Lab 2.2 Container Deletion

We are going to see how we can manipulate our containers

## Step 1: Start up a new alpine container

docker container run -it alpine

You will find yourself a root prompt in the new container. Try executing the command "ls /"

# ls /

bin dev etc home lib media mnt proc   
root run sbin srv sys tmp usr var

## Step 2: Delete all files in the container

Now let's delete all files in the container. This is very dangerous... you must make sure you are doing this **IN THE CONTAINER**, and not on a root prompt on your local system or your cloud VM.

# rm -rf /

**DO NOT TRY THIS AT HOME!!**

Now let's see if the files are gone

# ls

/bin/sh ls: not found

Everything's gone. You've trashed your system. Congradulations!

**=> Exit out of your container by hitting control-D**

Now, let's see what happens if we run our container again.

docker container run -it alpine

Ok, let's see if our files are there

ls -l

bin dev etc home lib media mnt proc   
root run sbin srv sys tmp usr var

=>\*\* Why are all the files back after we deleted them all?\*\*

Go ahead and exit by hitting CNTRL-D or typing exit.

## Step 3: Deleting Containers

Now, see if you can see your command:

docker container ls -as

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES SIZE  
c183852a7215 alpine "/bin/sh" 6 minutes ago Exited (0) 2 minutes ago adoring\_joliot 5B (virtual 4.15MB)

Notice the size. 5 bytes. Not exactly taking up a lot of space. The container IMAGE takes up 4.15MB.

But, we may want to clean up our images anyway. We can use docker container rm for that.

docker container rm <container-id>

And you can now run docker container ls -as

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES SIZE

It shouldn't be there anymore.

What if you want to run the docker container and just have it auto-delete after you are done.

```bash