

Reproducible Environments with Docker



Doug Ashton

Overview

Reproduce in Cloud

Reproducible Environments



VAGRANT



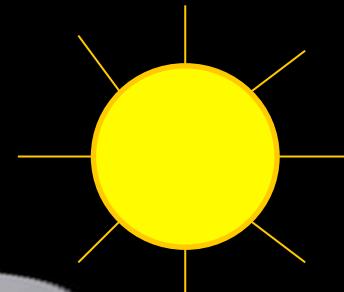
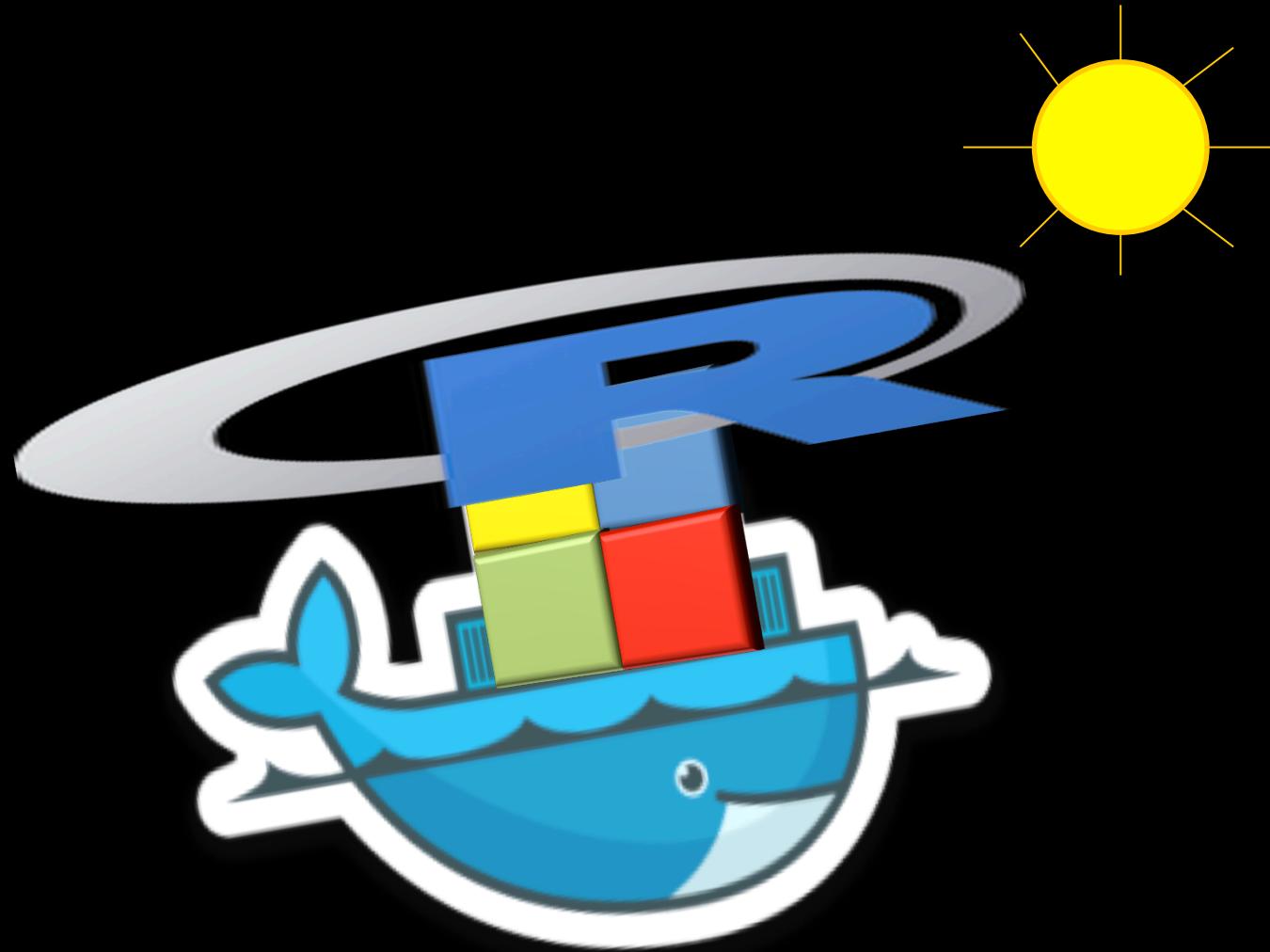
Reproducible Packages



Reproducible
Code







Reproducible Code



Can you just re-run
those numbers for
me?



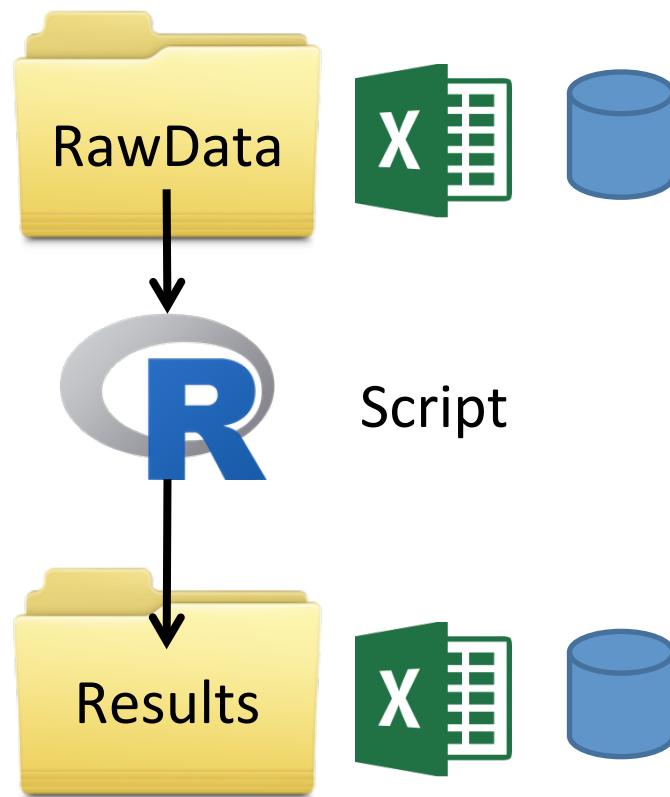


Two Warnings

- Retracted cancer research –
<http://bit.ly/repres01> (video)
- Reinhart, Rogoff, and the Excel Error That Changed History -
<http://bit.ly/repres02>

the solution code

A Reproducible Flow



Demo

- <https://github.com/dougmet/repEnvDemo>

Dependencies

```
library(zoo)
library(xts)
library(copula) # packrat headache

# Data Wrangle -----
```

Input data

```
# Retrieve the processed daily data from csv
allClose <- read.zoo("data/allClose.csv",
                      format="%Y-%m-%d", header=TRUE)
```

Manipulation

```
# Get log daily returns
allLog <- diff(log(allClose))

# Sum to log weekly returns
allLogWk <- apply.weekly(allLog, columns)
```

Plots

```
# Plot of the time series -----
pdf("results/ts.pdf")
plot(allLogWk, main="weekly Log Returns")
dev.off()
```

Calculation

```
# Correlation matrix -----
corMat <- cor(allLogWk, use="pairwise.complete.obs")
```

Output data

```
# Write out the correlation matrix
write.csv(corMat, "results/correlation.csv")
```

Version Control

- Edit with confidence
- ~~script-v1a.R~~ use tags
- Backup to remote
- Work together

Reproducible Reports

- Text and
- **code**
- All together
- **run demo report.Rmd in repo**

Market Correlation

Doug Ashton

06 October, 2016

Reproduce this document

```
seed <- 909
gitCommit <- system2("git", c("rev-parse", "HEAD"), stdout = TRUE)
gitBranch <- system2("git", c("rev-parse", "--abbrev-ref", "HEAD"), stdout = TRUE)
set.seed(seed)
repData <- data.frame(Git.Commit = gitCommit, Git.Branch = gitBranch,
                      Random.Seed = seed, R.Version = R.version$version.string,
                      stringsAsFactors = FALSE)
knitr::kable(repData, format = "markdown")
```

Git.Commit	Git.Branch	Random.Seed	R.Version
37b1435fad26e116adb459858486b3b3e147f635	odsc	909	R version 3.2.3 (2015-12-10)

It is nice to layout all dependencies at the top.

```
library(zoo)
library(xts)
library(quantmod)
```

Literate Programming Tools

- Knitr/Sweave
- rmarkdown
- Jupyter (Ipython) Notebook
- RCloud
- git →

```
> ./a.out
Built from: 5c1a5f6cd58ee0409aed447f2fff0c92dc079844
Branch: master

Random Seed
D9BE237B 7D63F1B2 AE882E77 B1C8F592

Starting simulation...
```

Data Pipelines Interlude (my next talk)



Data Pipelines

`raw -> clean -> prepared`

- “Make” for data
 - Drake, Airflow (Python)
 - Remake (R)
 - Many more
- Awesome-pipline list: <http://bit.ly/repres03>

Reproducible Code Summary

Raw data + 6c30934a0ea2c0473d37b6d
= reproducible results

Reproducible Packages



the new problem

Your Laptop



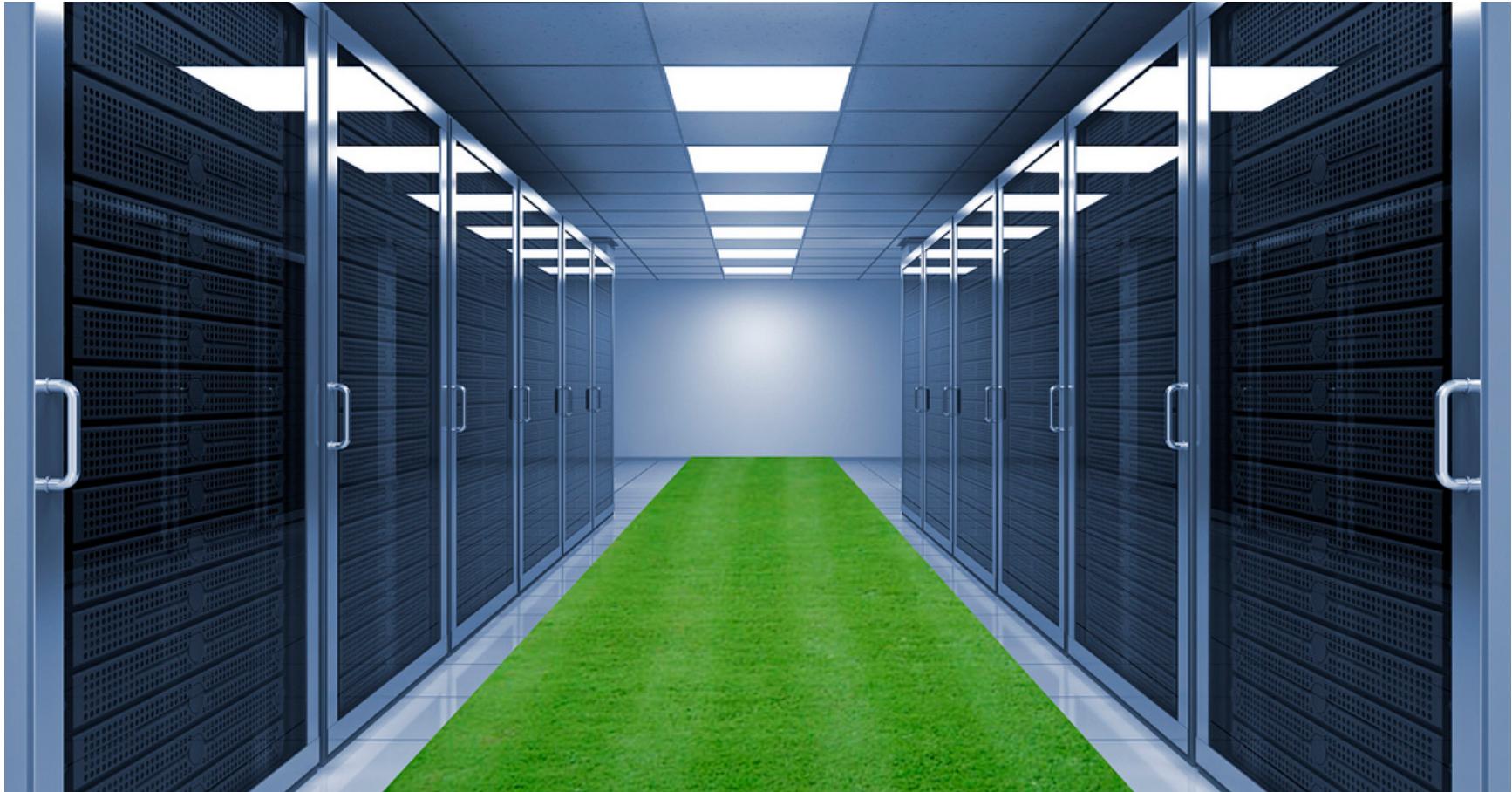


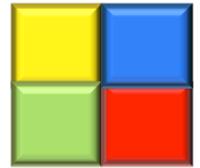
No, no, leave it. I have it
how I like it.



How you like it?! No, no, Jen, it's
infected. If this was a human being,
I'd shoot it in the face.

Their server





Package Versions

- R: Packrat/checkpoint/switchr + conda
- R: MRAN mirror snapshots
 - E.g. <https://mran.revolutionanalytics.com/snapshot/2015-03-01>
- Python: Conda / pip

Package Management Packages

Packages	Pros	Cons
Packrat	Built into Rstudio	Bulky in repo (eg git) Needs foresight
Checkpoint	Easy Rescues old scripts	Lots of libraries Only one date
Switchr	Package manifests Repo friendly	Not easy

Packrat

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
> packrat:::init()  
Initializing packrat project in directory:  
-("~/Documents/Talks/RepEnvironments/packrat")
```

Adding these packages to packrat:

ADGofTest	0.3
Matrix	1.2-2
TTR	0.23-0
colorspace	1.2-6
copula	0.999-14
gsl	1.9-10
lattice	0.20-33
mvtnorm	1.0-3
packrat	0.4.6-1
pspline	1.0-17
quantmod	0.4-5
stabledist	0.7-0
xts	0.9-7
zoo	1.7-12

```
Fetching sources for ADGofTest (0.3) ... OK (CRAN current)  
Fetching sources for Matrix (1.2-2) ... OK (CRAN current)  
Fetching sources for TTR (0.23-0) ... OK (CRAN current)  
Fetching sources for colorspace (1.2-6) ... OK (CRAN current)  
Fetching sources for copula (0.999-14) ...
```

Packrat

Pros

- Baked into RStudio
- Works how you expect
`install.packages(...)`
`install_github(...)`
- Not much else to do

Cons

- Saves entire source into repo (potentially bulky)
 - There is a workaround
- Plan in advance

Checkpoint

The screenshot shows the RStudio interface with the 'Packages' tab selected in the top menu bar. The main area displays a list of installed packages, their descriptions, and versions. The list includes packages like ADGofTest, compiler, copula, gsl, lattice, Matrix, mvtnorm, pspline, quantmod, stabledist, TTR, xts, and zoo. The 'Description' column provides a brief summary of each package's purpose, and the 'Version' column shows the current version number.

Name	Description	Version
ADGofTest	Anderson-Darling GoF test	0.3
compiler	The R Compiler Package	3.2.2
copula	Multivariate Dependence with Copulas	0.999-12
gsl	wrapper for the Gnu Scientific Library	1.9-10
lattice	Lattice Graphics	0.20-30
Matrix	Sparse and Dense Matrix Classes and Methods	1.1-5
mvtnorm	Multivariate Normal and t Distributions	1.0-2
pspline	Penalized Smoothing Splines	1.0-16
quantmod	Quantitative Financial Modelling Framework	0.4-3
stabledist	Stable Distribution Functions	0.6-6
TTR	Technical Trading Rules	0.22-0
xts	eXtensible Time Series	0.9-7
zoo	S3 Infrastructure for Regular and Irregular Time Series (Z's ordered observations)	1.7-11

Switchr

- No demo

SwitchR

Pros

- Save versioned package inventory
- Very repository friendly

Cons

- Not as easy to use
- Might struggle with custom sources

Python Package Management

- conda
- Listing packages
 - conda list
- Save environment
 - conda env export > environment.yml
- Install environments
 - conda env create -f environment.yml
- pip / virtualenv
- Listing packages
 - pip list
- Save/freeze packages
 - pip freeze > requirements.txt
- Install
 - pip install -r requirements.txt

Reproducible Environments

A close-up photograph of a sea turtle swimming in dark, rippling water. The turtle's head and front flippers are visible, showing its patterned skin. A small, colorful fish is seen near the turtle's head. The background is dark and textured.

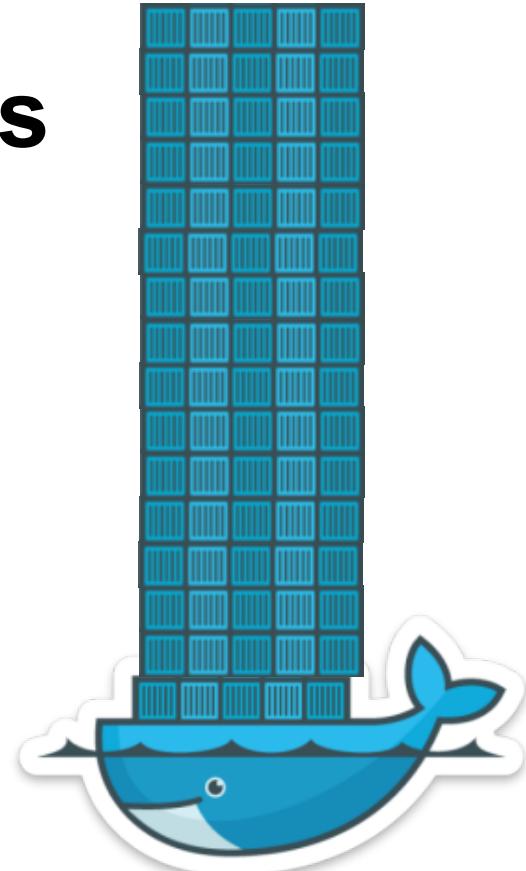
Reproducible Environments



1. System Dependencies

2. Managed Environments

3. Scalability



Vagrant

- www.vagrantup.com
- Primarily for development
- Each project gets a virtual machine
- <https://github.com/dougmet/vagrantR>



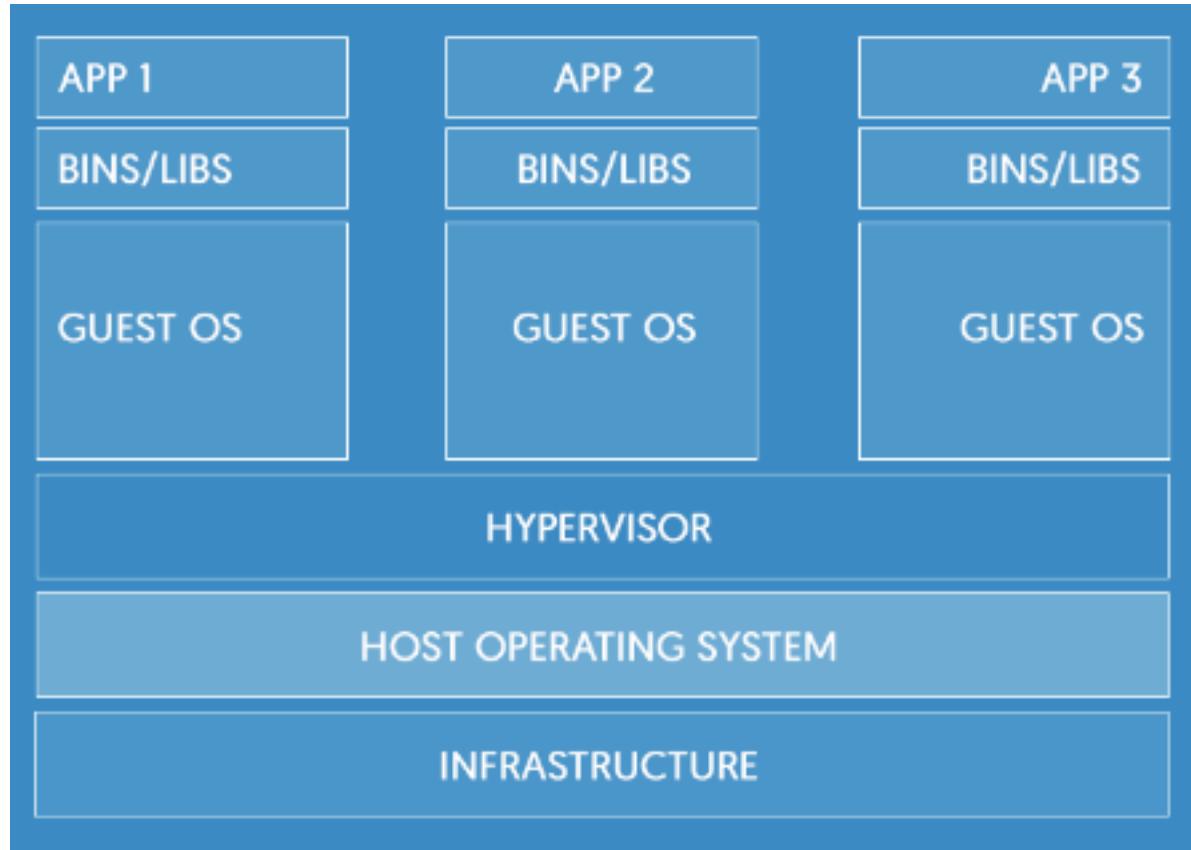
Docker

- www.docker.com
- Primarily for deployment



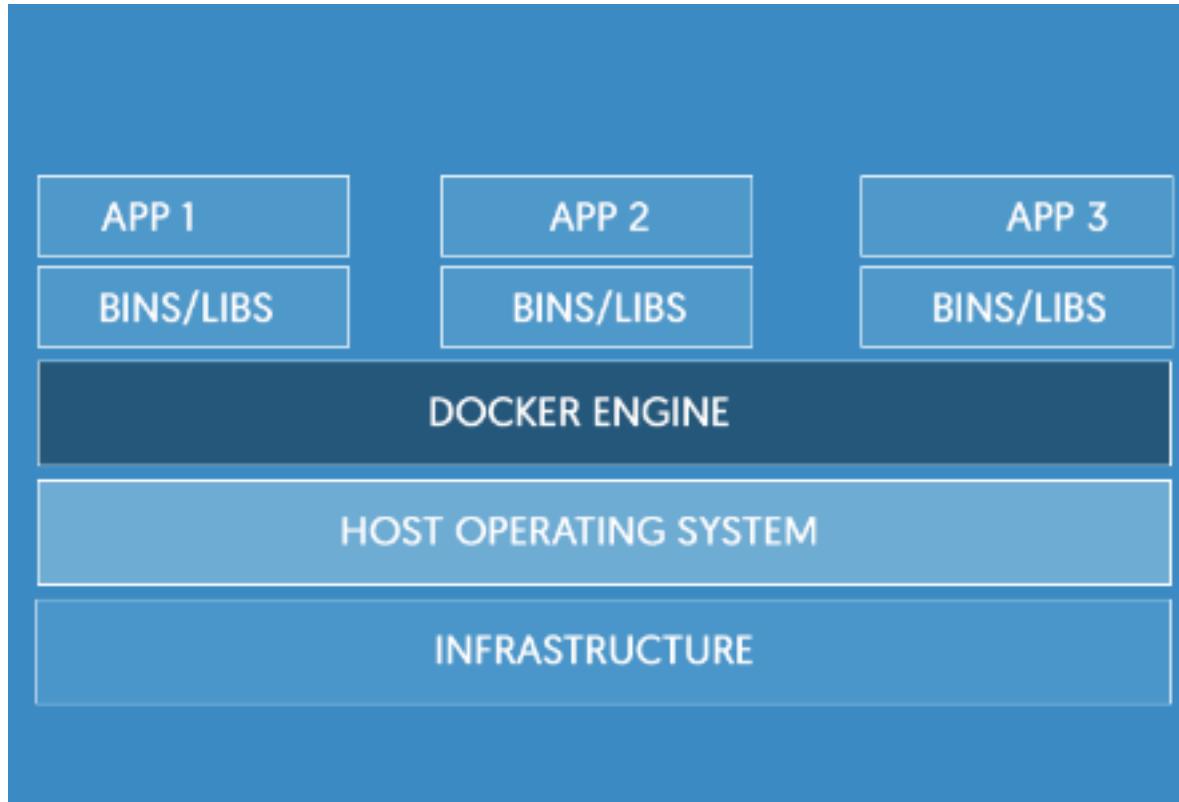
* OK, except Windows and Mac

How's it different to a VM?



Source:
Docker Website

How's it different to a VM?



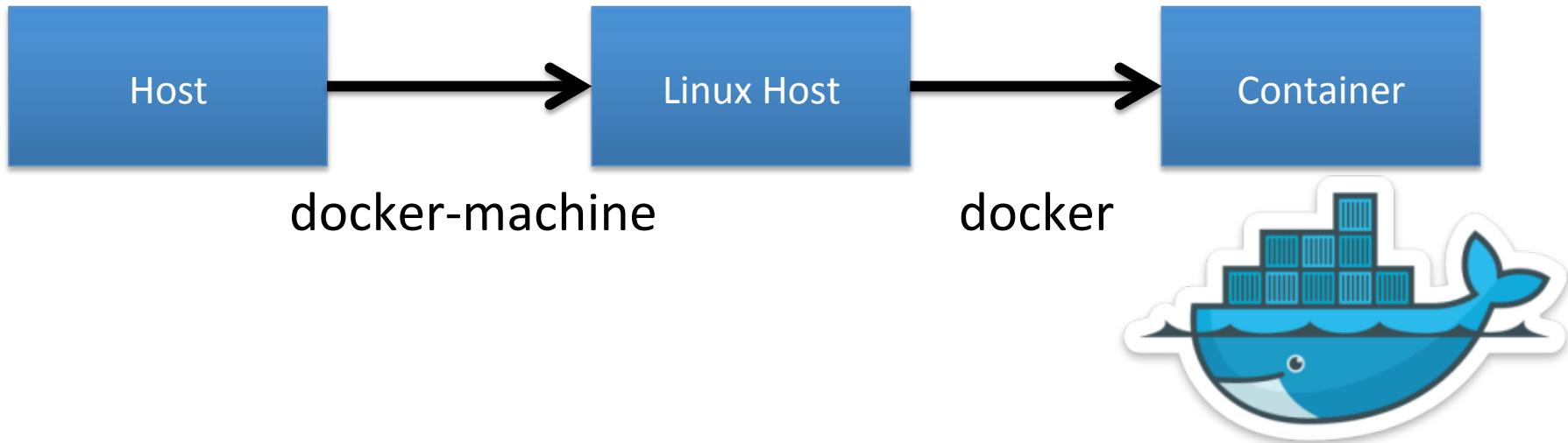
Source:
Docker Website

How's it different to a VM?

- Shares resources (same kernel)
- Starts nearly instantaneously
- Destroy just as quickly
- Launch as many as you want

Old Docker For Windows / Mac

- Linux on Virtual Box
- docker-machine application



New Docker For Windows / Mac

- Mac
- Linux on **xhyve**
- Can use alongside old Docker
- Windows
- Linux on **Hyper-V**
- **Cannot** use alongside old Docker

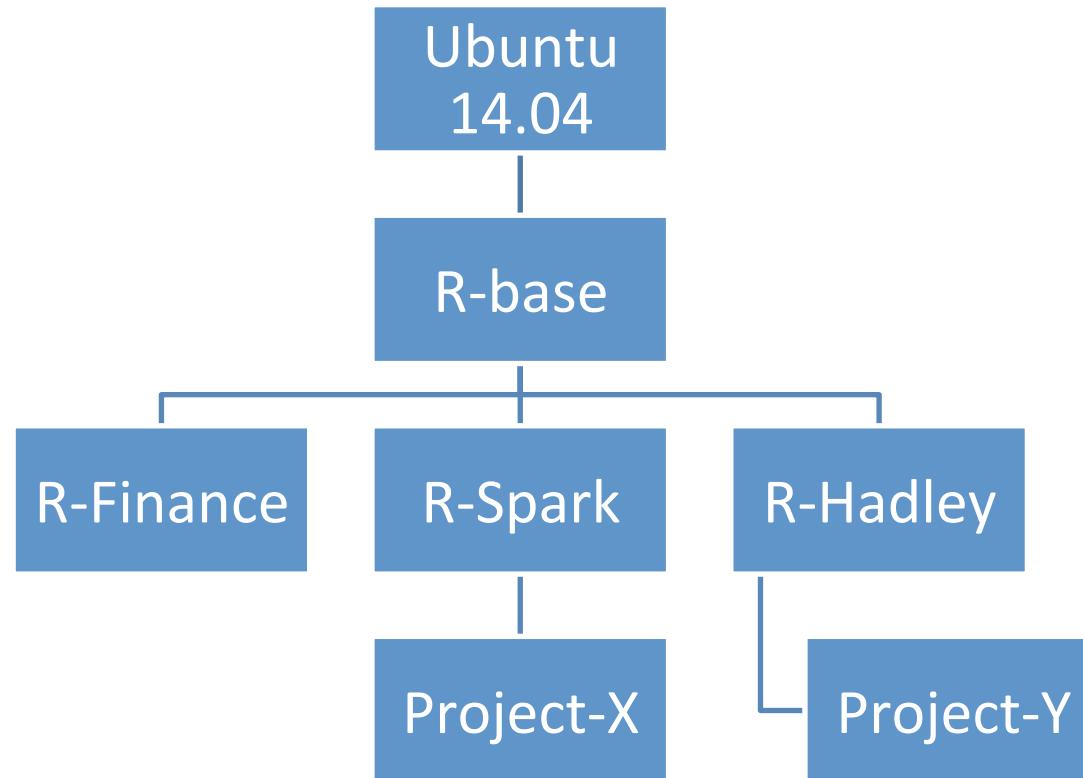
There's also Windows
on Windows
Containers
<http://bit.ly/repres05>



Key Components

- **Images**
 - Big (>200Mb). A **whole system**. docker images
- **Registry**
 - Repository of images docker pull <image>
- **Docker File**
 - Recipe to build an **image** docker build <dockerfile>
- **Containers**
 - An instance of an **image**. docker run <image>

Docker Images are Hierarchical



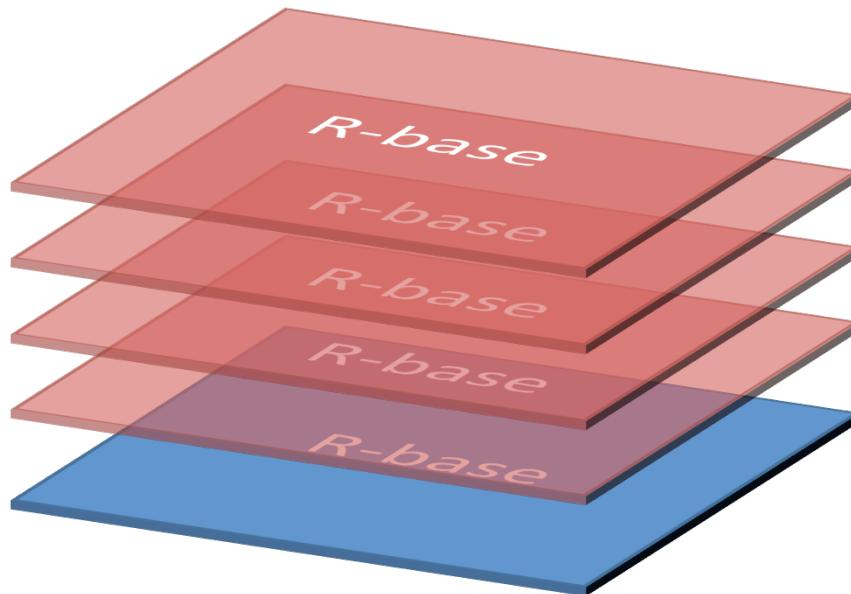
Docker Images are Hierarchical

ubuntu R-base R-studio

```
dougmet/r-base      3.1.2          72bdd2099619    10 months ago
591.4 MB
Rosita:~ douglas$ docker pull dougmet/r-studio:3.1.2
3.1.2: Pulling from dougmet/r-studio
863735b9fd15: Already exists
4fbaa2f403df: Already exists
44be94a95984: Already exists
a3ed95caeb02: Pull complete
3d6eeeea631a7: Already exists
614e027b8663: Already exists
cd9acb1119a1: Already exists
cebfcaa4deee0: Already exists
66a4ea8165a9: Already exists
0743342e74bc: Pull complete
67af1488119c: Pull complete
d209ae5e9e36: Pull complete
ac823c05692e: Pull complete
b195b9a5bbea: Pull complete
b437272b7085: Pull complete
172e001c3d7d: Pull complete
Digest: sha256:5337d913fddc84aebf506248b4507aca44cde454ca688ae620eeab8c63562f8
Status: Downloaded newer image for dougmet/r-studio:3.1.2
Rosita:~ douglas$
```

Docker Containers

Multiple containers from one image, launch rapidly



Sandbox

- Demo rocker/hadleyverse <http://bit.ly/repres06>
- jupyter/all-spark-notebook <http://bit.ly/repres07>
- ```
docker run -d -p 8787:8787 -v $PWD:/home/rstudio/host dougmet/r-studio:3.1.2
```

  - -d keep running in background
  - -p map local port to container port
  - -v map a local volume to container

# Rocker Warning



cboettig commented 27 days ago

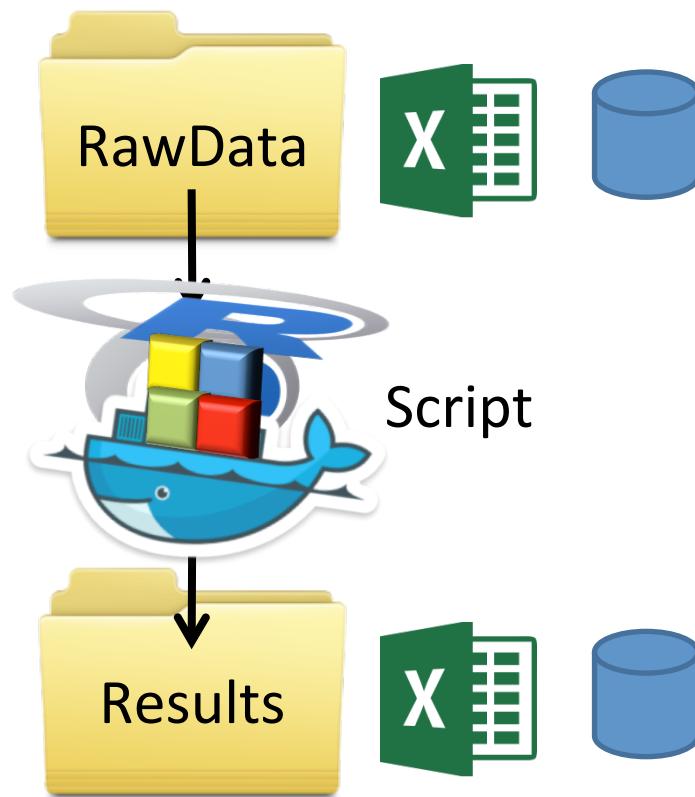
Owner

@dougmet Yeah, sorry about that. The docker container pulls whatever it gets from the debian:testing repos, so once those updated, 3.1.2 updated as well.

Future tags shouldn't do this (I think), because the version number is now explicit in the `apt-get` line.

Nevertheless, because our image builds R from the Debian binaries rather than from source (but see rocker/r-devel for the latter), the r-base images will probably never be well suited for running older R versions. That is, you couldn't come back and run the Dockerfile for 3.2.2 a few years later and get R 3.2.2, because apt-get simply won't be able to find that version on the debian:testing repos in a few years, which will have moved on.

# A Reproducible Flow



# Docker Demo

- <https://github.com/dougmet/dockerR>

```
FROM dougmet/r-base:3.1.2
```

```
RUN apt-get -y install libgsl0ldbl=1.16*1 libgsl0-dev=1.16*
```

```
Install R package manifest
COPY loadPackages.R /tmp/
COPY packages.csv /tmp/
RUN Rscript /tmp/loadPackages.R
```

```
CMD ["R"]
```

# A Reproducible Flow

- `docker run --rm -v $PWD:/home/docker/host repenv Rscript /home/docker/host/script.R`
- `--rm` : Delete container when done
- `-v` : Mount local volume to container
- `repenv` : The image to run
- `Rscript` : The command to run

# Scale Up



# Moving to the cloud

- Digital Ocean Demo
  - Use the Docker “one click app”
  - Clone repo. Build image. Done.
- You could use anything (AWS, Azure, etc)

# Docker Swarm

- “turns a pool of Docker hosts into a single, virtual Docker host”
- Launch lots of containers
- Out of my league!

# Summary

- Use code
  - Version Control
  - Keep text with code (notebooks / Rmd)
- Save package versions in repo
  - R: Packrat / pkgsnap / MRAN
  - Python: Conda
- Save the environment in repo
  - Docker / Vagrant File

# Get in touch

- GitHub
  - MangoTheCat
  - dougmet
- Twitter
  - @dougashton
- Email
  - dashton@mango-solutions.com