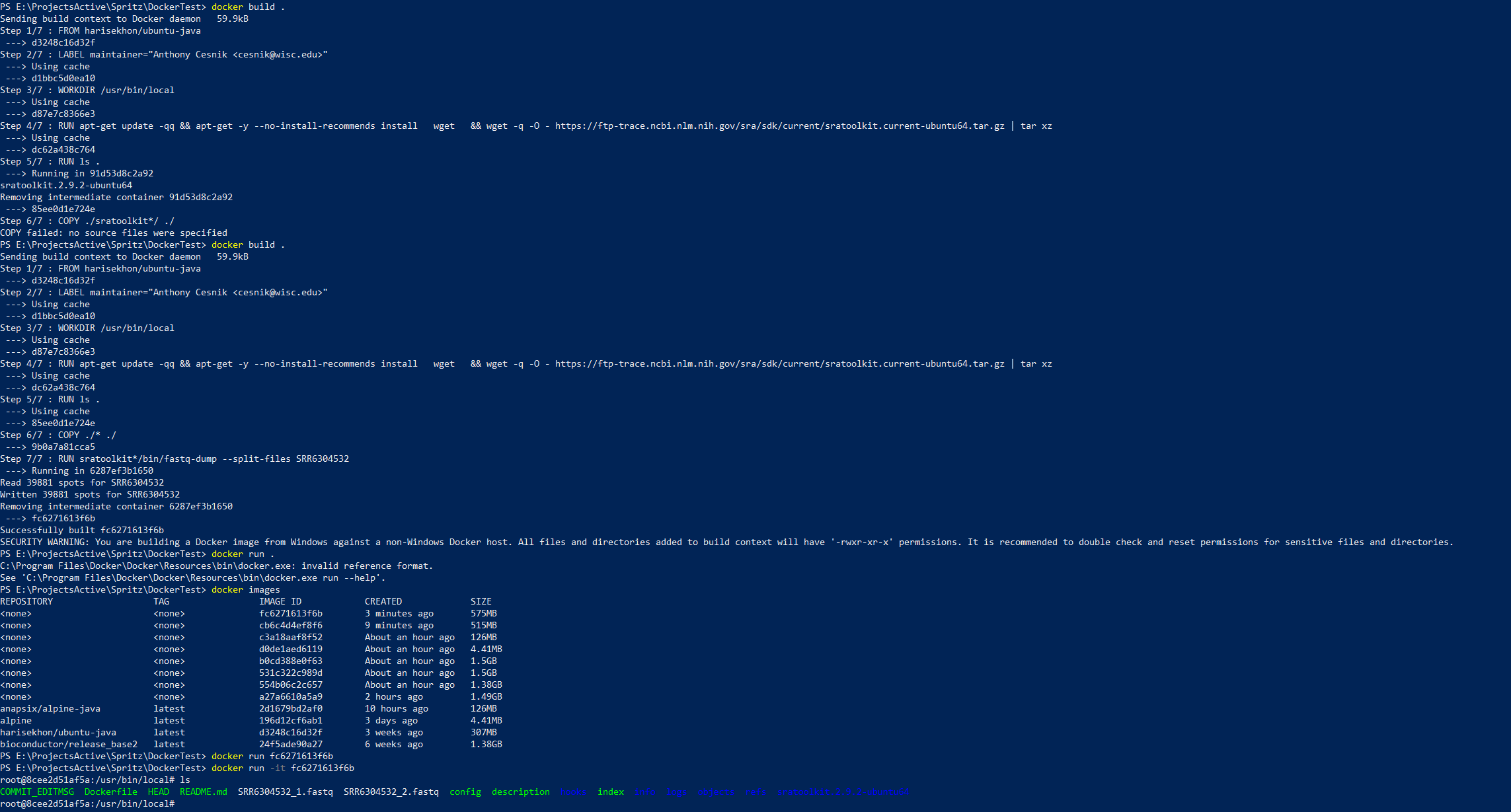
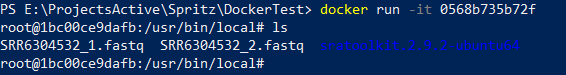
**180914 Getting Docker to run SRA toolkit**

After getting the RUN commands down, it does actually work, and it can easily download the files into an image. Using the “-it” flag gets me into the terminal of the image after the run.

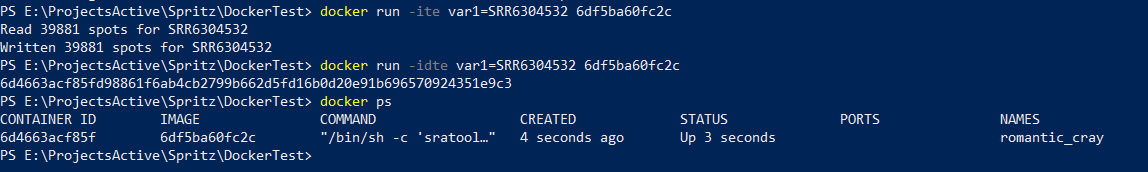


And when I don’t copy everything “./\*” to the image, I still see the right files:



Oh man, yeah, so when you run an “image,” it makes a thin writable layer called a “container” that goes away after the run.

So how do you keep it around? Using the –d flag.



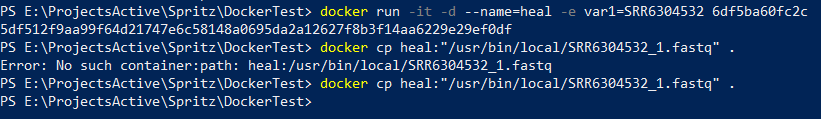
**TO RESTART a stopped docker container:**

Again for those arriving from Google, if [**@BretFisher**](https://github.com/BretFisher)'s command doesn't work for you, try this:

docker commit CONTAINER NEWIMAGENAME

docker run -ti --entrypoint /bin/bash NEWIMAGENAME

Here, I finally got this to work somehow how I intended:



It runs the image with the SRR accession input, and it downloads the read archive, and I can access the container to copy the file to the current directory in PowerShell.

My confusion is this:

1. As illustrated by the two copy commands, I can’t tell when the container has finished the process.
2. I can’t use a wildcard to copy files back over. This is probably okay, since I usually keep track of filenames pretty precisely in Spritz, but it’s annoying.

# In conclusion

I think Docker is really powerful. It would definitely help keep the application as lightweight as possible, while keeping the binaries organized. It would be a major overhaul, though! There are many kinks in the chain of just integrating this into my pipelines.

I suppose there might be some clever structuring I could do: only keep around the tools that I would need to wrap for a certain workflow.

But wow, the Dockerfile thing is really buggy coming over from bash scripts. It’s not a simple transition. The “FROM ubuntu” thing is amazing as a base. I could imagine using the SRA toolkit image as a base for my other workflows. Making all of the bash commands into one-liners, which is recommended, would definitely be a big transition.

The biggest issue for me is how to schlep files around from containers. I did a pretty good job of keeping track of the output files that matter in the Spritz code, so it could probably be an easier transition that way.

The remove feature of the containers is pretty great. When it’s done, you can just wipe it from the face of the earth and just keep what you need.