Do-It-Yourself: Post-Mortem Inspection

In this lesson, you will explore some container management commands on your hello-world container.

The hello-world container we previously ran stopped because its only job was to output text. We'll soon run more advanced containers, but before we do that, let's play with that stopped container in order to get a better grip of container tooling.

Run the following command on a command-line:

```
docker ps
```

There is no output because there is no container currently running on your machine. Now run the following command on a command-line:

```
docker ps -a
```

You should see output similar to the one below, although the exact container ID varies:

Container ID	Image	Status
48bab7f673b3	hello-world	Exited

The container ID should be noted, as it allows you to run commands on the ID's corresponding container.

Run the following command replacing the ID with your container ID:

```
docker logs 48bab7f673b3
```

The above command outputs exactly the same text you saw when you ran the container. This is the standard output of the container. In real-world scenarios, you'll have containers running in the background (web servers for instance) and this command will allow you to have a look at their output.

Run the following command replacing the ID with your container ID:

```
docker inspect 48bab7f673b3
```

The above command gets you an abundance of information about many aspects of the container - whether it is still running or stopped. Again, in real-world scenarios, this may prove useful for understanding what happens inside your containers.

All of the information you obtain from the above commands takes up space, and once you're completely done with a stopped container, you may want to reclaim that space. This is where the *docker rm* command comes in.

Run the following command replacing the ID with your container ID:

```
docker rm 48bab7f673b3
```

The container with the ID you ran the above command on, should now be deleted. To confirm that the container does not exist anymore, run the following commands replacing the ID with your container ID:

```
docker ps -a
docker logs 48bab7f673b3
docker inspect 48bab7f673b3
```

Docker doesn't know about that container anymore:

```
Error: No such object: 48bab7f673b3
PS C:\>
```

In the next lesson, we will take a closer look at the *docker run* command.