On Building Open Source Programs

A Few Somewhat Disjoint Comments

Dirk Eddelbuettel

Invited Talk, Gretl Conference 2021

3 Jun 2021

https://dirk.eddelbuettel.com/papers/gretlconf_building_jun2021.pdf

ILLINOIS

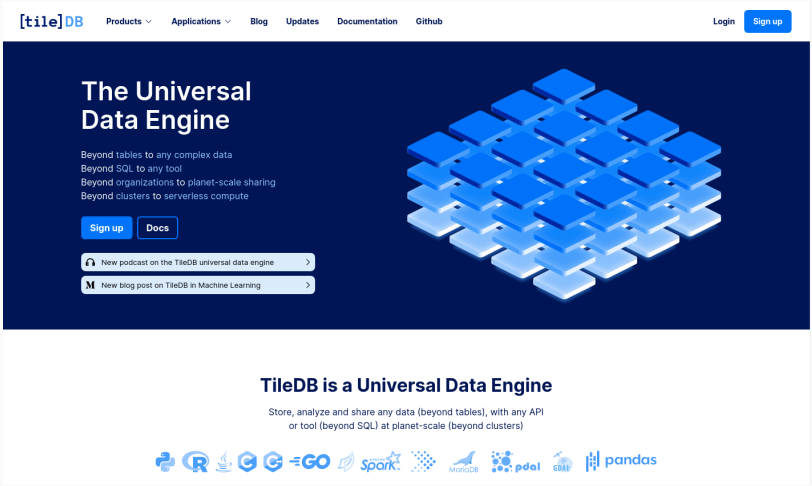
Outline

Today's Talk

- Gretl (and Gretl from Debian's View)
- A Little Docker
- The Action around GitHub Actions
- Open Build Service

Quick Bio for Context

My day job


[tile]DB
Products
Applications
Blog
Updates
Documentation
Github
Login
Sign up
The Universal Data Engine
Beyond tables to any complex data
Beyond SQL to any tool
Beyond organizations to planet-scale sharing
Beyond clusters to serverless compute
Sign up
Docs
New podcast on the TileDB universal data engine
New blog post on TileDB in Machine Learning
TileDB is a Universal Data Engine
Store, analyze and share any data (beyond tables), with any API
or tool (beyond SQL) at planet-scale (beyond clusters)
pandas

My day job

- Open Source (plus Service) as a business
- Earlier 20+ years as a Quant in Finance (coming in as Econometrician)
- Some relevant experience on different tools and approaches
- Influenced by ‘commonly seen’ best practices:
  - incremental changes
  - continued integration
  - test, test, test

My other jobs

Academic

- (Adjunct) Clinical Professor, University of Illinois
  - teaching STAT 447, a Data Science Programming Methods class

Open Source

- Debian developer
  - since 1995, currently maintaining about 170+ packages (including `gretl`)
- R contributor / package author
  - since 2002, author / maintainer of 60+ CRAN packages, R Foundation Board Member
- Rocker Project co-founder
  - Docker for R, including official 'r-base' image

Gretl

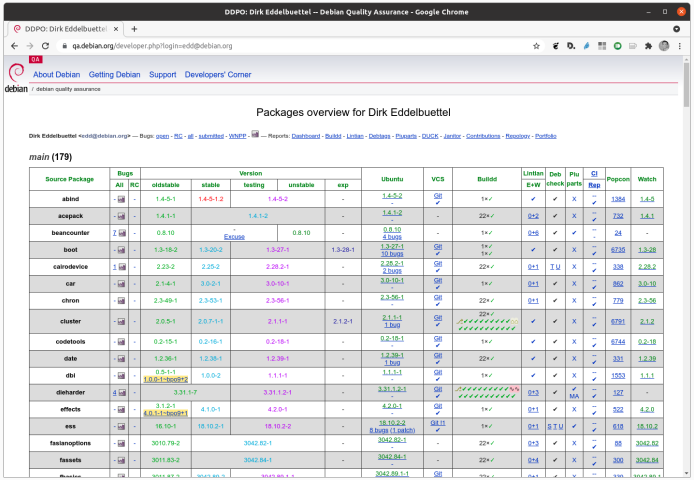
Gretl as an Open Source Package

Key Points

- Really impressive breadth of build features
- Really thorough use of build tools
- Broad reach of build targets and platforms
- Impressive continued development over many years
- Creation of a lively community supporting Gretl
- Using a well-understand and widely-used copyleft license
- My one claim to fame ...
- ... may have been to suggest to Allin to add a reader for R data

Debian (and Gretl)

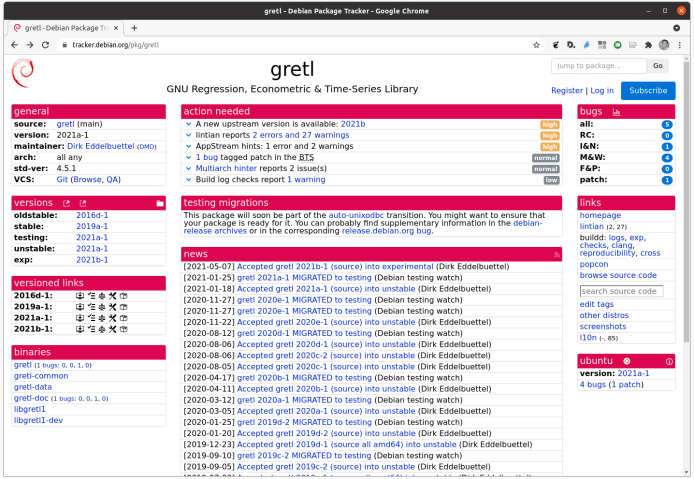
Maintainer View



Per Maintainer

- very dense display
- each package per row
- version status per release
- issue count
- build status
- QA check status
- `qa.debian.org/developer.php?login=edd@debian.org`

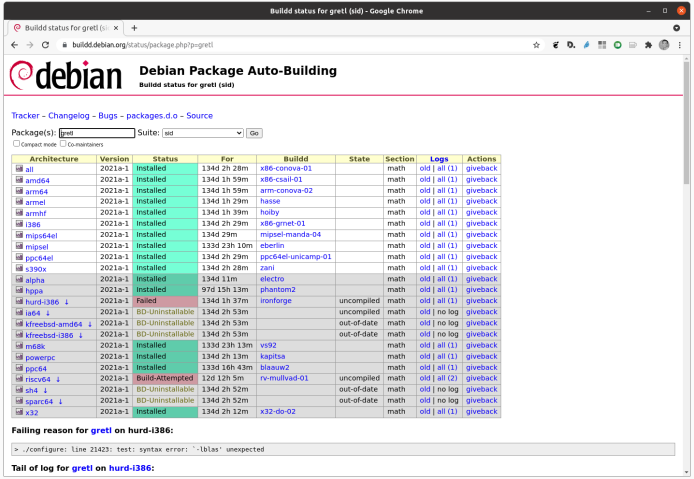
Package (Tracker) View



Package View

- version, maintainer, issues (by severity)
- links to builds logs (see below) and more
- version log
- binaries
- `tracker.debian.org/pkg/gretl`

Build Status - Distro

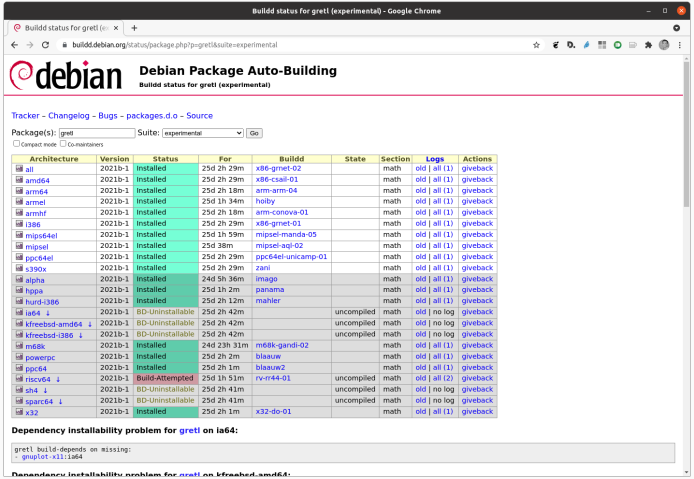


Architecture	Version	Status	For	Buildd	State	Section	Logs	Actions
all	2021a-1	Installed	134d 2h 28m	x86-conova-01		math	old | all (1)	giveback
amd64	2021a-1	Installed	134d 1h 59m	x86-csail-01		math	old | all (1)	giveback
arm64	2021a-1	Installed	134d 1h 59m	arm-conova-02		math	old | all (1)	giveback
armel	2021a-1	Installed	134d 1h 29m	hasse		math	old | all (1)	giveback
armhf	2021a-1	Installed	134d 1h 39m	hoiby		math	old | all (1)	giveback
i386	2021a-1	Installed	134d 2h 29m	x86-grnet-01		math	old | all (1)	giveback
mips64el	2021a-1	Installed	134d 29m	mipsel-manda-04		math	old | all (1)	giveback
mipsel	2021a-1	Installed	133d 23h 10m	eberlin		math	old | all (1)	giveback
ppc64el	2021a-1	Installed	134d 2h 29m	ppc64el-unicamp-01		math	old | all (1)	giveback
s390x	2021a-1	Installed	134d 2h 28m	zani		math	old | all (1)	giveback
alpha	2021a-1	Installed	134d 11m	electro		math	old | all (1)	giveback
hppa	2021a-1	Installed	97d 15h 13m	phantom2		math	old | all (1)	giveback
hurd-i386 ↓	2021a-1	Failed	134d 1h 37m	ironforge	uncompiled	math	old | all (1)	giveback
ia64 ↓	2021a-1	BD-Uninstallable	134d 2h 53m		uncompiled	math	old | no log	giveback
kfreebsd-amd64 ↓	2021a-1	BD-Uninstallable	134d 2h 53m		out-of-date	math	old | no log	giveback
kfreebsd-i386 ↓	2021a-1	BD-Uninstallable	134d 2h 53m		out-of-date	math	old | no log	giveback
m68k	2021a-1	Installed	133d 23h 13m	vs92		math	old | all (1)	giveback
powerpc	2021a-1	Installed	134d 2h 13m	kapitsa		math	old | all (1)	giveback
ppc64	2021a-1	Installed	133d 16h 43m	blaauw2		math	old | all (1)	giveback
riscv64 ↓	2021a-1	Build-Attempted	12d 12h 5m	rv-mullvad-01	uncompiled	math	old | all (2)	giveback
sh4 ↓	2021a-1	BD-Uninstallable	134d 2h 52m		out-of-date	math	old | no log	giveback
sparc64 ↓	2021a-1	BD-Uninstallable	134d 2h 52m		out-of-date	math	old | no log	giveback
x32	2021a-1	Installed	134d 2h 12m	x32-do-02		math	old | all (1)	giveback

Build Release

- by architecture
- showing official and 'candidate' architectures
- links to build logs
- `buildd.debian.org/status/package.php?p=gretl`

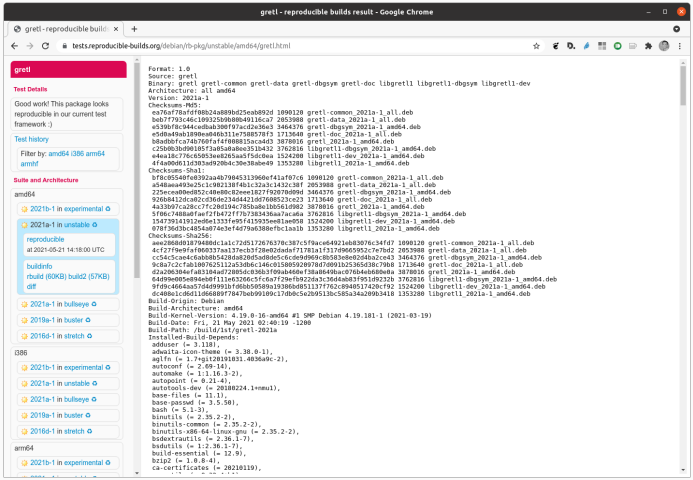
Build Status - 'Experimental'



Build of Experimental

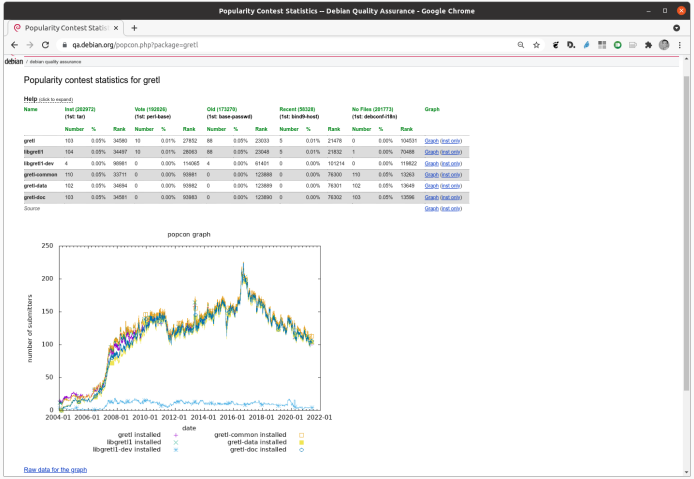
- again by architecture
- showing official and 'candidate' architectures
- links to build logs
- `buildd.debian.org/status/package.php?p=gretl&suite=experimental`

Reproducible Builds



Reproducible

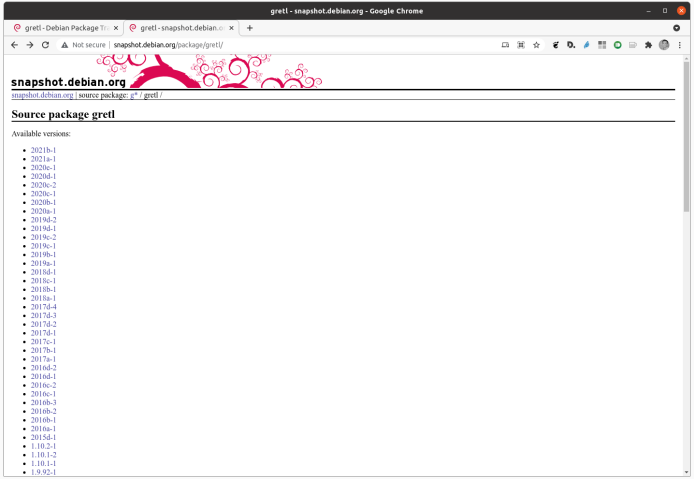
- an aside but interesting
- initially Debian only
- now across distros
- important for trust in reproducibility
- ‘bit by bit’ identical
- tests.reproducible-builds.org/debian/rb-pkg/unstable/amd64/gretl.html



Popularity Context

- older *opt-in* iniative
- unclear statistical ‘survey properties’
- baseline is *e.g.* the glibc package at 200k
- `qa.debian.org/popcon.php?package=gretl`

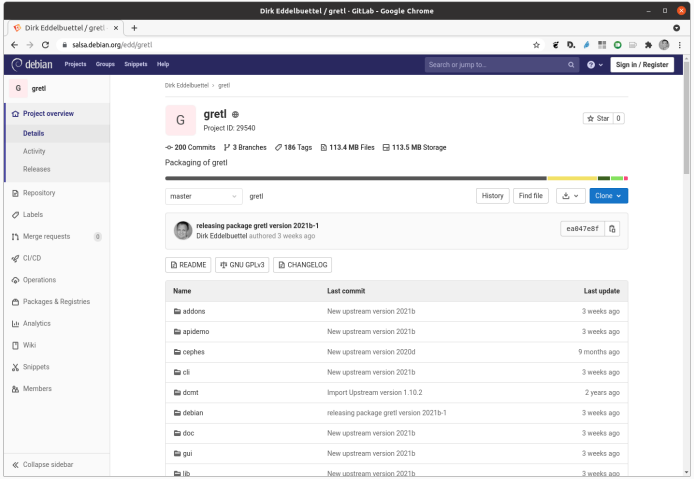
Snapshot



Snapshots

- useful (if little known)
- access to old source versions *and binaries*
- useful for comparison, regression testing, …
- `snapshot.debian.org/package/gretl`

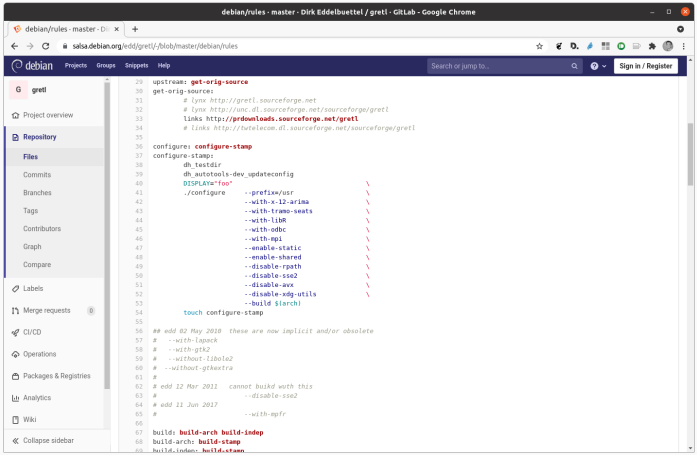
Packaging



Package Repo

- upstream plus packaging
- mostly internal and used by package builds
- but all packaging ‘visible’
- `salsa.debian.org/edd/gretl`

Packaging Details



Package Repo

- all details of packaging a little overwhelming
- but at least starting point for mods
- `salsa.debian.org/edd/gretl/-/blob/master/debian/rules`

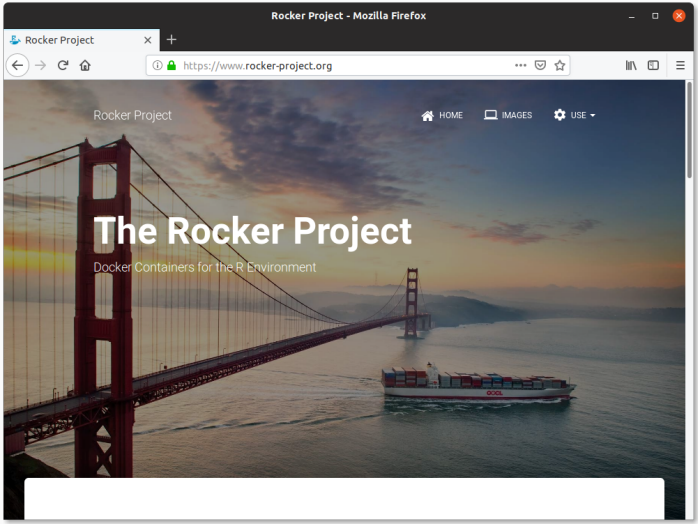
Docker

Docker

Key Points

- Big fan and early user/contributor
- Part of the Rocker Project offering different R containers
- Our rocker/r-base is also the official r-base
- Possibly way for gretl to just offer gretl contains
- With tagged releases: gretl:2021a, gretl:2021b, ...

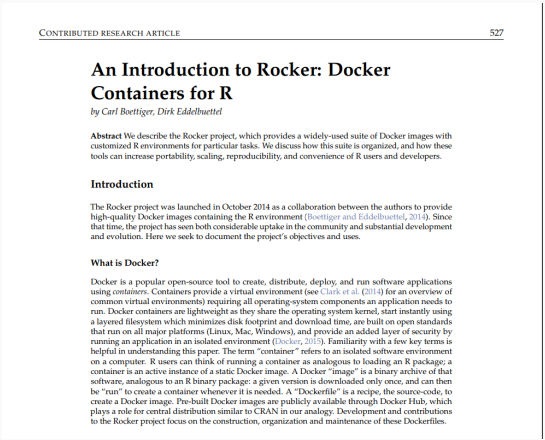
Rocker Project



Rocker

- base containers used for derived containers
- 'versioned' container popular for snapshots
- now also ML flavours
- popular containers with over 500k downloads
- our rocker/r-base is official r-base
- `rocker-project.org`

Rocker Paper: R Journal, 2017, Issue 65



CONTRIBUTED RESEARCH ARTICLE 527

An Introduction to Rocker: Docker Containers for R

by Carl Boettiger, Dirk Eddelbuettel

Abstract We describe the Rocker project, which provides a widely-used suite of Docker images with customized R environments for particular tasks. We discuss how this suite is organized, and how these tools can increase portability, scaling, reproducibility, and convenience of R users and developers.

Introduction

The Rocker project was launched in October 2014 as a collaboration between the authors to provide high-quality Docker images containing the R environment (Boettiger and Eddelbuettel, 2014). Since that time, the project has seen both considerable uptake in the community and substantial development and evolution. Here we seek to document the project's objectives and uses.

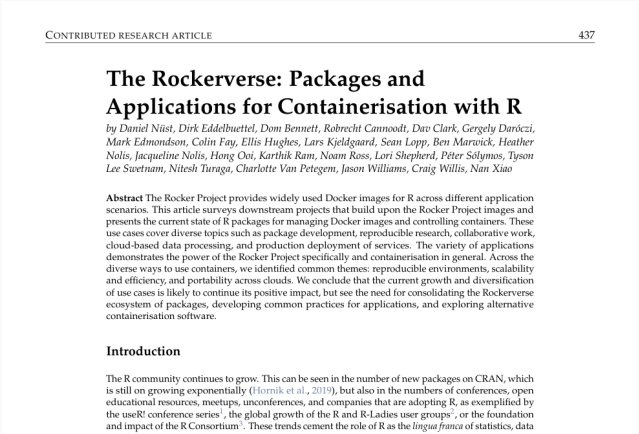
What is Docker?

Docker is a popular open-source tool to create, distribute, deploy, and run software applications using *containers*. Containers provide a virtual environment (see Clark et al. (2014) for an overview of common virtual environments) requiring all operating-system components an application needs to run. Docker containers are lightweight as they share the operating system kernel, start instantly using a layered filesystem which minimizes disk footprint and download time, are built on open standards that run on all major platforms (Linux, Mac, Windows), and provide an added layer of security by running an application in an isolated environment (Docker, 2015). Familiarity with a few key terms is helpful in understanding this paper. The term "container" refers to an isolated software environment on a computer. R users can think of running a container as analogous to loading an R package; a container is an active instance of a static Docker image. A Docker "image" is a binary archive of that software, analogous to an R binary package: a given version is downloaded only once, and can then be "run" to create a container whenever it is needed. A "Dockerfile" is a recipe, the source-code, to create a Docker image. Pre-built Docker images are publicly available through Docker Hub, which plays a role for central distribution similar to CRAN in our analogy. Development and contributions to the Rocker project focus on the construction, organization and maintenance of these Dockerfiles.

Early Paper

- describes project
- and two (early) stacks
- `journal.r-project.org/archive/2017/RJ-2017-065/`

Rockerverse Paper: R Journal, 2020, Issue 7



CONTRIBUTED RESEARCH ARTICLE 437

The Rockerverse: Packages and Applications for Containerisation with R

by Daniel Nüst, Dirk Eddelbuettel, Dom Bennett, Robrecht Cannoodt, Dav Clark, Gergely Daróczi, Mark Edmondson, Colin Fay, Ellis Hughes, Lars Kjeldgaard, Sean Lopp, Ben Marwick, Heather Nolis, Jacqueline Nolis, Hong Ooi, Karthik Ram, Noam Ross, Lori Shepherd, Péter Sólymos, Tyson Lee Swetnam, Nitesh Turaga, Charlotte Van Petegem, Jason Williams, Craig Willis, Nan Xiao

Abstract The Rocker Project provides widely used Docker images for R across different application scenarios. This article surveys downstream projects that build upon the Rocker Project images and presents the current state of R packages for managing Docker images and controlling containers. These use cases cover diverse topics such as package development, reproducible research, collaborative work, cloud-based data processing, and production deployment of services. The variety of applications demonstrates the power of the Rocker Project specifically and containerisation in general. Across the diverse ways to use containers, we identified common themes: reproducible environments, scalability and efficiency, and portability across clouds. We conclude that the current growth and diversification of use cases is likely to continue its positive impact, but see the need for consolidating the Rockerverse ecosystem of packages, developing common practices for applications, and exploring alternative containerisation software.

Introduction

The R community continues to grow. This can be seen in the number of new packages on CRAN, which is still on growing exponentially (Hornik et al., 2019), but also in the numbers of conferences, open educational resources, meetups, unconferences, and companies that are adopting R, as exemplified by the useR! conference series[1], the global growth of the R and R-Ladies user groups[2], or the foundation and impact of the R Consortium[3]. These trends cement the role of R as the *lingua franca* of statistics, data

Recent Paper

- recent 'applications' paper
- very broad overview
- journal.r-project.org/archive/2020/RJ-2020-007/

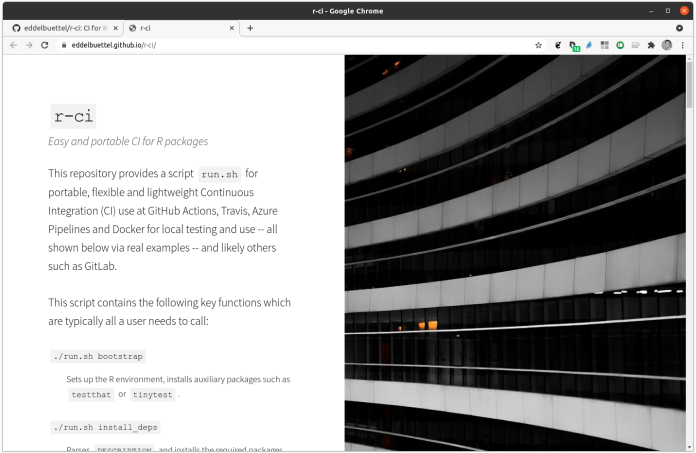
GitHub and GitHub Actions

GitHub (and GitHub Actions)

Status

- `git` won the distributed version control systems competition
- Will admit having been a slow-ish converter (maybe eight years ago)
- But am all in now and keep code, papers, websites, course notes, … there
- `GitHub` has won the UI competition and is now synonymous to `git` for many
- Personally, I was not that much on the ‘quit because of MSFT’ train
- Not too concerned about `GitHub` for code: GitLab, Gitea.io, … offer alternatives
- But I still really dislike vendor lock-in
- Which makes rather sceptical about the current craze of complicated and sophisticated GitHub Actions ‘YAML’ programming

CI: r-ci

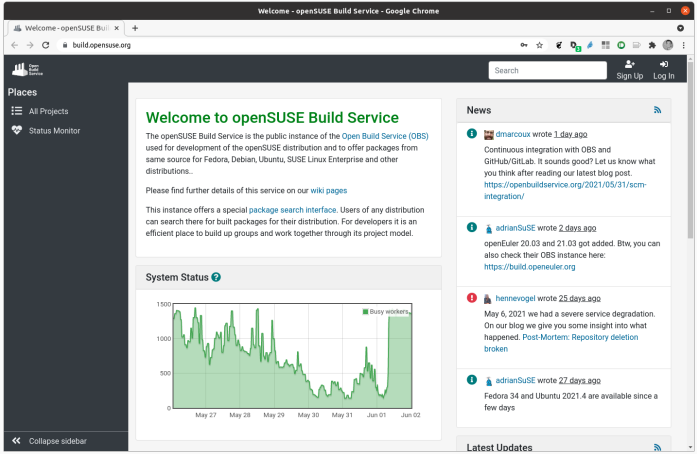


Features

- portable
- independent of CI host
- works GitHub, Azure, Travis
- likely works elsewhere
- simple, comprehensible
- “works for me”
- `eddelbuettel.github.io/r-ci/`

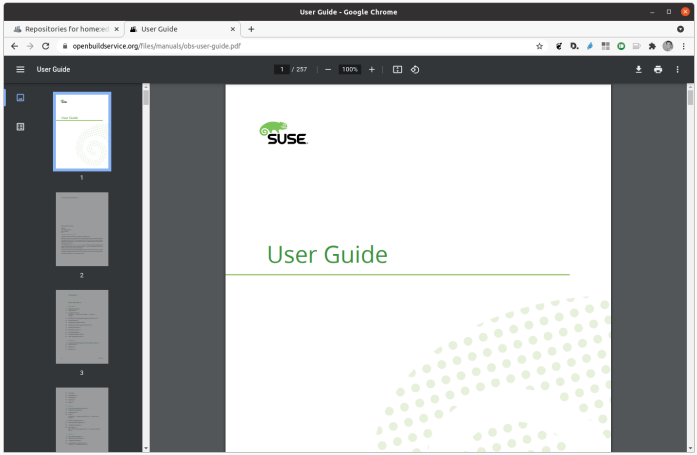
OpenSUSE Build Service

OpenSUSE Build Service



OBS

- free service, no restrictions
- fairly resourceful
- started exploring / testing only recently
- `build.opensuse.org`



Manual

- one of the detailed documentation piece
- also available is a wiki
- plus a mailing list
- `openbuildservice.org/files/manuals/obs-user-guide.pdf`

Key Points

- New(ish) to me too
- Using it currently to experiment with the 3rd or 4th attempt at converting all / most CRAN packages into .deb binaries
- Build service for *multiple distro* and OSs, even windoze
- Free as in beer and cpu cycles, can be combined with GitHub and GitLab
- Pretty well documented: 250+ page pdf, wiki, responsive (low volume) mailing list
- Build releases and/or dev snapshots for OpenSUSE, Fedora, Debian, Ubuntu, ...
- This may be of interest to 'Team Gretl'

Thanks!

Thank you!

slides	https://dirk.eddelbuettel.com/presentations/
web	https://dirk.eddelbuettel.com/
mail	dirk@eddelbuettel.com
github	@eddelbuettel
twitter	@eddelbuettel