Week 5

Learning

Problems

Minikube is not using the latest image when I load an image with the same name and tag

Running `$(eval docker-env)` in terminal helps with connecting your current docker implementation with minikube. Now whenever I build an image, it goes directly into the minikube image registry, and the images can be used from there.

Unknown directive in nginx

It looks like the nginx docker container I'm using, doesn't have the ability to "proxy_pass". I find this hard to believe since proxy_pass is a very basic nginx command.

Turns out the directive I was using was `proxy_pass:` when it needed to be `proxy_pass`.

504 Gateway Timeout when redirecting to backend

I had a few issues with ports on the backend side. For one, I was setting my `targetPort` in my configuration to "http" which defaults to port 80. I learned that `targetPort` is the port that the service exposes internally in the port, and `port` is the port that the container is to expose externally. I was only exposing my spring boot application on port 8080, so this it made sense why my `targetPort` wasn't working when mapping it to port 80.

Liveness vs Readiness Probes

These are important in determining the rolling out of an application. Many applications take a while to start up, and these probes help Kubernetes to determine whether an application is in an acceptable state.

Liveness probes are used to determine whether an application is in an acceptable state. For example, if you program was live, and then something happened to it where it

is now broken, the liveness probe could help to determine that the probe is no longer in an acceptable state.

Readiness probes determine when an application is done starting up. For example, a program may need to fetch a bunch of files or parse through some configuration before becoming ready to serve traffic.

Both probes start immediately when the container runs. If either of the probes fail, the pod will be restarted until the probes succeed.