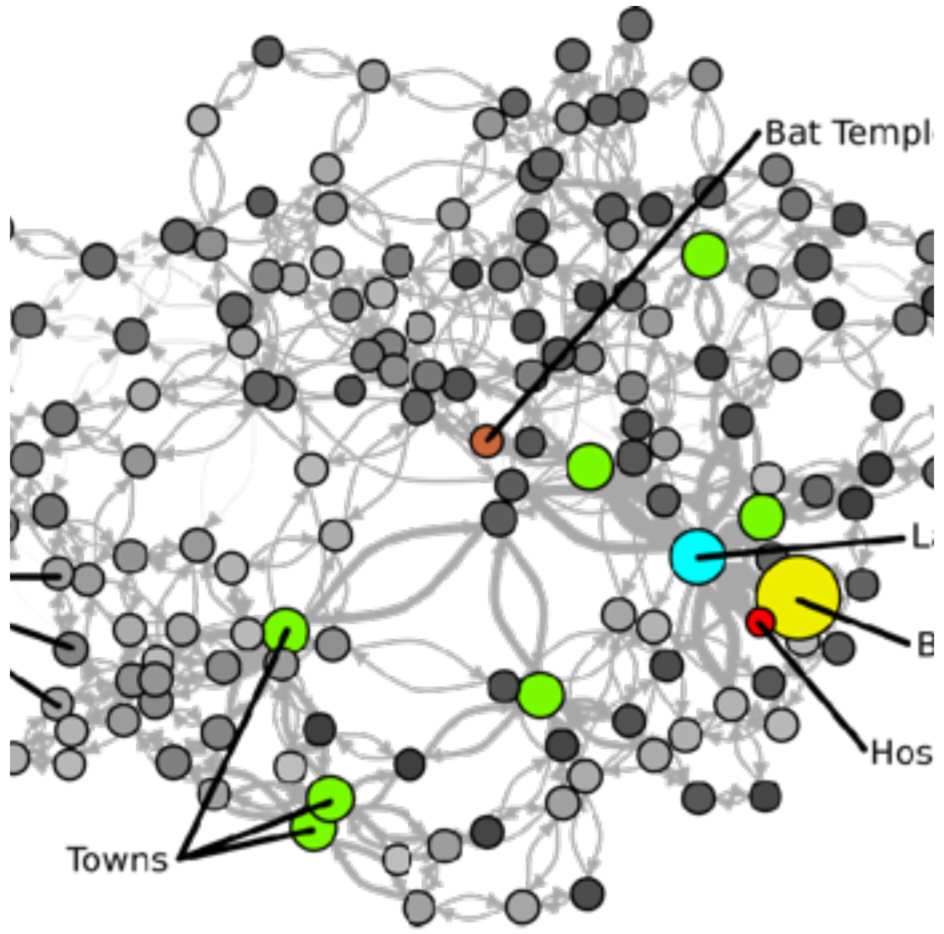
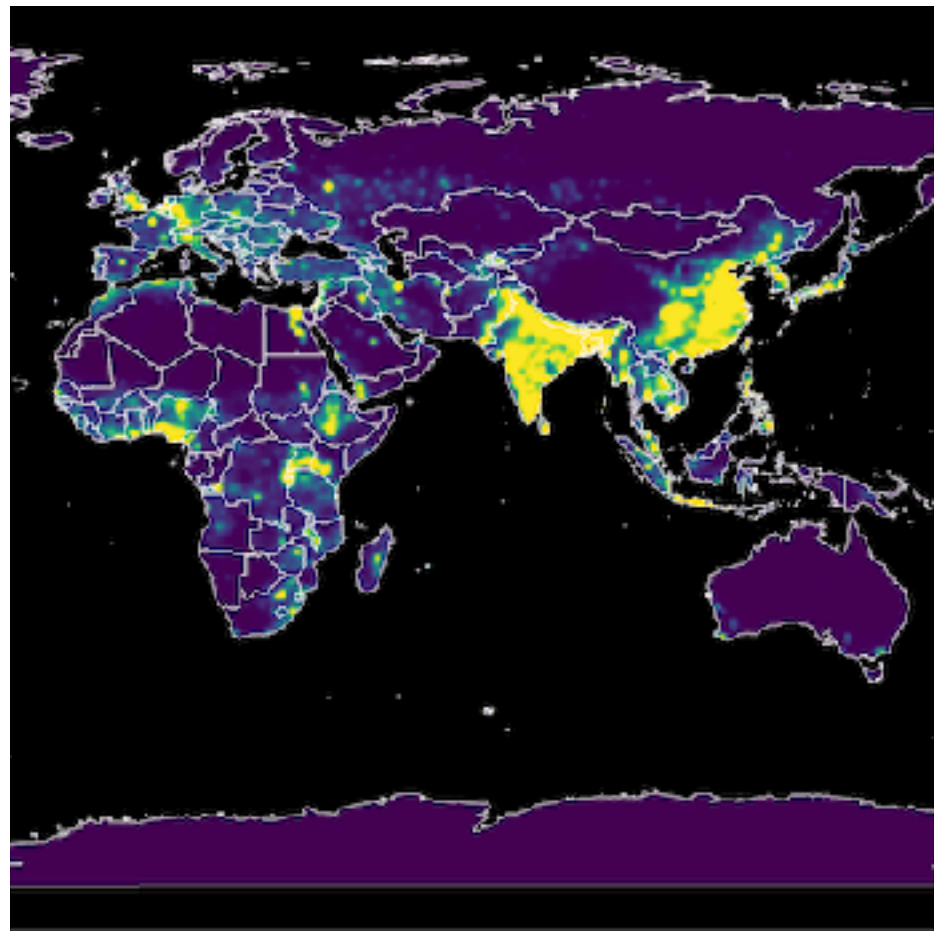
Docker for the UseR

Noam Ross, nyhackr, 2018-07-10

@noamross





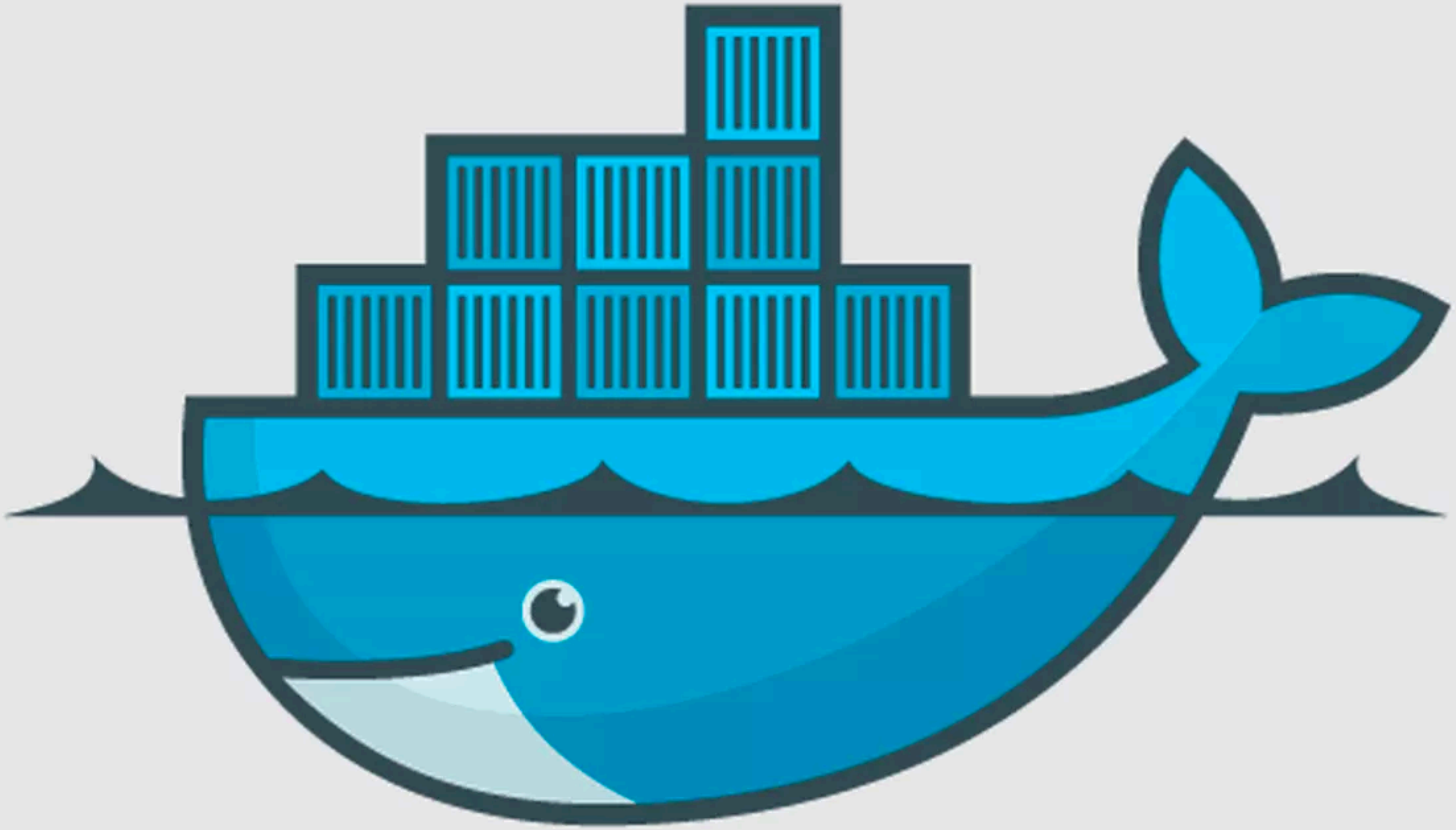
EcoHealth Alliance

@EcoHealthNYC ecohealthalliance.org

ROpenSci

Building technical and community infrastructure for R to support open, reproducible science

@rOpenSci
ropensci.org

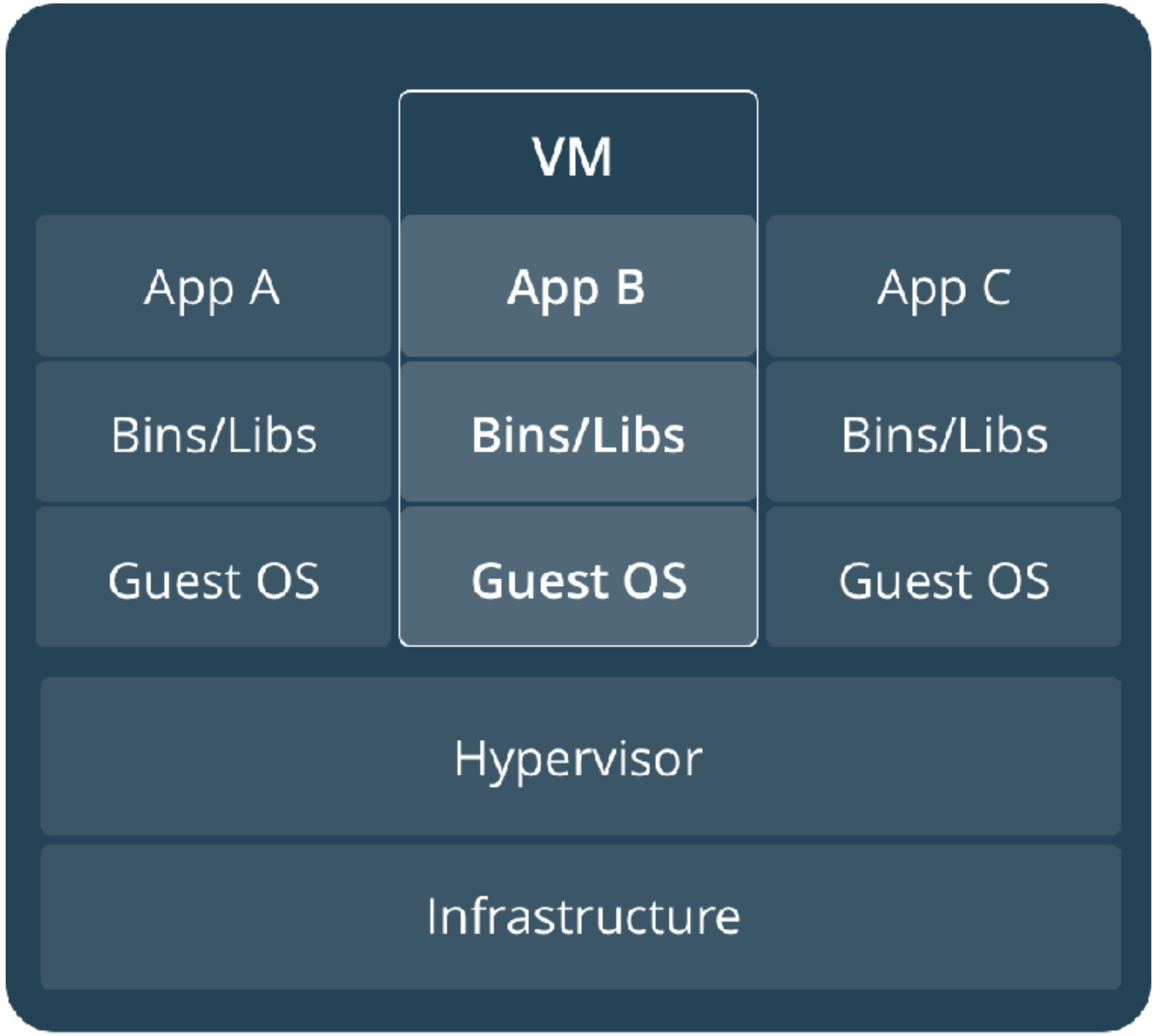


docker ?

Virtual Machine Technologies

A lighter weight framework for scalability

Virtual Machine



VM

App A App B App C

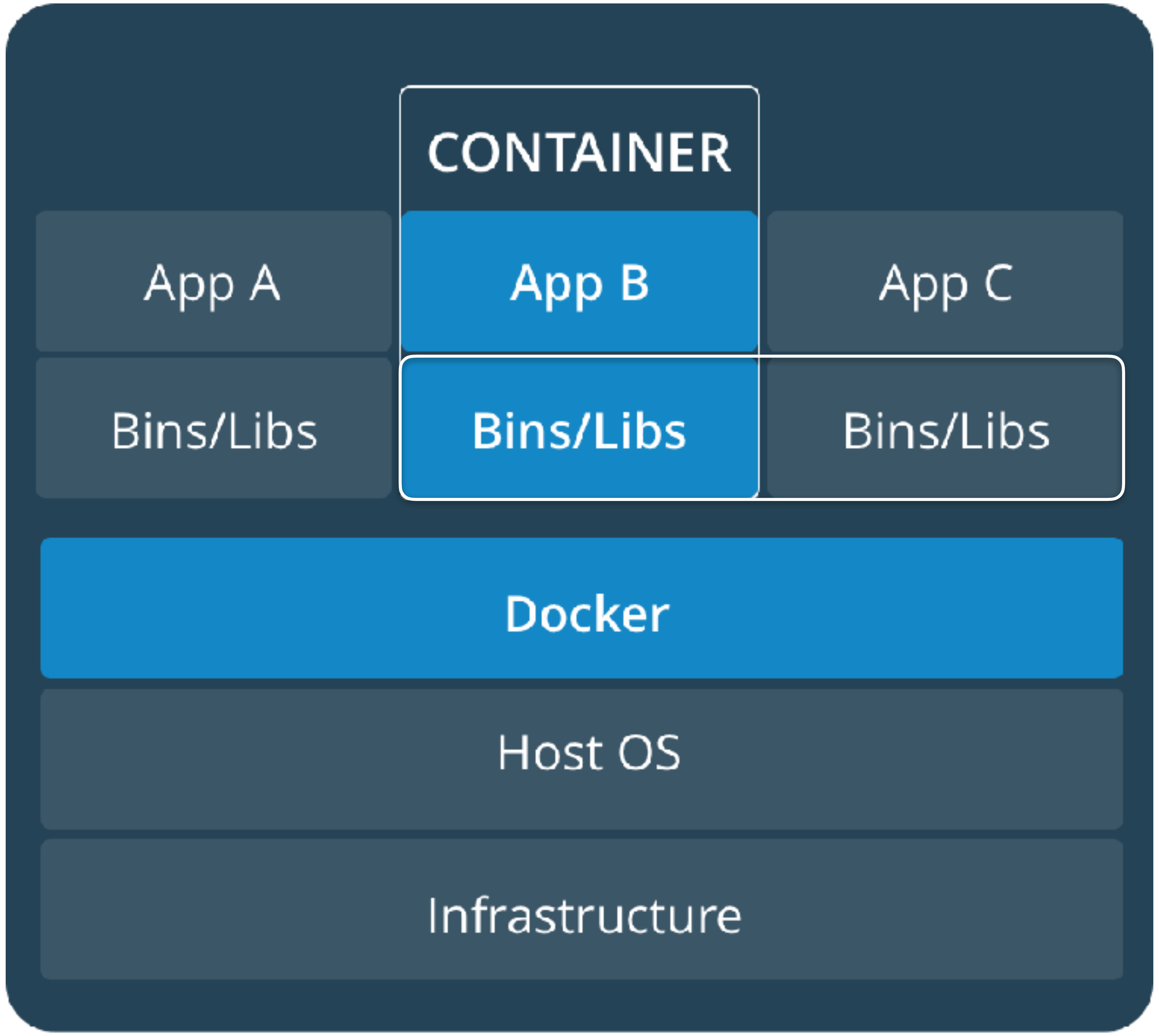
Bins/Libs Bins/Libs Bins/Libs

Guest OS Guest OS Guest OS

Hypervisor

Infrastructure

Docker



CONTAINER

App A App B App C

Bins/Libs Bins/Libs Bins/Libs

Docker

Host OS

Infrastructure

Scalability → Ubiquity → Galactic Conquest

A bare minimum of **vocabulary**

image	a package defining a working environment
container	an instance of an image / a running image
Dockerfile	a file with a series of commands to build an image
Docker Hub	a website/repository for sharing Docker images

A bare minimum of **commands**

`docker pull [user/repo]`	Get image from a repository

`docker build [directory]`
`-t [tag]`
	Make an image from a Docker file
	and give it a tag/name

`docker run [image]`
`-p [ports]`
`-v [volumes]`
`[command] -d`
	Start a container from an image
	opening a set of ports
	linking a set volumes
	run a non-default command, detach

`docker stop [container]`	Stop a running container

A bare minimum of **commands in R**

```
library(stevedore) # github.com/richfitz/stevedore
                     `harbor`, and `docker` also options
dc <- docker_client()

dc$image$pull(name = [user/repo])

dc$image$build(context = [directory])

my_container <-
  dc$container$run(image = [image], ports = [ports],
                   volumes = [volumes], cmd = [command],
                   detach = TRUE)

my_container$stop()
```

Now, let's use Docker to ...

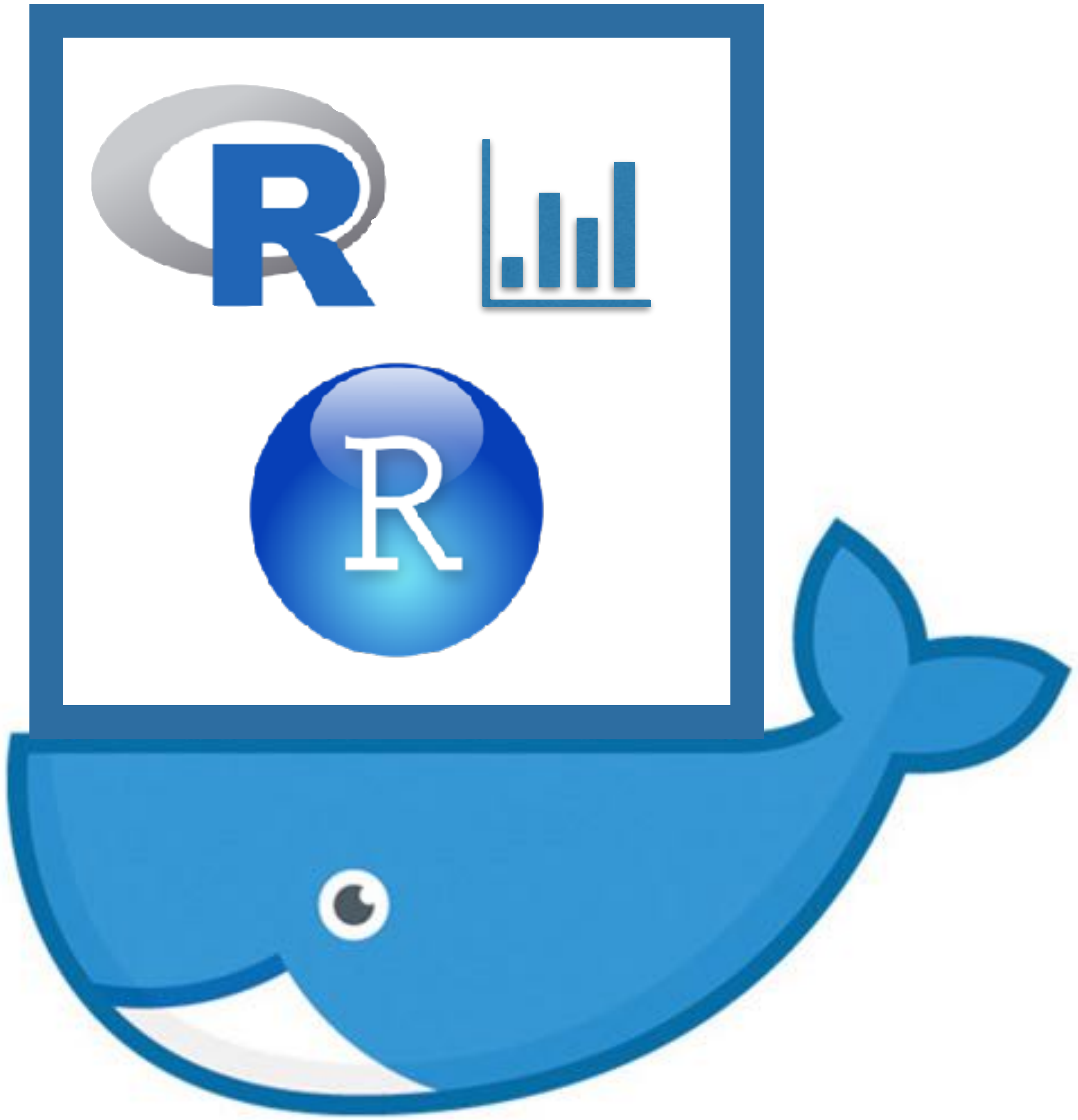
Make a fixed working environment

Access a service outside of R

Create an R based service

Send our compute job somewhere else

Docker as a Fixed Working Environment



GitHub, Inc. [US] https://github.com/ecohealthalliance/HP3

```
│   ├── 08-make-Figure03-ExtendedFigs-stitch-maps.R    | Assemble maps together into Figure 3 and Extended
│   ├── 09-make-Figure04-viral-traits.R                | Generate Figure 4
│   ├── 10-make-ExtendedFigure02-heatmap.R             | Generate heat map for Extended Figure 2
│   ├── 11-make-ExtendedTable01-models.R               | Generate Extended Table 1 of model summaries
│   └── 12-make-SuppTable01-predictions.R              | Generate supplemental table of oberved and predict
│
├── intermediates/                                     | Holds intermediate fitted model objects when proje
├── shapefiles/                                        | Holds large shapefiles downloaded when project is
├── packrat/                                           | Holds all R package dependencies
└── .Rprofile                                          | Configures R to use packrat dependencies
```

Reproducing the analysis

The `Makefile` in this repository holds the project workflow. Running `make all` in the directory will re-build the project. `make clean` will remove shapefiles, intermediate data, fit models, and all figures and maps. If this project is opened in RStudio, this can also be accomplished with the "Build All" and "Clean" buttons in the Build tab.

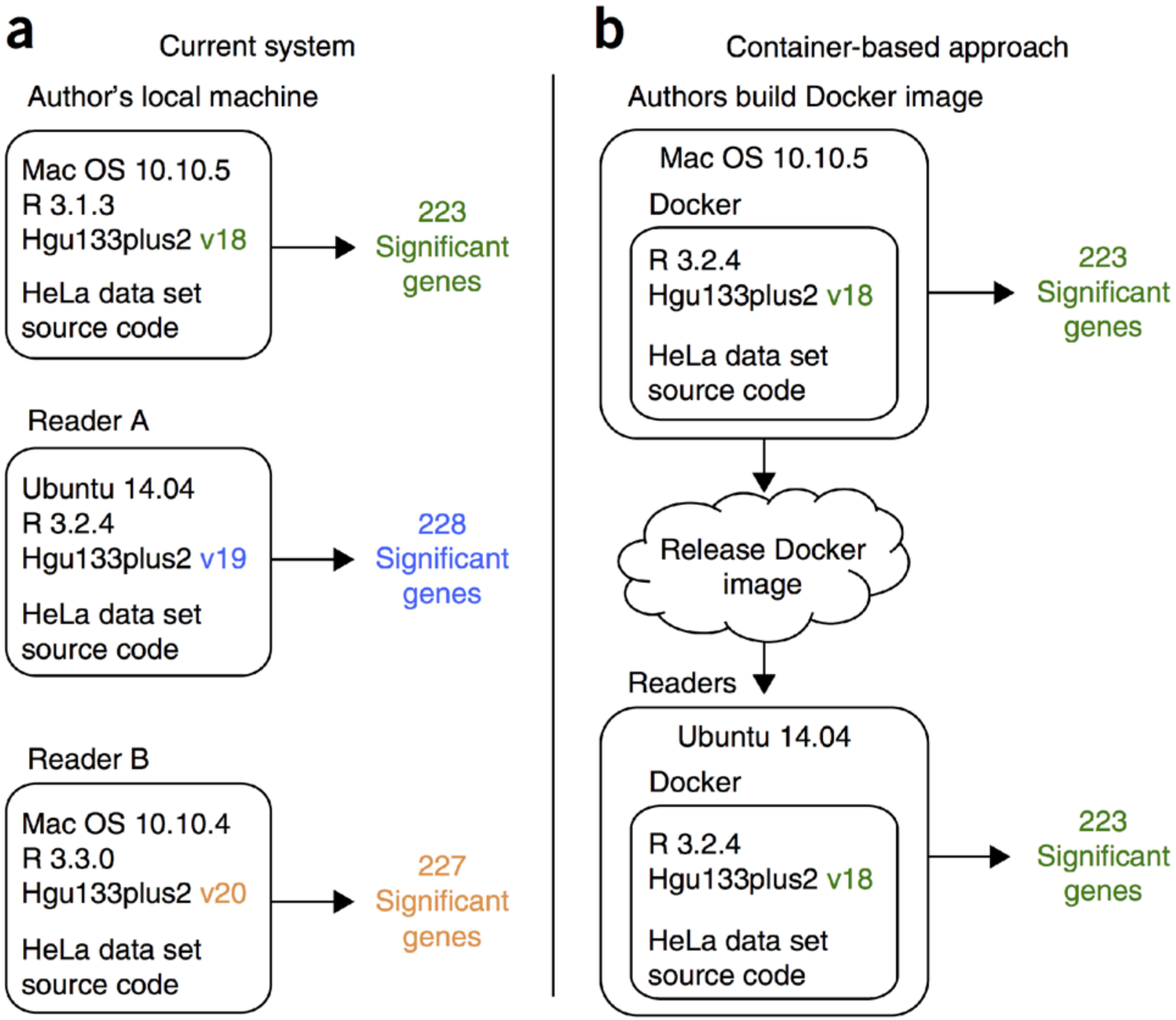
This project uses packrat to manage R package dependencies. Running `packrat::restore()` will unpack the versions of packages used in this project. In addition, these packages have the following system requirements: `cairo`, `gdal`, `GEOS`, `libmagick++-`, `jave`, `libcurl`, `libpng`, `libxml2`, `OpenSSL`, and `pandoc`. All analyses were performed using R 3.3.2 under Ubuntu 14.04. Complete build takes approximately 1 hour with 40 cores and 256GB of memory, or approximately 8 hours on a 2-core Macbook Pro with 16GB of memory.

Getting people on the same page - within and across teams

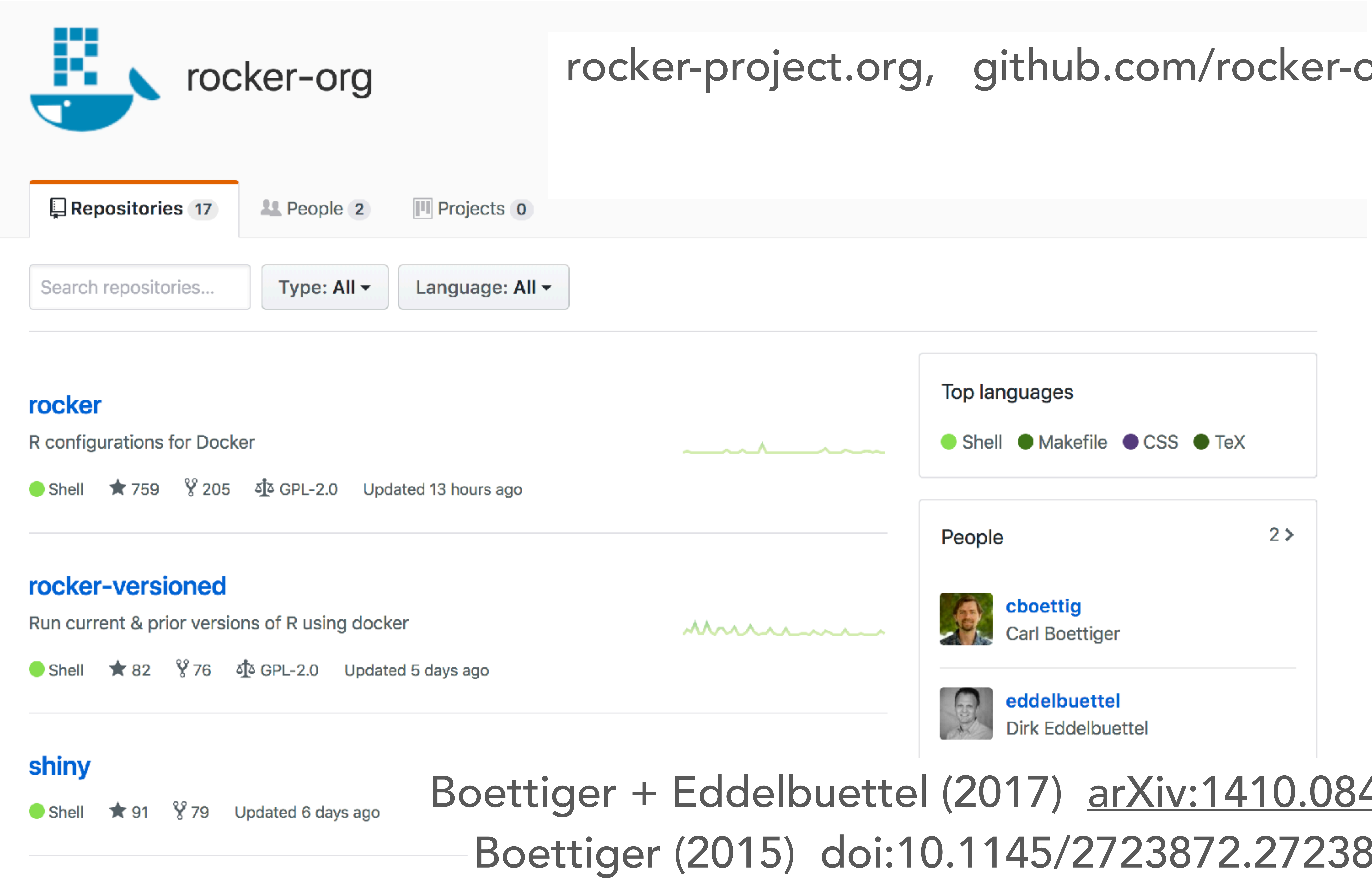
nature biotechnology

Reproducibility of computational workflows is automated using continuous analysis

Brett K Beaulieu-Jones[1] & Casey S Greene[2]



Beaulieu-Jones & Greene (2017) doi:10.1038/nbt.3780 rocker-project.org, github.com/rocker-org



rocker-org

Repositories 17 | People 2 | Projects 0

Search repositories... | Type: All | Language: All

rocker

R configurations for Docker

● Shell ★ 759 205 GPL-2.0 Updated 13 hours ago

rocker-versioned

Run current & prior versions of R using docker

● Shell ★ 82 76 GPL-2.0 Updated 5 days ago

shiny

● Shell ★ 91 79 Updated 6 days ago

Top languages

● Shell ● Makefile ● CSS ● TeX

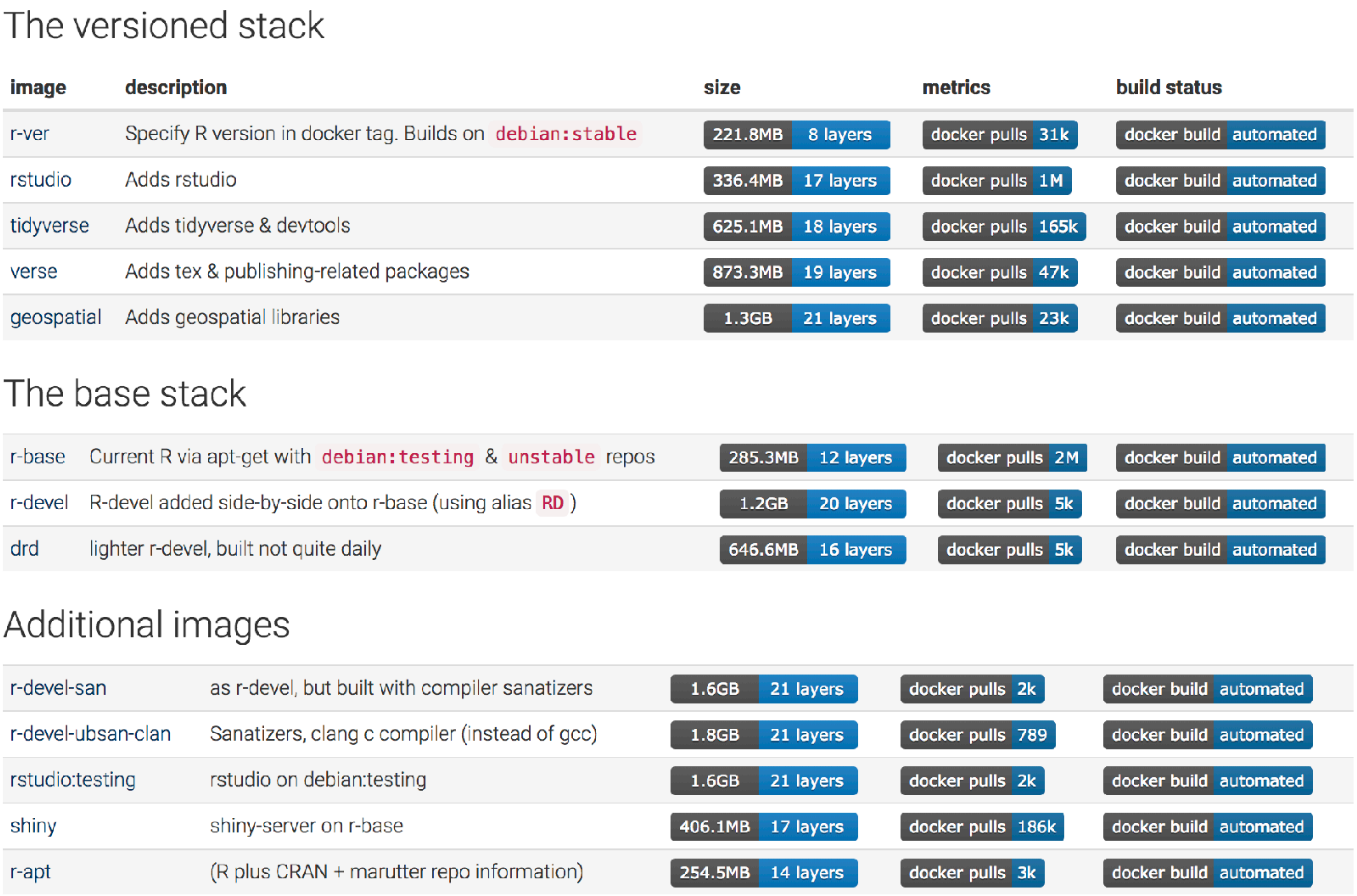
People 2 >

cboettig
Carl Boettiger

eddelbuettel
Dirk Eddelbuettel

Boettiger + Eddelbuettel (2017) arXiv:1410.0846

Boettiger (2015) doi:10.1145/2723872.2723882



The versioned stack

image	description	size	metrics	build status
r-ver	Specify R version in docker tag. Builds on `debian:stable`	221.8MB 8 layers	docker pulls 31k	docker build automated
rstudio	Adds rstudio	336.4MB 17 layers	docker pulls 1M	docker build automated
tidyverse	Adds tidyverse & devtools	625.1MB 18 layers	docker pulls 165k	docker build automated
verse	Adds tex & publishing-related packages	873.3MB 19 layers	docker pulls 47k	docker build automated
geospatial	Adds geospatial libraries	1.3GB 21 layers	docker pulls 23k	docker build automated

The base stack

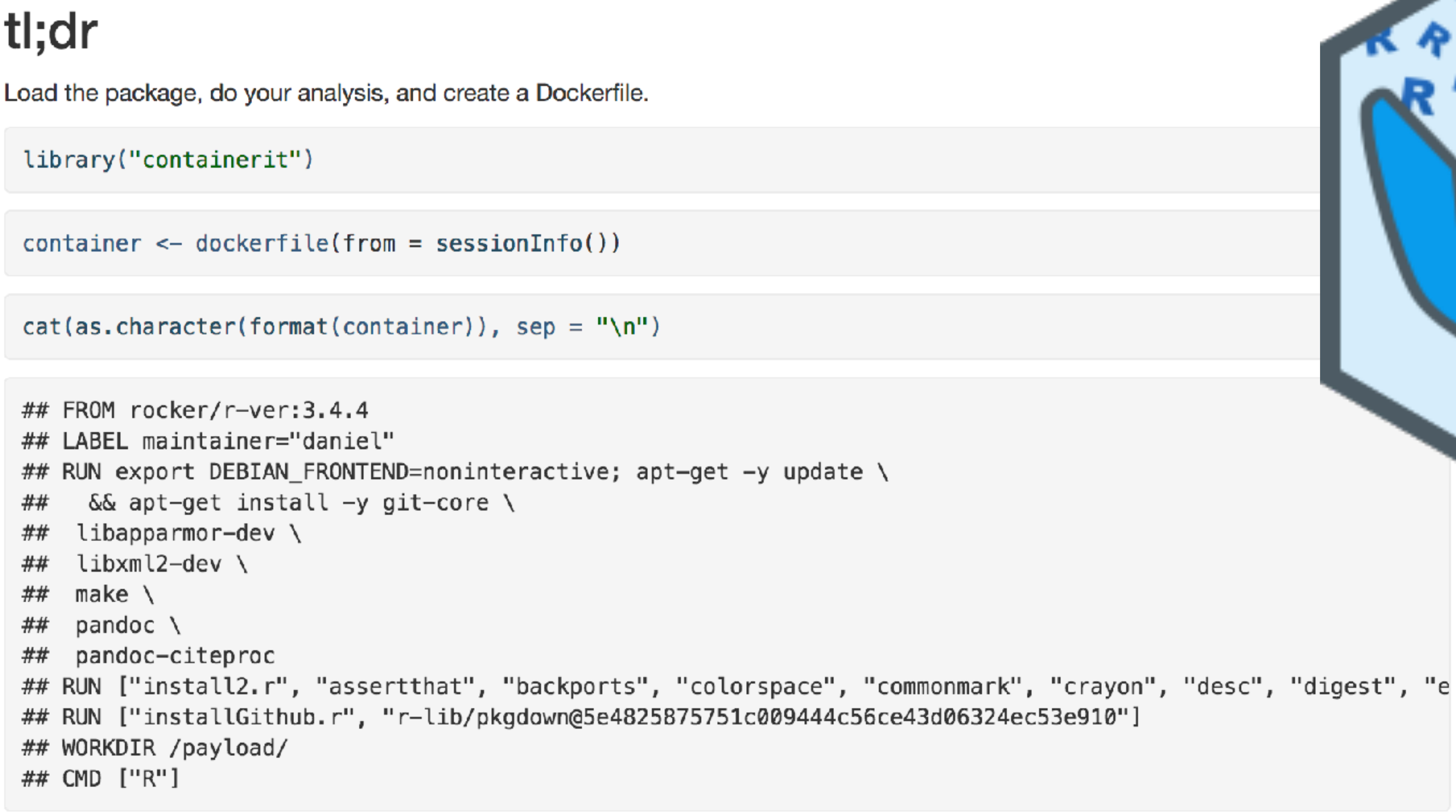
r-base	Current R via apt-get with `debian:testing` & `unstable` repos	285.3MB 12 layers	docker pulls 2M	docker build automated
r-devel	R-devel added side-by-side onto r-base (using alias `RD`)	1.2GB 20 layers	docker pulls 5k	docker build automated
drd	lighter r-devel, built not quite daily	646.6MB 16 layers	docker pulls 5k	docker build automated

Additional images

r-devel-san	as r-devel, but built with compiler sanatizers	1.6GB 21 layers	docker pulls 2k	docker build automated
r-devel-ubsan-clan	Sanatizers, clang c compiler (instead of gcc)	1.8GB 21 layers	docker pulls 789	docker build automated
rstudio:testing	rstudio on debian:testing	1.6GB 21 layers	docker pulls 2k	docker build automated
shiny	shiny-server on r-base	406.1MB 17 layers	docker pulls 186k	docker build automated
r-apt	(R plus CRAN + marutter repo information)	254.5MB 14 layers	docker pulls 3k	docker build automated

Let's Demo!

Reproducibility Helpers: Containerit to Capture Dependencies



tl;dr

Load the package, do your analysis, and create a Dockerfile.

```
library("containerit")
```

```
container <- dockerfile(from = sessionInfo())
```

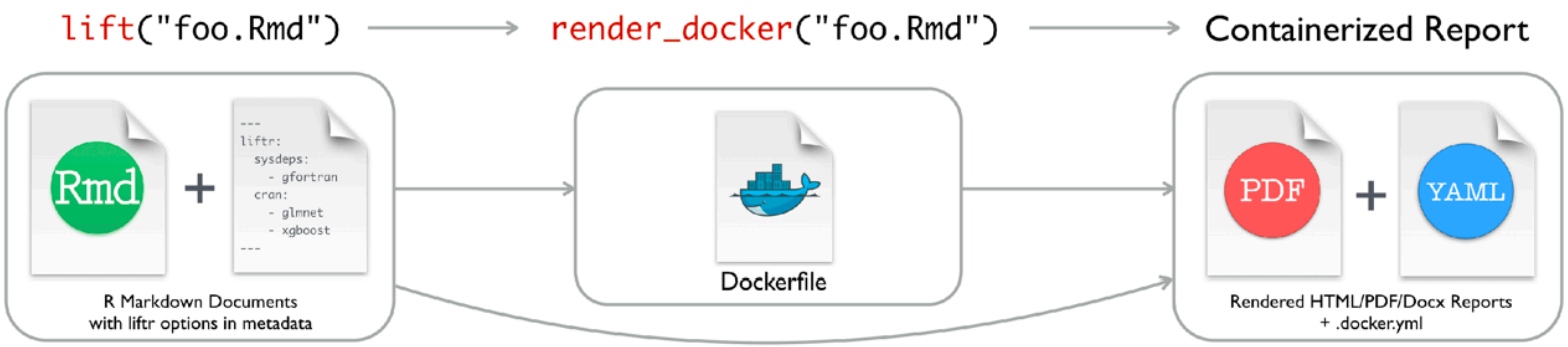
```
cat(as.character(format(container)), sep = "\n")
```

```
## FROM rocker/r-ver:3.4.4
## LABEL maintainer="daniel"
## RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
##   && apt-get install -y git-core \
##  libapparmor-dev \
##  libxml2-dev \
##  make \
##  pandoc \
##  pandoc-citeproc
## RUN ["install2.r", "assertthat", "backports", "colorspace", "commonmark", "crayon", "desc", "digest", "e
## RUN ["installGithub.r", "r-lib/pkgdown@5e4825875751c009444c56ce43d06324ec53e910"]
## WORKDIR /payload/
## CMD ["R"]
```

o2r.info/containerit

Reproducibility Helpers: Liftr for Dockerized R Markdown

```
title: "The Missing Example of liftr"
author: "Author Name"
date: "2018-06-26"
output: rmarkdown::html_document
liftr:
  maintainer: "Maintainer Name"
  email: "name@example.com"
  from: "rocker/r-base:latest"
  pandoc: true
  texlive: false
  sysdeps:
    - gfortran
  cran:
    - glmnet
  bioc:
    - Gviz
  remotes:
    - "road2stat/liftr"
```



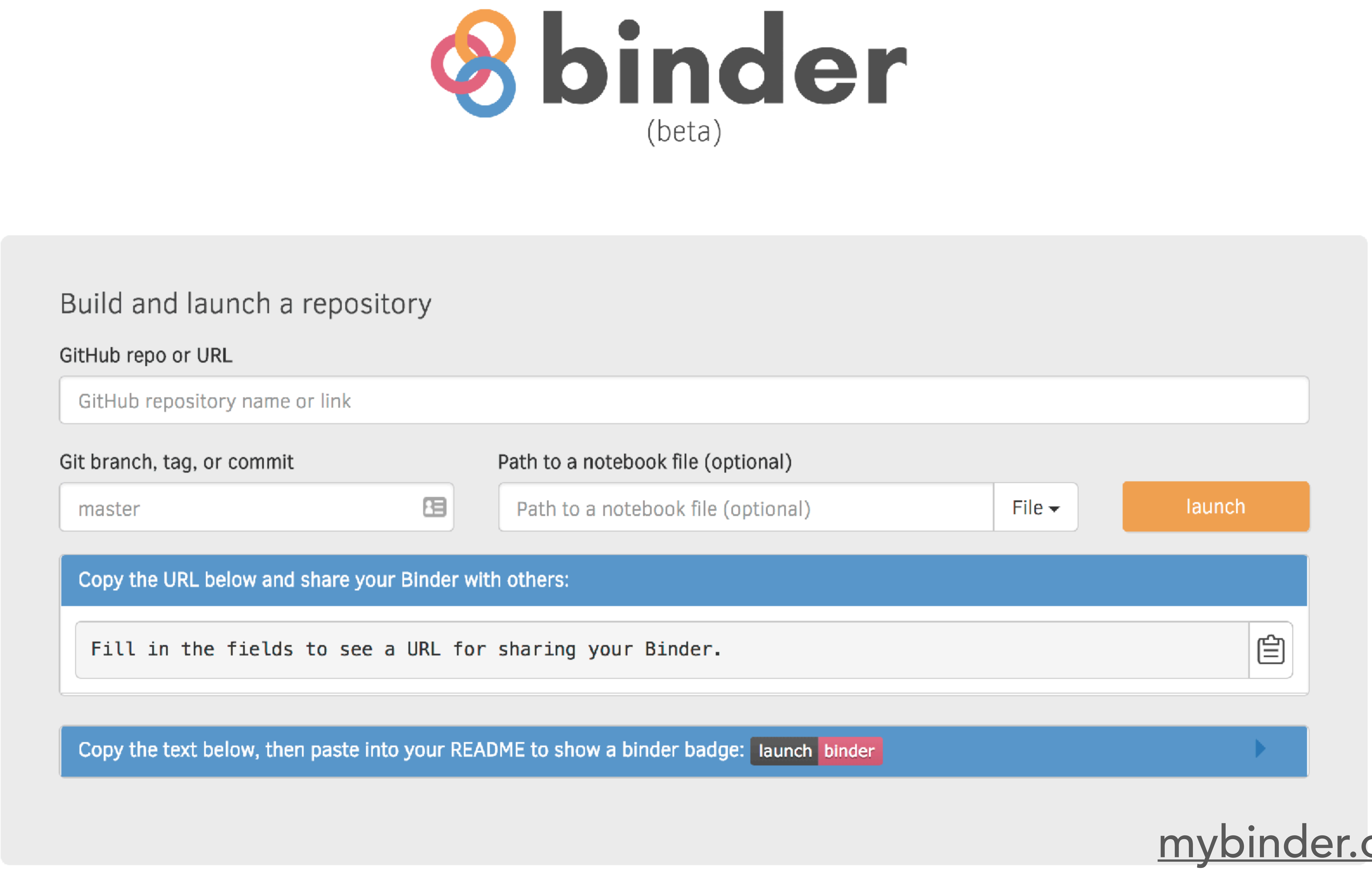
```
---
liftr:
  sysdeps:
    - gfortran
  cran:
    - glmnet
    - xgboost
---
```

liftr extends the R Markdown metadata format, introducing additional options for containerizing and rendering reports.

By running lift() on the RMD file, liftr parses the metadata fields appeared in the R Markdown document; then generates the Dockerfile.

By running render_docker(), liftr will build the Docker image, run the container, and render the R Markdown document.

https://nanx.me/liftr



Build and launch a repository

GitHub repo or URL

GitHub repository name or link

Git branch, tag, or commit

master

Path to a notebook file (optional)

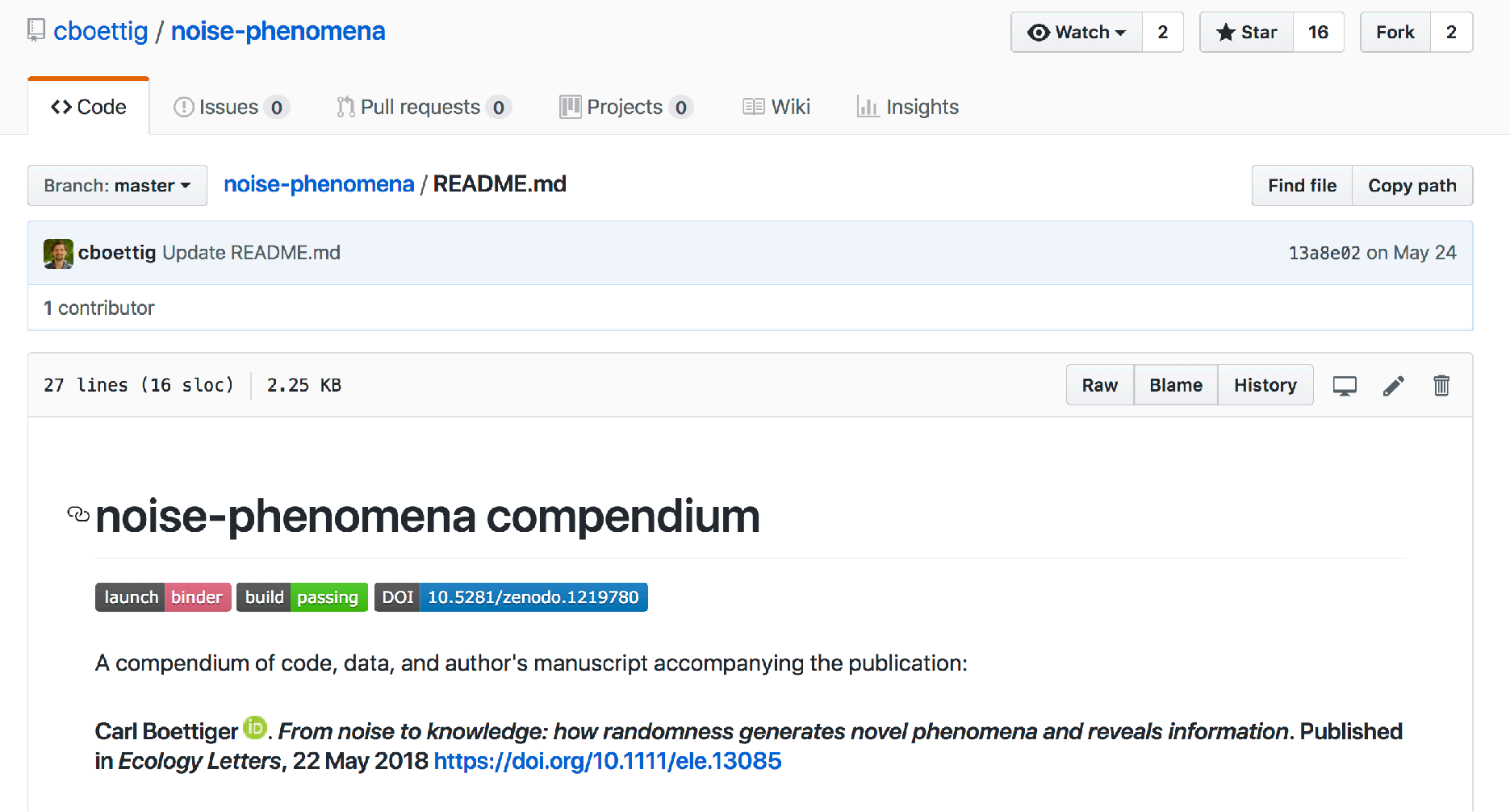
Path to a notebook file (optional)

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

Copy the text below, then paste into your README to show a binder badge:

mybinder.org



cboettig / noise-phenomena

noise-phenomena / README.md

cboettig Update README.md 13a8e02 on May 24

1 contributor

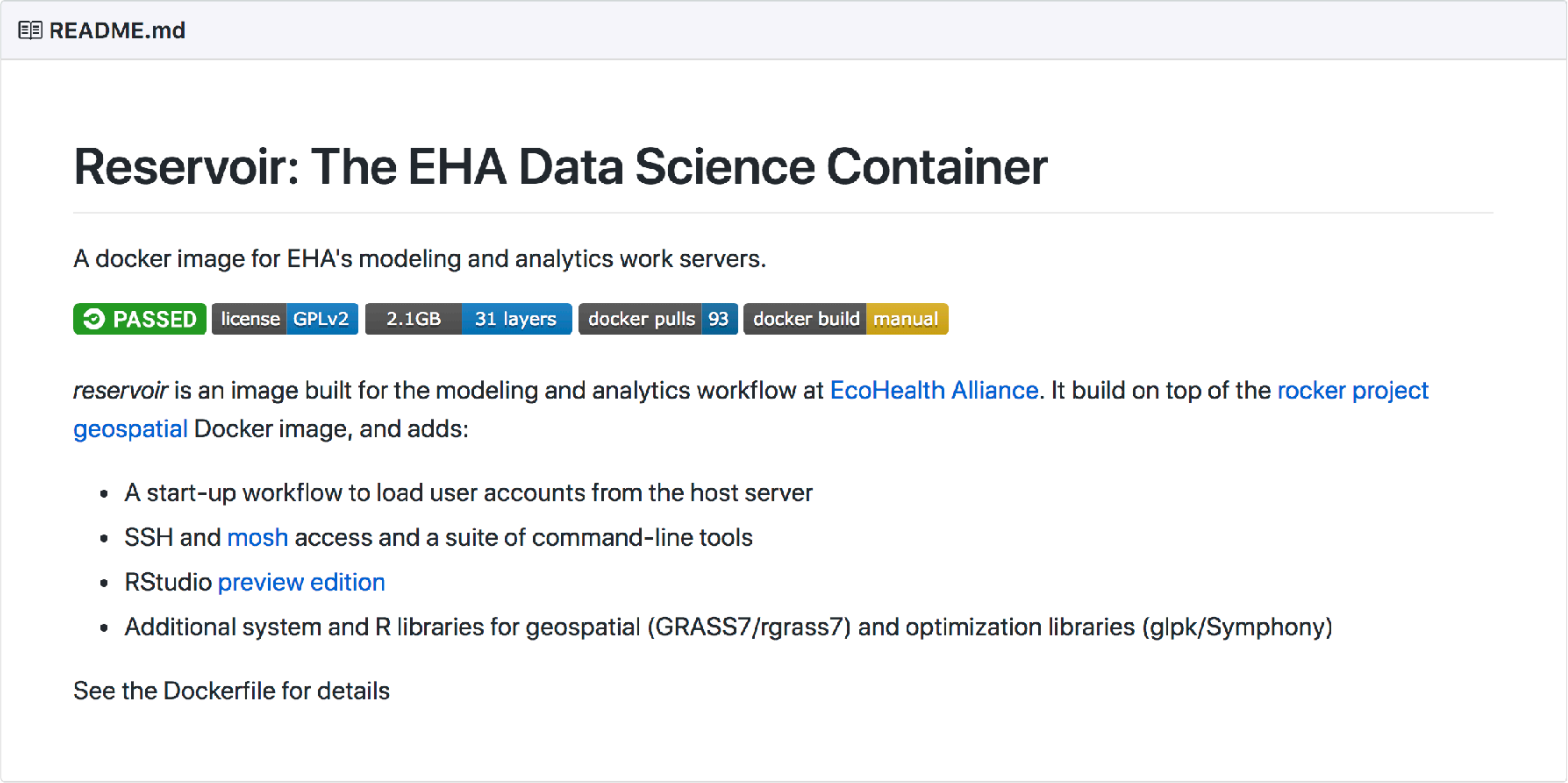
27 lines (16 sloc) 2.25 KB

noise-phenomena compendium

DOI 10.5281/zenodo.1219780

A compendium of code, data, and author's manuscript accompanying the publication:

Carl Boettiger. ***From noise to knowledge: how randomness generates novel phenomena and reveals information*. Published in *Ecology Letters*, 22 May 2018 https://doi.org/10.1111/ele.13085**



Reservoir: The EHA Data Science Container

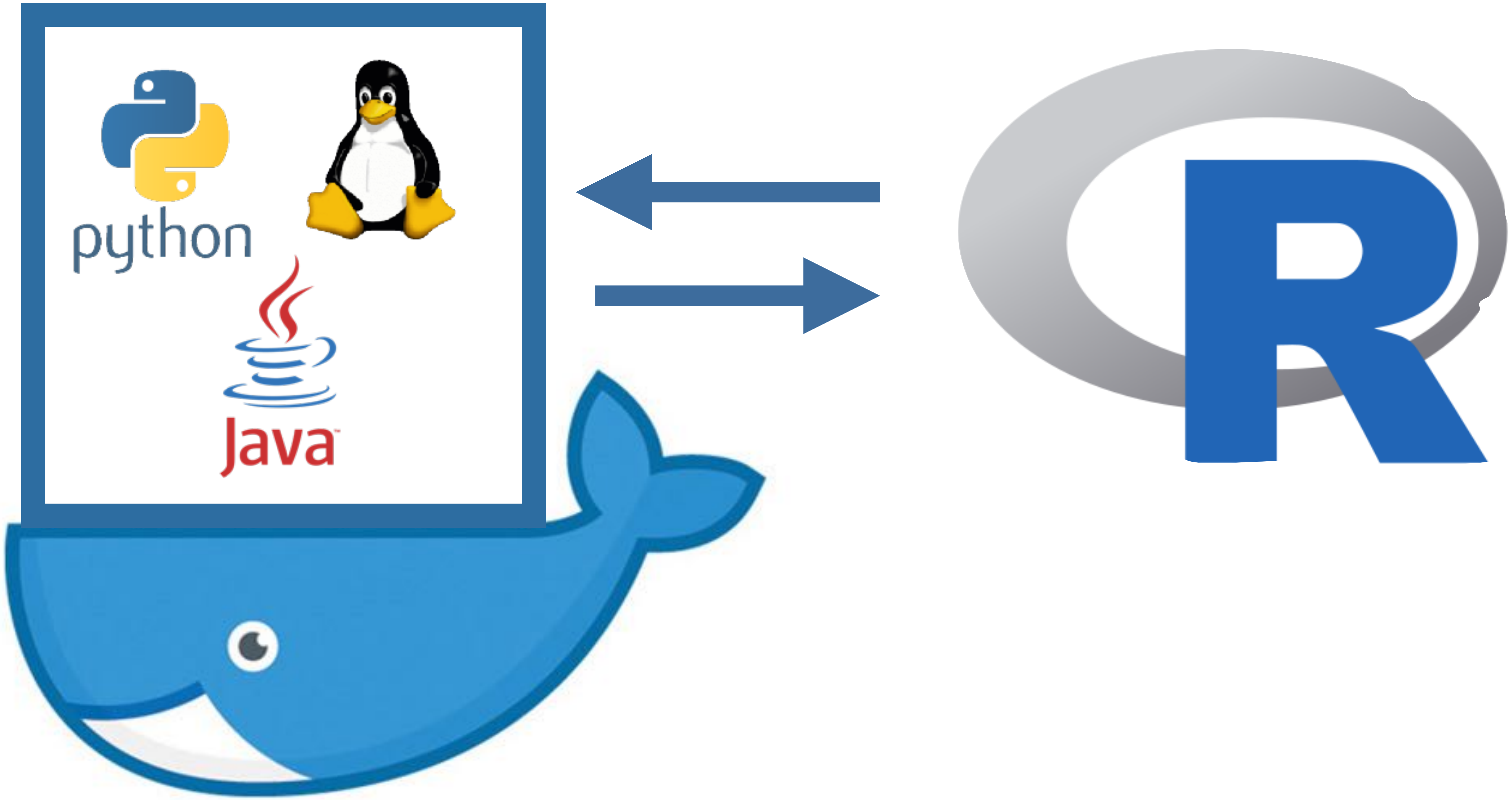
A docker image for EHA's modeling and analytics work servers.

reservoir is an image built for the modeling and analytics workflow at EcoHealth Alliance. It build on top of the rocker project geospatial Docker image, and adds:

- A start-up workflow to load user accounts from the host server
- SSH and mosh access and a suite of command-line tools
- RStudio preview edition
- Additional system and R libraries for geospatial (GRASS7/rgrass7) and optimization libraries (glpk/Symphony)

See the Dockerfile for details

Docker as provider of libraries/services



External library/service examples

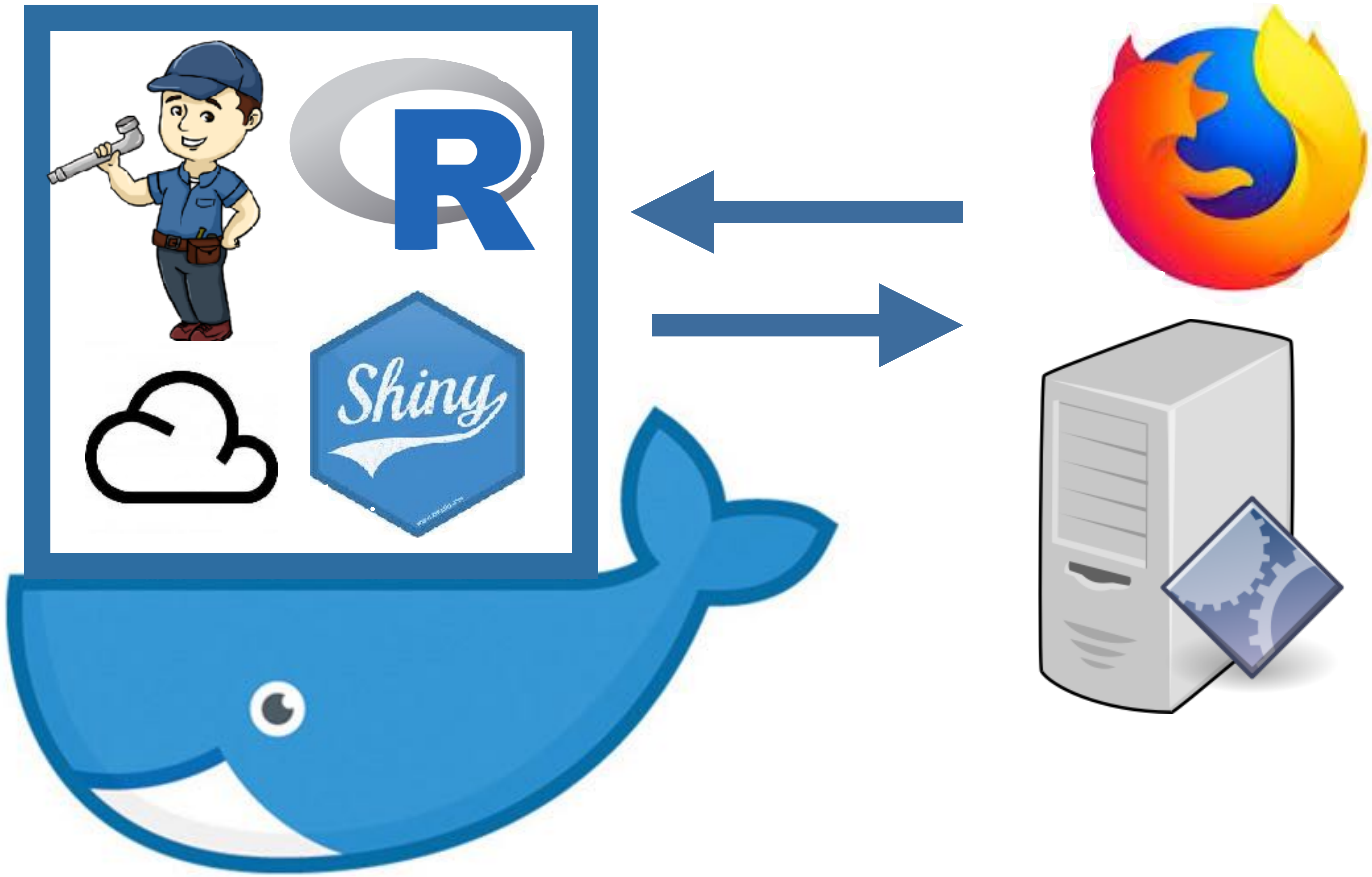
Browser rendering services for web scraping (**splashr**, **RSelenium**)

Geospatial Services (**osrm**)

ML Services (**h2o**)

Let's Demo!

Docker as a platform for R services/apps



Types of R Services

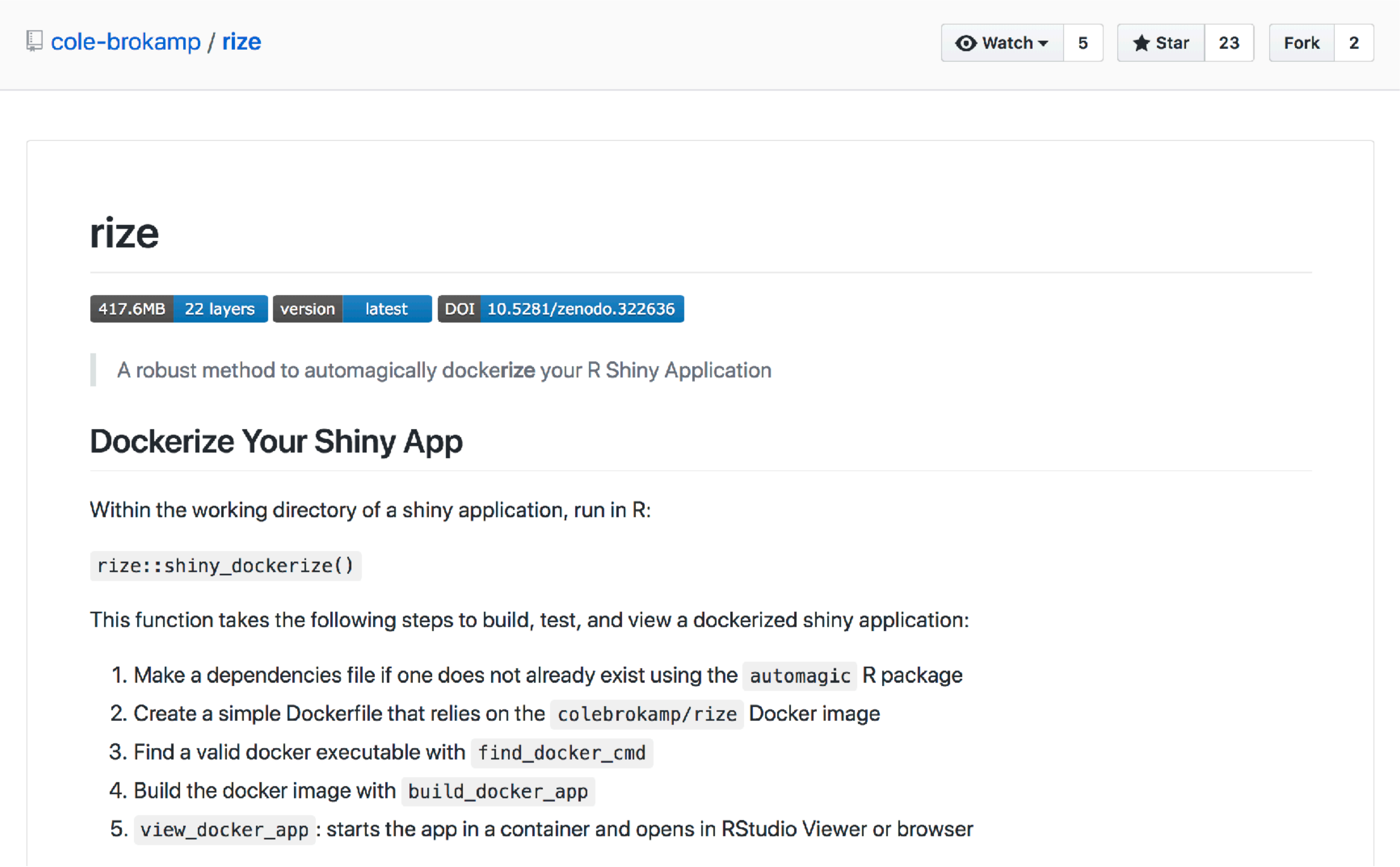
Fiery
Shiny
www.rstudio.com
R
jupyter

Web Apps

APIs

IDEs

Let's Demo!



rize

> A robust method to automagically dockerize your R Shiny Application

Dockerize Your Shiny App

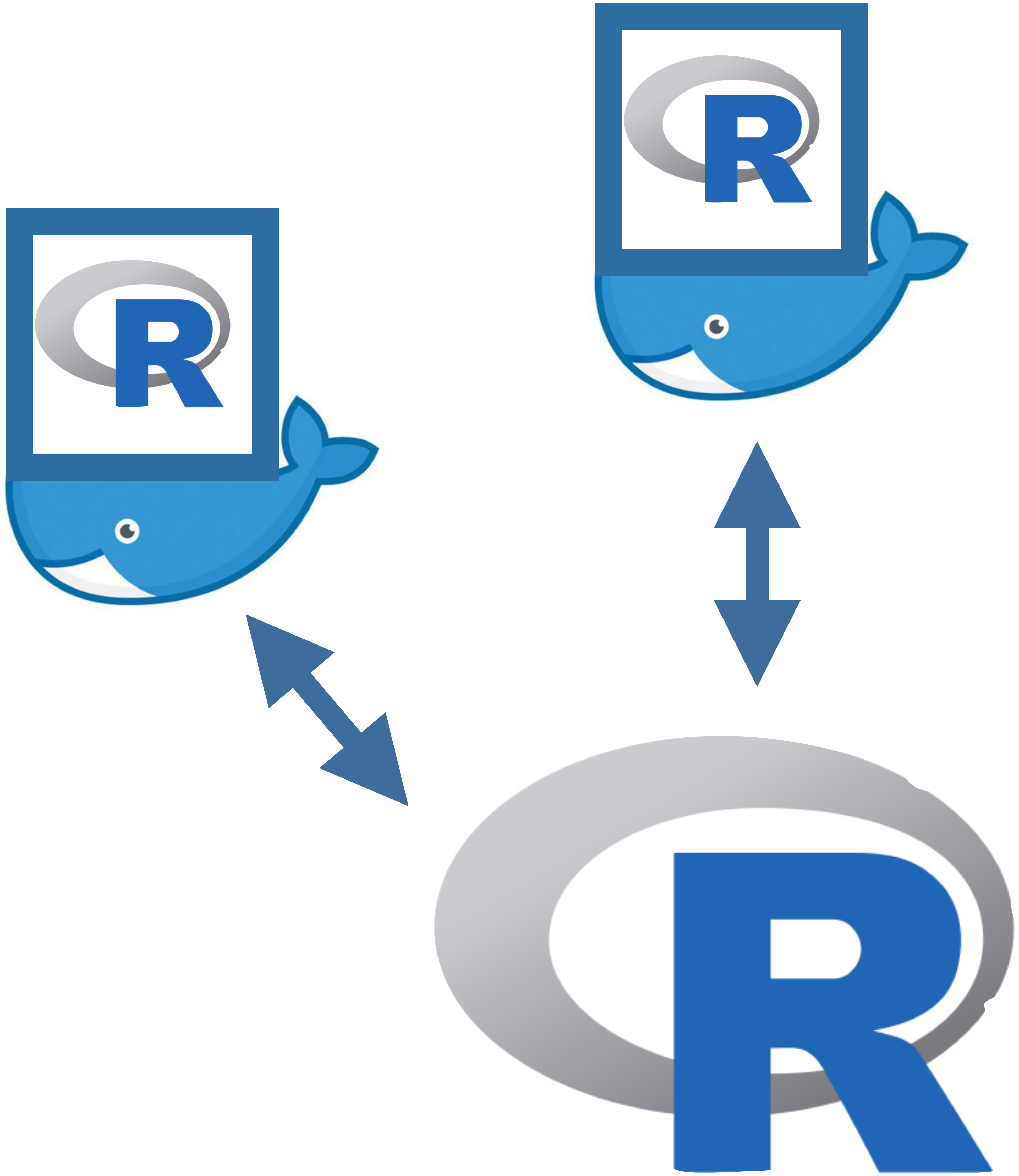
Within the working directory of a shiny application, run in R:

```
rize::shiny_dockerize()
```

This function takes the following steps to build, test, and view a dockerized shiny application:

1. Make a dependencies file if one does not already exist using the `automagic` R package
2. Create a simple Dockerfile that relies on the `colebrokamp/rize` Docker image
3. Find a valid docker executable with `find_docker_cmd`
4. Build the docker image with `build_docker_app`
5. `view_docker_app` : starts the app in a container and opens in RStudio Viewer or browser

Docker for deploying compute jobs

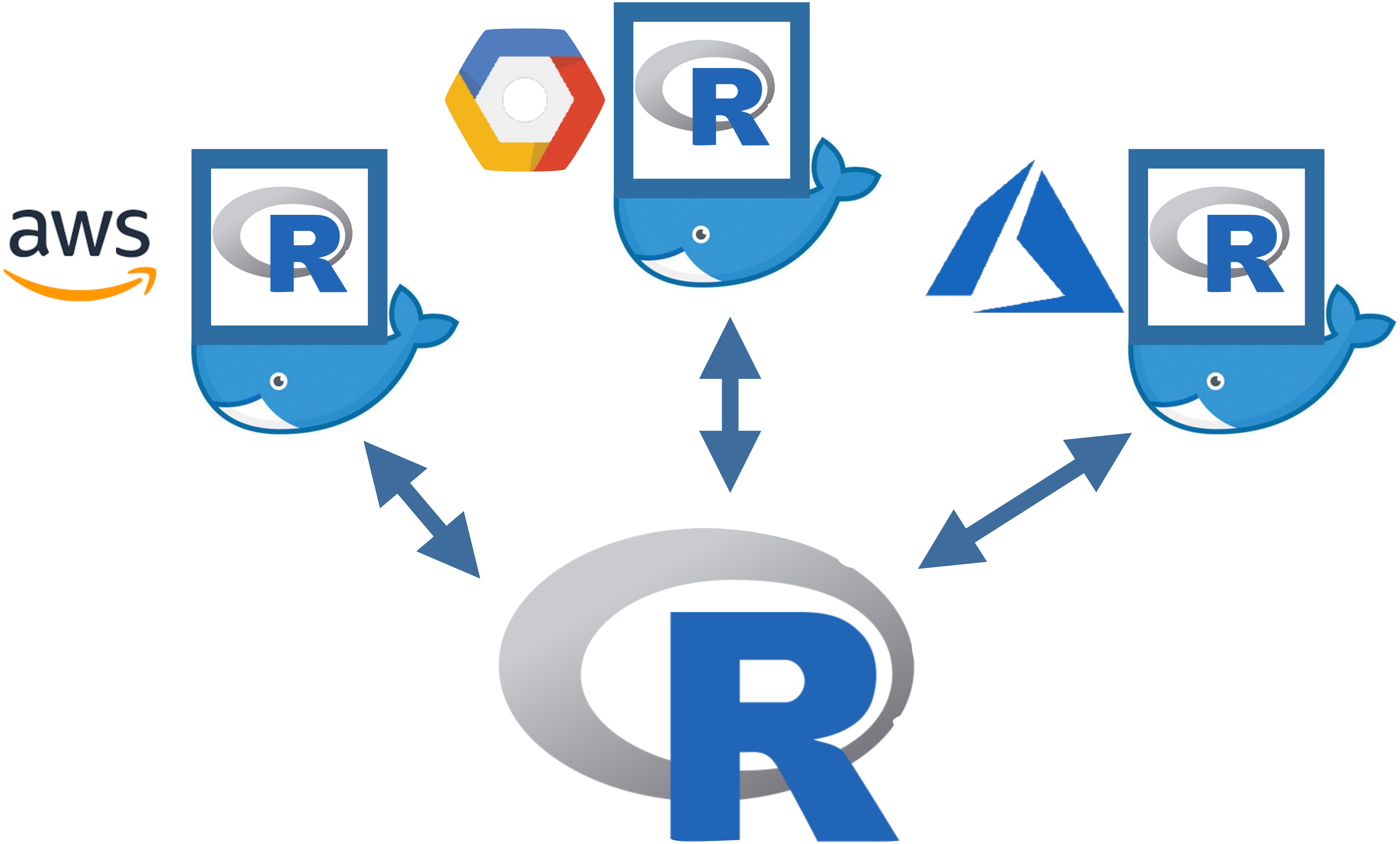


An example with the **future** package

```
library(future)
library(future.apply)
cl <- makeClusterPSOCK(
  workers = c("10.0.0.40", "10.0.0.43"),
  rscript = c("docker", "run",
              "henrikbengtsson/r-base-future", "Rscript"))

plan(cluster, workers = cl)
result <-future.apply(1:10, a_long_function_run_in_docker)
```

Docker for deploying compute jobs



Singularity

Don't need admin rights on computer to use

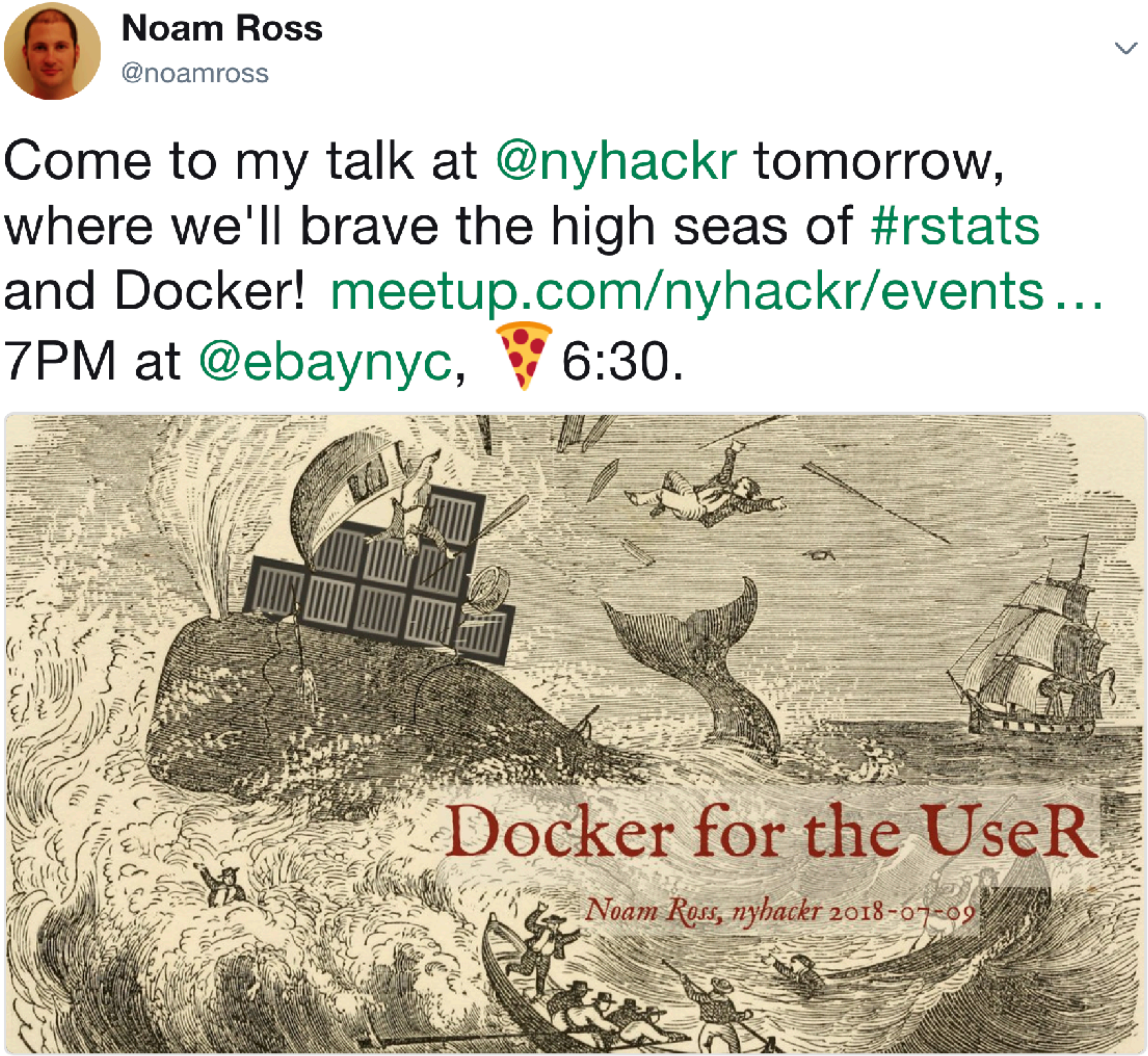
Less isolation - shares files with your system

For packaging up workflows, scripts, utilities

Targets scientific applications and computing clusters

Runs Docker images, too

syslab.io



Noam Ross
@noamross

Come to my talk at @nyhackr tomorrow, where we'll brave the high seas of #rstats and Docker! meetup.com/nyhackr/events… 7PM at @ebaynyc, 🍕 6:30.

Docker for the UseR

Noam Ross, nyhackr 2018-07-09

Henrik Bengtsson
@henrikbengtsson

Replying to @noamross @nyhackr @ebaynyc

Couldn't resist - here's the @SingularityApp version 😝

2:54 AM - 10 Jul 2018

More Resources!

The Dockumentation: docs.docker.com

An rOpenSci Lesson:

ropenscilabs.github.io/r-docker-tutorial/

All links and resources mentioned here will be live tomorrow at:

github.com/noamross/nyhackr-docker-talk

Thanks!

@noamross

@EcoHealthNYC

@rOpenSci