Lab] Mini Project: Riyadh Climate Data - Cron Job
Monday - AWS Intro
[Lecture Material] AWS Intro
[Lab] AWS Account Setup
[Lab] Workshop AWS EC2
[Lab] Workshop S3
Tuesday - Lambda
[Lecture Material] Lambda
[Lab] Mini Project: Lambda
Wednesday - Docker Basics
[Lecture Material] Docker Intro
[Quiz] Docker Commands Quiz
[Lab] Software Installation: Docker
[Lab] Exercise: Basic Docker Commands
[Lab] Workshop: Install Zepplin with Docker
Thursday - Docker Compose and Dockerfile

[Lecture Material] Docker Compose and dockerfile

[Lab] Mini Project: Dockerfile

[Lab] Exercise: Bash Commands