

# Introduction to Docker

**Objective:** An overview of Docker's functionality working with containers<br>

**Preparation:** Open up two instances of your favorite shell, and a browser window<br>

**Outcome:** Each student will replicate a highlight demo of some basic functionality<br>

**Data Files:** Ask Instructor<br>

As we are learning, Docker is a container technology. When we talk about Docker, we are talking about containers. Remember, containers are spun up off of images, and images are spun up off of Dockerfiles. We will talk more about creating an image from a Dockerfile later, but for now let's grab an existing image on Docker Hub and run a container.

## Step 1. Exploring Docker Help

1. First, let's look at Docker's help menu in our shell

```
$ docker -h
```

2. Next, let's check to see if we currently have any running containers

```
$ docker ps
```

3. Finally, let's see if we have any images in our local repository

```
$ docker images
```

4. [OPTIONAL] Go through and try other Docker commands from the help list. If you will type `docker <command>` you will get to see the structure syntax.

## Step 2. Pulling Images Down From Docker Hub

1. Type `docker pull` into our terminal and note the structural syntax to enacting the pull command:

Output:

```
docker pull [OPTIONS] NAME[:TAG|@DIGEST]
```

2. Now, let's make our own pull request using the appropriate syntax we just learned. The name of the image we want to `pull` is `docker/whalesay`, we don't need to enact any options, and we won't need to specify a tag this time.

NOTE: Whenever you don't issue a specific tag, Docker defaults to latest.

```
$ docker pull docker/whalesay
```

| NOTE: If you do not have access to the internet, consult your instructor.

3. Verify we have successfully pulled our image down using the previous command:

```
$ docker images
```

## Step 3. Launching Containers From Images

1. Now, we still don't have a container running. We can verify that by running the command:

```
$ docker ps
```

2. So, let's spin up a container. By typing `docker run` into our terminal, we can review the run command's structure, just like we did with `docker pull`.

Output:

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

3. Using our new `docker/whalesay` image, let's spin up a container, referring back to the previous step, if needed, for the appropriate structuring. We won't need to enact any options and we know the image name. The command we are going to use is `cowsay` and we are going to input `hello-world`. It should look like this:

```
$ docker run docker/whalesay cowsay hello-world
```

[illegible]

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```
Output:
< hello-world >

-----
 \
  \
   \

          ##                .
        ## ## ##          ==
       ## ## ## ##      ===
 /"oooooooooooooooooooo"/ ===
~~~~ {~~ ~~~~ ~~~ ~~~~ ~ ~ / === - ~~~
    \_____ 0 _____/
     \      \      _/
      \_____\_____/
```

[illegible][illegible][illegible]

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

[illegible][illegible]

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

```

Output:
< hello-world >

-----
\
\
\

##
.

## ## ## ==
## ## ## ## ===
/""""""""""""""""""""___/ ===
~ ~ {~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~
\_____ 0 _____/
\         \       /
\         \_____/
\         \_____/

```

## Conclusion

Great job! Believe it or not, you just: Pulled down an image from Docker Hub, used it to spin up a free standing container, but you also ran a command inside that container and display the results to your monitor. You should be proud of yourself!