Lab 1 - Hello World

This is the lab to test drive Docker.

Before we begin, start the *cmder* command.

> cmder is a bash command prompt alike tool that brings the terminal experience of linux to windows users.

```
$ docker run hello-world
```

Let's look at this command. The **docker** is the command that we are running, and **run** is the specific command we are issuing to instruct docker. Lastly, **hello-world** is the specific image name that we want to run.

The list below is the complete list of available docker commands.

Command	Description
attach	Attach to a running container
build	Build an image from a Dockerfile
commit	Create a new image from a container's changes
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
diff	Inspect changes on a container's filesystem
events	Get real time events from the server
exec	Run a command in a running container
export	Export a container's filesystem as a tar archive
history	Show the history of an image
images	List images
import	Import the contents from a tarball to create a filesystem image
info	Display system-wide information
inspect	Return low-level information on a container or image
kill	Kill a running container
load	Load an image from a tar archive or STDIN
login	Register or log in to a Docker registry

logout	Log out from a Docker registry
logs	Fetch the logs of a container
network	Manage Docker networks
pause	Pause all processes within a container
port	List port mappings or a specific mapping for the CONTAINER
ps	List containers
pull	Pull an image or a repository from a registry
push	Push an image or a repository to a registry
rename	Rename a container
restart	Restart a container
rm	Remove one or more containers
rmi	Remove one or more images
run	Run a command in a new container
save	Save an image(s) to a tar archive
search	Search the Docker Hub for images
start	Start one or more stopped containers
stats	Display a live stream of container(s) resource usage statistics
stop	Stop a running container
tag	Tag an image into a repository
top	Display the running processes of a container
unpause	Unpause all processes within a container
update	Update resources of one or more containers
version	Show the Docker version information
volume	Manage Docker volumes
wait	Block until a container stops, then print its exit code

After issuing the command, you will see a message similar to the one below.

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
03f4658f8b78: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc1ca36966a7
```

```
Status: Downloaded newer image for hello-world:latest
```

This is normal, it simply means that docker fails to find the latest image of our **hello-world** image in our local file system (yes, because we have not downloaded it previously).

Once the image is downloaded, you shall see

```
Hello from Docker.
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker Hub account:
 https://hub.docker.com

For more examples and ideas, visit:
 https://docs.docker.com/userguide/
```

This means that the **hello-world** image has been downloaded successfully, and the message is the output from executing it. If you want to double check, just run the same command again, and you will see the same output, but it will not download the image anymore, since it is already been downloaded.

Using External Images

Locate docker images from http://hub.docker.com.

Sign up for an account, and login to it.

In the search bar on top, enter “whalesay”, scroll down and look for “docker/whalesay”. This is a public image available for use while learning Docker.

```
$ docker run docker/whalesay cowsay boo
```

This command will run **whalesay** image in a container. The terminal should be downloading the image as it is our first time.

You should see this.

```
 _____
< boo >
 -----
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
```

Check images

While inside the Docker quick start terminal, we can type

```
$ docker images
```

to list the images that we have downloaded. You should see a similar result to the list below.

```
REPOSITORY                    TAG                 IMAGE ID            CREATED
SIZE
ubuntu                        latest              b549a9959a66        4 days ago
188 MB
hello-world                   latest              690ed74de00f        5 months ago
960 B
docker/whalesay               latest              6b362a9f73eb        10 months ago
247 MB
```