

# Do-It-Yourself: Run a Container

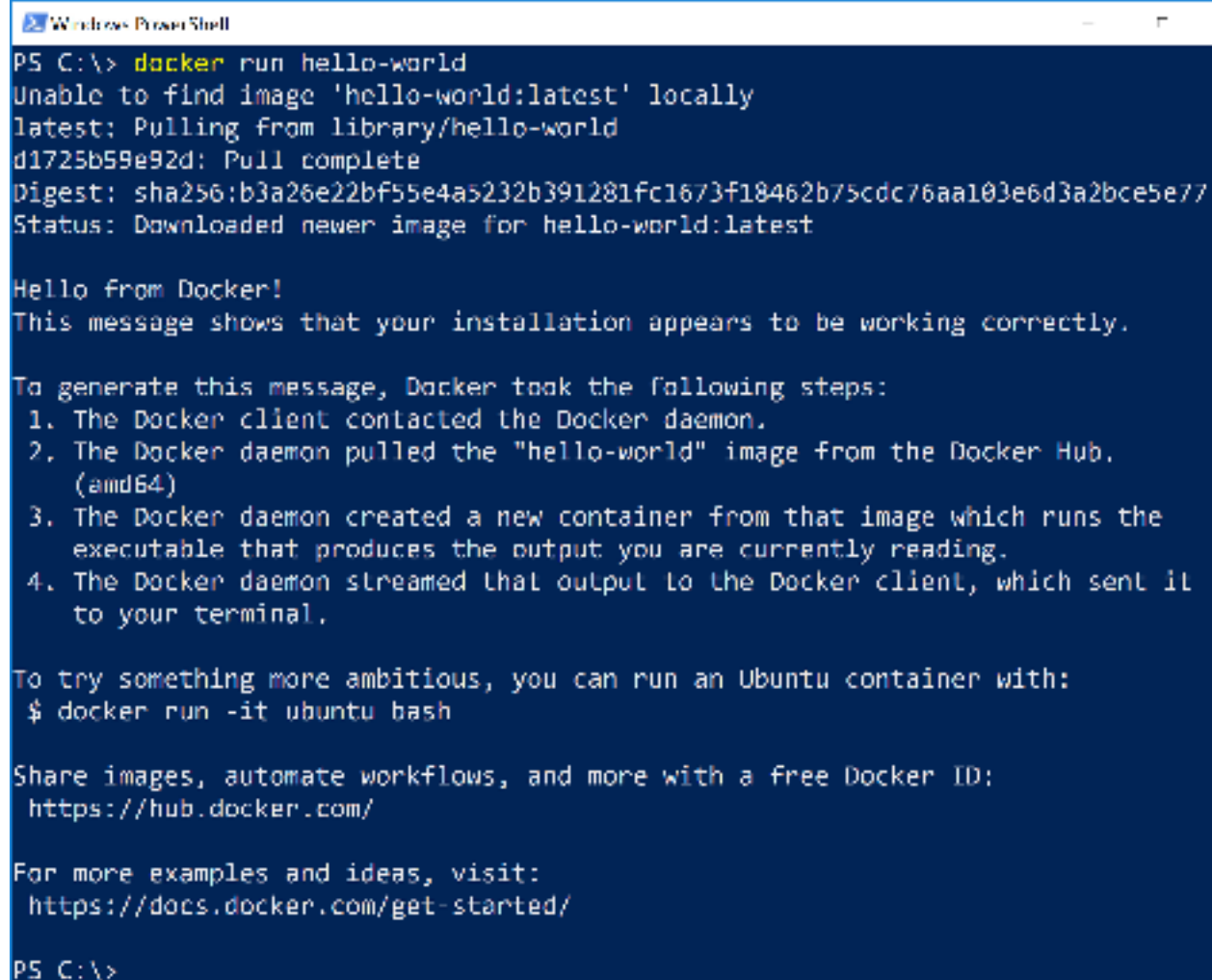
In this lesson, you will learn how to run a container based on the already created hello-world image.

Note: The commands used in this course will be the same regardless of if you are using a Windows, Linux, or Mac system.

Run the following command on a command-line:

```
docker run hello-world
```

You can see the following text output:



```
Windows PowerShell
PS C:\> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:b3a26e22bf55e4a5232b391281fc1673f18462b75cdc76aa103e6d3a2bce5e77
Status: Downloaded newer image for hello-world:latest

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.
    (amd64)
 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 ID:
https://hub.docker.com/

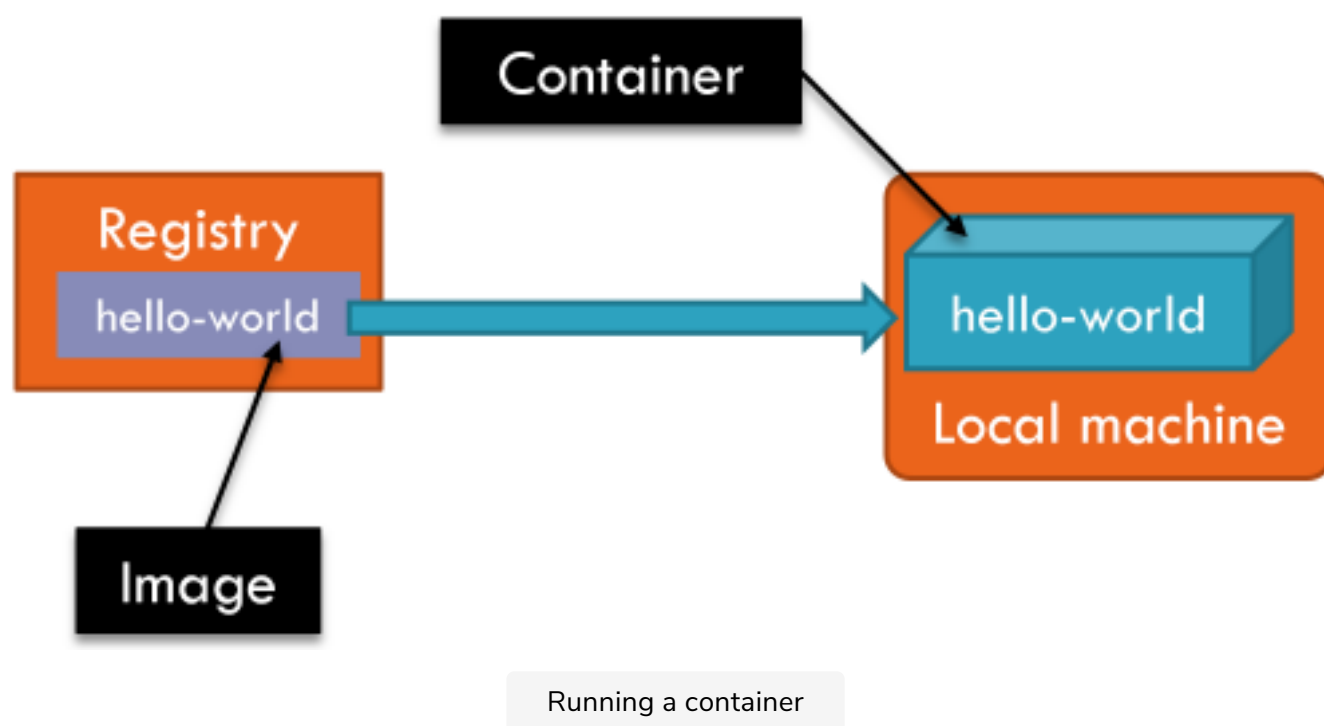
For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\>
```

Congratulations, you just ran your first container! Here's what just happened in detail:

1. Your command asks Docker to create and run a container based on the *hello-world* image.
2. Since the *hello-world* image wasn't already present on your disk, Docker downloaded it from a default registry, the *Docker Hub*. More about that [later](#).
3. Docker created a container based on the *hello-world* image.
4. The *hello-world* image states that, when started, it should output some text to the console, so this is the text you see as the container is running.
5. The container stopped.

Here's what you did, slightly simplified:



If you run the same command again, you'll see that all the above steps are being repeated except for step 2; this is because the image does not need to be downloaded as it is already present on your machine from the first time you ran the command. This is a simple optimization, but you'll see later that Docker optimizes many more steps. As such, Docker makes scarce use of a machine's resources.

---

In the next lesson, you will be introduced to some Docker commands.

