Defeating Distance

- Make everything REPL-enabled
- https://github.com/markbastian/clj-cloud-playground

https://github.com/markbastian/clj-cloud-playground

clj-cloud-playground

A Clojure demonstration project designed to enable you to rapidly spin up and experiment with a Clojure application using Docker, Containerization, and the like.

The Deployment Options Zoo

Run Locally

To run locally, do one of:

- Launch a REPL and evaluate `(clj-cloud-playground.core/start)`
- lein run
- lein uberjar, java -jar target/clj-cloud-playground-0.1.0-SNAPSHOT-standalone.jar

Run with Docker

To run using Docker:

1. Build the standalone app with `lein uberjar`
2. Build the image using `docker build --tag=clj-cloud-playground .`
3. Run the app using `docker run -p 3000:3000 clj-cloud-playground` . This will run you app in a local container that exposes port 3000 to port 3000 locally. You might also try these invocations:
   - `docker run -e NREPL_PORT=3001 -p 80:3000 -p 3001:3001 clj-cloud-playground` : Sets the `nrepl-port` variable to to 3001 and map the container's port 3000 to local port 80. This allows you to connect to your running image and do interactive development.
   - `docker run -e IS_PRODUCTION=true -p 3000:3000 clj-cloud-playground` : Sets the `is-production` environment variable to true so you can modify your internal app as appropriate.

DockerHub

1. Create a repo at Docker's Cloud Site. In this example my repo name is `markbastian/clj-cloud-playground` .
2. Build the image using `docker build --tag=$REPO:$TAG .` where `$REPO` and `$TAG` are your repository and tag names. In this case the exact command I am using is `docker build --tag=markbastian/clj-cloud-playground:latest .` I selected a tag of `latest` arbitrarily. It can be whatever you want.
3. Push the image using `docker push markbastian/clj-cloud-playground:latest` .
4. TODO: Deploy commands...