

If You Build It Will They Come? Lessons From the Development and Use of a Containerized Teaching and Learning Platform.

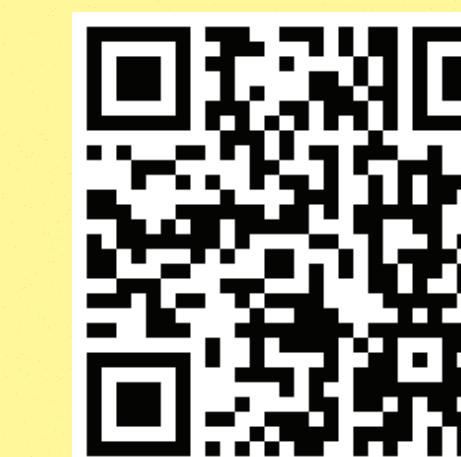
Karl Benedict (kbene@unm.edu) & Jonathan Wheeler (jwheel01@unm.edu) - College of University Libraries and Learning Sciences, University of New Mexico

IN5IF-0688

AGU100 ADVANCING EARTH AND SPACE SCIENCE

Abstract

In early 2018 the Research Data Services program at the University of New Mexico developed a teaching and learning platform that streamlined the installation and use of the tools that RDS's Coffee and Code workshop series focused on. The platform is based upon containerization of core analytic tools using Docker and the automated integration and updating of workshop content into the platform. Running an instance of the platform as a teaching and/or learning environment requires only the download of the GitHub repository¹ containing the Docker orchestration configuration files, and a copy of the Docker host and Docker Compose capabilities on the system. With these two requirements met, a simple one-line command will start the complete environment and download the content of all workshops included in the Coffee and Code series. After a year of use in multiple contexts - introductory and intermediate level workshops, diverse tool foci (command line, r, python, git, relational and non-relational databases), multiple instructors - the benefits of the platform have become more clear. Overall, while the platform has proven very useful as a hosted teaching and learning environment, there has been little uptake on the part of learners to take the additional step of running the learning environment on their own computers as originally intended. This presentation will outline the current status of the platform, the lessons learned from the pattern of platform use by learners, and the planned steps for moving forward with the continued development of the platform as a more effective teaching and learning environment.

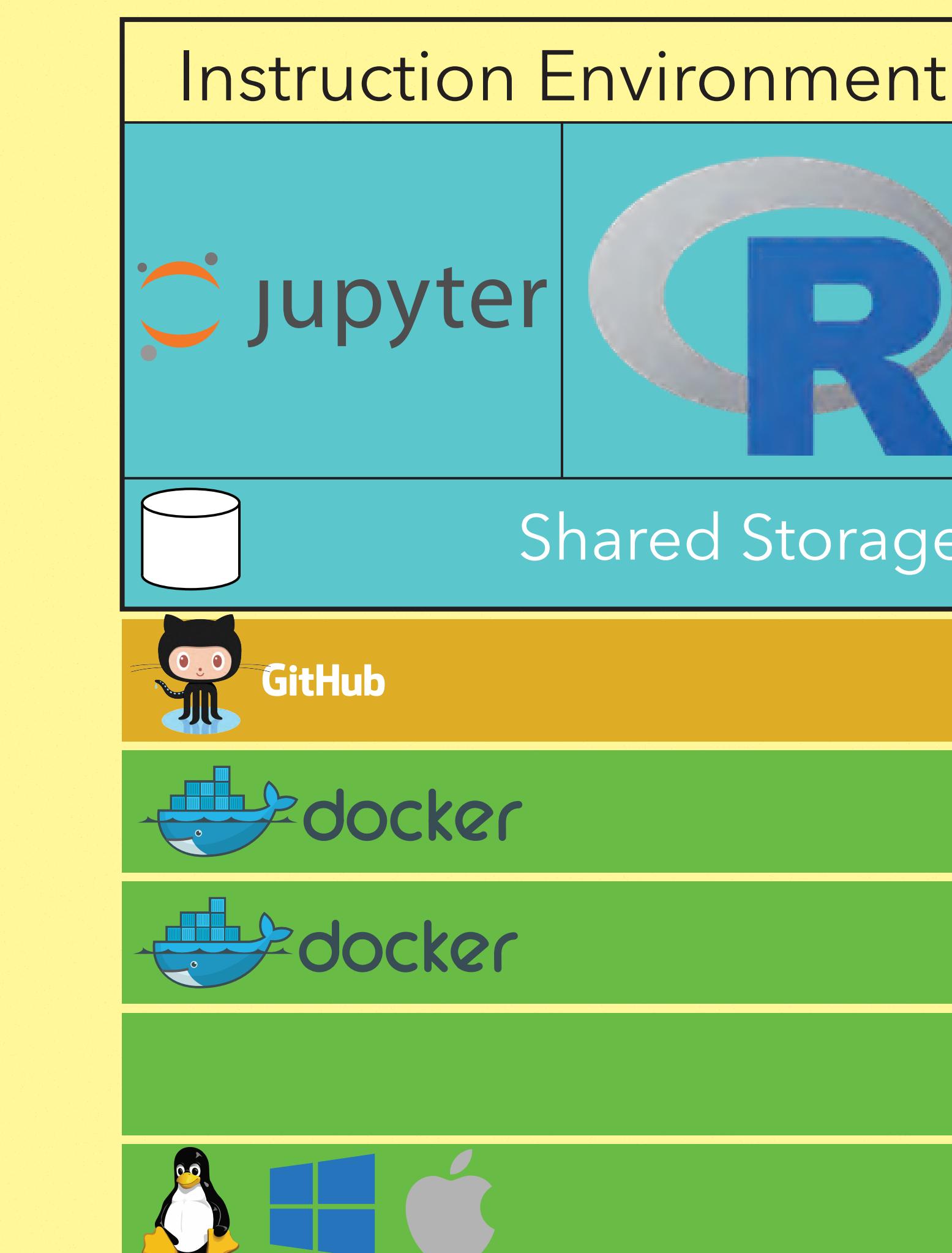


GitHub Repositories - github.com/unmrds

- cc-version-control - introduction to version control with Git
- cc-pandoc - document production
- cc-taskjuggler - project management using TaskJuggler
- cc-containers - introduction to Docker containers
- cc-command-line - introduction to the UNIX command line
- jupyter cc-jupyter - introduction to Jupyter Notebooks
- jupyter cc-python-data-apis - using data APIs in python
- jupyter cc-nlp - natural language processing in python
- jupyter cc-noSQL - introduction to noSQL databases
- jupyter cc-database-basics - introduction to database concepts
- jupyter cc-database-2 - additional database concepts
- jupyter cc-python - programming concepts using python
- jupyter cc-reproducibility-communication - Jupyter and R notebooks and markdown
- R cc-R-RStudio - introduction to RStudio
- R R-data-cleaning - introduction to data cleaning with R
- QGIS cc-digital-literacy - introduction to communication design principles and Adobe Spark
- QGIS cc-cartography_w_illustrator - cartography workflows with GIS and Adobe Illustrator

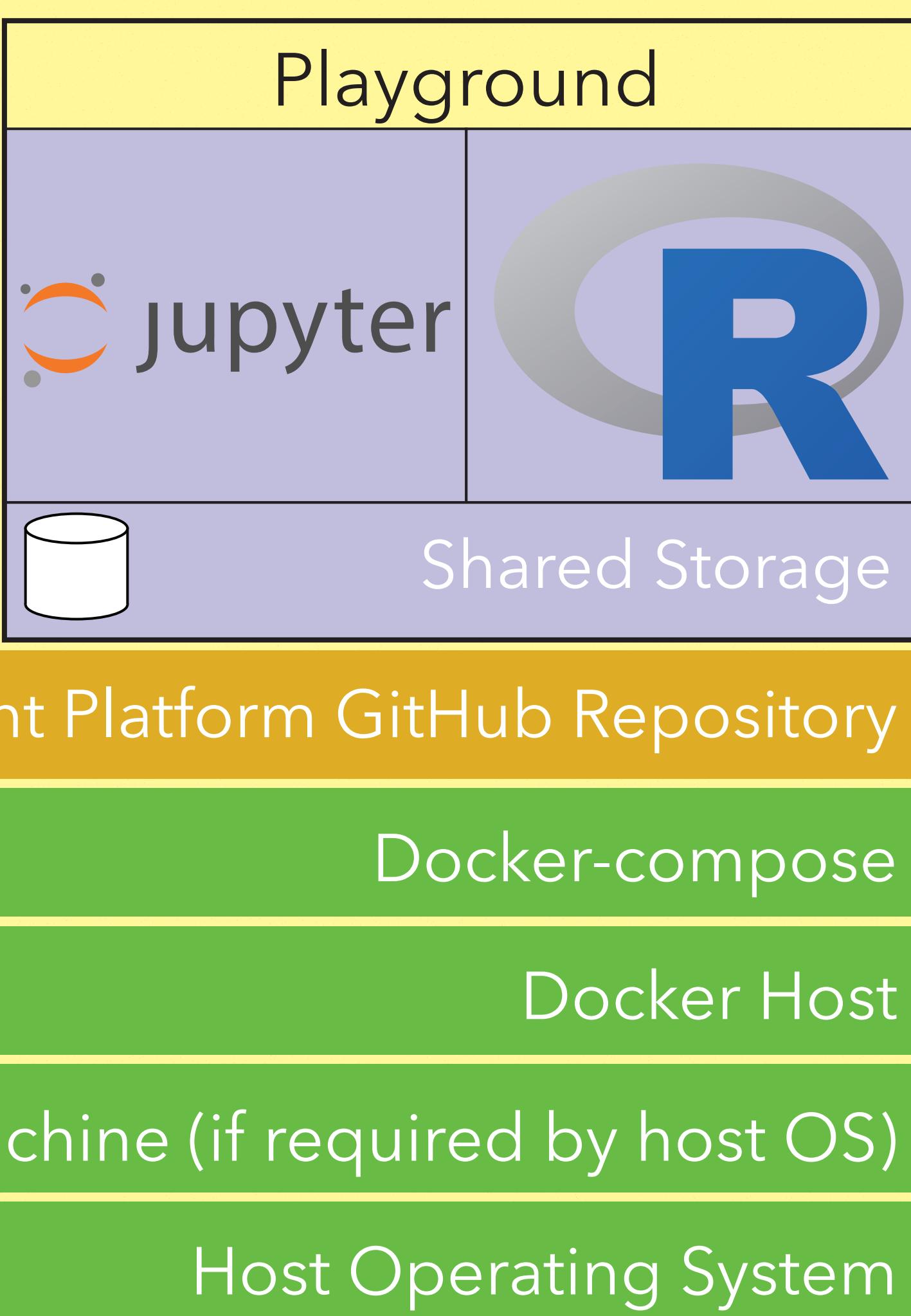
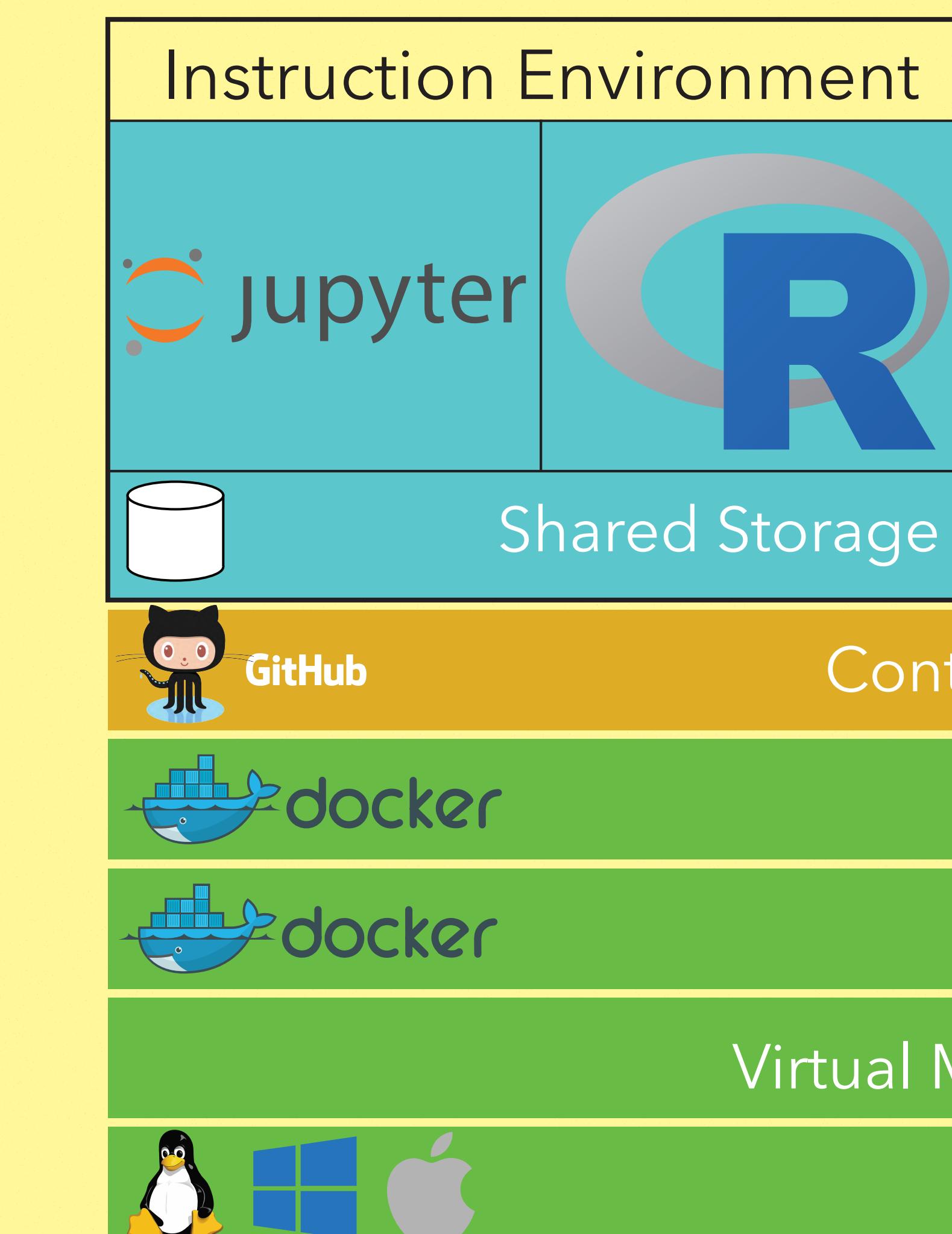
Emerging Training Material and Platform Practice

Python executed within hosted and local Jupyter Notebook environments
 Command Line executed in locally installed command line tools
 Presentation content written in Markdown in relevant environments
 R code and RMarkdown written and executed in hosted and locally run RStudio



Initial Release and Use

Command Line and Python executed within Jupyter Notebook environment
 Presentation content written in Markdown within Jupyter Notebooks
 R code and RMarkdown written and executed in hosted RStudio



Lessons (re)Learned

From the Learner's Perspective

- * The learner's primary focus is on the tool/method that they came to the workshop for - not the ability to install/run the full stack of workshop applications
- * Web-based tools (i.e. hosted Jupyter Notebooks, RStudio, terminal environment) are a useful backup, but not a replacement for locally running tools on the learner's computer - unless the hosted tools are persistent and powerful enough for "real" use
- * GitHub managed workshop repositories enable easy download of workshop materials - as Zip archives or through Git Pull command

From the Instructor's Perspective

- * The platform remains a useful system for rapid and iterative development and deployment of workshop content
- * GitHub for workshop content version control provides several advantages:
 - * Static preview for some Jupyter Notebooks within GitHub
 - * Ability to deploy repository-based Jupyter Notebook files or repositories into hosted platforms like Binder² and PanGeo³.
- * A teaching platform for RStudio remains valuable, but should be based on a multi-user hosted service
- * More targeted containers (i.e. streamlined for specific applications) may be more efficient moving forward
- * Markdown instructional materials may be better developed and managed outside of Jupyter Notebooks



Student and Instructor Installed Applications



Notes/References

- ¹ Platform GitHub Repository: <https://github.com/unmrds/cc-content-platform>
- ² Jupyter et al., "Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale." Proceedings of the 17th Python in Science Conference. 2018. 10.25080/Majora-4af1f417-OII : <https://binderhub.readthedocs.io/en/latest/index.html>
- ³ PanGeo Binder: <https://binder.pangeo.io>