How are folks adding computational replay in institutional repositories?

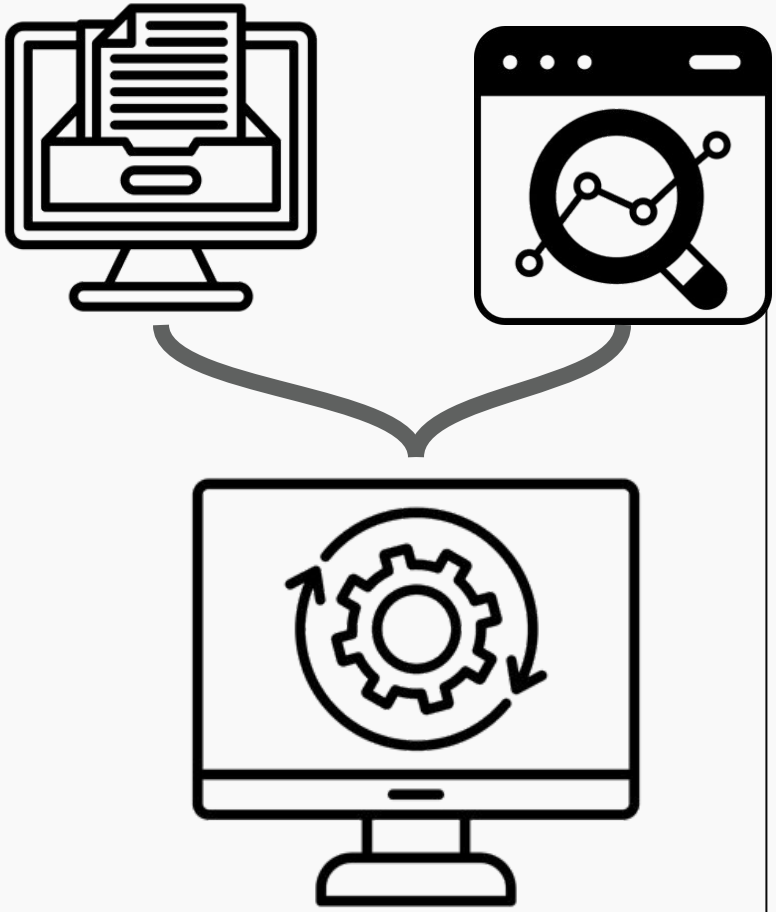
Vicky Rampin (she | they)

Librarian for Research Data Management & Reproducibility
New York University

vicky.rampin@nyu.edu

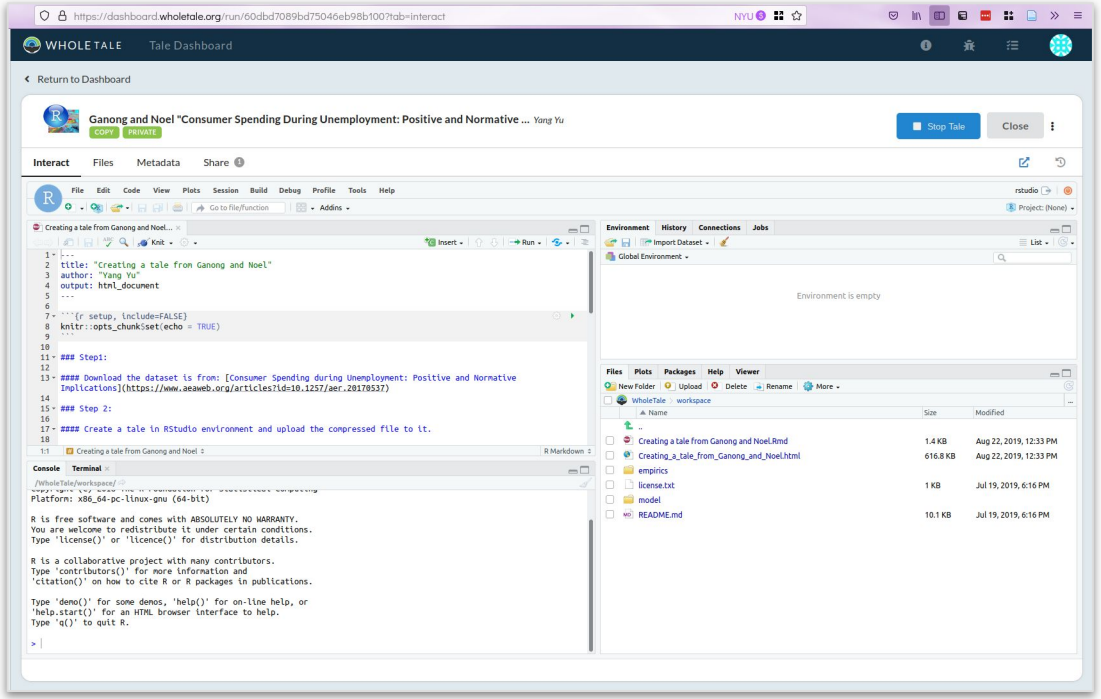
Meeting of two techs

- Computational research has several components, including software, it's dependencies, data, & docs
- Researchers tend to put these into various **repositories**, for open access & preservation
- Integrating **Web-based reproducibility tools** into repositories lowers the barrier to access and reuse of preserved computational research



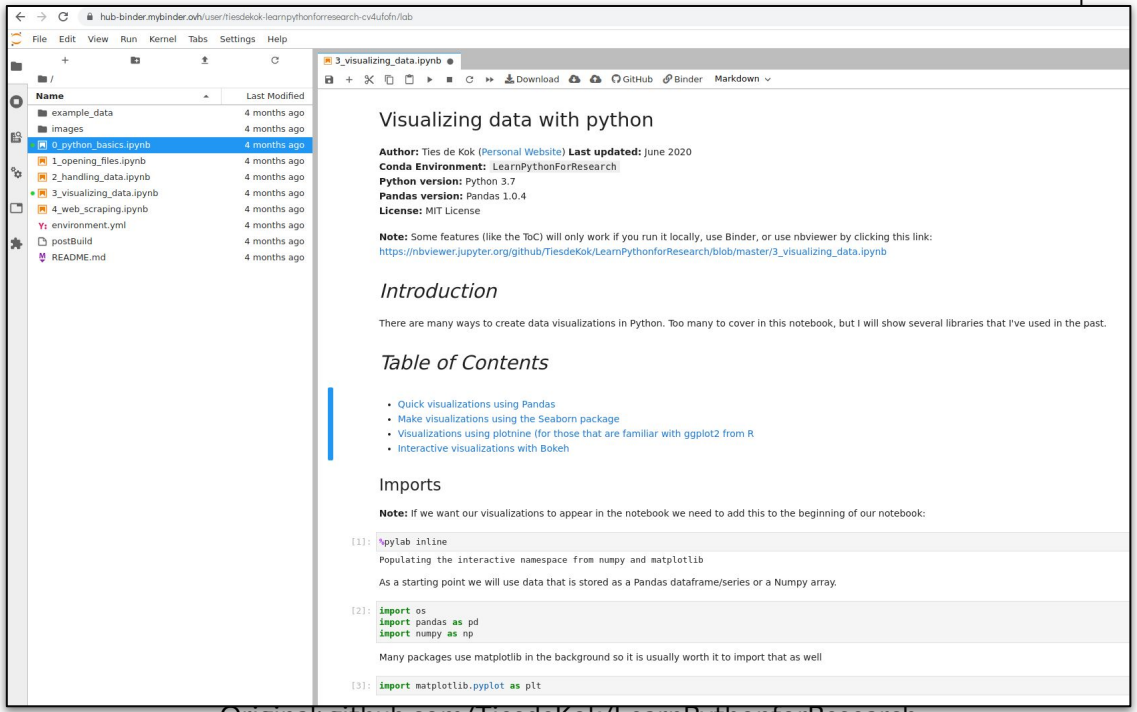
Web-based Integrated Development Environments (IDE)

- Built to optimize developing and running code
- Using IDEs with containers, Web IDEs provide researchers a place to do work and then share it reproducibly with others
- The catch — you have to work on the web in their platform the whole time



Web-based replay

- In-browser tool where you give a link to materials that are hosted somewhere else & it will build the computational environment in your own sandbox with all the files
- Use containers in the back-end for building the same computational environment



Original: github.com/TiesdeKok/LearnPythonforResearch
Replayed: hub-binder.mybinder.ovh/user/tiesdekok-learnpythonforresearch-cv4ufofn/lab

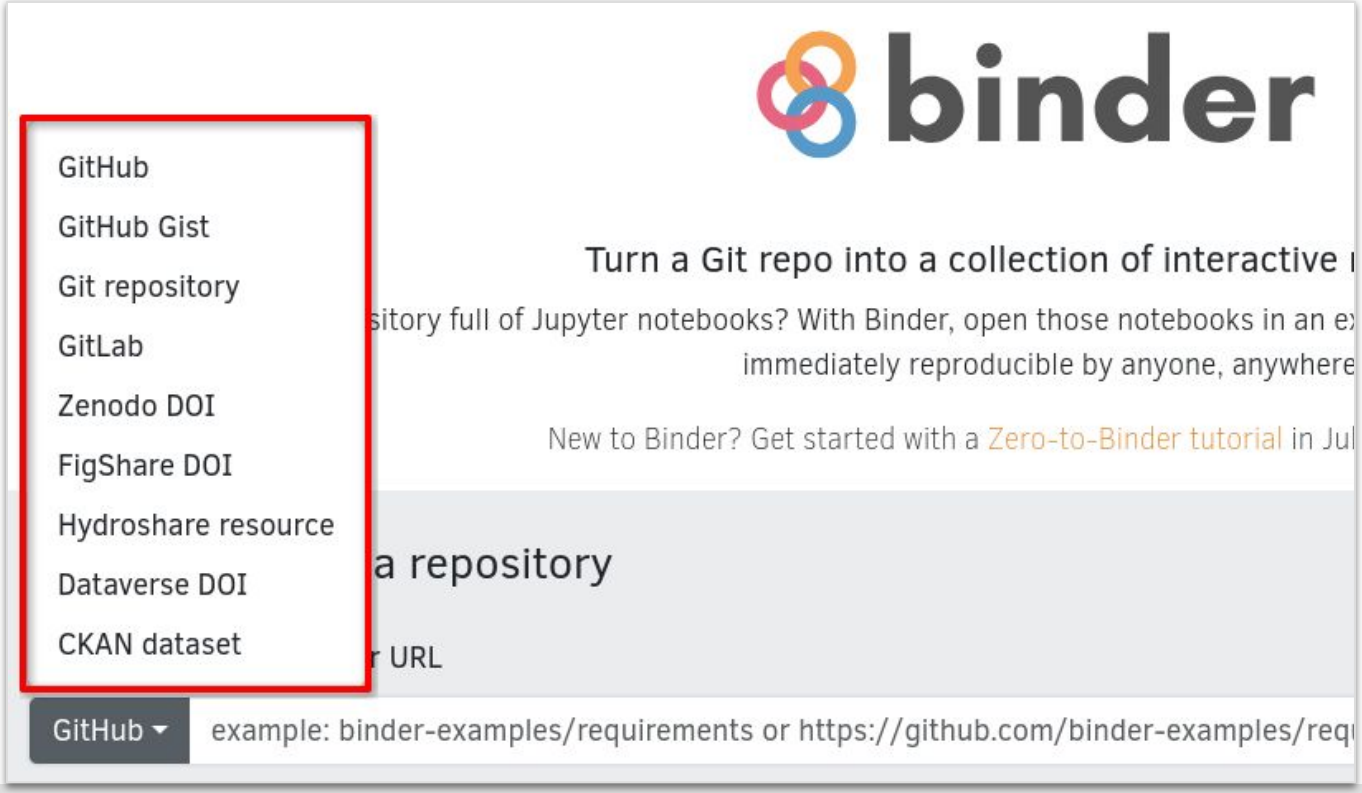
Selecting tools to examine for integration

Criteria	BinderHub	JupyterHub	WholeTale	EaaSI
Open source license	✔	✔	✔	✔
Established governance practices	✔	✔	✔	✔
Import from repositories	✔	✔	✔	•
Export to repositories	✖	✔	✔	•
Export files to local computer	✔	✔	✔	✔
Institutional authentication	✖	✔	✔	✔

✔ - implemented; • - potentially but not yet ; ✖ - not applicable

BinderHub: public sandbox, Web-based replay

- Under the hood: JupyterHub deployed in “API-only” mode
- Made for one-way interaction: replay only in sandbox environment



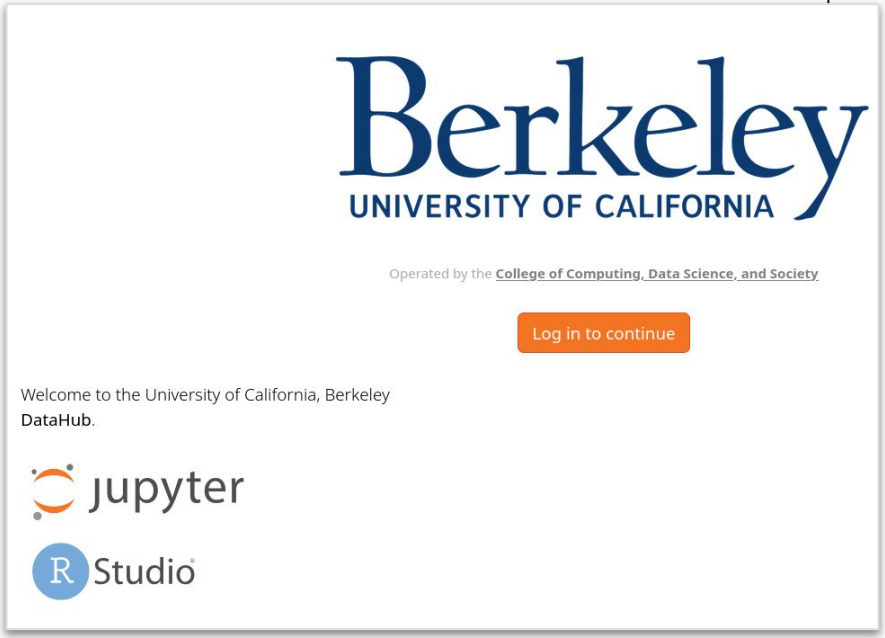
Harvard Dataverse-to-BinderHub from record pages

Dataverse changed their backend architecture AND front-end display on records!

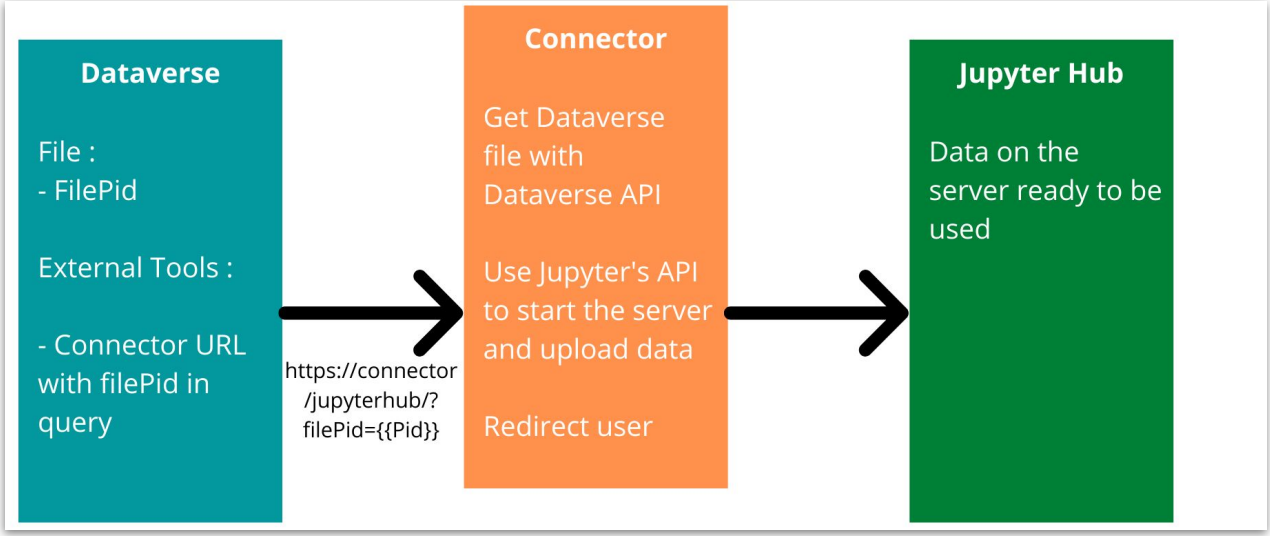
- Specific handling of Dockerfiles,
- replication-package metadata,
- New APIs,
- UI components for computational access

JupyterHub: institutional, Web-based IDE

- (typically) multi-user deployment of the Jupyter environment that allows administrators to create separate Jupyter environments for specific groups of users as needed, i.e., for-credit classes
- Environments can run for any type of language for which there is an accepted kernel
- Typically sits behind institutional authentication & is run by institutional IT



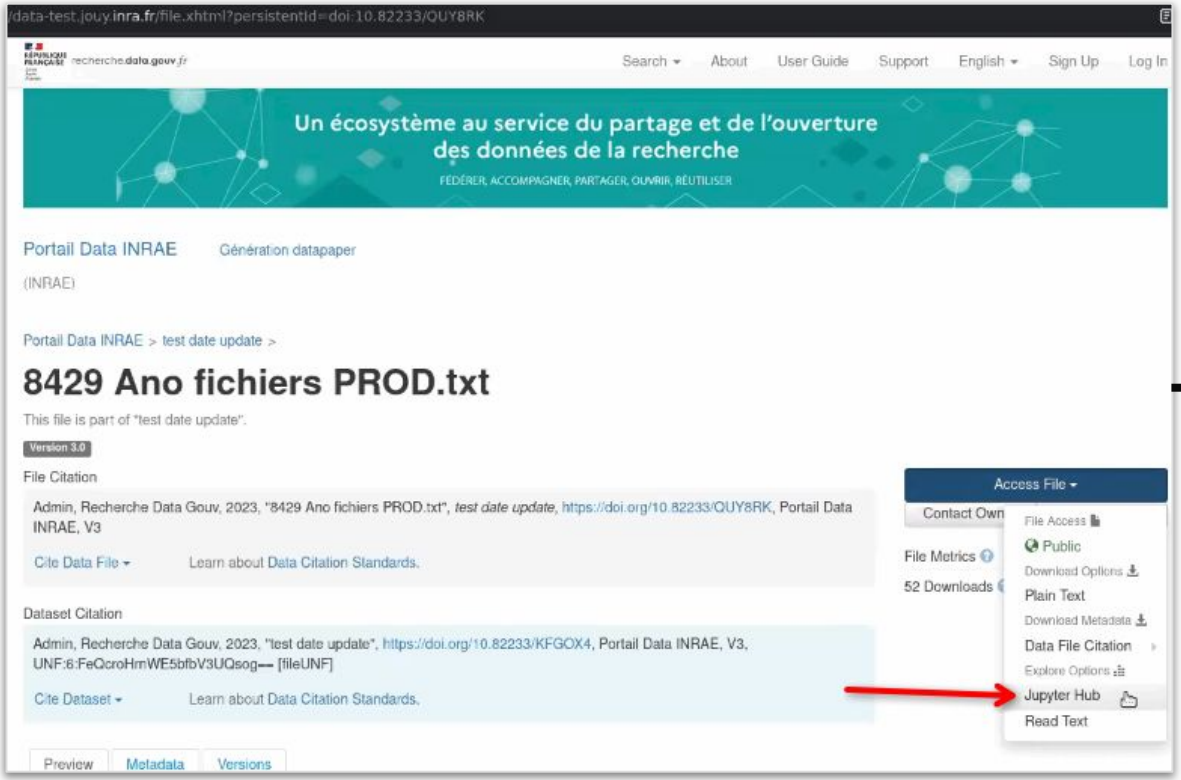
Dataverse-to-JupyterHub Data Transfer



- Created by EOSC-Pillar project
- From Dataverse to JupyterHub: based on Dataverse's External Tool feature, uses the APIs of both Dataverse and JupyterHub to move files & authenticate
- From JupyterHub to Dataverse: pre-loaded scripts enable transfer back

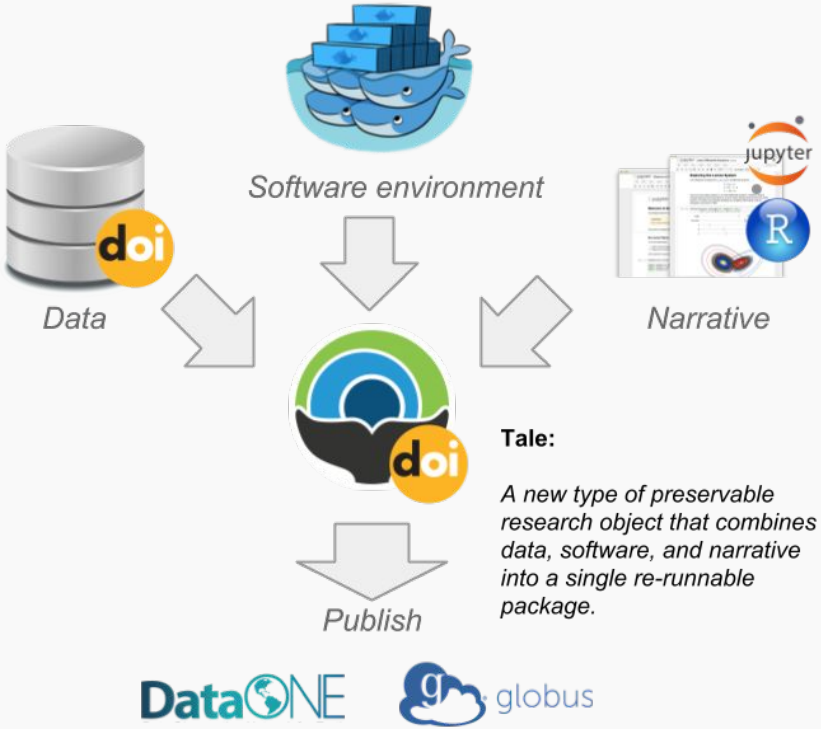
see presentation

European Open Science Cloud (EOSC)

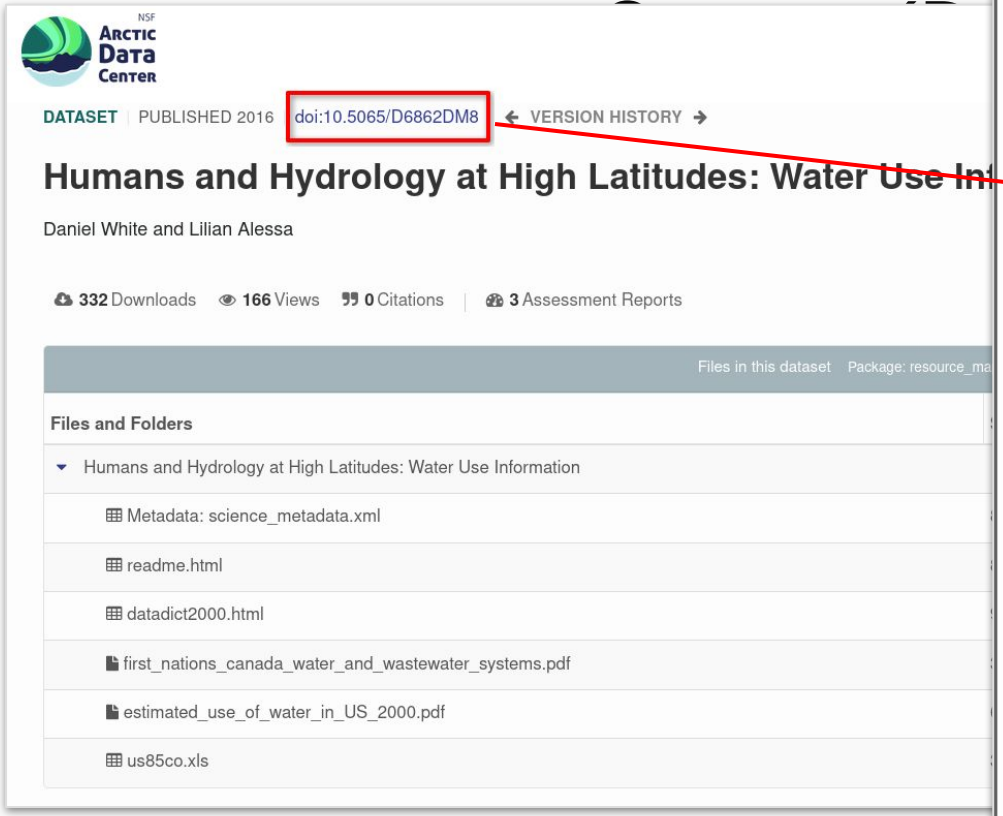


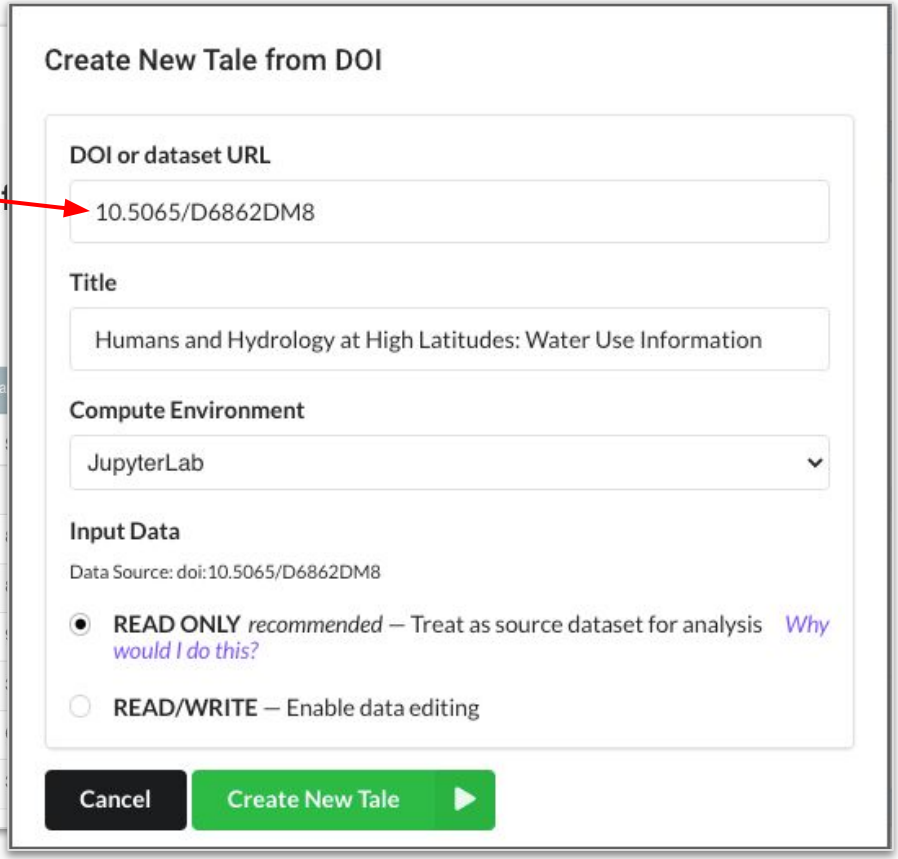
WholeTale: lots of software options, IDE & replay

- Web-based IDE **or** replay tool -- patrons can work within WholeTale entirely or just use it to try to reproduce other work
- Provides support for open source and proprietary software in the platform
- Two-way connections to: Zenodo, Dataverse, OpenICPSR, DataOne
  - “[...] WholeTale is not a repository and relies on integration with existing research data archives for persistent identifiers & long-term preservation”



Replaying Materials from the Arctic Data





EaaSI: essential underlying infrastructure

- EaaSI provides essential underlying preservation infrastructure we need long-term for access to remain viable
- Users configure the computational environment with specific software, import files, and interact with a virtual desktop
- Not currently integrated with repositories, but is a stated goal on the roadmap

"At the most immediate level, EaaSI needs to be able to **integrate with digital preservation/digital repository systems** so it can retrieve disk images and content from them for management and access.

It also needs to **integrate with access and discovery platforms to enable environments provided by EaaSI to be made available directly** to users via these existing platforms"

Summary

- Access is an important part of preservation and access to computational research requires computational tools to interact with the materials
- Integrating existing reproducibility tools into repositories encourages direct reuse of preserved materials
    - Helps patrons by pre-configuring computational dependencies
    - Familiar interfaces to many already (so fewer UI barriers)
- These 4 technologies (BinderHub, JupyterHub, WholeTale, EaaSI) have complementary scopes & approaches that would make them attractive to integrate into repositories

Thank you!
Let’s discuss!

Vicky Rampin (she | they)

Librarian for Research Data Management & Reproducibility
New York University

vicky.rampin@nyu.edu