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 a 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

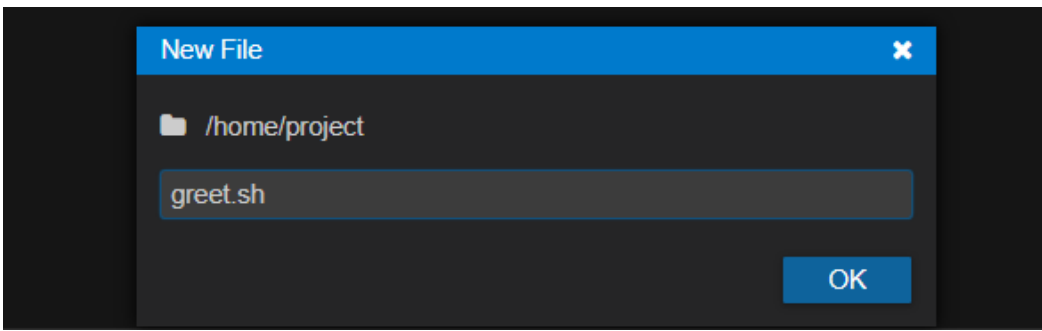You will also add comments to the script, which are lines starting with #. Comments are not executed by the shell.

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.
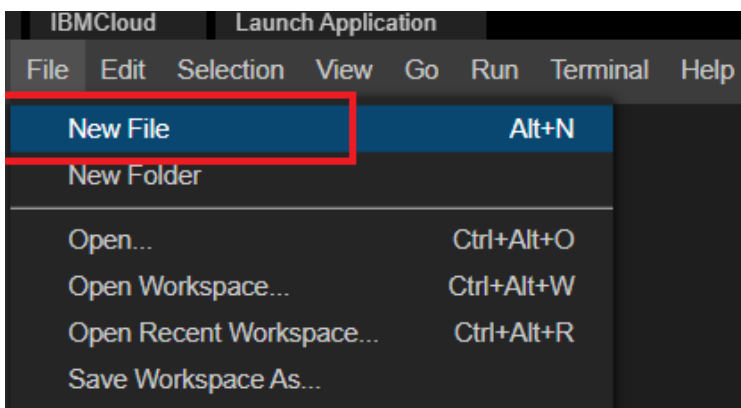
Step 2: Name it as `greet.sh` and click **OK**

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

```
# This script accepts the user\'s name and prints
# a message greeting the user
# Print the prompt message on screen
echo -n "Enter your name :"
# Wait for user to enter a name, and save the entered name into the variable \'name\'
read name
# Print the welcome message followed by the name
echo "Welcome $name"
# 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"
```
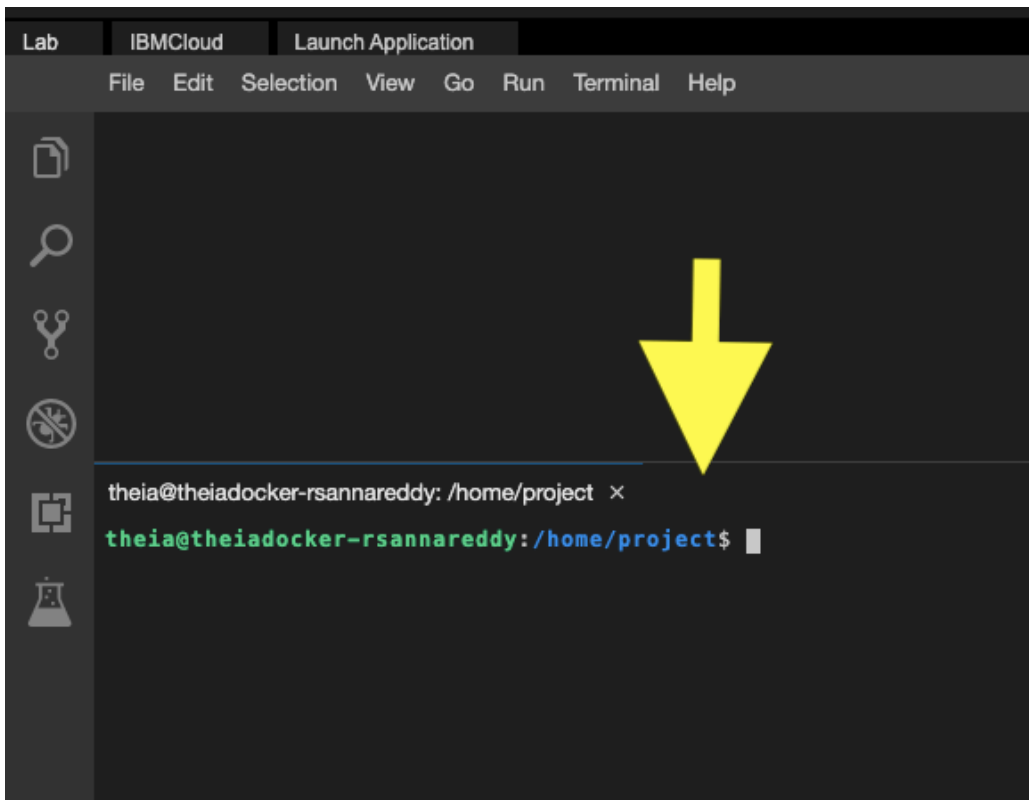
Step 4: Save the file using the **File->Save** menu option.

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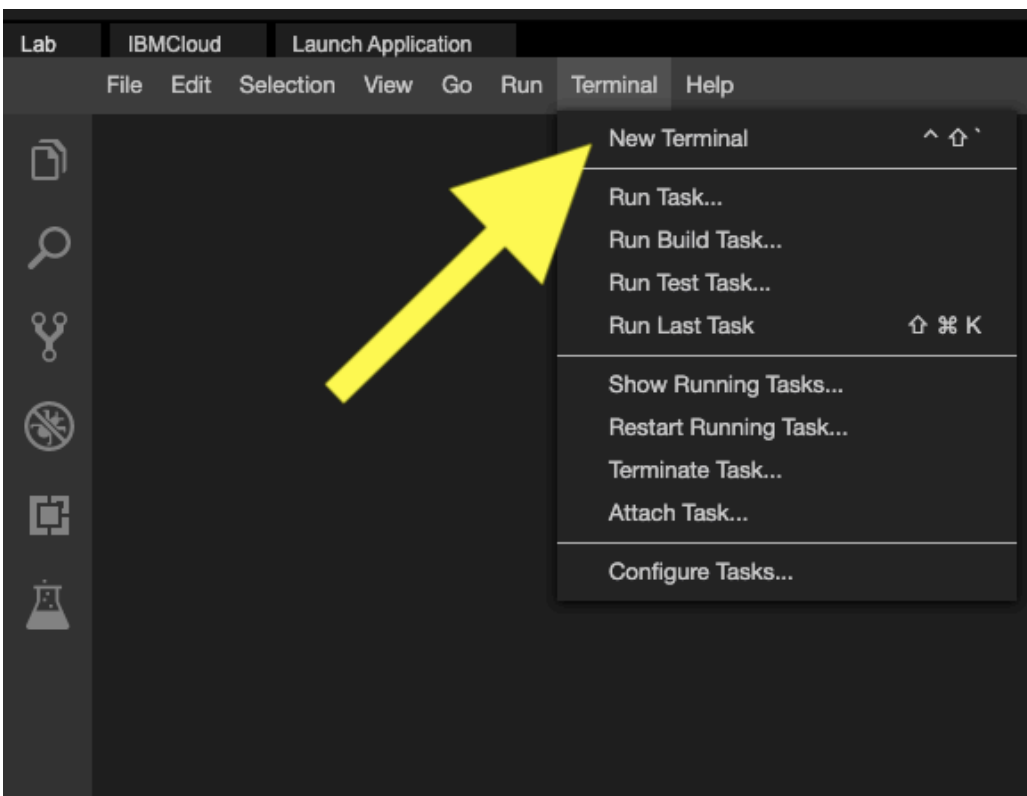
This will open a new terminal at the bottom of the screen.

Run the commands below in the newly opened terminal.

Let's check the permissions for this new file by entering the following:

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

If the file exists and has read permissions, run the following command to execute it:

```
bash greet.sh
```

The message `Enter your name :` appears on screen.

Type your name and press the `Enter` key.

You should now see the welcome messages displayed on screen with your entered name.

Congratulations! You have succesfully executed your first Bash shell script.

Exercise 2 - Using a shebang line

In this exercise, you will edit the `greet.sh` script 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 - in this case, the *Bash shell*.

Follow the steps below to learn how to add a shebang to your script.

2.1. Find the path to the interpreter

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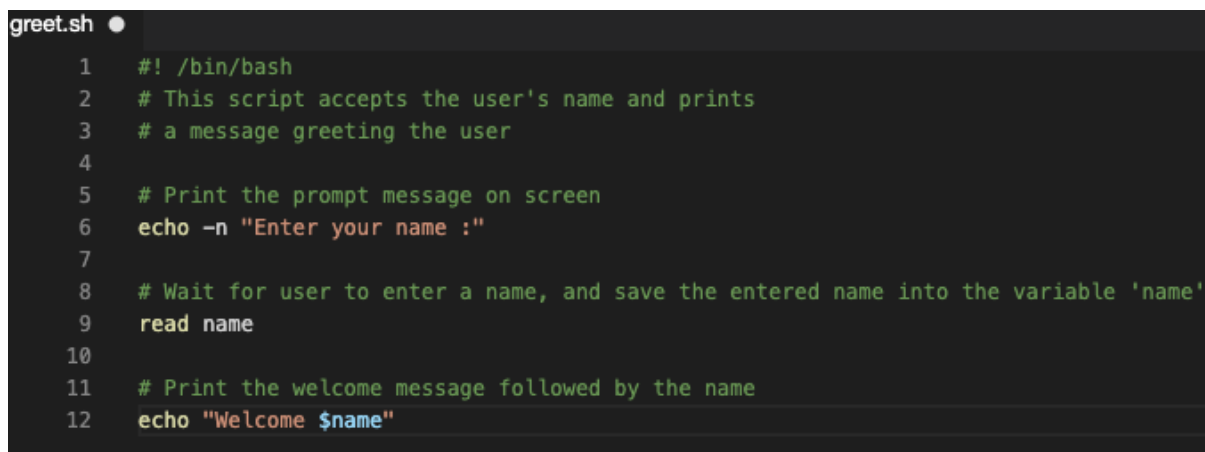
```
which bash
```

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:

```
#! /bin/bash
```

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.

To add the execute permission for the user on `greet.sh`, enter the following:

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

Verify whether the execute permission is granted.

> **Tip**: Generally it's not a good idea to grant permissions to a script for all users, groups, and others. It's more appropriate to limit the execute permission to only the owner, or the user who created the file (you).

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

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

Verify the permissions using the command below:

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

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.

```
./greet.sh
```

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 exercise

1. Create a script named `greetnew.sh` that takes the first and last names of the user, saves them in corresponding variables `firstname` and `lastname`, and prints a welcome message, such as "`Hello <firstname> <lastname>`".

▶ Click here for Hint
▶ Click here for Solution

Summary

In this lab, you learned how to:

- Create and execute a simple Bash shell script
- Implement the shebang directive `#! /bin/bash` in a Bash shell script

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja