Hands-On Lab: Informational, File, and Directory Commands

Estimated time needed: **40** minutes

Objectives

In this lab, you will be introduced to the use of basic Unix commands related to the following categories:

* Informational commands
* File and Directory Management Commands
* Access control commands

About Skills Network Cloud IDE

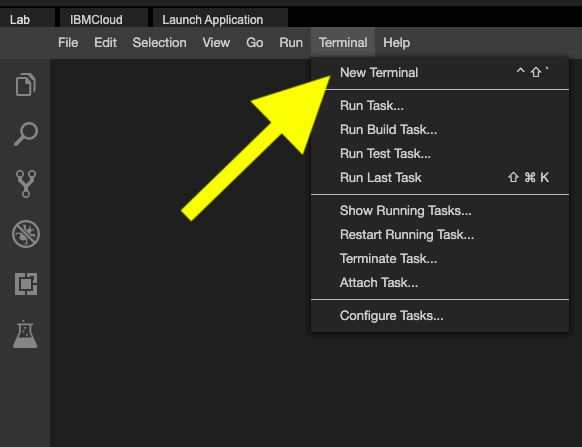
Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, you will be using the Cloud IDE based on Theia.

Important notice about this lab environment

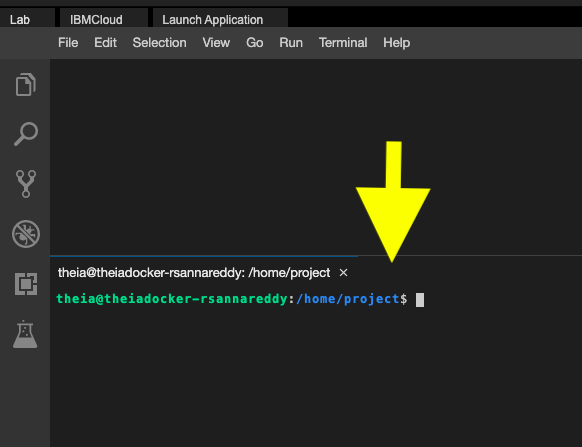
Please be aware that sessions for this lab environment are not persisted. Thus, every time you connect to this lab, a new environment is created for you and any data or files you may have saved in a previous session will be lost. To avoid losing your data, plan to complete these labs in a single session.

Setup

Open a new terminal by clicking on the menu bar and selecting **Terminal**->**New Terminal**, as in the image below.



This will open a new terminal at the bottom of the screen as seen below.



You can run the commands provided in the following excercises in your newly opened terminal. You can copy the code to your clipboard if you like by clicking on the little copy button on the bottom right of each codeblock, and then paste it on the command line.

Exercise 1 - Informational Commands

In this exercise, you will familiarize yourself with useful commands for providing system and user information.

**1.1. Display the name of the current user**

**whoami**

**whoami**

It will display the user name as theia. You are logged into this lab as theia.

You can get a list of currently logged in users using the command who. But this command doesn't work in the theia environment yet.

**1.2. Get basic information about the operating system**

**uname**

By default the command prints the kernel name.

**uname**

You will see Linux printed in the output.

Using the -a opton prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.

uname -**a**

**1.3. Obtain the user and group identity information**

**id**

This command displays the user id and group id information of the current user.

id

It will display the uid (user id) and gid (group id) for the user theia.

**1.4 Get available disk space**

**df**

The df command is used to display available disk space. Entering

**df**

It will display available disk space in 512-byte blocks. To get available disk space in a "human-readable" format, enter:

**df -h**

**1.5. View currently running processes**

**ps**

The ps command lists each processes that is currently running and its PID (process id).

**ps**

However, the output only contains the processes that are owned by you.

By using the -e option, you can display all of the processes running on the system. The includes processes owned by other users.

**ps -e**

**1.6. Get information on the running processes and system resources**

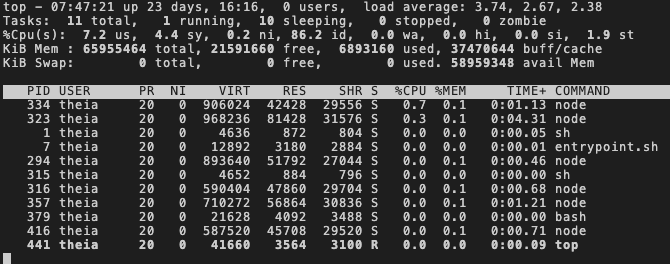
**top**

The top command provides a dynamic, real-time view of your system.

It shows summary information of the system and a table of more detailed information related to the processes or threads which are currently running and managed by the kernel. This includes information related to cpu and memory usage per process.

top

Here is a sample output.



When you start top, you'll be presented with the following elements on the main top screen.

1. Summary area - shows information like system uptime, number of users, load average, and overall memory usage
2. Column header - attribute names
3. Task area - displays the data for each process, or PID

The output keeps refreshing until you press q or Ctrl+c.

If you want to exit automatically after a specified number of repetitions, use the -n option as follows:

**top** -n 10

You can press the following keys while top is running to sort the table:

| **Key** | **Sorts by** |
| --- | --- |
| M | Memory Usage |
| P | CPU Usage |
| N | Process ID (PID) |
| T | Running Time. |

For example, you can find out which process is consuming the most memory by entering shift + m.

**1.7. Display Messages**

**echo**

The echo command displays the given text on the screen. For example, entering:

echo "Welcome to the linux lab"

prints Welcome to the linux lab.

These special characters help you better format your output.

| **Special Character** | **Effect** |
| --- | --- |
| \n | start a new line |
| \t | insert a tab |

Use the -e option of the echo command when working with special characters. For example:

echo -e "This will be printed \nin two lines"

**1.8. Display date and time**

**date**

The date command displays the current date and time.

date

It has several options which help you display the current date and time in your favourite format.  
For example, the following command displays the current date in mm/dd/yy format.

date "+%D"

Here are some of the popular format specifiers that you can try out:

| **Specifier** | **Explanation** |
| --- | --- |
| %d | Display the day of the month (01 to 31) |
| %h | Displays the abbreviated month name (Jan to Dec) |
| %m | Displays the month of year (01 to 12) |
| %Y | Displays the four-digit year |
| %T | Displays the time in 24 hour format as HH:MM:SS |
| %H | Displays the hour |

date "+%"

**1.9. View the Reference Manual For a Command**

**man**

The man command displays the user manual for any command that you provide as its argument.

For example, to see the manual page for the ls command, enter:

man date

Scroll through the command's manual to find any info you may need. When you're done, press qto quit.

Excercise 2: File and directory management commands

In this exercise, you will practice using commands for managing files and directories.

**2.1. Get the location of the current working directory**

**pwd**

When working in a Linux terminal, you will always be working from a directory. By default, you will start in your home directory. To get the absolute path of your current working directory, enter:

pwd

**2.2. List the files and directories**

**ls**

To list the files and directories in the current directory, enter:

**ls**

If your directory happens to be empty, ls will not return anything.

The following command will list the many binary and executable files which are present in your /bin directory.

**ls** /bin

The /bin directory happens to be where Linux commmands such as ls and pwd are stored. For example, you can see that ls is present by entering:

ls /bin/ls

To list all files starting with b in the /bin directory, try entering:

ls /bin/b\*

To list all files ending in r in the /bin directory, enter:

ls /bin/\*r

To print a long list of files that has additional information compared to the simple ls command, such as the last modified date, enter:

**ls -l**

Here are some popular options that you can try with the ls command.

| **Option** | **Description** |
| --- | --- |
| -a | list all files, including hidden files |
| -d | list directories only, do not include files |
| -h | with -l and -s, print sizes like 1K, 234M, 2G |
| -l | include attributes like permissions, owner, size, and last-modified date |
| -S | sort by file size, largest first |
| -t | sort by last-modified date, newest first |
| -r | reverse the sort order |

To get a long listing of all files in /etc, including any hidden files, enter:

**ls** -la /etc

Here we combined the options -l and -a using the shorter notation, -la.

**2.3. Create a directory**

**mkdir**

The mkdir command is used to create a new directory.

To create a directory named scripts in your current directory, run the following command:

mkdir scripts

Use the ls command to verify whether the scripts directories got created:

**ls**

**2.4. Change your current working directory.**

**cd**

To get into the directory scripts directory, run the following command:

cd scripts

Now use the pwd command to verify whether your current working directory has changed as expected:

pwd

If you use cd without any directory name as follows, it will move you back to your home directory:

cd

Again, enter the pwd command to verify whether your current working directory has changed:

pwd

The syntax .. is a shortcut that refers to the parent directory of your current directory. Run the following command to change dictories up one level:

**cd** ..

**2.5. Create an empty file**

**touch**

First, return to your home directory by entering:

cd

Next, use the touch command to create an empty file named myfile.txt:

**touch myfile**.txt

Now use the ls command to verify creation of myfile.txt:

**ls**

If the file already exists, the touch command updates the access timestamp, or last-modified date of the file. To see this, enter:

**touch myfile**.txt

and use use the date command to verify the date change:

date -r myfile.txt

**2.6. Search and locate files**

**find**

The find command is used to search for files in a directory. You can search for files based on different attributes, such as the file's name, type, owner, size, or timestamp.

The find command conducts a search of the entire directory tree starting from the given directory name.

For example, the following command finds all .txt files in the subfolders of the /etc directory:

find /etc -name '\*.txt'

Along with the listing of all .txt files, you may get some Permission denied errors.

These error are normal, as you have limited access on the lab machine.

**2.7. Remove files**

**rm**

The rm command is used to delete files, and is ideally used along with the -i option, which makes it ask for confirmation before every deletion.

To remove the file myfile.txt, enter the follwing command and press y to confirm deletion, or n to cancel:

rm -**i** myfile.txt

Use the ls command to verify removal:

**ls**

**2.8. Move, Rename a file**

**mv**

First, make a file called users.txt:

**touch users**.txt

The mv command moves a file from one directory to another.

Use caution when moving a file, because if the target file already exists, it will be overwritten by the source file.

Conveniently however, if the source and target directories are the same, you can employ mv as a rename operation.

To illustrate this, rename users.txt as user-info.txt by entering the following command:

**mv** users.txt user-info.txt

Now use the ls command to verify the name change:

**ls**

Now move user-info.txt to the /tmp directory, as follows:

**mv** user-info.txt /tmp

Now use the ls command twice to verify the move, as follows:

**ls**

**ls** -l /tmp

**2.9. Copy files**

**cp**

You can use the cp command to copy user-info.txt, which is now in your /tmp directory, to your current working directory as follows:

cp /tmp/**user**-**info**.txt **user**-**info**.txt

Now use the ls command to verify whether the copy was successful:

**ls**

The following command copies the content of /etc/passwd to a file named users.txt under the current directory:

cp /etc/passwd users.txt

Now use the ls command to verify whether the copy was successful:

**ls**

Exercise 3 - Access control commands

Required files:

Run the following code to acquire the required files for this exercise:

wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/usdoi.txt

Each file and each directory has permissions set for three permission categories: the 'owner', the 'group' and 'all users'.

The following permissions are set for each file and directory:

| **Permission** | **symbol** |
| --- | --- |
| read | r |
| write | w |
| execute | x |

To see the permissions currently set for a file, run the command ls -l.

For example, to see the permissions for the file named usdoi.txt in your current directory, run:

**ls -l usdoi**.txt

A sample output looks like:

-rw-r--r-- 1 theia theia 8121 May 31 16:45 usdoi.txt

The permissions set here are rw-r--r--. The - preceeding these permissions indicates that usdoi.txt is a file. If it were a directory, you would see a d instead of the -.

The first three entries correspond to the owner, the next three correspond to the group, and the last threee are for all others. You can see the owner of the file has read and write permissions, while the user group only has read permissions, and all other users have read permission. No users have execute permissions, as indicated by the - instead of an x in the third position for each user category.

**3.1. Change permissions**

**chmod**

The chmod (change mode) command lets you change the permissions set for a file.

The change of permissions is specified with the help of a combination of the following characters:

| **Option** | **Description** |
| --- | --- |
| r, w and x | permissions: read, write and execute, respectively |
| u,g and o | user categories: owner, group and all others, respectively |
| +, - | operations: grant and revoke, respectively |

The command below removes read permission for all users (user, group and other) on the file usdoi.txt:

**chmod -r usdoi**.txt

You can verify the changed permissions by entering:

**ls -l usdoi**.txt

To add read access to all users on usdoi.txt, enter:

**chmod** +r usdoi.txt

Now verify the changed permissions:

**ls -l usdoi**.txt

To remove the read permission for 'all other users' category, enter:

**chmod o-r usdoi**.txt

Verify the changed permissions:

**ls -l usdoi**.txt

Practice exercises

1. Problem:

*Display the content of /home directory.*

Click here for Hint

Click here for Solution

1. Problem:

*Ensure that you are in your home directory.*

Click here for Hint

Click here for Solution

1. Problem:

*Create a new directory called final in your home directory.*

Click here for Hint

Click here for Solution

1. Problem:

*View the permissions of the newly created directory, final.*

Click here for Hint

Click here for Solution

1. Problem:

*Create a new blank file named 'display.sh' in the final directory*

Click here for Hint

Click here for Solution

1. Problem:

*Create a copy of display.sh called report.sh.*

Click here for Hint

Click here for Solution

1. Problem:

*Delete the file 'display.sh'.*

Click here for Hint

Click here for Solution

1. Problem:

*List the files in /etc directory in the ascending order of their access time.*

Click here for Hint

Click here for Solution

1. Problem:

*Display the current time.*

Click here for Hint

Click here for Solution

1. Problem:

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

Click here for Hint

Click here for Solution

Authors

Ramesh Sannareddy Sam Prokopchuk

Other Contributors

Rav Ahuja Jeff Grossman

Change Log

| **Date (YYYY-MM-DD)** | **Version** | **Changed By** | **Change Description** |
| --- | --- | --- | --- |
| 2021-05-30 | 0.1 | Ramesh Sannareddy | Created initial version of the lab |
| 2021-11-29 | 0.2 | Sam Prokopchuk | Update lab contents and split |
| 2021-12-02 | 0.3 | Jeff Grossman | Review and Update lab |

Copyright (c) 2021 IBM Corporation. All rights reserved.