Notes on setting up shiny on cloud.gov via dockerfile

1. install cloud foundry (if not already done) and log into cloud.gov with cloud foundry

- https://cloud.gov/docs/getting-started/setup/

2. get access to cloud.gov and login.

- if using ID, select 2nd certificate (at least that what it was on Jeff's ID)
- You can login via the browswer to see the dashboard, but most of cloud foundry tools you need to access via the command line interface (CLI).

- Log-in via the CLI is accomplished with:
  - `cf login -a api.fr.cloud.gov  --sso`
- You will be directed to go to https://login.fr.cloud.gov/passcode to get a temporary authorization code. If you have already logged in it will take you straght there, otherwise you will need to jump through the EPA authentication hoops.
- A page will show your temporary authorization code.
- Copy this code and paste it (or just type it) onto the line in your command prompt after `Temporary Authentication Code ( Get one at https://login.fr.cloud.gov/passcode )>`. Nothing will show up on the screen. This is normal. Hit return.
- If it accepts the code, you will need to select the appropriate org to use for your cloud.gov session.
- You should now be logged into cloud.gov and be ready to use the cloud foundry tools to push up your app.

3. Build your app.

- Mostly on your own here, but a few considerations
- Apps with just CRAN available packages are going to be much more straightforward.
- External dependendices (e.g. gdal) are not yet figured out. Jeff is still working on this.
- Use the absolute minimum number of packages. Adds considerably to the time to deploy the app and there is a 15 minute time out on cloud.gov that is pretty easy to hit. Vendoring the packages might help. Jeff is still working on this.

4. Set-up files needed for the cloud.gov push

- r.yml
  - This file tells the r-buildpack what R packages are needed. it looks like:

```
---
packages:
  - cran_mirror: https://cran.r-project.org
    packages:
      - name: dplyr
      - name: stringr
```

- manifest.yml
  - This provides cloud.gov with all the clound foundry bits and pieces. You can include almost all of these via the command line, but the manifest.yml provides a level of reproducibility so is preferrable. A bare bones example looks like:

```
---
applications:
- name: myshinyapp
  buildpacks:
  - https://github.com/cloudfoundry/r-buildpack.git
```

- Procfile

  - This is a one-line file that gets run once your cloud.gov instance has started. For Shiny apps this starts the Shiny application. It looks like:

```
web: R -f shiny.R
```

- shiny.R
  - This is the R script that starts the Shiny app is what the Procfile points to. It looks like

```
library(shiny)
runApp(host="0.0.0.0", port=strtoi(Sys.getenv("PORT")))
```

5. Issues still to be worked out.

- cf ssh fails on EPA network
  - It's available (https://cloud.gov/docs/apps/using-ssh/)
  - Port 2222 appears to be blocked somewhere
  - works fine on my home network
- External dependencies
  - rgdal app is example of attempts
  - uses the experimental apt-buildpack
  - Might work for most, but gdal and rgdal are problems
  - rgdal looks for gdal in /usr/lib or /usr/bin (can't remember which at the moment) but the apt-buildpack stashes these elsewher
  - Fix is probably going to require installing rgdal differently
    - https://stackoverflow.com/questions/34333624/trouble-installing-rgdal might help… esepcially no 7 in selected answer.
- Speeding up deployment with vendored packages
  - Made some progress
  - make dir vendor_R/src/contrib in app
  - use miniCRAN::makeRepo(pkgDep("pkg_name"),"vendor_r", type="source"")
  - saves time on downloading pacakges as they are installed from local sources.