

Docker for R Users



useR! 2022

Rahul Sangole, Rami Krispin

June 20, 2022

Agenda

- Kick Off
- Introduction to Docker

Break [5 min]

- Dockerfile
- Docker CLI

Break [10 min]

- Docker + R Basics
- Intermediate Concepts
- Summary, Q&A

About Us



Rami
raa-me

- Data Science and Engineering Manager
- Time series Analysis and Forecasting
- Open source contributor



Rahul
raa-hul

- Data Scientist
- Time Series Forecasting & Anomaly Detection
- R in Production

Poll

- In the “Polls” Section on the left, you’ll see:
 - What is your experience with Docker?
 - What is your experience with shell, bash, command line tools?
 - Do you have Docker Desktop installed?
 - What type of OS are you using?
 - What type of CPU in your machine?
- On the chat:
 - Which country are you from? And what is your favorite animal?

Code Of Conduct

- Everyone's expected to follow guidelines published here: <https://user2022.r-project.org/about/coc/>
- tl;dr:
 - Be considerate of all participants
 - Be kind and inclusive

Engagement Model

- Ask questions in the “Q&A” section

We'll have Q&A at the end of each sub-section.

We'll answer your questions in the window or live.

- You *don't* have to code along.

In fact, it's more important to absorb the fundamentals, frameworks and design patterns.

All code bases are fully reproducible and available in GitHub repo.

[https://github.com/rsangole/
user2022-r-for-docker](https://github.com/rsangole/user2022-r-for-docker)

Installing Docker Desktop

Installing Docker Desktop

Go to: <https://docs.docker.com/get-docker/>

Get Docker

 Update to the Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) now requires a paid subscription.

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

You can download and install Docker on multiple platforms. Refer to the following section and choose the best installation path for you.



Docker Desktop for Mac

A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.



Docker Desktop for Windows

A native Windows application which delivers all Docker tools to your Windows computer.



Docker Desktop for Linux

A native Linux application which delivers all Docker tools to your Linux computer.

Testing Docker Desktop

Run from the CLI : *docker pull docker/whalesay*

```
docker pull docker/whalesay
```

```
Using default tag: latest
latest: Pulling from docker/whalesay
Image docker.io/docker/whalesay:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better fu
ture compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed3712d2cf: Pull complete
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest
docker.io/docker/whalesay:latest
```

<https://hub.docker.com/r/docker/whalesay/>

Testing Docker Desktop

Run from the CLI : *docker run docker/whalesay cowsay Hello R Users*

```
docker run docker/whalesay cowsay Hello R Users
< Hello R Users >
-----
\ \
  \
    \
      ##          .
      ## ## ##      ==
      ## ## ## ##      ===
      /":-----"\_/_/ ===
~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
      \____ o      _/
          \ \      _/
              \ \_/_/
```

Docker Desktop

Home

Containers

Images

Volumes

Dev Environments PREVIEW

Extensions BETA

Add Extensions

Containers [Give Feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Showing 3 of 3

Search

	NAME	STARTED	STATUS	
<input type="checkbox"/>	docker/whalesay 5a42cd1a4ba5 <small>(practical_robinson)</small>		exited	▶ trash
<input type="checkbox"/>	rkrispin/coronavirus_dashboard_rstudio:dev 8bc26acaeef0e <small>(bold_lumiere)</small>		exited	▶ trash
<input type="checkbox"/>	rkrispin/coronavirus_dashboard_rstudio:dev fc93852301f0 <small>(friendly_chandrasekhar)</small>		exited	▶ trash

Docker Desktop

Preferences X

General Resources Advanced

Resources

- ADVANCED
- FILE SHARING
- PROXIES
- NETWORK

Docker Engine

- Experimental Features
- Kubernetes
- Software Updates
- Extensions

Resources

CPUs: 3

Memory: 8.00 GB

Swap: 1 GB

Disk image size: 64 GB (38.6 GB used)

Cancel Apply & Restart

RAM 4.94GB CPU 0.12% Not connected to Hub v4.9.1 Q

Docker Hub

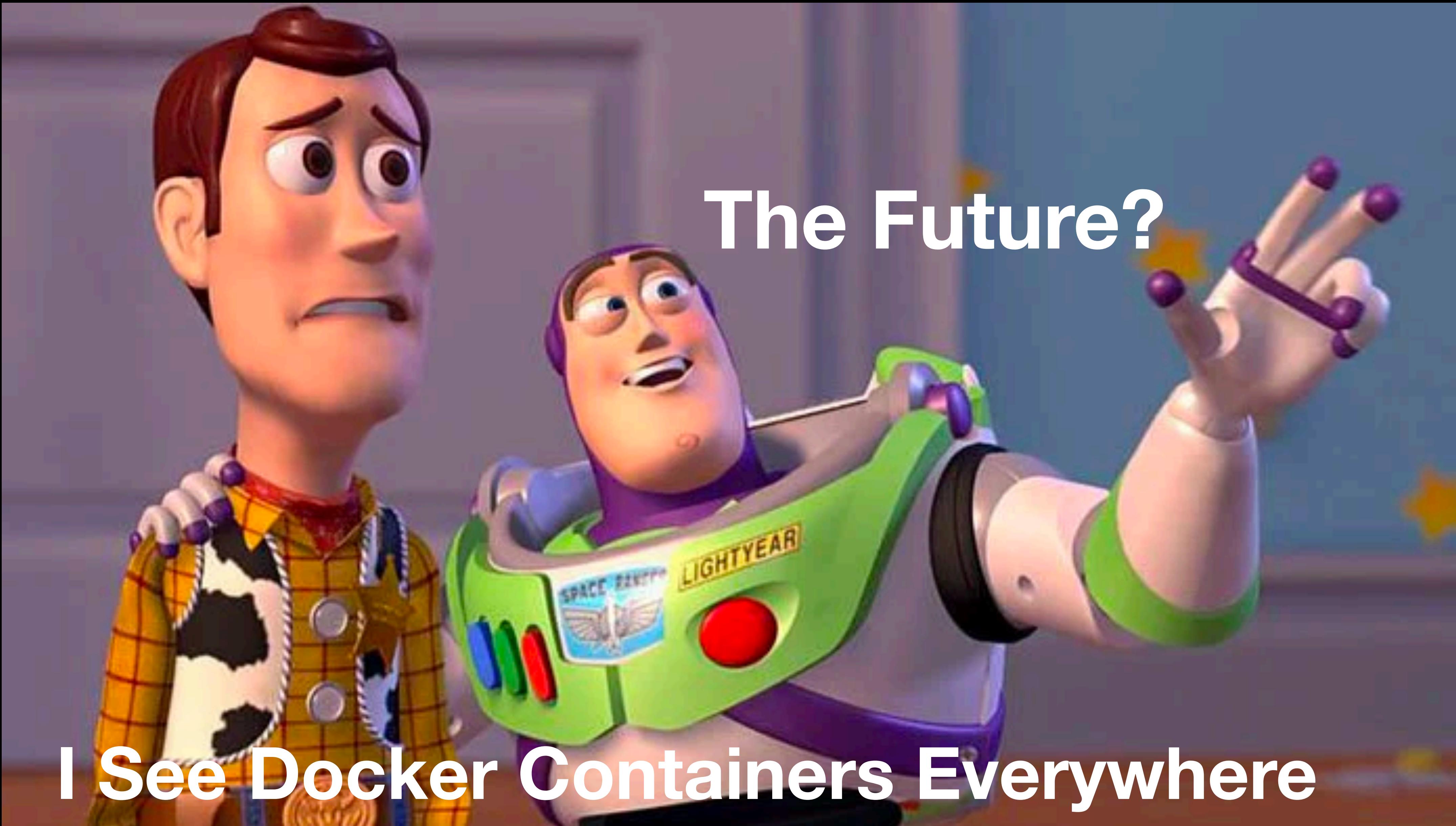
Screenshot of the Docker Hub homepage showing search results for popular images.

Filters: Products (Images, Plugins), Trusted Content (Docker Official Image, Verified Publisher, Open Source Program), Operating Systems (Linux, Windows), Architectures (ARM, ARM 64, IBM POWER, IBM Z, PowerPC 64 LE, x86, x86-64).

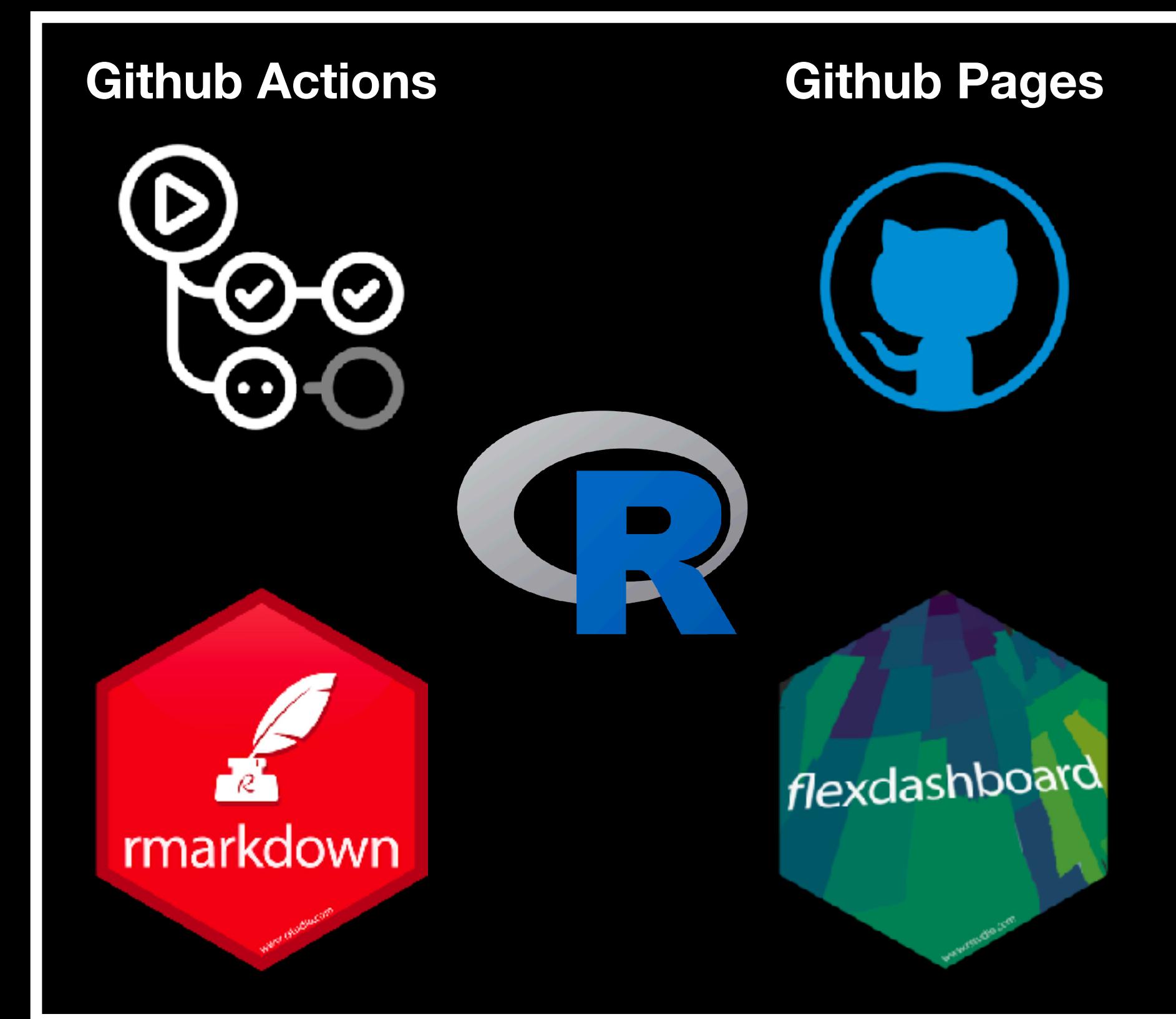
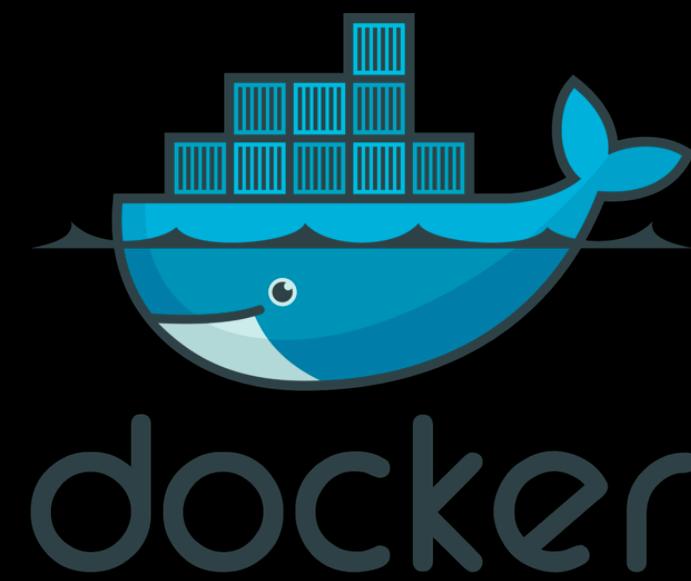
Search results:

- ubuntu** (DOCKER OFFICIAL IMAGE)
Updated 12 days ago
Ubuntu is a Debian-based Linux operating system based on free software.
Architectures: Linux, ARM, ARM 64, PowerPC 64 LE, riscv64, IBM Z, 386, x86-64
Downloads: 1B+ Stars: 10K+
- alpine** (DOCKER OFFICIAL IMAGE)
Updated a month ago
A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!
Architectures: Linux, IBM Z, riscv64, x86-64, ARM, ARM 64, 386, PowerPC 64 LE
Downloads: 1B+ Stars: 8.9K
- redis** (DOCKER OFFICIAL IMAGE)
Updated 6 days ago
Redis is an open source key-value store that functions as a data structure server.
Architectures: Linux, Windows, mips64le, PowerPC 64 LE, IBM Z, k86-64, ARM, ARM 64, 386
Downloads: 1B+ Stars: 10K+
- postgres** (DOCKER OFFICIAL IMAGE)
Updated 12 days ago
The PostgreSQL object-relational database system provides reliability and data integrity.
Architectures: Linux, ARM 64, 386, PowerPC 64 LE, IBM Z, mips64le, x86-64, ARM
Downloads: 1B+ Stars: 10K+

Introduction to Docker

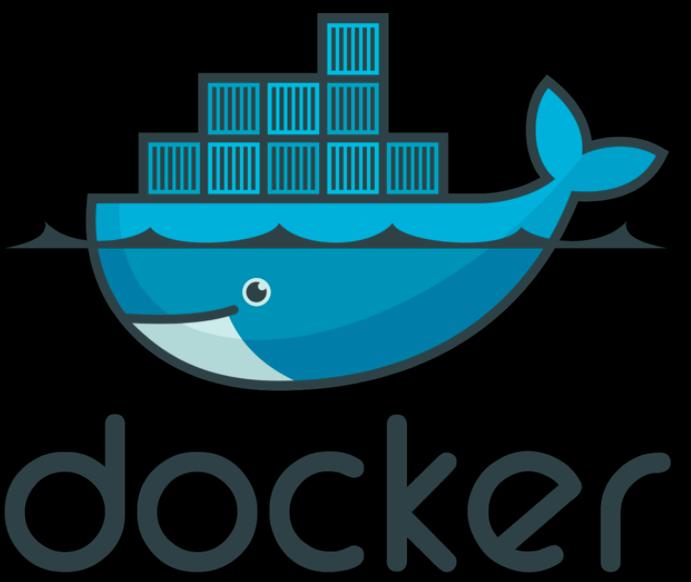


Docker + R = ❤️ ❤️ ❤️



Docker + R = ❤️ ❤️ ❤️

- CI/CD
- Deploy Shiny and flex dashboards
- Data automation
- R package automation
- Modeling
- Reproducibility



Github Actions



Github Pages



Data Pipeline

Rmarkdown

Coronavirus Dataset Data Pipeline

Rami Krispin

17 June, 2022

Functions

```
```{r}
library(magrittr)

function for parsing the JH covid19 cases csv file
parse_url <- function(url, type){
 raw <- readr::read_csv(file = url) %>%
 as.data.frame()

 # Transforming the data from wide to long
 # Creating new data frame
 df <- raw[, 1:4]

 names(df) <- gsub(pattern= "/", replacement = "_", x = names(df))
 # Take diff
 for(i in 5:ncol(raw)){
 raw[,i] <- as.integer(raw[,i])

 if(i == 5){
 df[[names(raw)[i]]] <- raw[, i]
 } else {
 df[[names(raw)[i]]] <- raw[, i] - raw[, i - 1]
 }
 }

 df1 <- df %>% tidyr::pivot_longer(cols = dplyr::starts_with(c("1", "2", "3", "4", "5", "6", "7", "8", "9")),
 names_to = "date_temp",
 values_to = "cases_temp") %>%
 dplyr::mutate(date = lubridate::mdy(date_temp)) %>%
 dplyr::group_by(Province.State, Country.Region, Lat, Long, date) %>%
 dplyr::summarise(cases = sum(cases_temp),
 .groups = "drop") %>%
 dplyr::ungroup() %>%
 dplyr::mutate(type = type,
 Country.Region = trimws(Country.Region),
 Province.State = trimws(Province.State))

 if(type == "recovered"){
 df1$cases <- ifelse(df1$date > as.Date("2021-08-04") & df1$cases == 0 |
 df1$date > as.Date("2021-08-04") & df1$cases < 0, NA, df1$cases)
 }
 }

 return(df1)
}
```

```

Github Actions

| Data Pipeline | | | |
|------------------------------|--------------------------------|--------------|---------|
| data_refresh.yml | | | |
| 4,281 workflow runs | | | |
| Event ▾ | Status ▾ | Branch ▾ | Actor ▾ |
| ✓ Data Pipeline | Data Pipeline #4281: Scheduled | 4 hours ago | ... |
| ✓ Data Pipeline | Data Pipeline #4280: Scheduled | 12 hours ago | ... |
| ✓ Data Pipeline | Data Pipeline #4279: Scheduled | 19 hours ago | ... |
| ✓ Data Pipeline | Data Pipeline #4278: Scheduled | yesterday | ... |
| ✓ Data Pipeline | Data Pipeline #4277: Scheduled | 2 days ago | ... |
| ✓ Data Pipeline | Data Pipeline #4276: Scheduled | 2 days ago | ... |
| ✓ Data Pipeline | Data Pipeline #4275: Scheduled | 2 days ago | ... |
| ✓ Data Pipeline | Data Pipeline #4274: Scheduled | 3 days ago | ... |

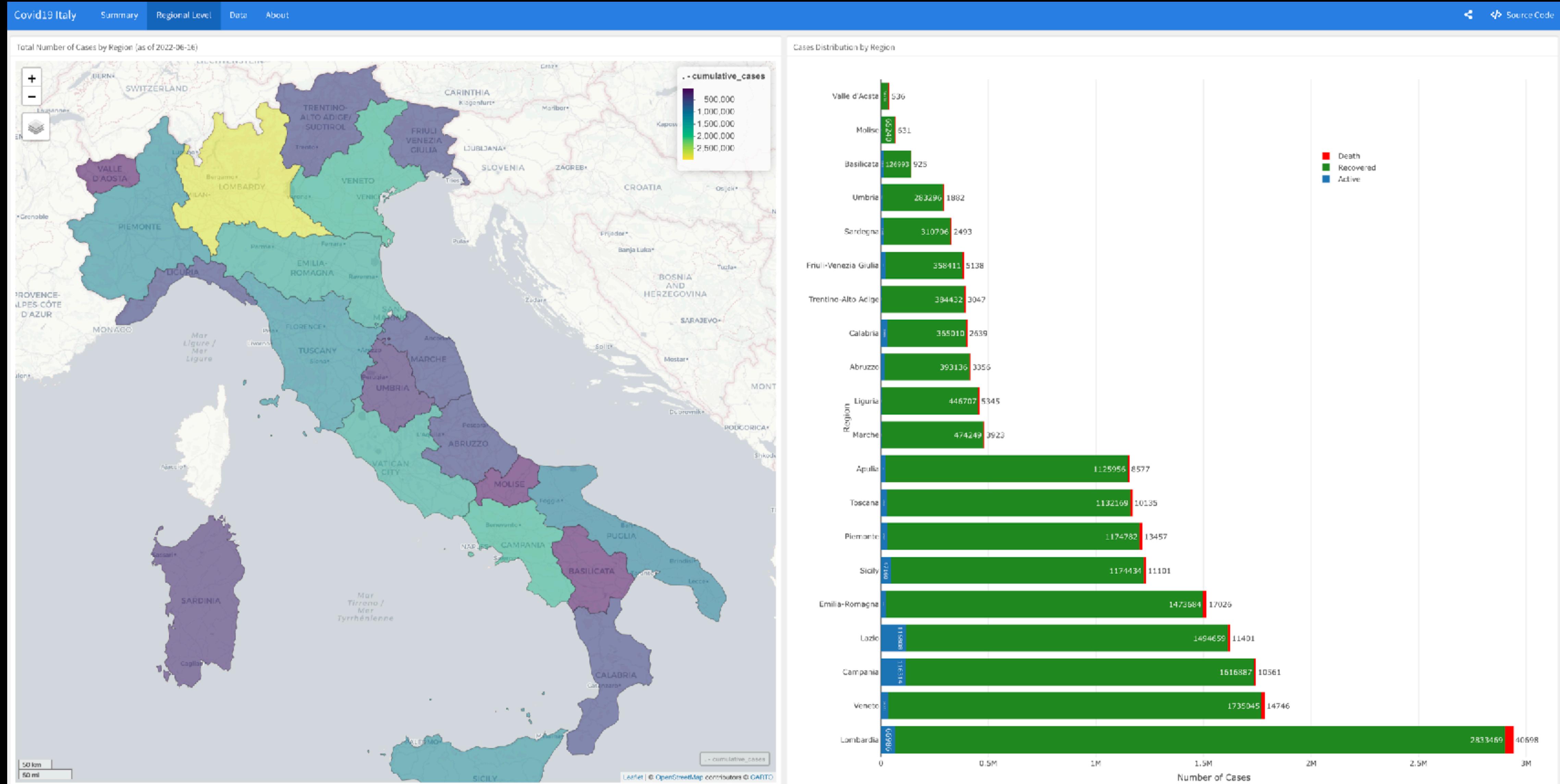
R CMD Check

The screenshot shows a GitHub repository page for 'RamKrispin/coronavirus'. At the top, there's a banner indicating potential security vulnerabilities in dependencies. Below the banner, the repository's main page displays a list of recent commits from the 'master' branch. The commits are as follows:

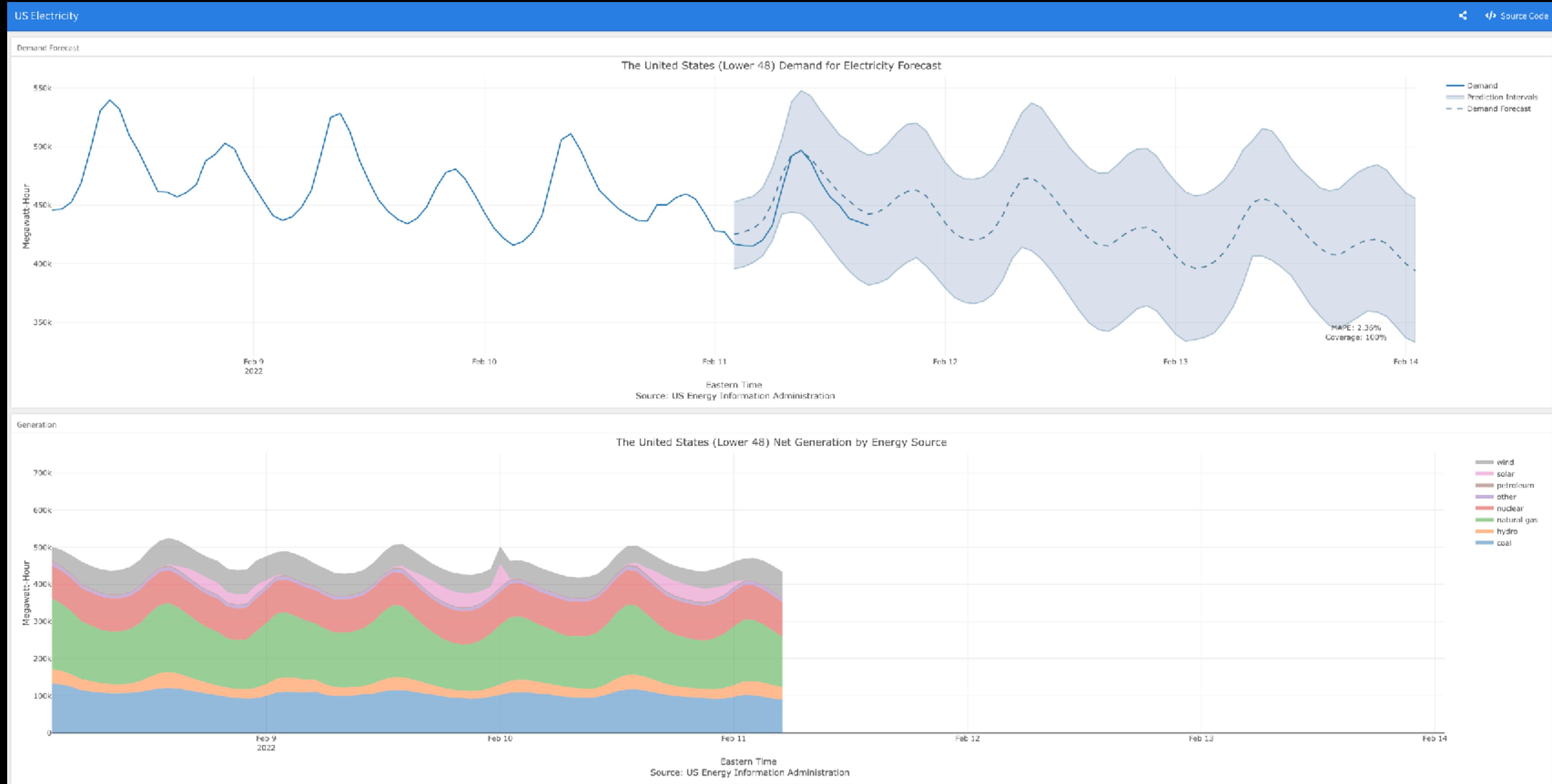
- RamiKrispin Auto update coronavirus data (44e87af) - 4 hours ago
- .github/workflows Update data_refresh.yml (8 months ago)
- R fix typo (8 months ago)
- covid19_env set julia env (12 days ago)
- csv Auto update coronavirus data (4 hours ago)
- data Auto update coronavirus data (4 hours ago)
- data_pipelines Auto update coronavirus data (4 hours ago)
- data_raw update the pipeline (8 months ago)
- docker Fix merge conflict (7 hours ago)
- docs Auto update coronavirus data (4 hours ago)
- man update the docs (12 days ago)
- tests add tests for the vaccine dataset (10 months ago)
- vignettes update the docs (12 days ago)
- .Rbuildignore set julia env (12 days ago)
- .gitignore Updated so vignettes are built (2 years ago)
- CRAN-RELEASE cran release notes (8 months ago)
- DESCRIPTION auto update roxygen (4 months ago)
- LICENSE Updating the package license (2 years ago)
- LICENSE.ind Updating the package license (2 years ago)
- NAMESPACE update the refresh function (17 months ago)
- NEWS.md update the package version and news (8 months ago)
- README.Rmd I corrected some typo in your README.RMD files (4 months ago)
- README.ind update the docs (13 days ago)
- _pkgdown.yml update the title (8 months ago)
- coronavirus.Rproj init commit (2 years ago)

On the right side of the page, there are sections for 'About', 'Releases', 'Packages', 'Contributors', 'Environments', and 'Languages'. The 'About' section includes details like the repository URL (`ramikrispin.github.io/coronavirus/`), license (Unknown, MIT licenses found), and statistics (462 stars, 29 watching, 202 forks). The 'Languages' section shows a breakdown where HTML is 98.1%, R is 1.4%, and Other is 0.5%.

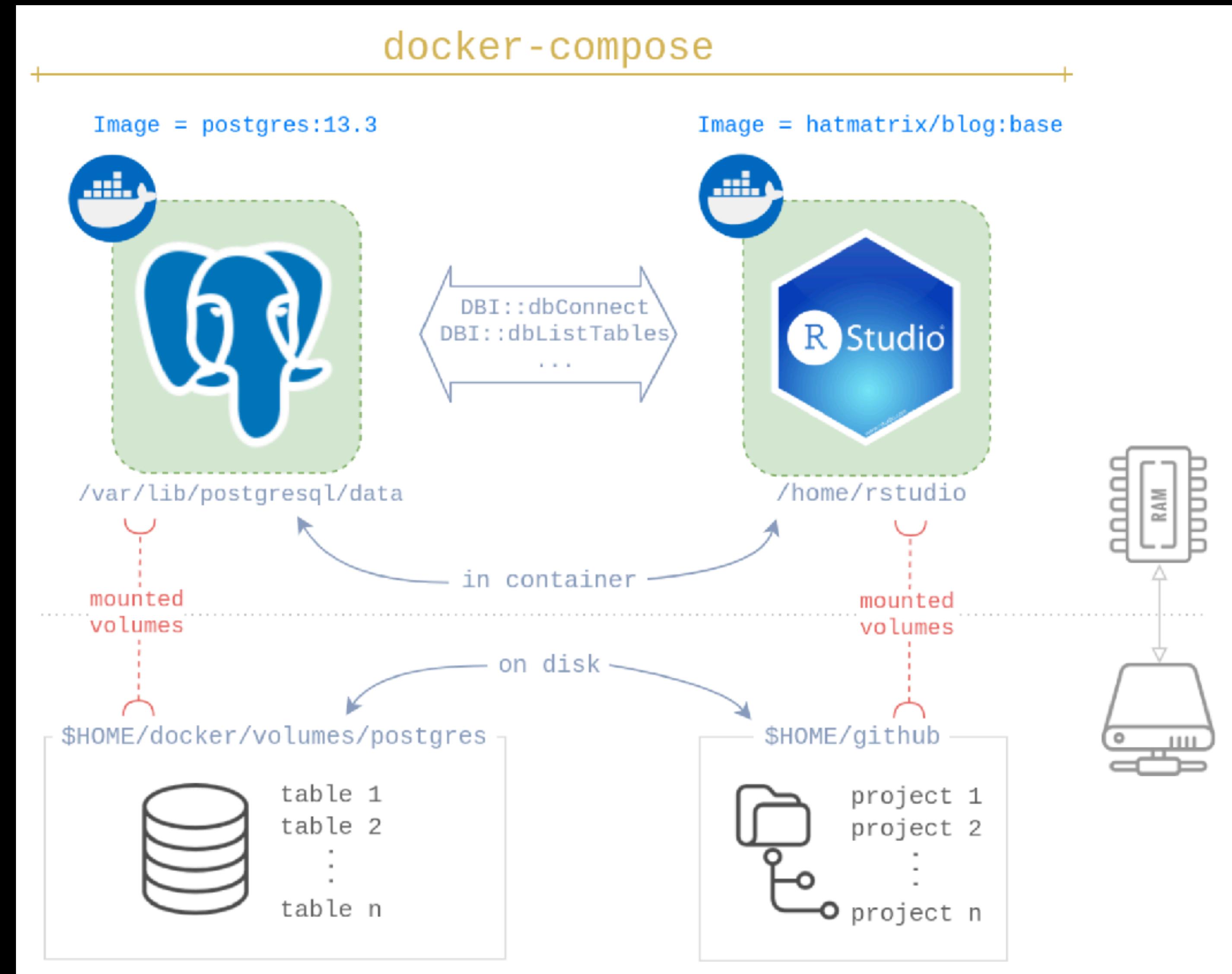
Flexdashboard



Machine Learning

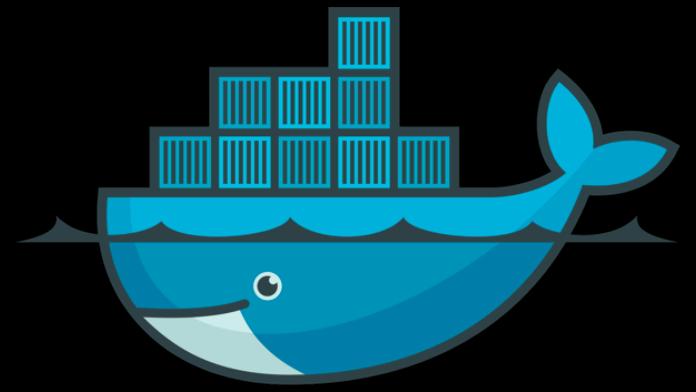


Postgres & R



Seamless Reproducibility & Production

Reproducibility



docker

```
> set.seed(12345)
> rnorm(n = 2)
[1] 0.5855288 0.7094660
> rnorm(n = 2)
[1] -0.1093033 -0.4534972
```

set.seed()



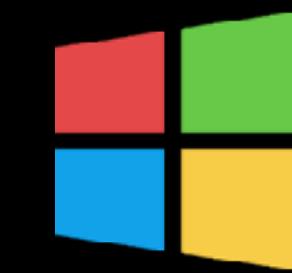
Version Control



Package/IDE Version



R Version



OS/Version



Hardware

The Reproducibility Problem



`set.seed(1234)`



Error



Prof.
R 3.5.0
dplyr 0.8.5
tidyr 0.8.0



X = 5.297



X = 5.298



X = 5.3



Student 1
R 3.6.2
dplyr 1.0.0
tidyr 1.0.0

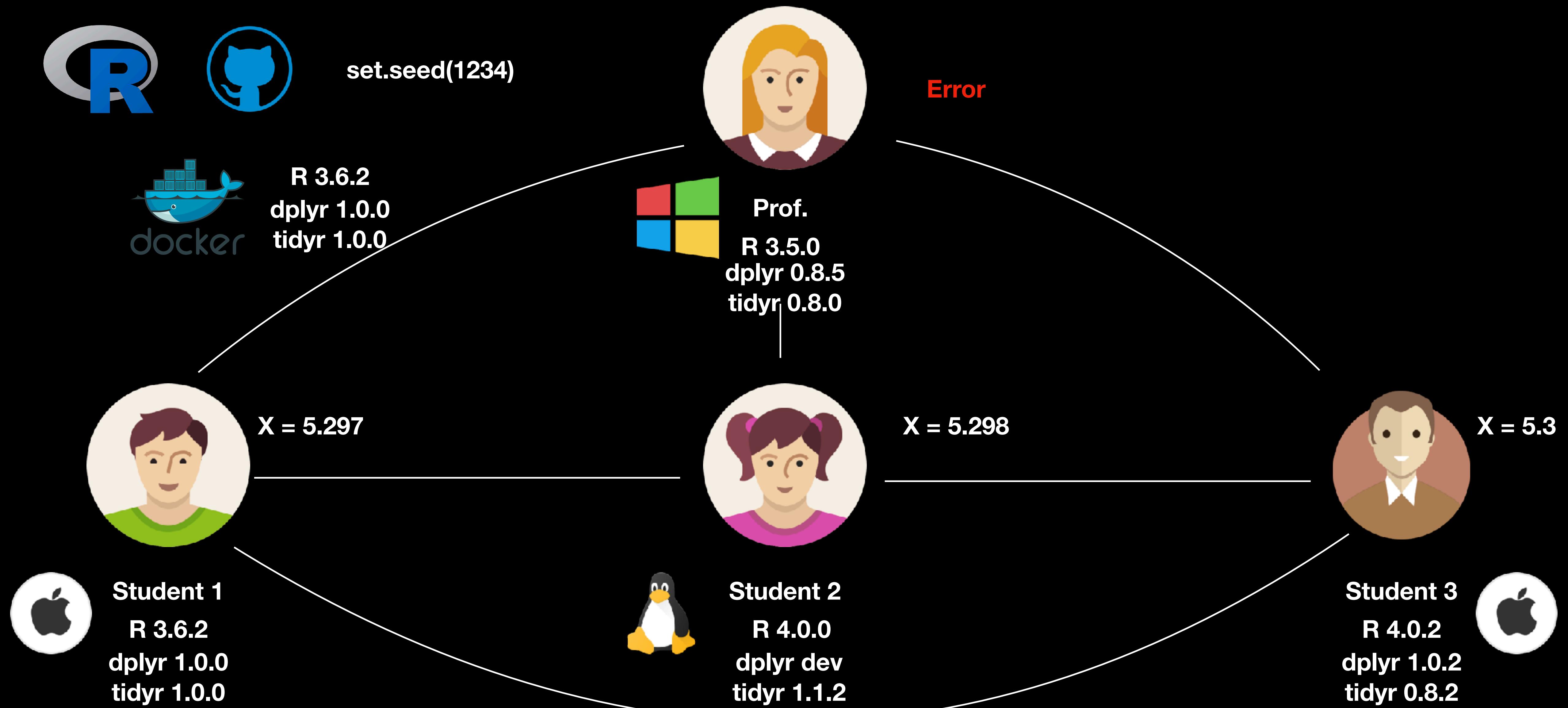


Student 2
R 4.0.0
dplyr dev
tidyr 1.1.2

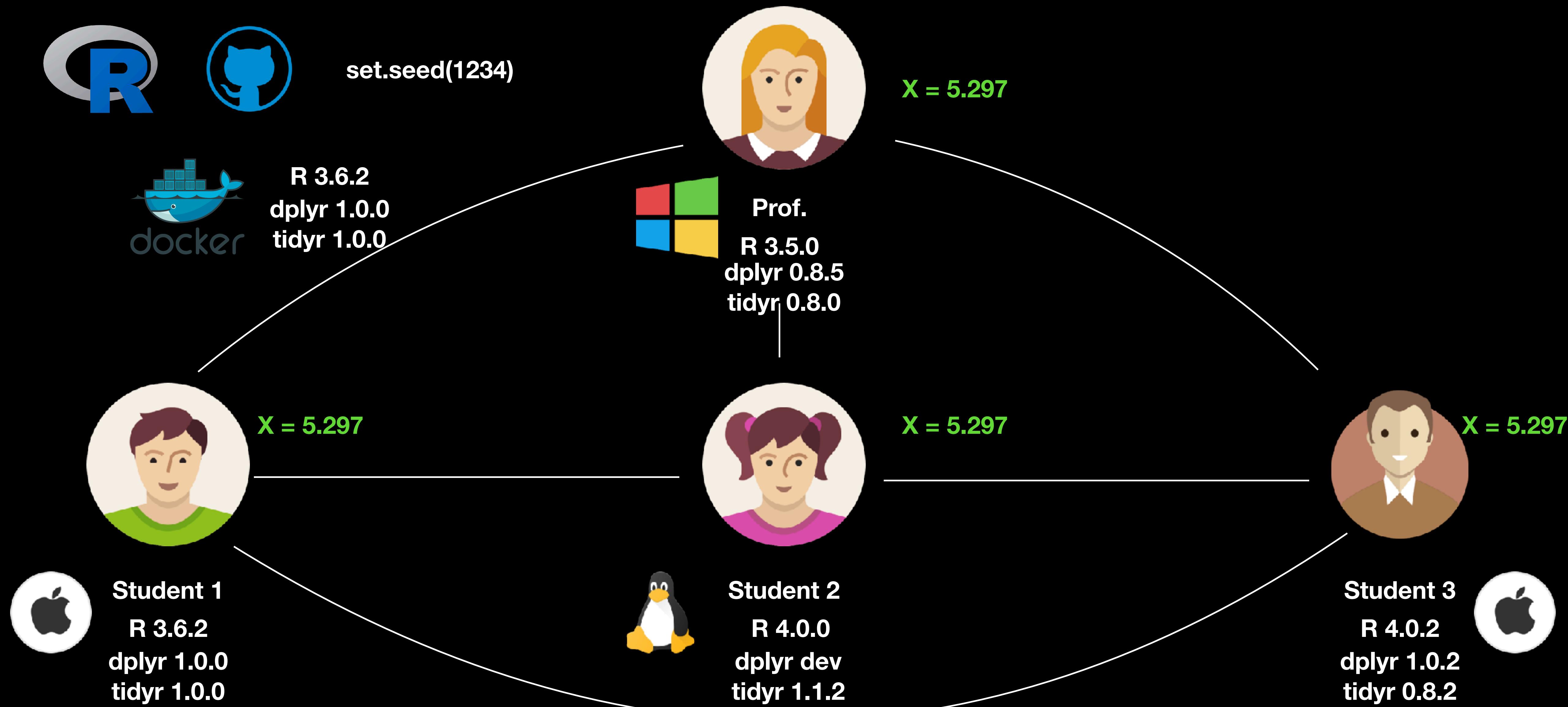


Student 3
R 4.0.2
dplyr 1.0.2
tidyr 0.8.2

The Reproducibility Problem



The Reproducibility Problem



The Production Problem

Shiny app runs locally but fails when deployed due to differing locale

Asked 9 months ago Active 9 months ago Viewed 37 times

I have a Shiny app where you can upload a file, and the data is then processed and relevant outputs appear. This is working totally fine locally. However, I've deployed it with shinyapps.io and now although the app appears fine at first, clearly something is going wrong with the data processing.

I use `data.table:::fread()` for .csv files and `readxl:::read_excel` for excel files. If you upload a .csv file, the app grays out, whereas if you upload an excel file, the outputs appear but basically blank as if no data is there.

The key errors in the logs are all like this:

Warning in `grepl(pattern, vector, ignore.case = ignore.case, fixed = fixed)` : input string 2 is

The Overflow Blog

Podcast 276: Best question on Stack Overflow

The Overflow #42

Featured on Meta

Responding to the commitments made

Linked

R Shiny: works locally but failed on server

Asked 2 months ago Active 1 month ago Viewed 27 times

My shiny dashboard works successfully on R-studio. Recently, I moved it to the AWS EC2 Ubuntu server. I deployed a test app and it works fine. However, the main shiny app doesn't work at all. It says "The application failed to start. The application exited during initialization."

I checked the log and it seems the app cannot recognize any variable from Global Environment which fails the app. Since my data is over 8 GB, the Shiny app would not work if I put "readRDS" inside the app.R file. When I built this app under R-studio, I always load all the files and variables to the global environment before I start my shiny app. It seems this method is not working under the Shiny server.

Is there any other method that I can let my shiny app recognize all the variables that I preloaded to the Global Environment under the shiny-server?

If no, is there any alternative way that I can make my shiny app work and avoid loading 8GB files every time I start it?

Thank you.

Shiny server does not work with my app, which is working in local #2

Open EnricowithR opened this issue on Dec 11, 2016 · 3 comments



EnricowithR commented on Dec 11, 2016

...

I have installed Shiny server on AWS Ubuntu. The default test page works both for rmarkdown and shiny server. However, if I upload a shiny app, which is working in local, it does not work on the shiny server. I tried with index.Rmd and with index.html; in the first case the page only shows the following message:
Error: An error has occurred. Check your logs or contact the app author for clarification.
However the logs, in var/log/shiny-server.log, do not report anything for that.
in the second case the page opens with an empty frame without running the shiny app.

What should I do?

The Production Problem

 **Wrong path in `addResourcePath`** 

shiny

 RamiKrispin 2  2019-09-26

Hello,

I am getting the following error on a Shiny app:

```
Warning: Error in value[[3L]]: Couldn't normalize path in `addResourcePath`,  
with arguments: `prefix` = 'crosstalk-1.0.0';  
`directoryPath` = '/Library/Frameworks/R.framework/Versions/3.5/Resources/li  
[No stack trace available]
```

The error is coming from the `addResourcePath` function when calling by the `crosstalk` package. For some reasons, it assigned wrong path (using version 3.5 instead of version 3.6). Any suggestion how can I modify the path reference?

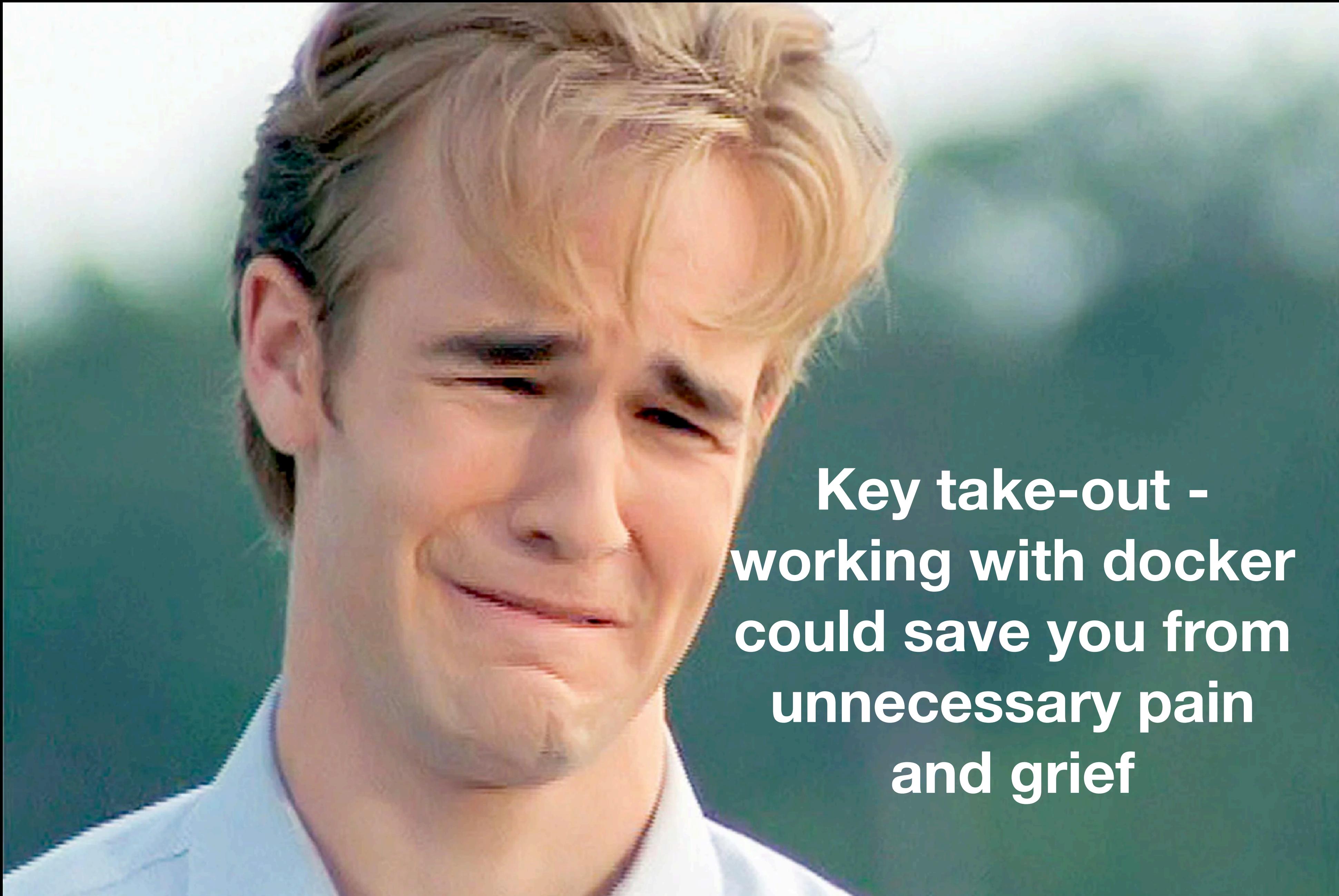
Below is the output of the `.libPaths()` on my machine:

```
> .libPaths()  
[1] "/Library/Frameworks/R.framework/Versions/3.6/Resources/library"
```

Thanks,
Rami

The Production Problem



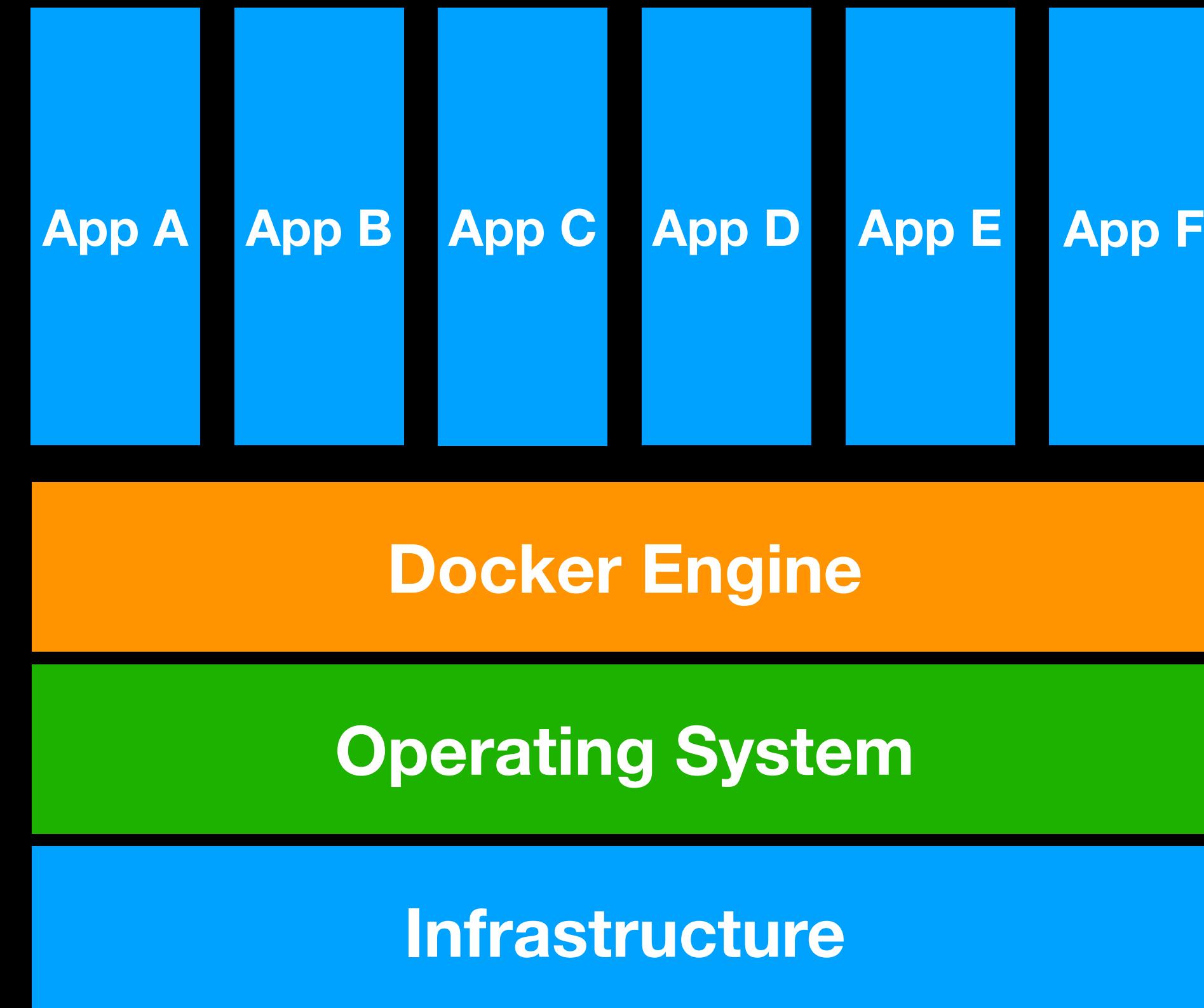


**Key take-out -
working with docker
could save you from
unnecessary pain
and grief**

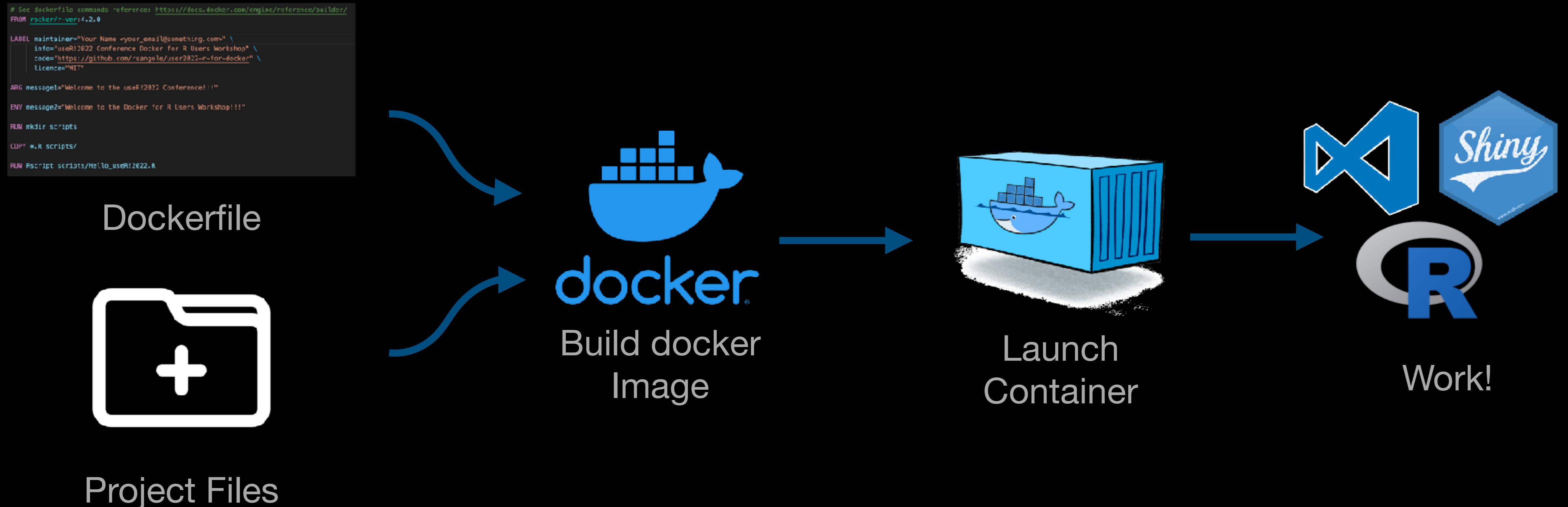
So, what is Docker?

Introduction to Docker

- A platform for OS-level virtualization
- Based on isolated containers
- Package different components of a software/app
- Enable seamless shipment and deployment
- Free and enterprise versions available



Workflow



Dockerfile

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"

ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts

COPY *.R scripts/

RUN Rscript scripts/Hello_useR!2022.R
```

```
02-dockerfile > ↵ Hello_useR!2022.R
1   print(Sys.getenv("message1"))
2   print(Sys.getenv("message2"))
```

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"

ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts

COPY *.R scripts/

RUN Rscript scripts/Hello_useR!2022.R
```

FROM - Defines the base image to use for the image's build. A valid docker will always start with FROM instruction.

Docker Hub - Container image library (or the “Github” of the images).

The screenshot shows the Docker Hub interface with a search bar at the top containing 'Search for great content (e.g., mysql)'. Below the search bar are navigation links: Explore, Repositories, Organizations, Help, Upgrade, and a user profile icon for rkrispin. On the left, there are filters for Products (Images, Plugins), Trusted Content (Docker Official Image, Verified Publisher, Open Source Program), Operating Systems (Linux, Windows), and Architectures (ARM, ARM 64, IBM POWER, IBM Z, PowerPC 64 LE, x86, x86-64). The main area displays four search results:

- ubuntu** (DOCKER OFFICIAL IMAGE): Updated 12 days ago. Description: Ubuntu is a Debian-based Linux operating system based on free software. Available architectures: Linux, ARM, ARM 64, PowerPC 64 LE, riscv64, IBM Z, 386, x86-64.
- alpine** (DOCKER OFFICIAL IMAGE): Updated a month ago. Description: A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size! Available architectures: Linux, IBM Z, riscv64, x86-64, ARM, ARM 64, 386, PowerPC 64 LE.
- redis** (DOCKER OFFICIAL IMAGE): Updated 6 days ago. Description: Redis is an open source key-value store that functions as a data structure server. Available architectures: Linux, Windows, mips64le, PowerPC 64 LE, IBM Z, x86-64, ARM, ARM 64, 386.
- postgres** (DOCKER OFFICIAL IMAGE): Updated 12 days ago. Description: The PostgreSQL object-relational database system provides reliability and data integrity. Available architectures: Linux, ARM 64, 386, PowerPC 64 LE, IBM Z, mips64le, x86-64, ARM.

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"

ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts

COPY *.R scripts/

RUN Rscript scripts/Hello_useR!2022.R
```

```
"Labels": {
    "code": "https://github.com/rsangole/user2022-r-for-docker",
    "info": "useR!2022 Conference Docker for R Users Workshop",
    "license": "MIT",
    "maintainer": "Your Name <your_email@something.com>",
    "org.opencontainers.image.authors": "Carl Boettiger <cboettig@ropensci.org>",
    "org.opencontainers.image.base.name": "docker.io/library/ubuntu:focal",
    "org.opencontainers.image.description": "Reproducible builds to fixed version of R",
    "org.opencontainers.image.licenses": "GPL-2.0-or-later",
    "org.opencontainers.image.revision": "d5e8891f7bebdc9deb12f4785ffcddf485ea0f31",
    "org.opencontainers.image.source": "https://github.com/rocker-org/rocker-versioned2",
    "org.opencontainers.image.title": "rocker/r-ver",
    "org.opencontainers.image.vendor": "Rocker Project",
    "org.opencontainers.image.version": "R-4.2.0"
}
```

LABEL - Enables to add information about the image to the image's metadata, such as authors, maintainers, license, etc.

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R
```

ARG - Defines variables that can be passed to the image during the build time.

ENV - Sets the image environment variables

```
02-dockerfile > 🐳 Hello_useR!2022.R
1 print(Sys.getenv("message1"))
2 print(Sys.getenv("message2"))
```

```
#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:bdaf45d418da2da3aa19b552d089f085b32f1aedb70b7408c599f35ecfd30813
#8 0.403 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.404 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s
```

```
docker build . --build-arg message1="We Love R" \
--progress=plain --no-cache -t user2022:hello.world
```

```
#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:3a64cbf72a2ba5e635b35e2a603b0deae1aee3d82b391cfec9b45dc5aa8106b
#8 0.311 [1] "We Love R"
#8 0.311 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.3s
```

```
"Env": [
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
  "R_VERSION=4.2.0",
  "R_HOME=/usr/local/lib/R",
  "TZ=Etc/UTC",
  "CRAN=https://packagemanager.rstudio.com/cran/__linux__/focal/latest",
  "LANG=en_US.UTF-8",
  "message2=Welcome to the Docker for R Users Workshop!!!"
],
```

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"

ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts

COPY *.R scripts/

RUN Rscript scripts/Hello_useR!2022.R
```

RUN - Enables to execute of CLI commands on the image during the image build time

Dockerfile Core Commands

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"

ENV message2="Welcome to the Docker for R Users Workshop!!!"

RUN mkdir scripts

COPY *.R scripts/

RUN Rscript scripts/Hello_useR!2022.R
```

COPY - Enables copying files from the local system to the image during the image build time

Dockerfile Core Commands

- **EXPOSE** - Defines the port on the container to listen during run time.
- **CMD** - Enables the execution of commands during the run time (unlike RUN, which executes during the build time).
- **ENTRYPOINT** - Enables the execution of commands during the run time with arguments.
- **VOLUME** - Sets a mount point inside the image which can be linked with external volume (e.g., local directory).
- **WORKDIR** - Defines the working directory inside the image for any **RUN**, **CMD**, **ENTRYPOINT**, **COPY**, and **ADD** commands
- **Reference** - <https://docs.docker.com/reference/>

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts

COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R
```

```
#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecfd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R

#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecfd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R

#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecfd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R

#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecfd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R

#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecfd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/rsangole/user2022-r-for-docker" \
      license="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR!2022.R

"Type": "layers",
"Layers": [
    "sha256:af7ed92504ae4c20128a0f01048d41d467fef5c795c38d0defdb998a187ed1d4",
    "sha256:1509dfbd9740d4d09dfd6bc670b34faccac1b493bf4dd21cd592ae6699829f89",
    "sha256:c973ae61543cd4e7e198e97a9ad35b0698356890fe8cfe456d604d8e9db58661",
    "sha256:29ce27ffdb05092f93eacf1b668f991e688ccedb6ce998cba0f0ea2f85fc404",
    "sha256:93f8861ed22aeff0641f92957912bf7ed78030c6c11873526e85ae49ead3ee2c",
    "sha256:a86140abb474bab3115413543340bcf31d24eb3502c76633d1b022a8bab54b15",
    "sha256:05332c0800ea84da01070813fcad2d71d459d5bb0e50a56874cccd3a7ddb70b71",
    "sha256:88439546c48910ff1c02139145f8fe7ae91c5539e2aab0e84fb68cb5769780c9"
]

#1 [internal] load build definition from Dockerfile
#1 sha256:74a23682865d62df9675cde9499e850f0e37eab8f63d2fdd5b53de3ba044b692
#1 transferring dockerfile: 37B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:d3cafb3294061efecd476b259df28deaae2a3d01a502ddbc8d2fe64b078b053e
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/rocker/r-ver:4.2.0
#3 sha256:266598a71b086c88f019f9187a5aaf6a74b2f977672d5ce8e758812d17a390ed
#3 DONE 0.0s

#4 [1/4] FROM docker.io/rocker/r-ver:4.2.0
#4 sha256:b23b1ebbd23f5243be99e448f51ea675f905e7f7e60b21308cf5c0c9a98e5884
#4 CACHED

#6 [internal] load build context
#6 sha256:82fa965674bf1e2d89d7716cf3cad5a14607fa3e0427b44c7949e8e3ebd96d4
#6 transferring context: 38B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir scripts
#5 sha256:94a4e316aebf3050c92033a3a28057a42ce82f1914bca08adc92246aa4a768d0
#5 DONE 0.3s

#7 [3/4] COPY *.R scripts/
#7 sha256:97e51fe5b96de7e6b15844ab025d9b894d3998046f61ffa16f56daf34375549
#7 DONE 0.0s

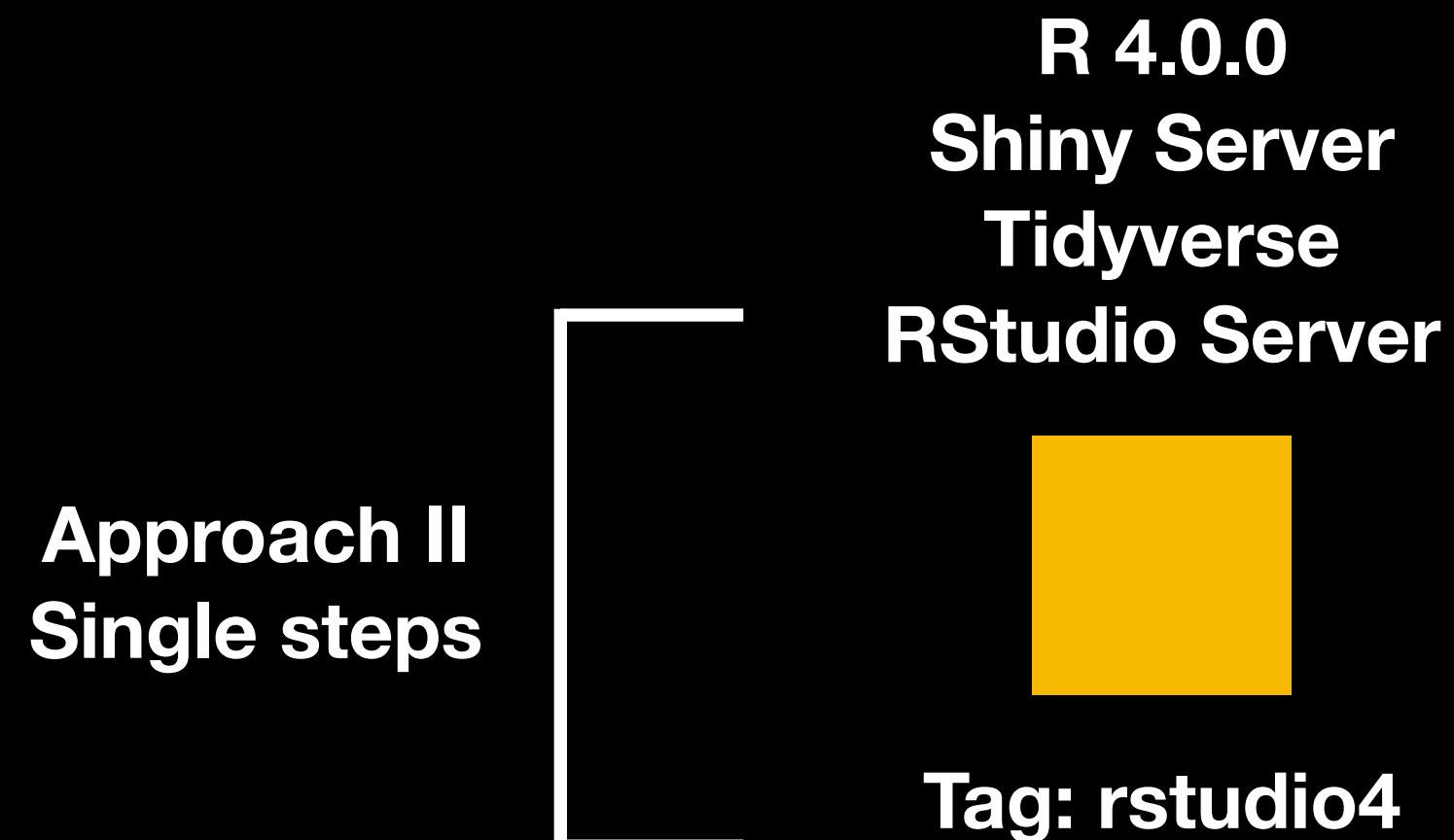
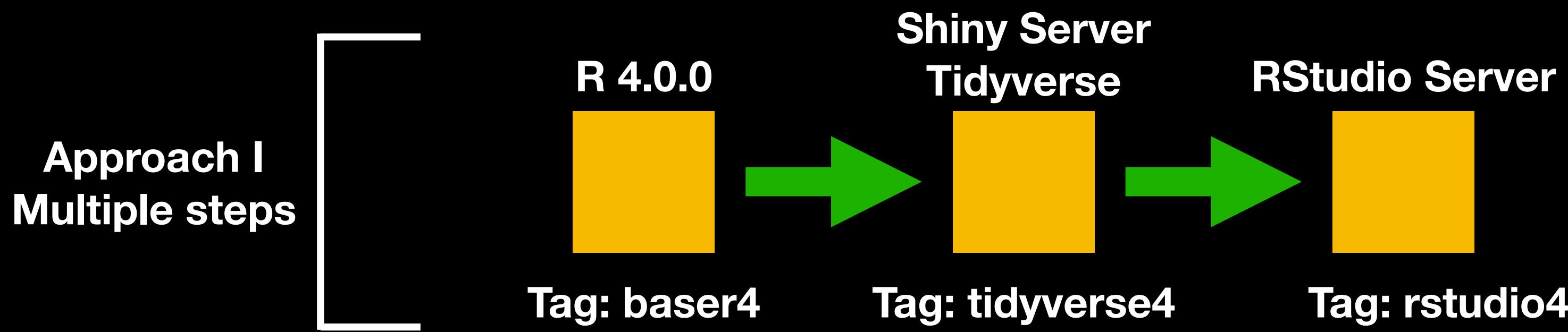
#8 [4/4] RUN Rscript scripts/Hello_useR!2022.R
#8 sha256:5f430e0e4011fa76f91e933aa188bb6998bcc3e673460ad4263278aff385f4f
#8 0.346 [1] "Welcome to the useR!2022 Conference!!!"
#8 0.346 [1] "Welcome to the Docker for R Users Workshop!!!"
#8 DONE 0.4s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:76365c7d84f772f47a266ed195ca65abcac2722016e7bd552086fa13cd533ceb done
#9 naming to docker.io/library/user2022:hello.world done
#9 DONE 0.0s
```

The Image Layers

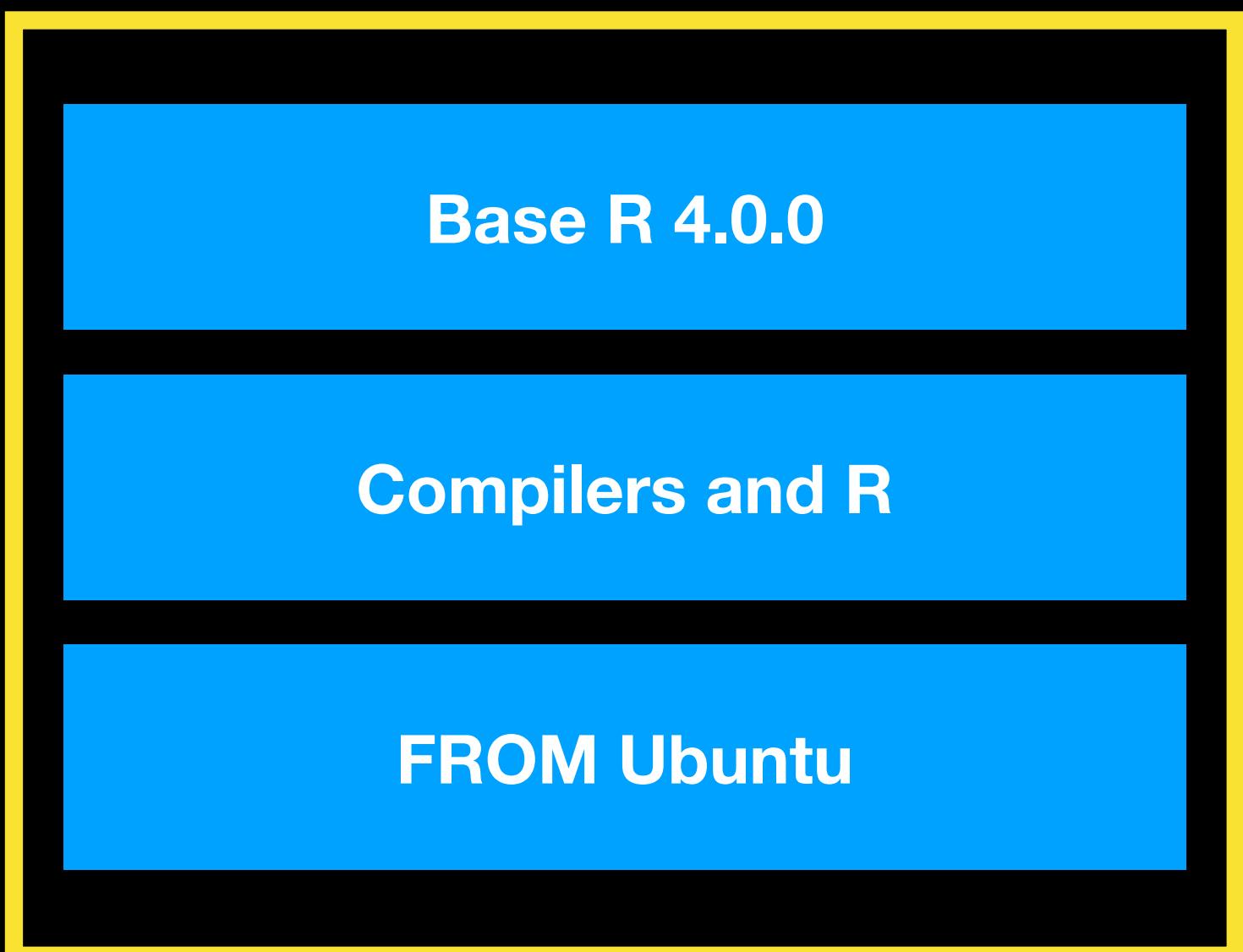
Docker requirements:

- R version 4.0.0
- Tidyverse packages
- Shiny Server
- RStudio Server



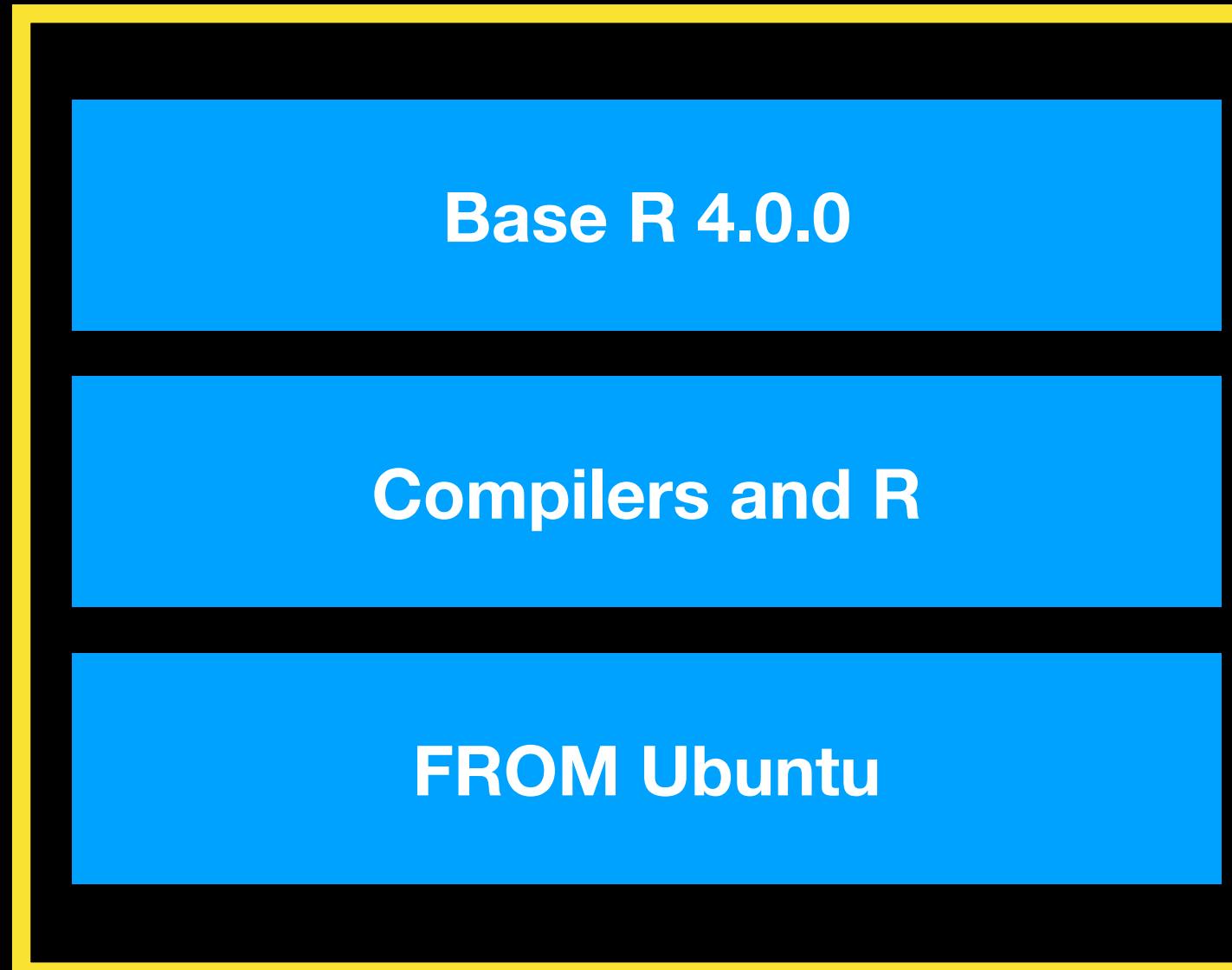
The Image Layers

Tag: baser4



The Image Layers

Tag: baser4

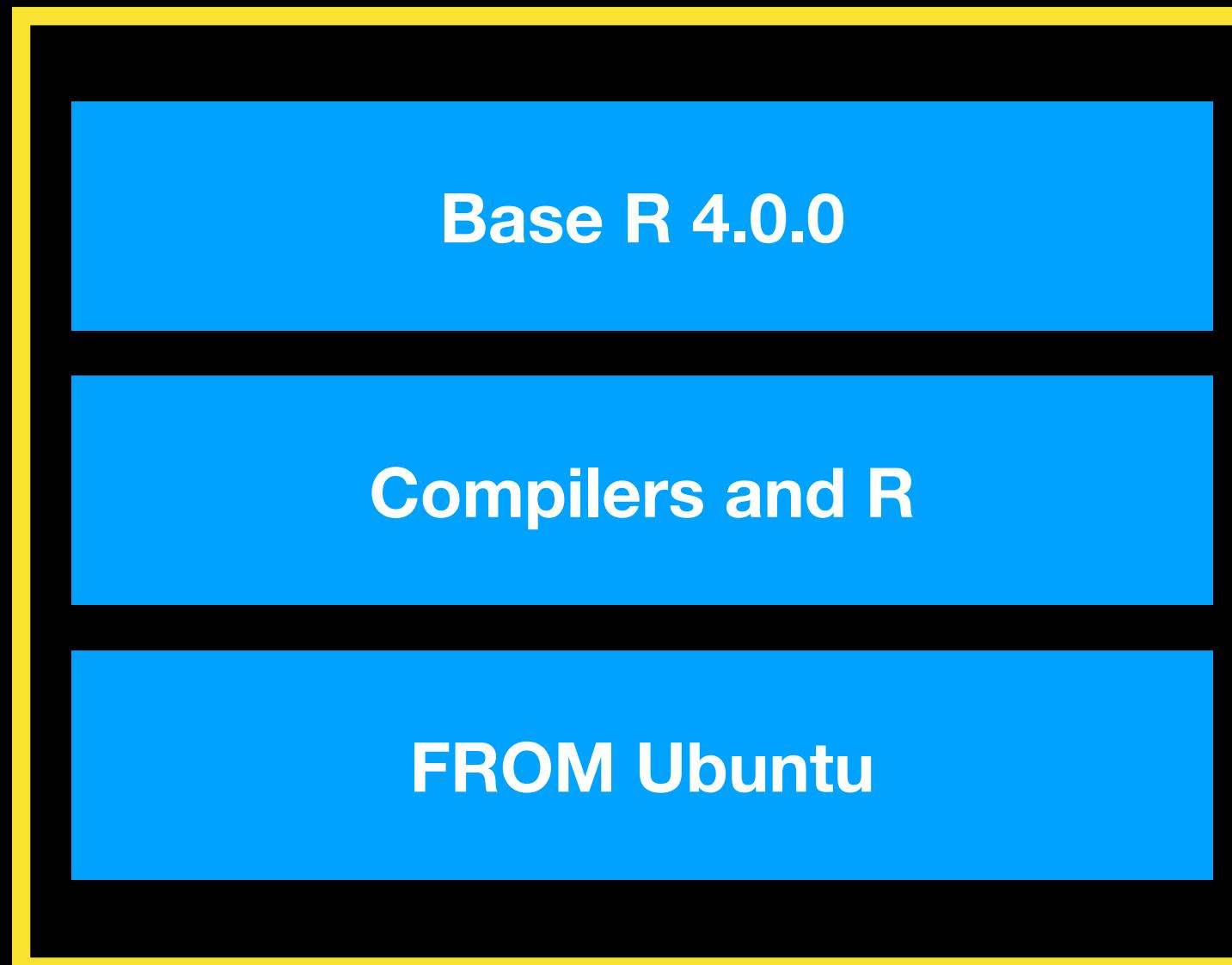


Tag: tidyverse4

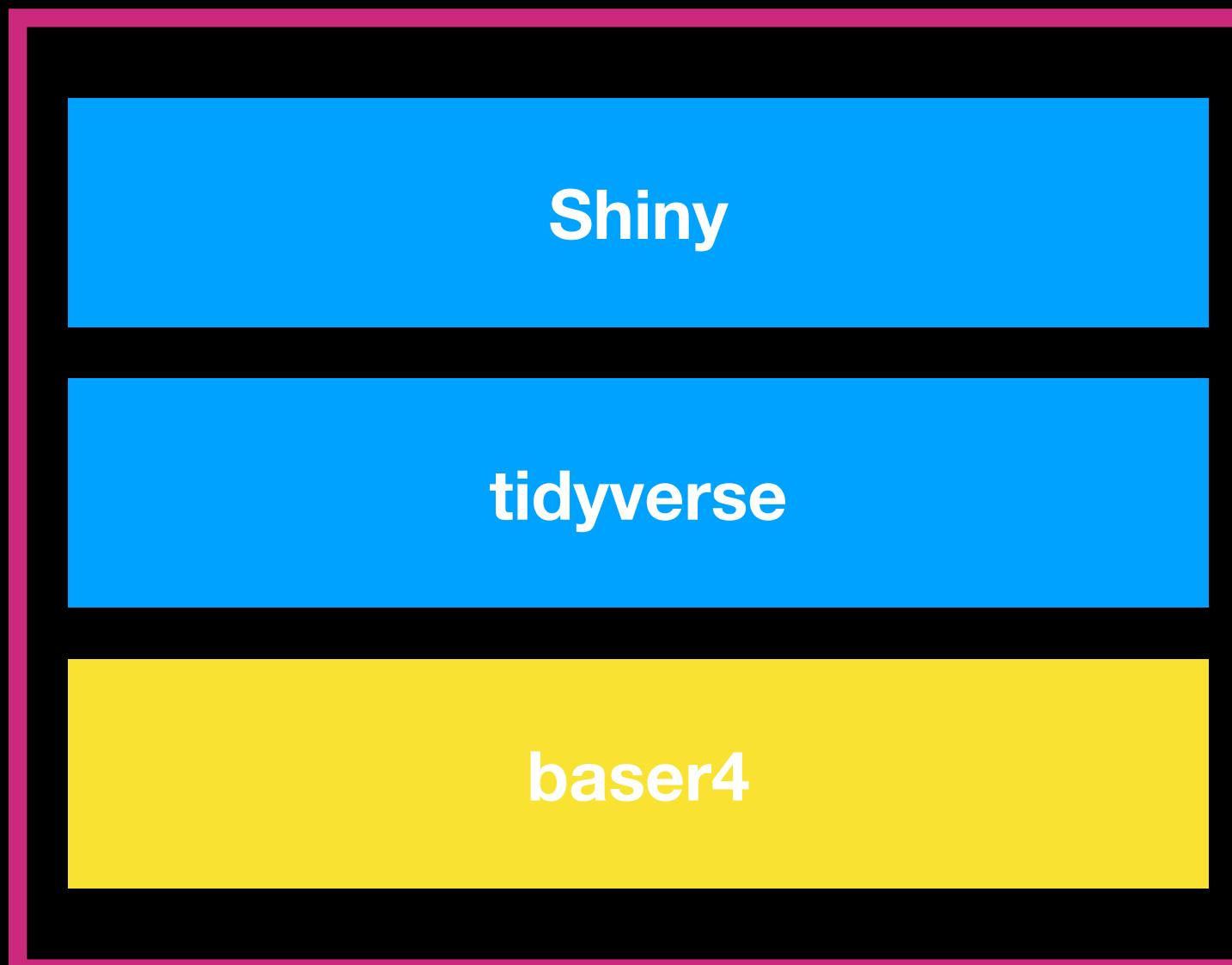


The Image Layers

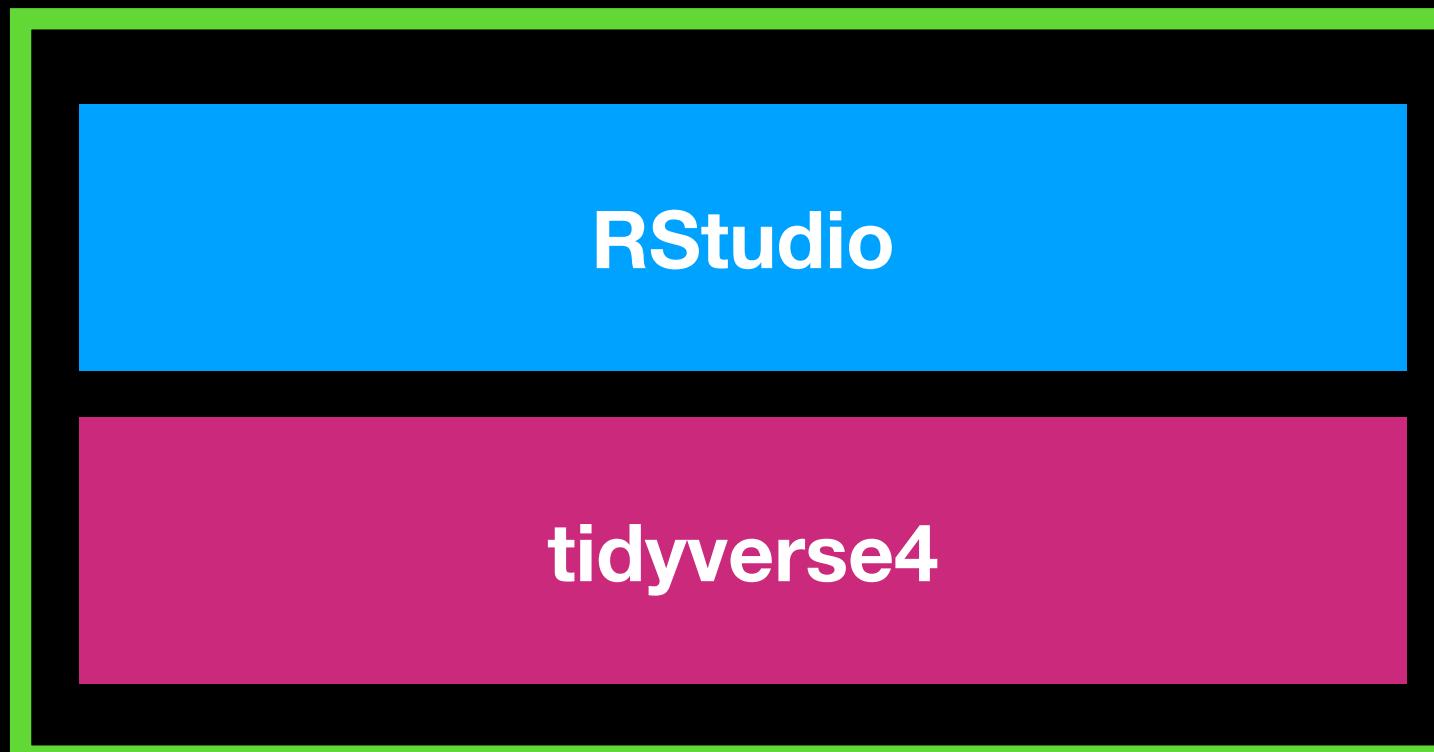
Tag: baser4



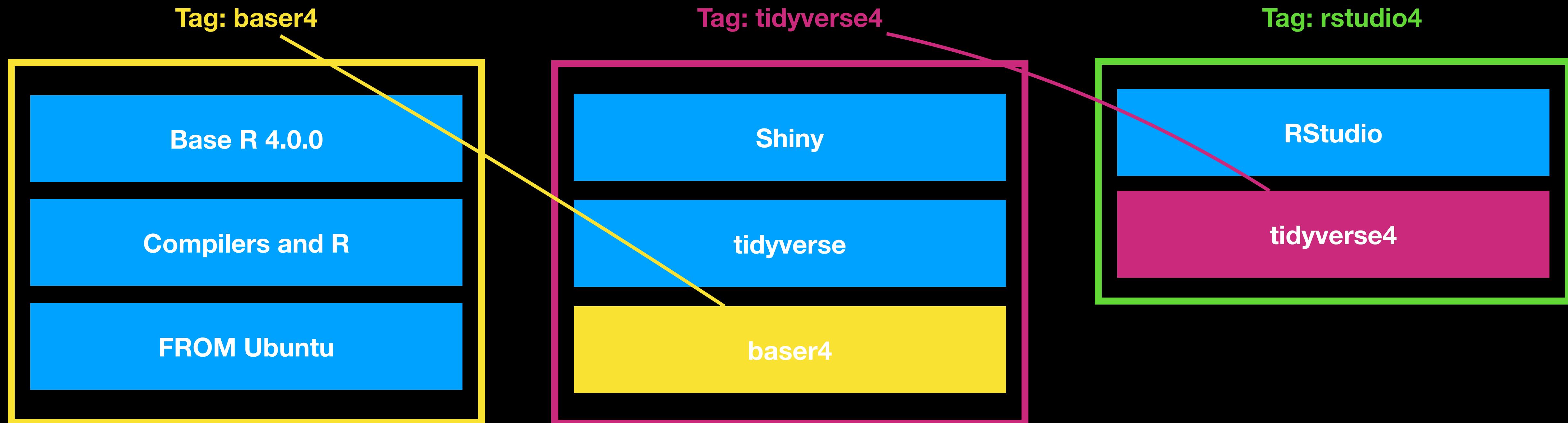
Tag: tidyverse4



Tag: rstudio4



The Image Layers



The Image Layers

Tag: rstudio4

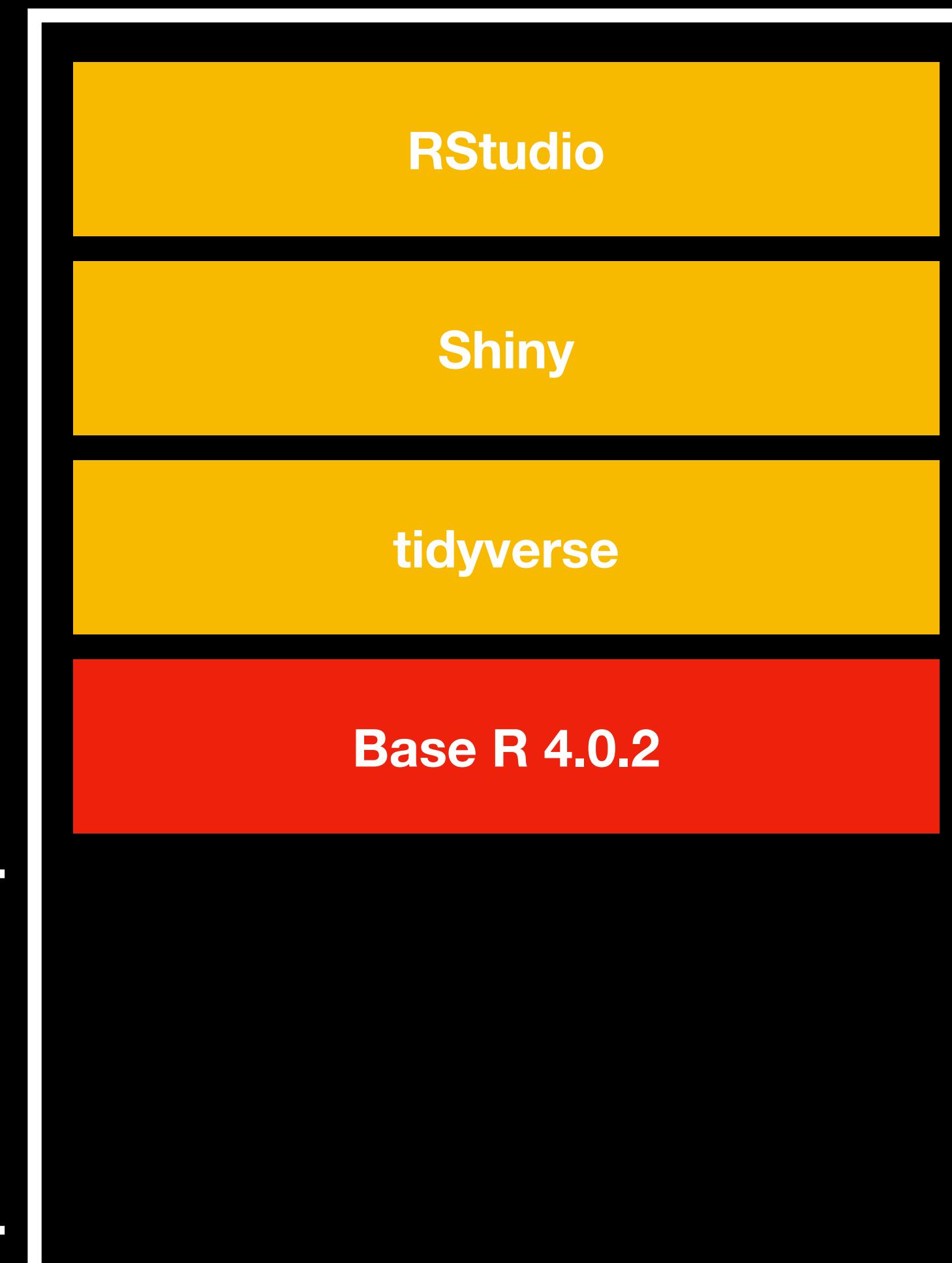


The Image Layers

Tag: rstudio4



Tag: rstudio402



Reinstall
From
Scratch

New
Layer

Cached
Layers

Upgrade the R version

The Image Layers

Approach I

- Convince during development
- Order of layers do matter, potentially enables faster build while applying changes and testing
- Image size might be bigger

Approach II

- Enables better optimization of the image size
- Run time might be longer

BREAK

Docker CLI

Docker CLI

```
# See dockerfile commands reference: https://docs.docker.com/engine/reference/builder/
FROM rocker/r-ver:4.2.0

LABEL maintainer="Your Name <your_email@something.com>" \
      info="useR!2022 Conference Docker for R Users Workshop" \
      code="https://github.com/sangole/useR2022-r-for-docker" \
      licence="MIT"

ARG message1="Welcome to the useR!2022 Conference!!!"
ENV message2="Welcome to the Docker for R Users Workshop!!!"
RUN mkdir scripts
COPY *.R scripts/
RUN Rscript scripts/Hello_useR2022.R
```

Dockerfile



Project Files

```
docker build
--file Dockerfile
--tag <namespace>/<image>:<tag>
```

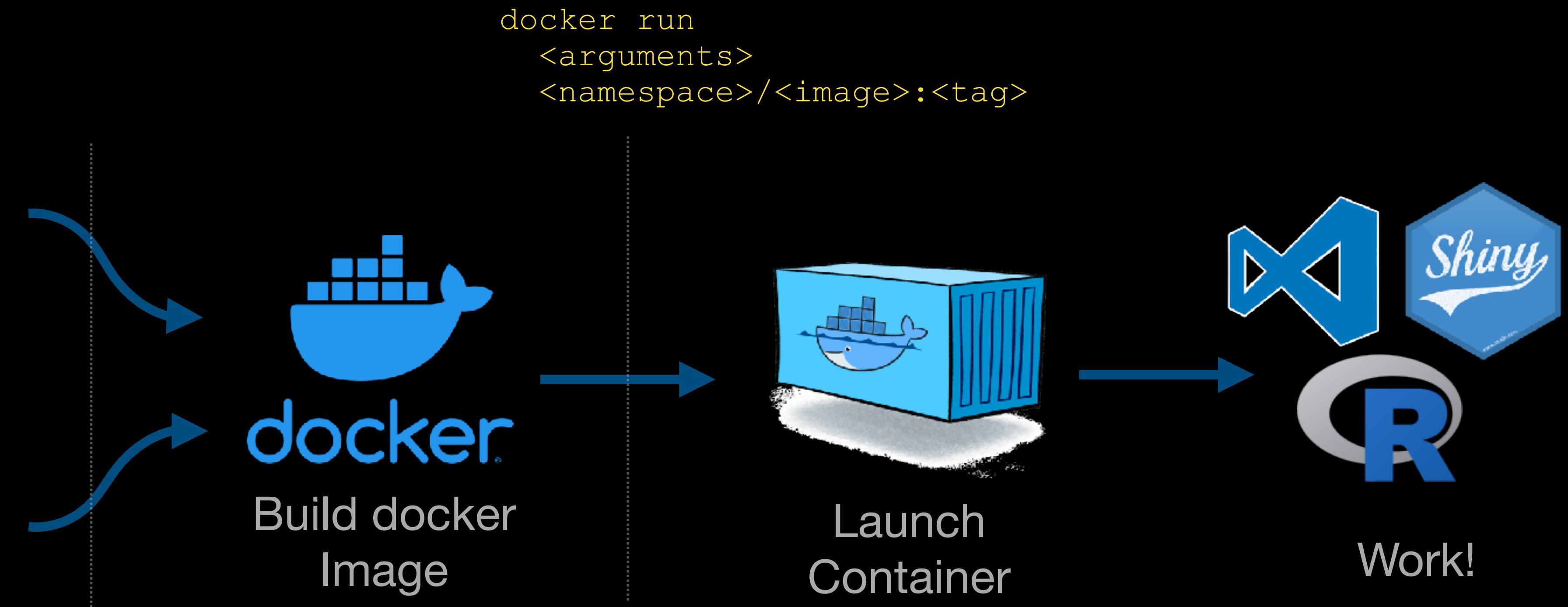
```
docker image ls
```

```
docker image inspect
```

```
docker ps
```

```
docker exec
```

```
docker compose
```



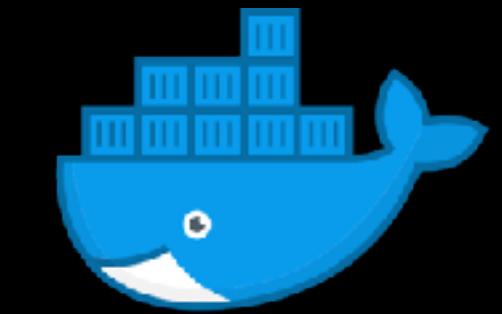
```
docker run
<arguments>
<namespace>/<image>:<tag>
```

BREAK

Docker + R Basics

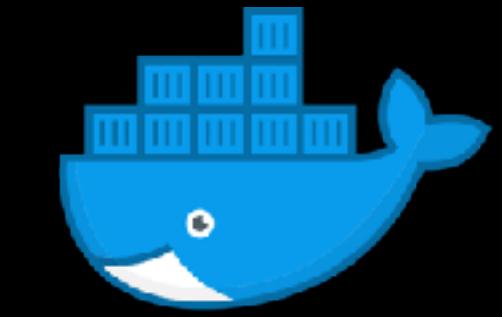
- Introduction to The Rocker Project
- Installing packages (*4 ways*)
- Develop in Docker (*with RStudio IDE*)
- Deploy Models in Docker (*with plumber API*)
- Deploy Shiny Apps in Docker

Installing R Packages



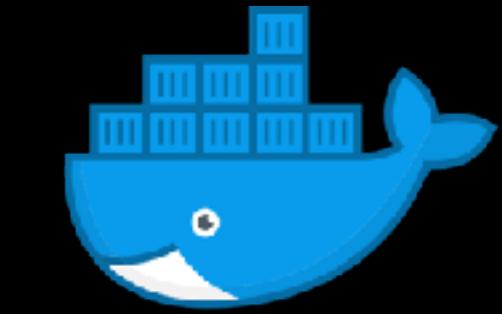
docker

`install.packages(...)`



docker

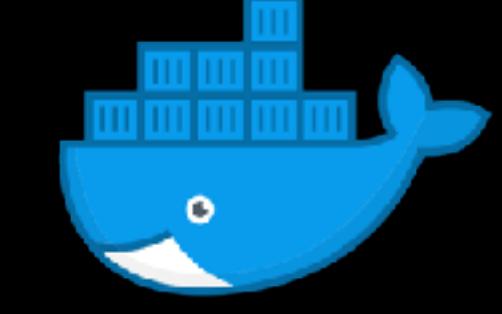
Use MRAN



docker

`{renv}`

Use Install packages
when Docker image is
built.



docker

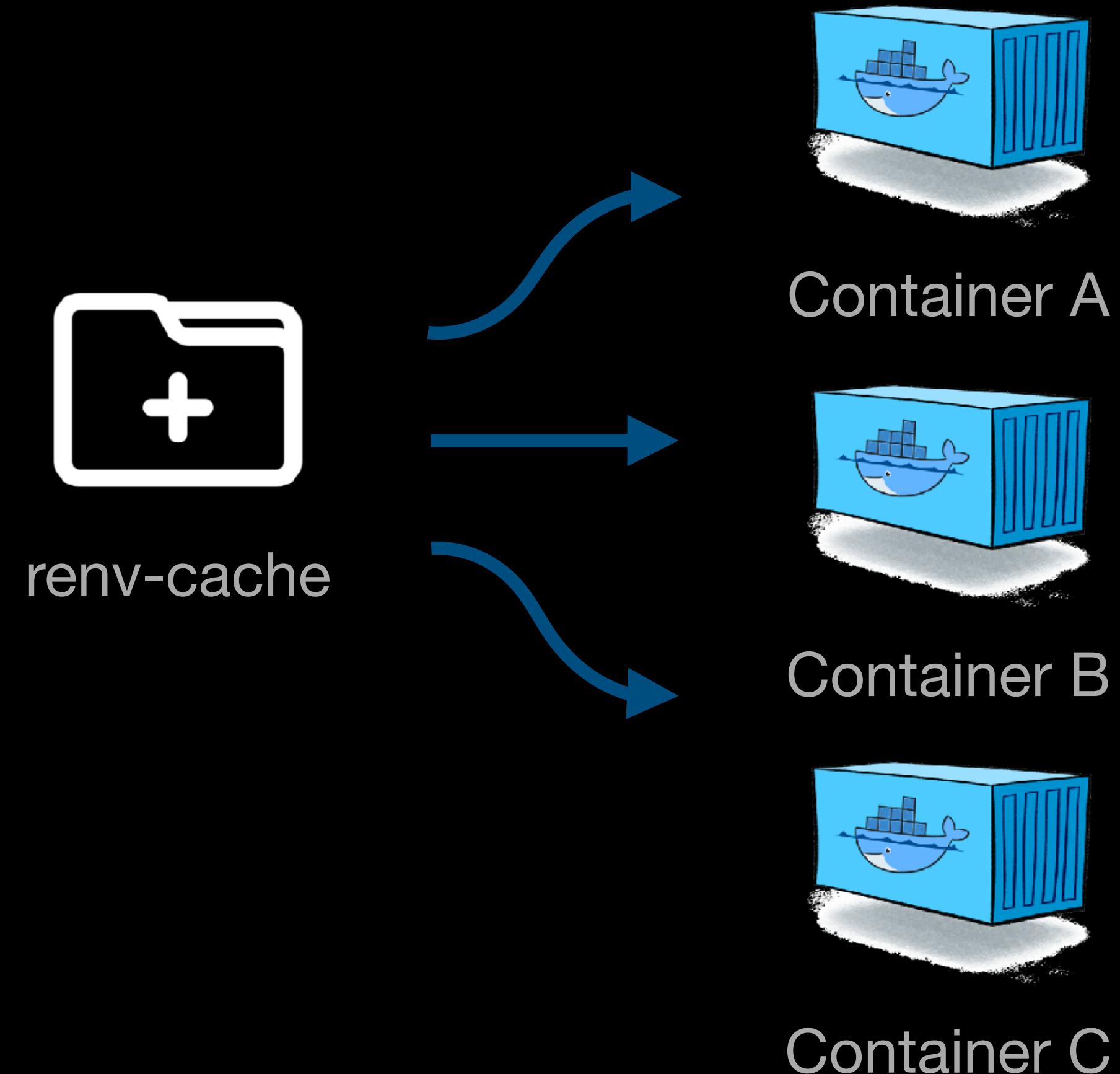
`{renv}`

Install packages when
Docker container is
run.

Installing R Packages

{renv}

Install packages when Docker container is run.



Intermediate Concepts

- Package Development with GitHub Actions [15 min]
- VS Code, R and Docker [10 min]
- *Stretch Goal - Postgres & R*

Some Resources

- Workshop materials - <https://github.com/rsangole/user2022-r-for-docker>
- Docker documentation - <https://docs.docker.com/get-started/overview/>
- Docker Hub - <https://hub.docker.com/>
- Rocker - <https://www.rocker-project.org/>
- Free Code Academy - <https://www.youtube.com/c/Freecodecamp/search?query=docker>

Also see: <https://github.com/rsangole/user2022-r-for-docker/tree/main/06-resources>

Thank You!



@ Rami_Krispin



@ rsangole