Hands-on Lab: Getting Started with Shell Scripting

Estimated time needed: **30** minutes

Learning objectives

After completing this lab you will be able to:

- Create and execute a simple bash shell script.
- Implement the 'shebang' directive in a bash shell script.

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, we will be using the Cloud IDE based on Theia running in a Docker container.

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

Exercise 1 - Create and execute a basic Shell Script

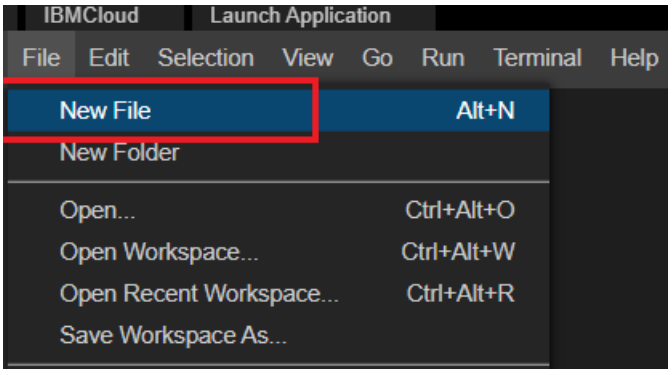
In this exercise, you will create a simple script which will do the following:

- Accept a user name
- Print a welcome message to the user

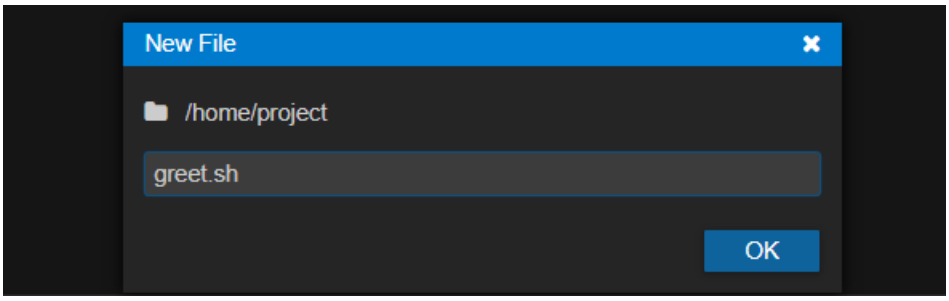
You will also add comments to the script. All the lines starting with # are comments. Comments are not executed by the shell.
When used appropriately, comments can make a shell script more readable, and help in debugging the script.

1.1 Create a new script file

Step 1: On the menu on the lab screen, use **File->New File** to create a new file.



Step 2: Give the name as 'greet.sh' and click 'OK'



Step 3: Copy and paste the following lines into the newly created file.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

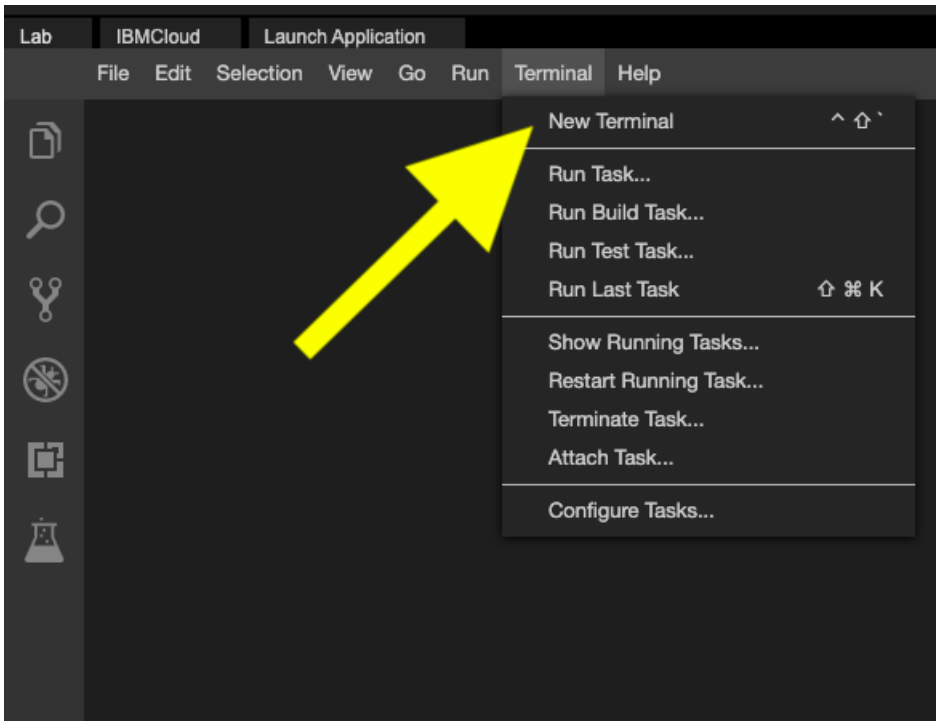
```
::page{title="This script accepts the user's name and prints"}

## a message greeting the user

::page{title="Print the prompt message on screen"}

echo -n "Enter your name :"

::page{title="Wait for user to enter a name, and save the entered name into the variable 'name'"}

read name

::page{title="Print the welcome message followed by the name"}

echo "Welcome $name"

::page{title="The following message should print on a single line. Hence the usage of '-n'"}

echo -n "Congratulations! You just created and ran your first shell script "
echo "using Bash on IBM Skills Network"
```

Copied!

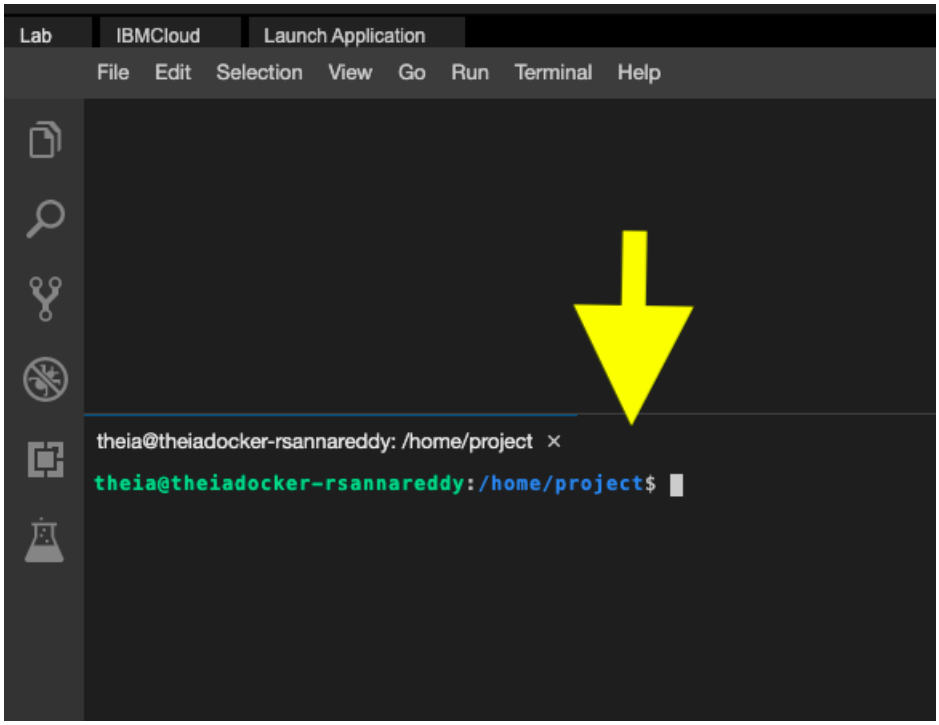
Step 4: Save the file using the **File->Save** menu option

1.2 Execute the script

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 in the image below.



Run the commands below in the newly opened terminal.

Let us check the permissions for this new file:

```
1. 1
1. ls -l greet.sh
```

Copied!

If the file exists and has read permission, run the following command to execute it.

```
1. 1
1. bash greet.sh
```

Copied!

The message ‘Enter your name :’ appears on screen.

Type your name and press the ‘enter’ key.

You should now see the message displayed on screen.

```
theia@theia-rsannareddy:/home/project$ bash greet.sh
Enter your name :Ramesh
Welcome Ramesh
Congratulations! You just created and ran your first shell script using Bash on IBM Skills Network
theia@theia-rsannareddy:/home/project$
```

You have succesfully executed your first bash shell script.

Exercise 2 - Using 'Shebang' line

In this exercise, you will edit the script ‘greet.sh’ you created in the previous exercise by adding a ‘shebang’ and making it an executable file.

This is done to ensure that the name of the script can be used like a command.

Adding this special ‘shebang’ line lets you specify the path to the interpreter of the script, which is the *Bash Shell* in this case.

Let us follow the below steps.

2.1 : Find the path to the interpreter.

The `which` command helps you find out the path of the command `bash`.

```
1. 1
```

```
1. which bash
```

Copied!

In this case it returns the path `/bin/bash`.

2.2 : Edit the script `greet.sh` and add the *shebang* line to the script.

Open the file and add the following line at the beginning of the script:

```
1. 1
```

```
1. #! /bin/bash
```

Copied!

The script should now look like the following:

greet.sh

```
1   #! /bin/bash
2   # This script accepts the user's name and prints
3   # a message greeting the user
4
5   # Print the prompt message on screen
6   echo -n "Enter your name :"
7
8   # Wait for user to enter a name, and save the entered name into the variable 'name'
9   read name
10
11  # Print the welcome message followed by the name
12  echo "Welcome $name"
```

2.3 Check the permissions of the script.

One more step needs to be completed to make `greet.sh` completely executable by name.

Let us add the execute permission for the user on `greet.sh`.

```
1
```

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

Copied!

Verify whether the execute permission is granted.

Generally, it is not a good idea to grant permissions to a script for all user, groups and others alike.
It is more appropriate to limit the execute permission to only the owner (the user who created the file, which is you).

To change permissions for `greet.sh` to make the file executable for the user, run the command below:

```
1
```

```
chmod u+x greet.sh
```

Copied!

Verify the permissions using the command below.

```
1
```

```
ls -l greet.sh
```

Copied!

If you wish to grant "execute" permission to everyone, you need to run the command `chmod +x greet.sh`

2.4 Execute the script.

Enter the command given below to run the shell script.

```
1
```

```
./greet.sh
```

Copied!

The `.` here refers to the current directory. You are telling Linux to execute the script `greet.sh` and that it can be found in the current directory.

Practice exercises

1. Problem:

   Create a script named `greetnew.sh` that reads the first and last names of the user into corresponding variables, `firstname` and `lastname`, and prints a welcome message like "Hello <firstname> <lastname>".

▼ Click here for Hint

Use the `read` command and `echo` commands. Write comments. Make sure to add the shebang line.

▼ Click here for Solution

Step 1: Create a new file named 'greetnew.sh'.

Step 2: Add the following lines to the file:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
```

```
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
```

```
1. #! /bin/bash
2.
3. ::page{title="This script accepts the user's name and prints"}
4.
5. ## a message greeting the user
6.
7. ::page{title="Print the prompt message on screen"}
8.
9. echo -n "Enter your firstname :"
10.
11. ::page{title="Wait for user to enter a name, and save the entered name into the variable 'name'"}
12.
13. read firstname
14.
15. ::page{title="Print the prompt message on screen"}
16.
17. echo -n "Enter your lastname :"
18.
19. ::page{title="Wait for user to enter a name, and save the entered name into the variable 'name'"}
20.
21. read lastname
22.
23. ::page{title="Print the welcome message followed by the name"}
24.
25. echo "Hello $firstname $lastname."
```

Copied!

Step 3: Save the file.

Step 4: Add the ‘execute’ permission to ‘greetnew.sh’ for the owner.

```
1. 1
```

```
1. chmod u+x greetnew.sh
```

Copied!

Step 5: Execute the file from the command prompt using the command below.

```
1. 1
```

```
1. ./greetnew.sh
```

Copied!

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-05-11	0.7	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	0.6	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	0.5	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-04	0.4	Benny Li	Republished
2023-02-12	0.3	Jeff Grossman	Add context, logo, edit grammar and formatting

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-09-12	0.2	Lavanya Rajalingam	Updated Image
2021-05-30	0.1	Ramesh Sannareddy	Created initial version of the lab