R and Docker

Dirk Eddelbuettel
dirk.eddelbuettel@R-Project.org
edd@debian.org
@eddelbuettel

Docker Chicago Meetup
Lightning Talk
25 September 2014
Outline

1. R
   - Context
   - Overview
   - Outline
> fortunes::fortune(92)

```r
##
## If you don't go with R now, you will someday.
## -- David Kane (on whether to use R or S-PLUS)
## R-SIG-Finance (November 2004)
```

A Good Forecast from About 10 Years Ago
A language and an environment (cf R FAQ)

Has forever altered the way people analyze, visualize and manipulate data (cf 1999 ACM citation)

A vibrant community and ecosystem: CRAN + BioConductor provide > 6k packages that “just work”

The lingua franca of (applied) statistical research

Reliable cross-platform + cross-operating system

Yet occasional challenges of getting R and code to collaborators, students, ...
CRAN and testing to ensure R “just works”
- There is a culture of fairly stringent testing
- Which sometimes poses issues for those unable to get the newest and greatest
- Example are the fresh-from-repo development version, as well as instrumented version (“sanitizers”)

Distribution of R and RStudio as an appliance
- Natural fit for Docker
- Short example later
Outline

2. Docker
   - Intro
   - Setup
   - Dockerfiles
   - Example: Sanitizers
   - Example: R Studio Server
What is Docker?

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications. Consisting of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows, Docker enables apps to be quickly assembled from components and eliminates the friction between development, QA, and production environments.

Ok, seriously, what is Docker?

Docker is a very lightweight abstraction using recent Linux kernel features which lets us to run code in cheap (to launch) and easy (to build) units: containers. We can share containers across OSs.

It changes how we build and test R (and R packages).
Getting started with Docker

Installation

- Most Unix variants have detailed (short) instructions on Docker website. Should just work.
- Or: `sudo apt-get install docker.io`
  
  (requires Ubuntu 14.04 or Debian testing; you also want to add yourself to group `docker`)

- On Windows or OS X: Use `boot2docker` which installs an appliances for you (with docker, git, virtualbox, ...).
  
  Tested on Windows at work. Appears to Just works too.
After installation, run

```
docker pull ubuntu
```

to pull a set of pre-built initial images.
### Getting started with Docker

#### First steps: Listing images

This one call gets us

```bash
edd@max:~$ docker images | head -8
REPOSITORY               TAG     IMAGE ID              CREATED             VIRTUAL SIZE
------                   ------   ----------              -------             --------
e..l/debian-rstudio     latest   aef9f264b093           3 hours ago        1.184 GB
eddelbuettel/rocker      latest   d82f0f7dc624           3 hours ago        430.2 MB
debian                  6.0      28e25859dccc8           6 weeks ago        78.44 MB
debian                  squeeze  28e25859dccc8           6 weeks ago        78.44 MB
debian                  6        28e25859dccc8           6 weeks ago        78.44 MB
debian                  6.0.10   28e25859dccc8           6 weeks ago        78.44 MB
debian                  wheezy   c1eec48018ed           6 weeks ago        85.18 MB
debian                  7        c1eec48018ed           6 weeks ago        85.18 MB
```
Dockerfiles are ‘recipes’ which create images.

Here is ’add-r’, a simple recipe to just add R:

```bash
## start with the Debian testing
FROM debian:testing
MAINTAINER Dirk Eddelbuettel edd@debian.org

## Remain current
RUN apt-get update -qq
RUN apt-get dist-upgrade -y

RUN apt-get install -y --no-install-recommends \
    r-base r-base-dev r-recommended littler
RUN ln -s /usr/share/doc/littler/examples/install.r \
    /usr/local/bin/install.r
```
I have created a few (partially nested) Docker images

- with basic R as a binary package
- with R-devel freshly built from svn source
- with R-devel and Address Sanitizer (ASAN) + Undefined Behavior Sanitizer (UBSAN)
- with R-studio

which are downloadable from hub.docker.com under eddelbuettel (though the repo layout is in flux and still changing).
Getting started with Docker
Sanitizer Test with R

$ docker run -v `pwd`:/mytmp -t b524252a3462 \\  
    R CMD check --no-manual --no-build-vignettes \\  
    /mytmp/sanitizers_1.0.tar.gz
* using log directory '//'sanitizers.Rcheck'
* using R version 3.1.0 (2014-04-10)
* using platform: i486-pc-linux-gnu (32-bit)
* using session charset: ASCII
* using option '--no-build-vignettes'
* checking for file 'sanitizers/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'sanitizers' version '1.0'
[...]

which tests (fine) under the release version.
Getting started with Docker

Sanitizer Test with R

```bash
$ docker run -v `pwd`:/mytmp -t b524252a3462 \
  Rdevel CMD check --no-manual --no-build-vignettes \
  /mytmp/sanitizers_1.0.tar.gz
edd@don:~/Dropbox/src/san-ubsan$ docker run -v `pwd`:/mytmp -t b524252a3462 \
  Rdevel CMD check --no-manual --no-build-vignettes \
  /mytmp/sanitizers_1.0.tar.gz
* using log directory '//sanitizers.Rcheck'
* using R Under development (unstable) (2014-06-20 r65987)
* using platform: i686-pc-linux-gnu (32-bit)
* using session charset: ASCII
* using option '--no-build-vignettes'
* checking for file 'sanitizers/DESCRIPTION' ... OK
[...]
```

which tests under the R-development version (which has been enabled for Sanitizer checks) and ...
Getting started with Docker
Sanitizer Test with R

[...]  
* checking tests ...  
  Running 'simple.R'  
  ERROR  
Running the tests in 'tests/simple.R' failed.  
Last 13 lines of output:  
  Freed heap region:  fd  
  Stack left redzone:  f1  
  Stack mid redzone:  f2  
  Stack right redzone:  f3  
  Stack partial redzone:  f4  
  Stack after return:  f5  
  Stack use after scope:  f8  
  Global redzone:  f9  
  Global init order:  f6  
  Poisoned by user:  f7  
  Contiguous container OOB:fc  
  ASan internal:  fe  
==258==ABORTING  
$
Getting started with Docker
Sanitizer Test with R

- Out of this grew a new (and still small) CRAN package “sanitizers”
- It regroups known tests which fail under ASAN and UBSAN
- This allows us to verify that the test program fails when it is supposed to fail (ie to avoid ’Type II errors’)
- The package currently contains two for ASAN and one for UBSAN
- Contributions welcome!
Getting started with Docker
RStudio Server via Docker

- Docker excels at containerizing applications
- It is particularly suitable for headless / networked app
- RStudio Server is one such application
- By shipping RStudio, plus whichever domain-specific package you may need, R is turned into a true cross-platform appliance
- Docker allows local use where RStudio Server would otherwise required a networked Linux server
Getting started with Docker

RStudio Server via Docker – running on Windows via Boot2Docker
Summary

- R has become the dominant applications platform
- Docker is a fabulous way to containerize applications
- Docker can change how we build, test, distribute, ... applications
- Docker offers fantastic new ways to ship and deploy
- Large upside to many Open Source communities
Acknowledgements

A shoutout to ...

- Matt Whiteley for a fabulous short tutorial on building Docker on i386
- Paul Tagliamonte for his excellent work on Debian’s docker package, and for lending an ear now and then
- Carl Boettiger for the ongoing joint work on Docker containers for R
- And of course Docker and GitHub for what they do