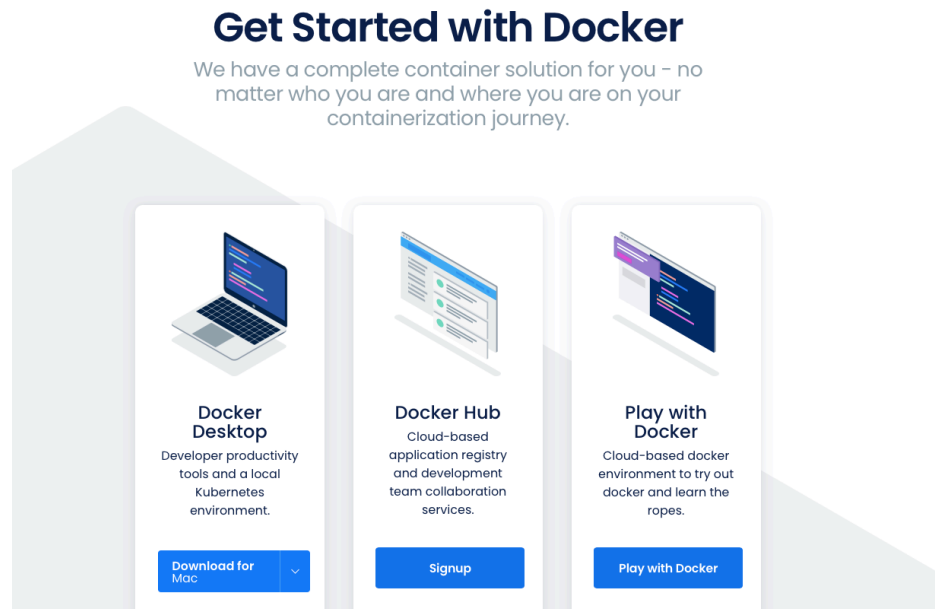


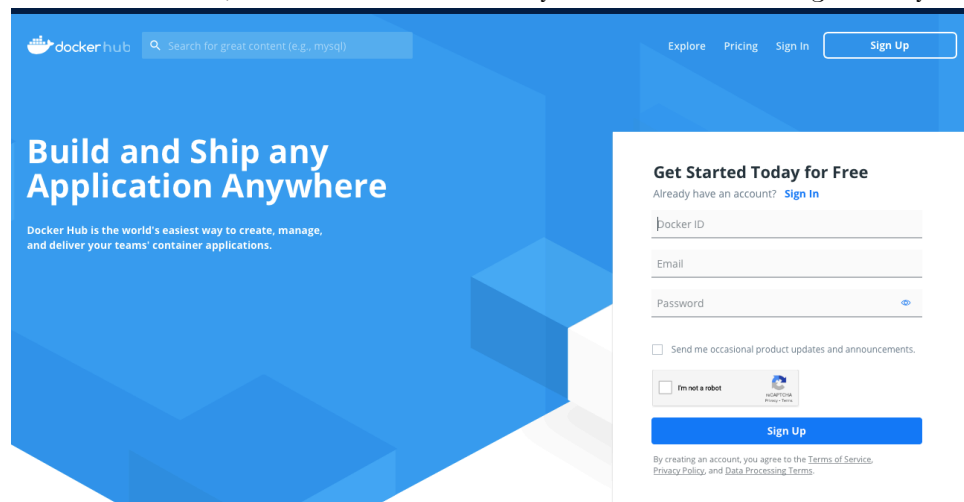
Microservice Docker Assignment

Casey Maloney and Jennifer Mohr

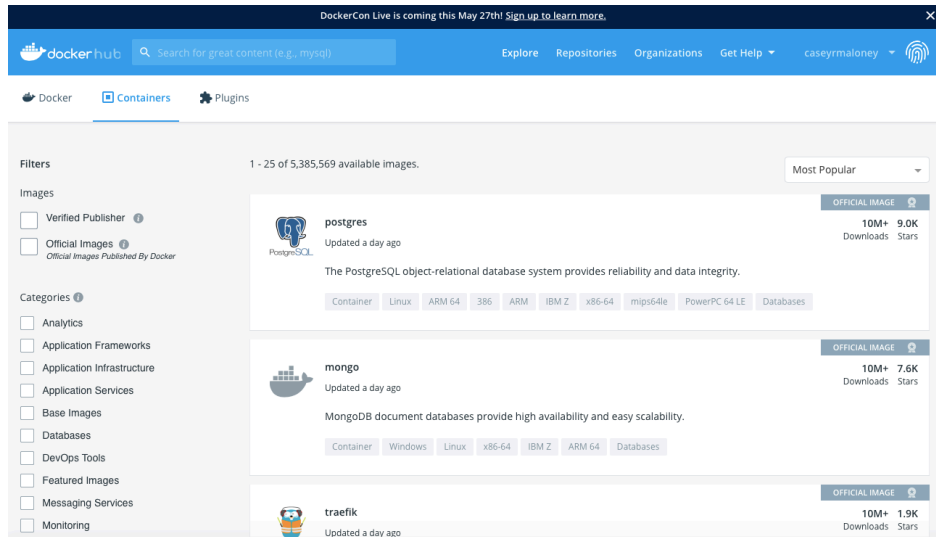
We installed docker onto our laptop using their website and clicking docker for desktops. After installing I had to move my download into my application.



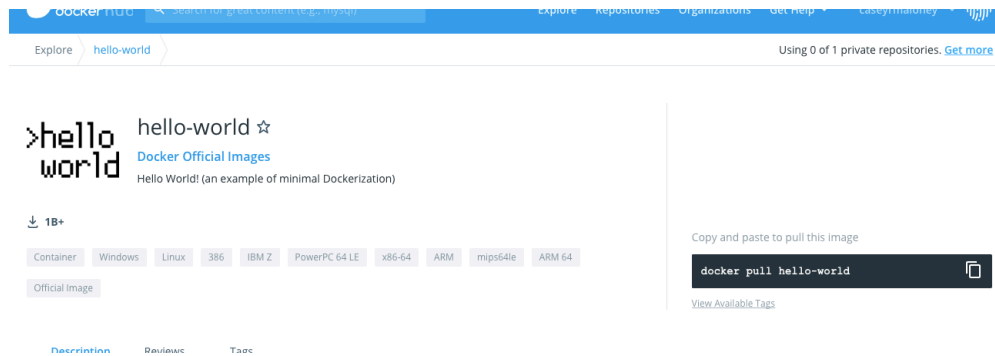
After I downloaded docker, I used dockerhub on my browser to create/log into my account.



Then after exploring the docker hub page we found a “hello-world” image that we could simply pull and push and run on our machines.



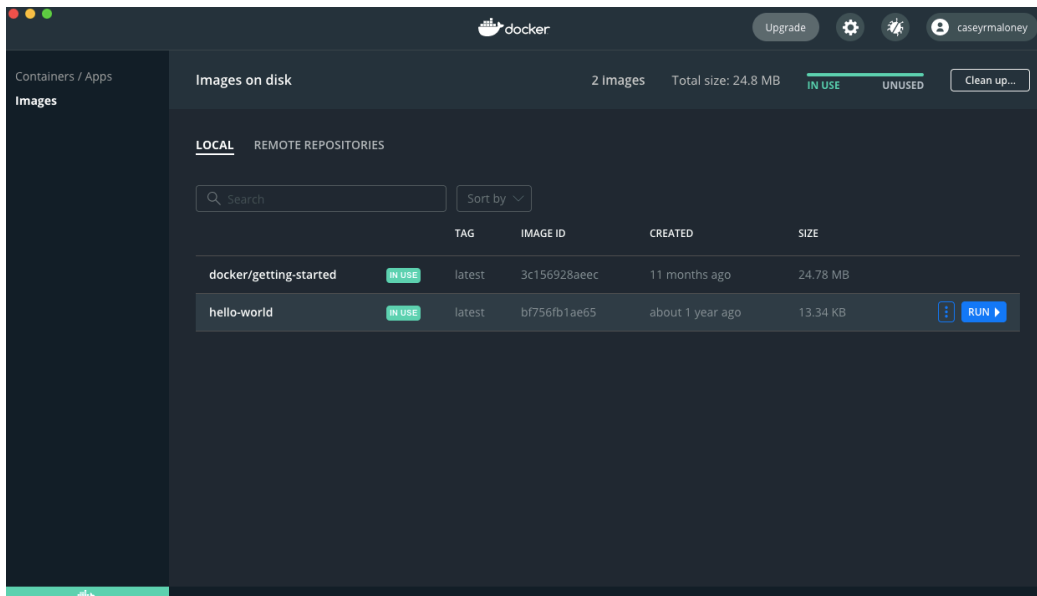
Then I went to the docker hello-world and followed then simple command line.



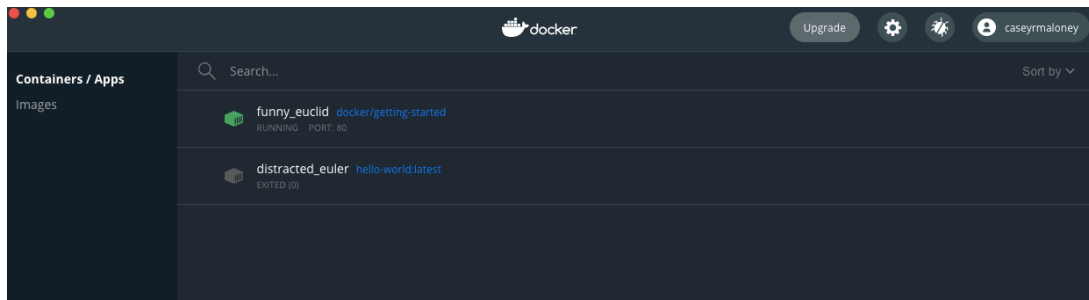
On my terminal I then ran docker pull hello-world, which cloned the image into my machine.

```
Caseys-MacBook-Air:~ caseymaloney$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:7e02330c713f93b1d3e4c5003350d0dbe215ca269dd1d84a4abc577908344b30
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
Caseys-MacBook-Air:~ caseymaloney$
```

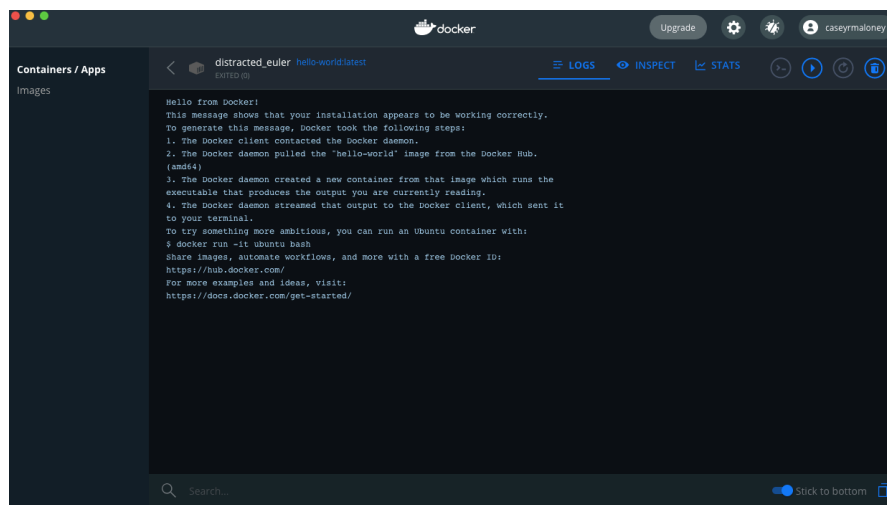
In my docker application it then came up in my images



I hit the run button and it ran hello-world and it went into my container/apps



When I inspected the distracted_euler it showed the output for the hello-world, showing that it worked correctly and what occurred on the application.



Then to double check that it worked I ran it on my terminal and the output was the same.

```
Caseys-MacBook-Air:~ caseymaloney$ docker run hello-world

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/

Caseys-MacBook-Air:~ caseymaloney$
```

What is going on with docker vs my computer?

Docker is a software development tool and a virtualization technology that makes it easy to develop, deploy, and manage applications by using containers. These containers are individual packages of a piece of software that contain all the libraries, config files, dependencies, and other parts needed to run the application. With docker, the application works on your local computer and will work anywhere that supports docker.

With virtual machine environments, images would be the snapshots of these environments. Docker Images can never be changed, and once made you can't modify them. Applications run the same no matter where they are and what machine they are running on because the container has the same environment throughout the software development cycle of the application. The container is like a print-out of the image we created.

These containers do not have an operating system within them. It shares the underlying kernel with the other containers. Each container running on a host is completely isolated as applications running on the same host are completely isolated. They are unaware of each other. The docker container runs within my computer.

Overall, the docker image is separated from the computer. The computer is handling docker itself, while docker is handling the application.