

# 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**

- #1: reproducibility helps to avoid disaster
- #2: reproducibility makes it easier to write papers
- #3: reproducibility helps reviewers see it your way
- #4: reproducibility enables continuity of your work
- #5: reproducibility helps to build your reputation

COMMENT | OPEN ACCESS

## Five selfish reasons to work reproducibly

Florian Markowetz 

*Genome Biology* 2015 16:274 | DOI: 10.1186/s13059-015-0850-7 | © Markowetz. 2015

Published: 8 December 2015

Abstract

Download PDF

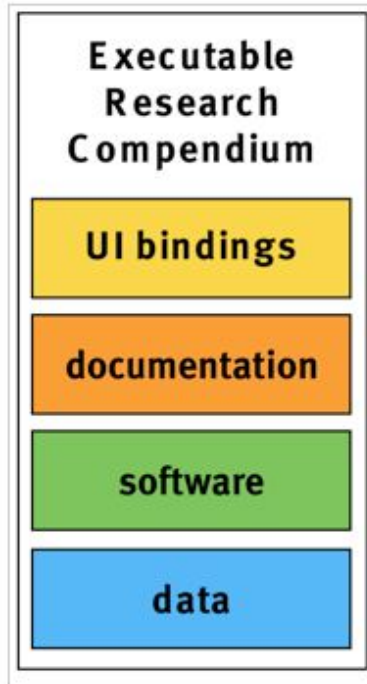
Export citations 

Table of Contents 

Abstract

Reproducibility: what's in it for me?

What's holding you



## Opening the Publication Process with Executable Research

**Compendia** Nüst D, Konkol M, Pebesma E, Kray C, Schutzzeichel M,

Przibytzin, H, Lorenz, J

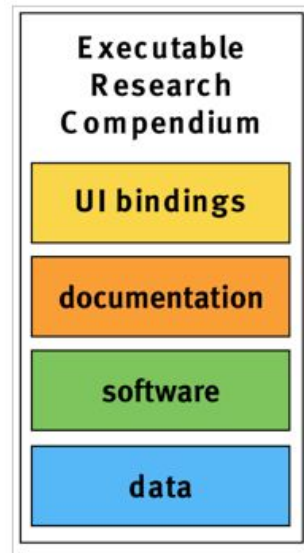
*Article(journal)*. *D-Lib Magazine*. 2017. doi: [10.1045/january2017-nuest](https://doi.org/10.1045/january2017-nuest);

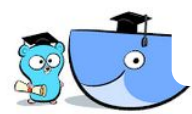
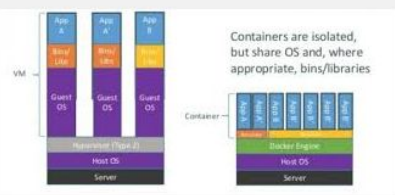
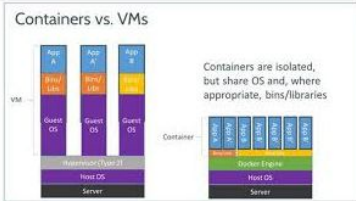
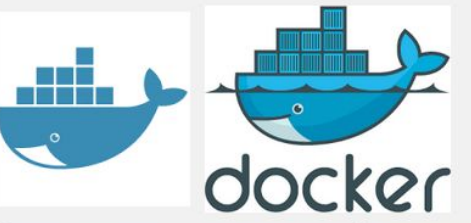
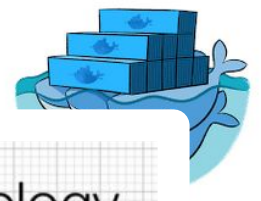
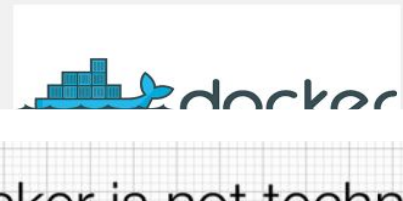
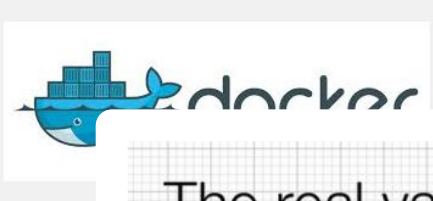
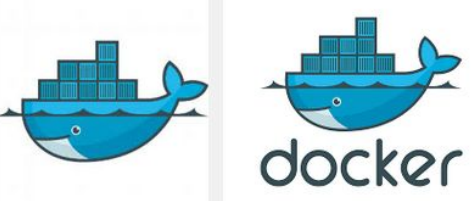
<http://www.dlib.org/dlib/january17/nuest/01nuest.html>

**o2r** opening  
reproducible  
research

# 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





[Docs hackathon results](#)[Get Docker](#)[Get started](#)[Get started with Docker](#)[Learn by example](#)[Docker overview](#)[User guide](#)[Admin guide](#)[Troubleshoot Docker Engine](#)[Manage a swarm](#)[Secure Engine](#)[Extend Engine](#)[Open source at Docker](#)[View the docs archives](#)

# 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 virtual

[Docker v17.06 \(current\)](#)[Edit this page](#)[Request docs changes](#)[Get support](#)

### On this page:

[The Docker platform](#)[Docker Engine](#)[What can I use Docker for?](#)[Docker architecture](#)[The Docker daemon](#)[The Docker client](#)[Docker registries](#)[Docker objects](#)[The underlying technology](#)[Namespaces](#)[Control groups](#)[Union file systems](#)[Container format](#)[Next steps](#)

science

data science

research

reproducibility

replication

package &

separate

applications and

their dependencies

for cloud

infrastructures

<https://docs.docker.com/engine/docker-overview/#the-docker-client>

<https://docs.docker.com/engine/docker-overview/>



Docs hackathon results

Get Docker

Get started

Get started with Docker

Learn by example

Docker overview

User guide

Admin

Troubleshooting

Managing

Secure

Extending

Open source at Docker

View the docs archives

<https://docs.docker.com/engine/docker-overview/#the-docker-client>

# Docker overview

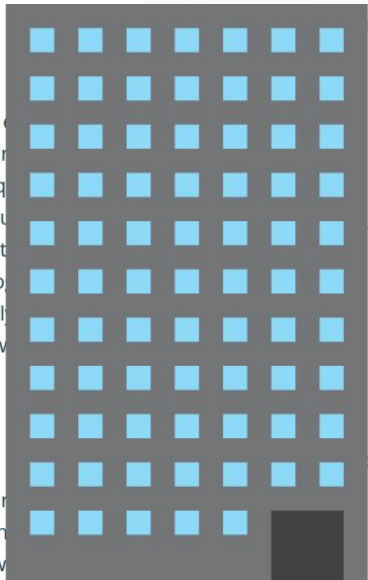
Estimated reading time: 10 minutes

Docker is an open platform for developing, shipping, and running applications. Docker lets you 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 methodology for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

## Docker platform

The Docker platform provides the ability to package and run applications in a loosely isolated environment called containers. 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

Docker v17.06 (current) →  
Edit this page

Container format

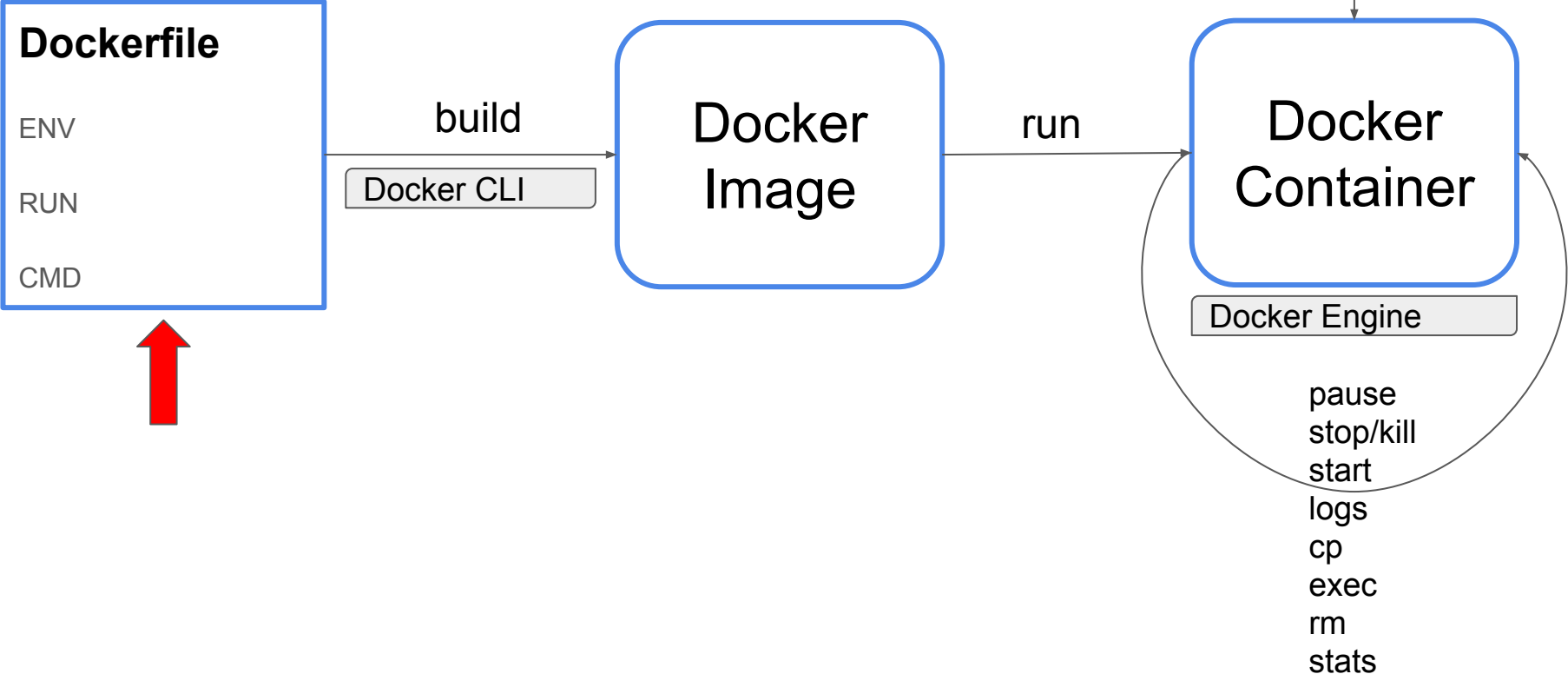
Next steps

science  
data science  
research  
reproducibility  
replication

package &  
separate  
applications and  
their  
dependencies for  
cloud  
infrastructures

<https://docs.docker.com/engine/docker-overview/>

# Docker basics



# 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>

The screenshot shows the Author Carpentry website. At the top, the logo 'Author Carpentry' is displayed. Below it is a navigation bar with links: Lesson, Getting started, RStudio, Jupyter, Transfer & Archive, Dockerfile, and Contact Us. The main heading is 'Author Carpentry : Docker for reproducible research'. A red warning message states: '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.' The main text explains that reproducibility is crucial in modern research and that the lesson introduces Docker as a tool for documenting and transferring computational environments. It lists content contributors (Daniel Nüst), lesson maintainers (Daniel Nüst), and the lesson status (In Development). The learning objectives are listed as: Docker basics, RStudio in a container, and Jupyter Notebook in a container.

Author Carpentry

Lesson Getting started RStudio Jupyter Transfer & Archive Dockerfile Contact Us

## 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 archivable. 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: [bloc-issue-bot@bioconductor.org](mailto:bloc-issue-bot@bioconductor.org)**

- [bioconductor/devel\\_base2](#)
- [bioconductor/devel\\_core2](#)
- [bioconductor/release\\_base2](#)
- [bioconductor/release\\_core2](#)

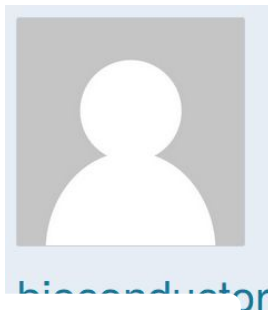
**Maintained by Steffen Neumann: [sneumann@ipb-halle.de](mailto:sneumann@ipb-halle.de)**









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

- [bioconductor/devel\\_protmetcore2](#)
- [bioconductor/devel\\_metabolomics2](#)
- [bioconductor/release\\_protmetcore2](#)
- [bioconductor/release\\_metabolomics2](#)

**Maintained by Laurent Gatto: [lg390@cam.ac.uk](mailto:lg390@cam.ac.uk)**

- [bioconductor/devel\\_proteomics2](#)
- [bioconductor/release\\_proteomics2](#)



 <a href="#">bioconductor/release_base</a> public	4 STARS	100K+ PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/release_core</a> public	4 STARS	50K+ PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/release_flow</a> public	0 STARS	10K+ PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/devel_core</a> public	5 STARS	10K+ PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/release_microarray</a> public	4 STARS	7.0K PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/devel_base</a> public	1 STARS	3.7K PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/release_sequencing</a> public	2 STARS	3.4K PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/release_proteomics</a> public	1 STARS	1.7K PULLS	<a href="#">&gt;</a> DETAILS
 <a href="#">bioconductor/devel_base2</a> public	1 STARS	1.7K PULLS	<a href="#">&gt;</a> DETAILS

<https://hub.docker.com/u/bioconductor/>



# containerit

<https://github.com/o2r-project/containerit>

<http://o2r.info/2017/05/30/containerit-package/>

o2r-project / containerit

Unwatch 10 Star 24 Fork 5

<> Code Issues 58 Pull requests 0 Projects 1 Settings Insights

Package an R workspace and all dependencies as a Docker container

docker reproducible-research reproducible-science r dockerfile Manage topics

111 commits 2 branches 0 releases 4 contributors GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

nuest committed on GitHub Merge pull request #71 from nuest/master Latest commit 7965258 7 days ago



Opening Reproducible Research is a DFG-funded research project by Institute for Geoinformatics (ifgi) and University and Regional Library (ULB), University of Münster, Germany

Home  
About  
Imprint  
Publications  
GitHub

## 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

7 days ago  
a month ago  
a month ago  
7 days ago  
a month ago  
7 months ago  
3 months ago  
a month ago  
3 months ago  
a year ago  
7 days ago  
7 months ago  
a month ago  
a year ago  
3 months ago  
3 months ago



# Packaging interactive session

```
> 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
```

# Packaging a script w/ sysreqs dependency resolving

```
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)

# curl https://sysreqs.r-hub.io/pkg/rgdal,sp,lattice/linux-x86\_64-debian-gcc
# ["libgdal-dev", "libproj-dev", "gdal-bin"]
```

```
> 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", "-r
'https://cloud.r-project.org'", "rgdal", "sp",
"lattice"]
WORKDIR /payload/
COPY [".", "."]
CMD ["R", "--vanilla", "-f",
"containerit_1a977e2dcdea.R"]
```

# Running the container

```
> write(dockerfile_object)
INFO [2017-07-06 10:10:05] Writing dockerfile to
/home/daniel/Documents/2017_useR/Dockerfile
```

```
$ 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>

```
daniel@gin-nuest:~/git/o2r$ containerit --help
container_it.R is a command-line interface to the R package containerit.
It packages R sessions, scripts, workspaces and vignettes together with all dependencies to ex

Usage:  container_it.R dir [options] [--copy arg] [-d <DIR>]
        container_it.R file [options] [--copy arg] [--cmd-render <FORMAT> | --cmd-R-file] <FIL
        container_it.R session [options] [-e <EXPR> ...]
        container_it.R [--help | --version]

Modes:
  dir      Searches the given directory for R / Rmarkdown files and uses the first encounte
  file     Packages a given R script or R markdown file.
           Optionally, the container can run a batch execution or rendering of the given fi
  session  Packages an empty R session in which a series of R commands may be executed (see

Options (for all modes):
  --force -f          Force writing output even if a file of the same name already exists
  --image <ARG>      Specify the Docker image that shall be used for the docker container (FR
                    By default, the image is determined from the given r_version,
                    while the version is matched with tags from the base image rocker/r-ver
                    see details about the rocker/r-ver at https://hub.docker.com/r/rocker/r-
  --maintainer -m <ARG>  Name / email of the dockerfile's maintainer (will be read from envir
  --no-write          Don't write dockerfile to output file
  --no-vanilla        Package a session / file without using the vanilla flag
                    (warning: site and environment files currently cannot be included in the
  --output -o FILE    Path and name of the output Dockerfile [default: ./Dockerfile]
  --print -p          Print dockerfile to the console
  --r_version -r <ARG> Specify an R version number that should run inside the container.
                    By default, the version of the currently linked R instance is used.
  --save -s <objects> ... Save a list of objects from the workspace to an .RData file (over
  --save-image -i     Save the current workspace to an .RData file
  --soft              Whether to include soft dependencies among the system dependencies of R
  --quiet -q          Run containerit as silent as possible (print only errors and warnings) [

Other:
  --copy -c script | script_dir | <copy_file> ...
                    Indicates whether and how a workspace should be copied.
                    For the modes 'dir' and 'file' containerit copies either the given input
                    as a list of individual files and directories that should be located bel
```

# CLI

```
daniel@gin-nuest:~/git/o2r/containereRit/tests/scripts$ containerit file -f sf.R
INFO [2017-05-15 17:38:36] Executing R script file in sf.R locally.
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/containereRit/tests/scripts/sf.R\", echo = TRUE)"
> source(file = "/home/daniel/git/o2r/containereRit/tests/scripts/sf.R", echo = TRUE)

> 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
```

■ ■ ■

# CLI

```
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] Created Dockerfile-Object based on sf.R
INFO [2017-05-15 17:38:38] Writing dockerfile to ./Dockerfile
daniel@gin-nuest:~/git/or/containers/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 \
      libudunits2-dev \
      make \
      wget
WORKDIR /tmp/gdal
RUN wget http://download.osgeo.org/gdal/2.1.3/gdal-2.1.3.tar.gz \
  && tar xzf 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.)

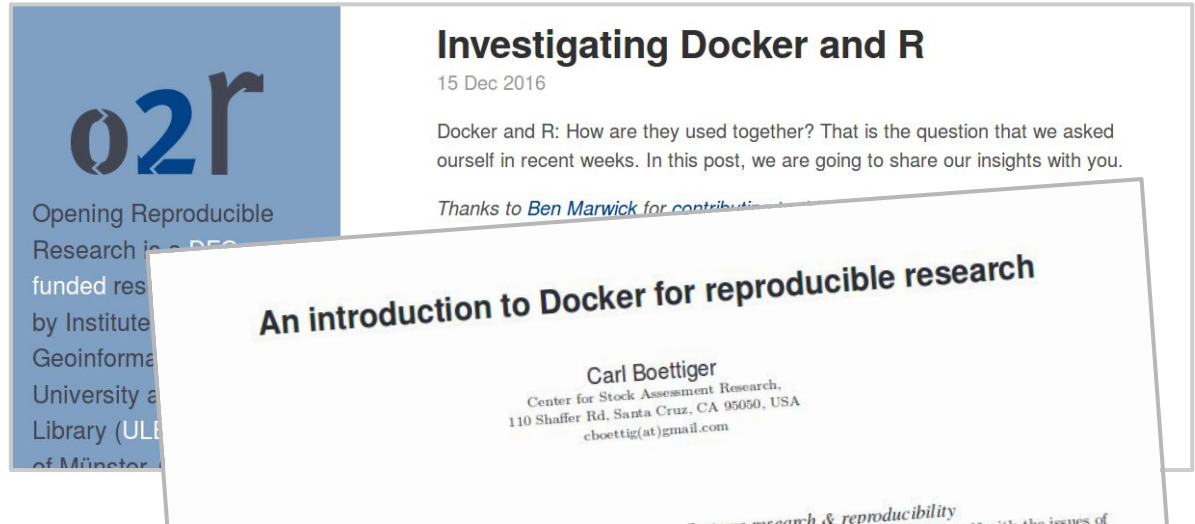
Custom **base images**

## Docker vs. R blog

<http://bit.ly/docker-r>

Boettiger, Carl. 2015. "An Introduction to Docker for Reproducible Research, with Examples from the R Environment." *ACM SIGOPS Operating Systems Review* 49 (January): 71–79.

doi:[10.1145/2723872.2723882](https://doi.org/10.1145/2723872.2723882)



# Summary

*Executable Research Compendia* are fun

**Docker** is great

**containerit** makes Docker easier

**BETA!**

(DRY, 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()`)

A screenshot of a GitHub Projects board for the repository 'o2r-project / containerit'. The board is organized into three columns: 'Current sprint 1', 'Backlog 18', and 'Ideas &amp; Discussion 14'. The 'Current sprint' column contains one item: 'Integrate with o2r-muncher'. The 'Backlog' column lists several items, including 'Add all dependencies to Dockerfile, even transitive ones', 'Create acbuild creation scripts', 'Put codemeta document for all installed software into container metadata (LABEL)', 'Load prepared session on container startup', 'Package a non-vanilla R session', and 'Selective session info'. The 'Ideas &amp; Discussion' column lists items like 'Integration with existing packages connected to Docker', 'Add automated tests, also for non-Unix systems', 'works\_with\_R', 'Read/parse Dockerfiles', 'Generate Vagrantfile', 'Check conformance to ROpenSci guidelines', and 'Add linter to testthat tests'. The interface includes navigation tabs for Code, Issues (59), Pull requests (0), Projects (1), Settings, and Insights.

<https://github.com/o2r-project/containerit/projects/1>



# 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 Kühnel (ifgi, student assistant)
- Lukas Lohoff (ifgi, student assistant)
- Jörg Lorenz (ULB)
- **Daniel Nüst** (ifgi)
- **Prof. Dr. Edzer Pebesma** (ifgi)
- Holger Przybytzin (ULB)
- Dr. Marc Schutzzeichel (ULB)
- Jan Suleiman (ifgi, student assistant)
- Dr. Beate Tröger (ULB)



The image shows a composite of two screenshots. The top screenshot is a Twitter profile for the account @o2r\_project. The profile bio states: "Opening reproducible research since 2016 by connecting digital curation with geosciences and interactive publications based on open science, #rstals and #Docker". It lists 209 tweets, 150 followers, and 102 people following. A tweet from Lorena Barba (@LorenaABarba) is visible, discussing primary research papers and reproducibility. The bottom screenshot is the website for "o2r Opening Reproducible Research". The website header includes navigation links for Features, Business, Explore, Marketplace, and Pricing. Below the header, there are sections for "Repositories" and "Pinned repositories". The "Pinned repositories" section shows two items: "architecture" (Architecture and overarching documentation for o2r microservices) and "erc-spec" (Executable Research Compendium specification and guides). Both repositories have a "Shell" icon.

**@o2r\_project**

**github.com/o2r-project**

**o2r.info**