Automatically Archiving Reproducible Studies with Docker

Daniel Nüst | University of Münster | @nordholmen

useR!2017 Brussels, Thu 12:12 pm, room 2.02

http://sched.co/AxqM
Contents

Motivation

Docker

containerit
Five selfish reasons to work reproducibly

Florian Markowetz

Published: 8 December 2015

Abstract

Five selfish reasons to work reproducibly: ask what
reproducibility helps to avoid disaster
reproducibility makes it easier to write papers
reproducibility helps reviewers see it your way
reproducibility enables continuity of your work
reproducibility helps to build your reputation
Opening the Publication Process with Executable Research Compendia

Nüst D, Konkol M, Pebesma E, Kray C, Schutzeichel M, Przibytzin, H, Lorenz, J

http://www.dlib.org/dlib/january17/nuest/01nuest.html
ERC creation process

- Submit workspace to publication platform
- Publication platform…
  - extracts metadata
  - executes analysis
    - check output vs. upload (syntax)
    - capture runtime environment (manifest + image)
- User checks metadata
- Publish ERC persistently
The real value of Docker is not technology

It’s getting people to agree on something

Slide by Docker inventor & Docker, Inc. CTO Solomon Hykes, DockerCon 2014
Docker overview

Estimated reading time: 10 minutes

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.

The Docker platform

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight because they don’t need the extra load of a hypervisor, but run directly within the host machine’s kernel. This means you can run more containers on a given hardware combination than if you were using virtualization.

https://docs.docker.com/engine/docker-overview/
Docker overview

Estimated reading time: 10 minutes

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 without compromising quality.

With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker’s methodology of shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing and running it in production.

Docker platform

Provides the ability to package and run applications in a loosely isolated environment. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight because they don’t need the extra load of a hypervisor, but run directly within the host machine’s kernel. This means you can run more containers on a given hardware combination than if you were using virtual...

science
data science
research
reproducibility
replication

package & separate applications and their dependencies for cloud infrastructures

https://docs.docker.com/engine/docker-overview/
Docker basics

**Dockerfile**
- ENV
- RUN
- CMD

**Docker Image**
- build
- Docker CLI
- run

**Docker Container**
- run
- Docker Registry
- Docker Engine
- pause
- stop/kill
- start
- logs
- cp
- exec
- rm
- stats
Docker for Data Science

(all the Docker advantages… write once, biz ops, cloud, etc.)

Reproducibility

**Project** separation + don’t clutter dev machine

**Environment** (re)creation, **documentation**

Adopt **good practices** on the way

Easy **collaboration**

Easy **transition** from testing to production
Docker for RR lesson

https://github.com/nuest/docker-reproducible-research

https://nuest.github.io/docker-reproducible-research

Author Carpentry: Docker for reproducible research

This course uses the Author Carpentry template but is not an Author Carpentry lesson yet! Find out more on the progress of this project at https://github.com/AuthorCarpentry/planning/issues/3.

Reproducibility of computational results is crucial in modern algorithm-based research. In this lesson, we introduce Docker as a useful tool to (a) document your computational environment and (b) make a computational environment transferable across machines and thus archiveable. The intention of this course is to showcase Docker as a useful tool for scientists, even if they are not regular users of the command line, which this course is completely based on.

Content Contributors: Daniel Nüst
Lesson Maintainers: Daniel Nüst
Lesson status: In Development

Learning Objectives:
- Docker basics (images, Dockerfiles, containers)
- RStudio in a container
- Jupyter Notebook in a container
Rocker: https://github.com/rocker-org

https://hub.docker.com/r/rocker/rstudio/

Base containers (r-base, r-devel, r-ver, ..)

Use case containers (r-devel-ubsan-clang, ..)

Stacks (tidyverse, geospatial, ..)

docker run -it -p 8787:8787 rocker/rstudio
http://localhost:8787/ (rstudio/rstudio)
https://bioconductor.org/help/docker/

Current Containers

Maintained by the Bioconductor Core Team: bioc-issue-bot@bioconductor.org

- bioconductor/devel_base2
- bioconductor/devel_core2
- bioconductor/release_base2
- bioconductor/release_core2

Maintained by Steffen Neumann: sneumann@ipb-halle.de

Maintained as part of the "PhenoMeNal, funded by Horizon2020 grant 654241"

- bioconductor/devel_protemcore2
- bioconductor/devel_metabolomics2
- bioconductor/release_protemcore2
- bioconductor/release_metabolomics2

Maintained by Laurent Gatto: lg390@cam.ac.uk

- bioconductor/devel_proteomics2
- bioconductor/release_proteomics2

https://hub.docker.com/u/bioconductor/
TAKE A DEEP BREATH
Generating Dockerfiles for reproducible research with R

30 May 2017

This post is the draft of the vignette for a new R package by o2r team members Matthias and Daniel. Find the original file in the package repository on GitHub.

1. Introduction
2. Creating a Dockerfile
3. Including resources
4. Image metadata
5. Further customization
6. CLI
7. Challenges
8. Conclusions and future work
Metadata

1. Introduction

Even though R is designed for open and reproducible research, users who want to share their work with others are facing challenges. Sharing merely the R script or R Markdown document should warrant reproducibility, but many...
Packaging interactive session

```r
> library(containerit); library("gstat"); library("sp")
> data(meuse)
> coordinates(meuse) = ~x+y
> data(meuse.grid)
> gridded(meuse.grid) = ~x+y
> v <- variogram(log(zinc)~1, meuse)
> m <- fit.variogram(v, vgm(1, "Sph", 300, 1))
> plot(v, model = m)

> dockerfile_object <- dockerfile()

INFO [2017-07-05 11:20:54] Trying to determine system requirements for the package(s) 'sp, gstat, zoo, futile.logger, xts, lambda.r, spacetime, futile.options, FNN, intervals, lattice' from sysreq online DB
INFO [2017-07-05 11:21:03] Adding CRAN packages: sp, gstat, zoo, futile.logger, xts, lambda.r, spacetime, futile.options, FNN, intervals, lattice
INFO [2017-07-05 11:21:03] Created Dockerfile-Object based on sessionInfo

> print(dockerfile_object)
FROM rocker/r-ver:3.4.1
LABEL maintainer="daniel"
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "sp", "gstat", "zoo", "futile.logger", "xts", "lambda.r", "spacetime", "futile.options", "FNN", "intervals", "lattice"]
WORKDIR /payload/
CMD ["R"]

> str(dockerfile_object, max.level = 2)
Formal class 'Dockerfile' with 4 slots
 ..@ image : Formal class 'From' with 2 slots
 ..@ maintainer : Formal class 'Label' with 2 slots
 ..@ instructions : List of 2
 ..@ cmd : Formal class 'Cmd' with 2 slots
```

17
Packaging a script w/ sysreqs dependency resolving

```r
library(rgdal); require(maptools)
nc <- rgdal::readOGR(system.file("shapes/", package="maptools"), "sids", verbose = FALSE)
proj4string(nc) <- CRS("+proj=longlat
+datum=NAD27")
plot(nc)
summary(nc)

scriptCmd <- CMD_Rscript("demo.R")
dockerfile_object <- dockerfile(
  from = "~/Documents/2017_useR/demo.R",
  cmd = scriptCmd)

print(dockerfile_object)
FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update 
&& apt-get install -y gdal-bin 
  libgdal-dev 
  libproj-dev
RUN ["install2.r", 
  "https://cloud.r-project.org'"", "rgdal", "sp",
  "lattice"]
WORKDIR /payload/
COPY [".", "]CMD ["R", "--vanilla", ",-f",
  "containerit_1a977e2dcdea.R"]
```

# curl https://sysreqs.r-hub.io/pkg/rgdal,sp,lattice/linux-x86_64-debian-gcc
# ["libgdal-dev", "libproj-dev", "gdal-bin"]
Running the container

```r
> write(dockerfile_object)

$ docker build -t user2017demo .

Sending build context to Docker daemon 6.054MB
Step 1/7 : FROM rocker/r-ver:3.4.1
3.4.1: Pulling from rocker/r-ver
c75480ad9aaf: Pull complete
[...
The following additional packages will be installed:
[...
* installing *source* package ‘foreign’ ...
[...
Successfully built e30936ac8687
Successfully tagged user2017demo:latest

$ docker run -it user2017demo

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
[...
> library(rgdal); require(maptools)
Loading required package: sp
> nc <- rgdal::readOGR(system.file("shapes/", package="maptools"), "sids", verbose = FALSE)
[...
> summary(nc)
Object of class SpatialPolygonsDataFrame
Coordinates:

          min        max
x -84.32385 -75.45698
y  33.88199  36.58965
Is projected: FALSE
[...]
```
CLI

Based on docopt

https://github.com/docopt/docopt.R
CLI

daniel@gin-nuest:~/.git/o2r/containeRit/tests/scripts$ containerit file -f sf.R
INFO [2017-05-15 17:38:36] Creating an R session with the following arguments:
  R --silent --vanilla -e "source(file = "/home/daniel/git/o2r/containeRit/tests/scripts/sf.R")"
> source(file = "/home/daniel/git/o2r/containeRit/tests/scripts/sf.R", echo = T

> cat("Hello from containerit!\n")
Hello from containerit!

> library("sf")
Linking to GEOS 3.5.1, GDAL 2.1.3, proj.4 4.9.2, lwgeom 2.3.2 r15302

> demo("meuse_sf", ask = FALSE)

demo(meuse_sf)
---- ~~~~~~~

> data(meuse, package = "sp") # load data.frame from sp
INFO [2017-05-15 17:38:37] Docker will try to install GDAL 2.1.3 from source
INFO [2017-05-15 17:38:37] Trying to determine system requirements for the package(s) 'sf, magrittr, DBI, units
m sysreq online DB
INFO [2017-05-15 17:38:38] Writing dockerfile to ./Dockerfile
daniel@gin-nuest:~/git/o2r/containeRit/tests/scripts$ cat Dockerfile
FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update 
    && apt-get install -y gdal-bin 
    libgeos-dev
    libproj-dev
    libbuunits2-dev
    make 
    wget
WORKDIR /tmp/gdal
RUN wget http://download.osgeo.org/gdal/2.1.3/gdal-2.1.3.tar.gz 
    && tar zxf gdal-2.1.3.tar.gz 
    && cd gdal-2.1.3 
    && ./configure 
    && make 
    && make install 
    && ldconfig 
    && rm -r /tmp/gdal
RUN ["install2.r", "-r https://cloud.r-project.org", "sf", "magrittr", "DBI", "units", "Rcpp", "udunits2"]
WORKDIR /payload/
COPY ["sf.R", "sf.R"]
CMD ["R"]
More

Labels for metadata
devtools session information (install from git under dev.)
Custom base images

Docker vs. R blog

doi:10.1145/2723872.2723882
Summary

Executable Research Compendia are fun

Docker is great

containerit makes Docker easier

(ready to go, less copy & paste, best practices, automatic system dependencies)

Benefits from Rocker (MRAN by default, …), harbor, …

Alternatives / potential for combination:

package management locally (packrat, pkgsnap, switchr/GRANBase) or remotely (MRAN timemachine/checkpoint), or install specific versions from CRAN or source (requireGitHub, devtools)
Outlook

Support our ERC creation service

Get feedback

Singularity
OCI/acbuild

CRAN

Docker + R paper for RJournal?

Package rplumber/jug web apps

Versioned packages and system libs (sf::sf_extSoftVersion())
Thanks!
What are your questions?

This is the o2r team and supporting university staff in alphabetical order:

- Rehan Chaudhary (ifgi, internship from 2017-01-17 to 2017-07-17)
- Matthias Hinz (ifgi, 2016-12 to 2017-03)
- Jim Jones (ULB)
- Dr. Stephanie Klötgen (ULB)
- Markus Konkol (ifgi)
- Jan Koppe (ifgi, student assistant, 2016-03 to 2016-08)
- Torben Kraft (ifgi, student assistant)
- Prof. Dr. Christian Kray (ifgi)
- Dr. Dirk Kussmann (ULB)
- Timm Kühl (ifgi, student assistant)
- Lukas Lohoff (ifgi, student assistant)
- Jörg Lorenz (ULB)
- Daniel Nüst (ifgi)
- Prof. Dr. Edzer Pebesma (ifgi)
- Holger Przybysz (ULB)
- Dr. Marc Schützeichel (ULB)
- Jan Suleiman (ifgi, student assistant)
- Dr. Beate Tröger (ULB)

@o2r_project

github.com/o2r-project

o2r.info