
How are folks adding computational replay in institutional repositories?

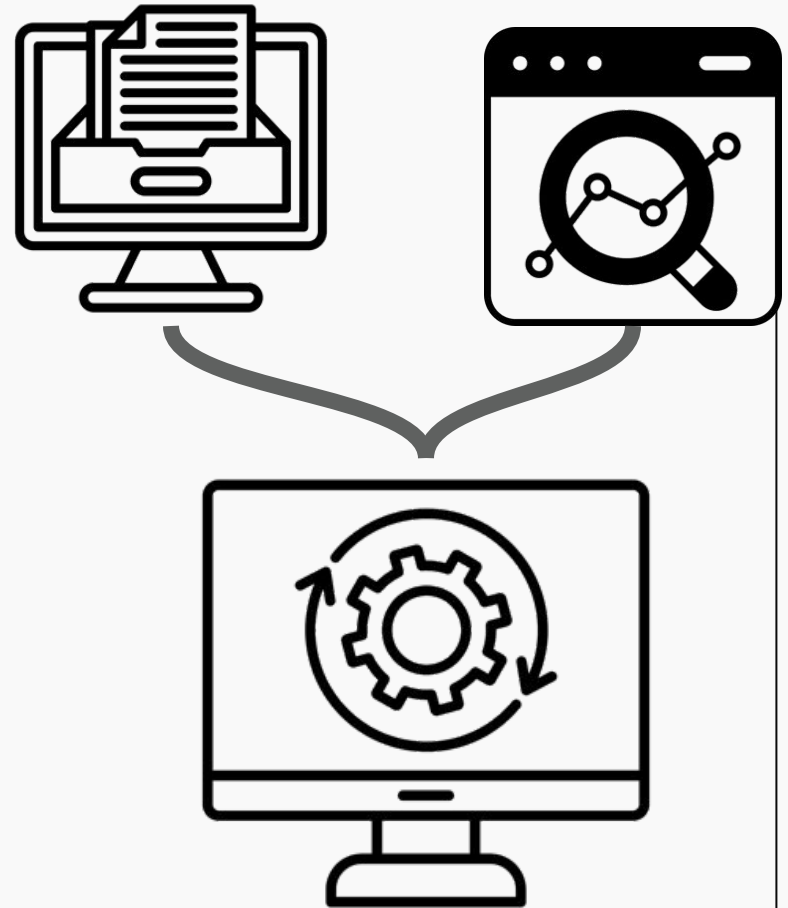
Vicky Rampin (she | they)

Librarian for Research Data Management & Reproducibility
New York University

vicky.rampin@nyu.edu

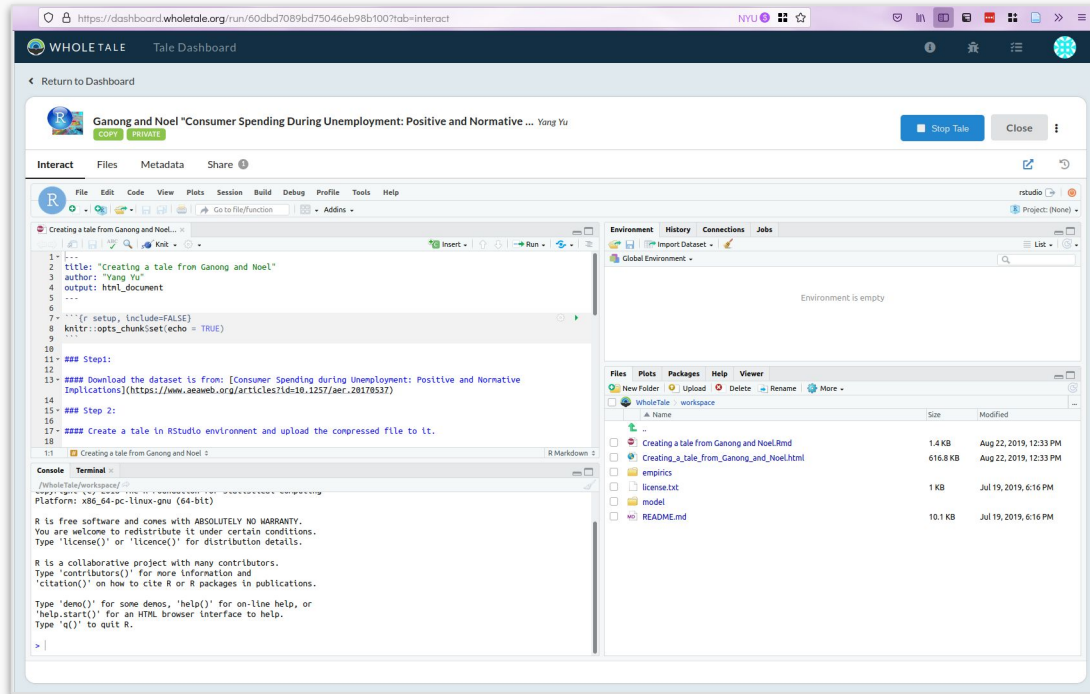
Meeting of two techs

- 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



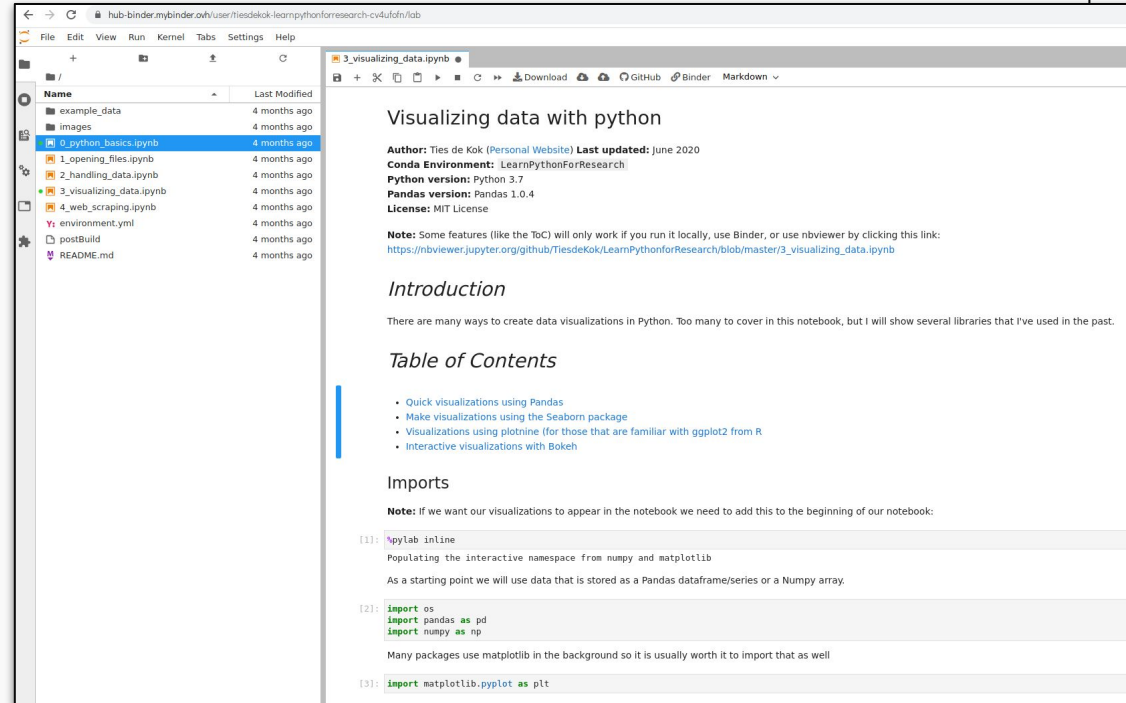
Web-based Integrated Development Environments (IDE)

- 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



The screenshot shows a web browser displaying a Jupyter notebook interface. The address bar shows the URL: `hub-binder.mybinder.ovh/user/tiesdekok-learnpythonforresearch-cv4ufofn/lab`. The interface is split into two main sections:

- File Browser (Left):** A sidebar showing a directory structure with files and folders. The file `3_visualizing_data.ipynb` is selected and highlighted in blue. Other files include `example_data`, `images`, `0_python_basics.ipynb`, `1_opening_files.ipynb`, `2_handling_data.ipynb`, `4_web_scraping.ipynb`, `environment.yml`, `postBuild`, and `README.md`. Each file has a timestamp of "4 months ago".
- Notebook Content (Right):** The main area displays the content of the selected notebook, titled "Visualizing data with python". It includes:
 - Metadata:** Author: Ties de Kok (Personal Website), Last updated: June 2020, Conda Environment: LearnPythonForResearch, Python version: Python 3.7, Pandas version: Pandas 1.0.4, License: MIT License.
 - Note:** A note stating that some features only work if run locally, via Binder, or nbviewer.
 - Introduction:** A paragraph explaining that there are many ways to create data visualizations in Python.
 - Table of Contents:** A list of topics: Quick visualizations using Pandas, Make visualizations using the Seaborn package, Visualizations using plotnine (for those familiar with ggplot2), and Interactive visualizations with Boken.
 - Imports:** A section explaining that imports are needed for visualizations to appear and listing the imports: `!pip install inline`, `import pandas as pd`, `import numpy as np`, and `import matplotlib.pyplot as plt`.

Original: github.com/TiesdeKok/LearnPythonforResearch

Replayed:

hub-binder.mybinder.ovh/user/tiesdekok-learnpythonforresearch-cv4ufofn/lab

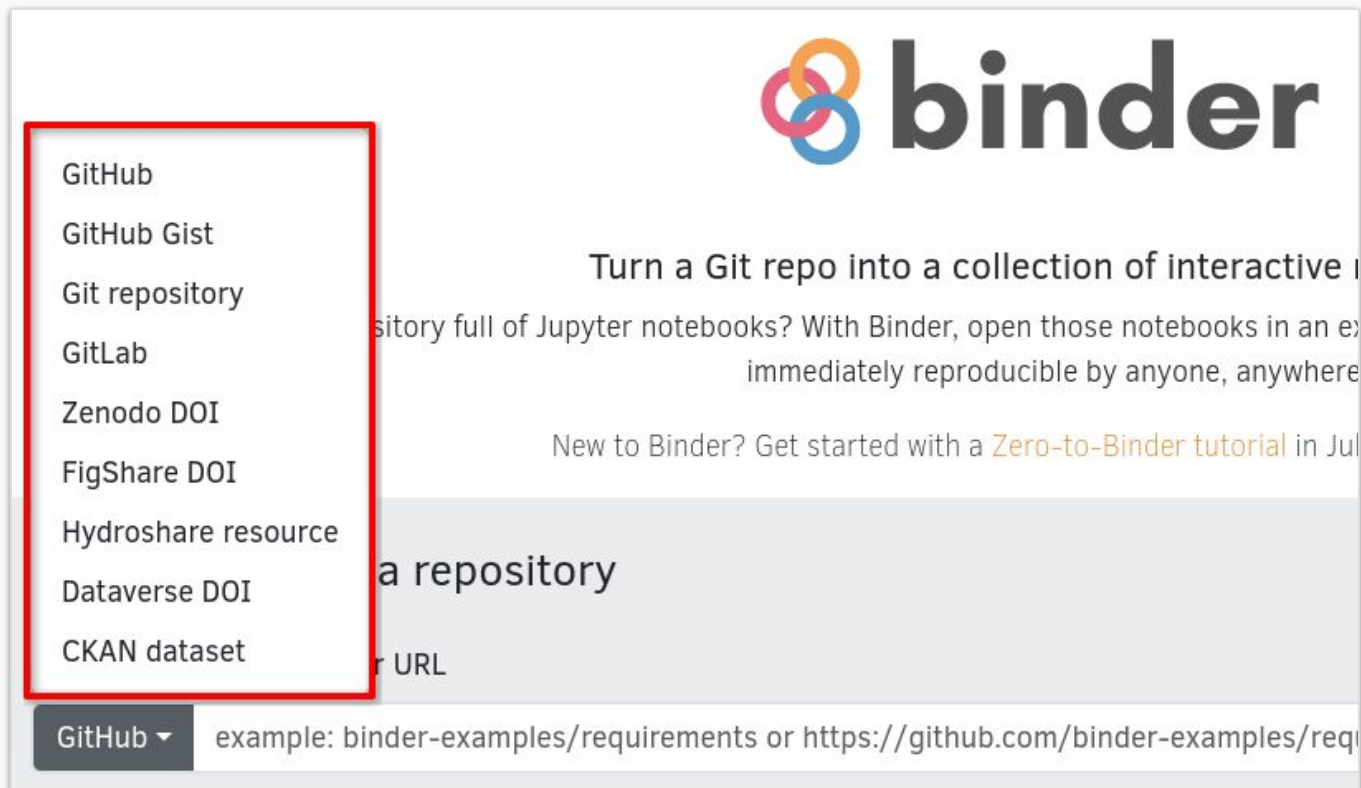
Selecting tools to examine for integration

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



The screenshot shows the Binder website interface. At the top right is the Binder logo, which consists of three interlocking circles in orange, pink, and blue, followed by the word "binder" in a bold, dark grey sans-serif font. Below the logo, the text reads: "Turn a Git repo into a collection of interactive Jupyter notebooks? With Binder, open those notebooks in an environment that is immediately reproducible by anyone, anywhere." Further down, it says: "New to Binder? Get started with a [Zero-to-Binder tutorial](#) in July." A search bar is visible with the placeholder text "a repository" and "r URL". A dropdown menu is open, listing various source providers: GitHub, GitHub Gist, Git repository, GitLab, Zenodo DOI, FigShare DOI, Hydroshare resource, Dataverse DOI, and CKAN dataset. The "GitHub" option is selected and highlighted with a dark grey background and a white downward arrow. Below the dropdown, the text "example: binder-examples/requirements or https://github.com/binder-examples/req" is partially visible.

binder

Turn a Git repo into a collection of interactive Jupyter notebooks? With Binder, open those notebooks in an environment that is immediately reproducible by anyone, anywhere.

New to Binder? Get started with a [Zero-to-Binder tutorial](#) in July.

a repository

r URL

GitHub ▾ example: binder-examples/requirements or https://github.com/binder-examples/req

Harvard Dataverse-to-BinderHub from record pages

Harvard Dataverse > Politics Population Space and Representation Lab / Grupo de Estudos sobre Política População Espaço e Representação >

Replication Data for: GeoDesp: A Database that Identifies where Candidates Spend their Campaign Expenses

Version 1.0



Guarnieri, Fernando, 2025, "Replication Data for: GeoDesp: A Database that Identifies where Candidates Spend their Campaign Expenses", <https://doi.org/10.7910/DVN/T32S14>, Harvard Dataverse, V1

Cite Dataset ▾

Learn about [Data Citation Standards](#).

Access Dataset ▾

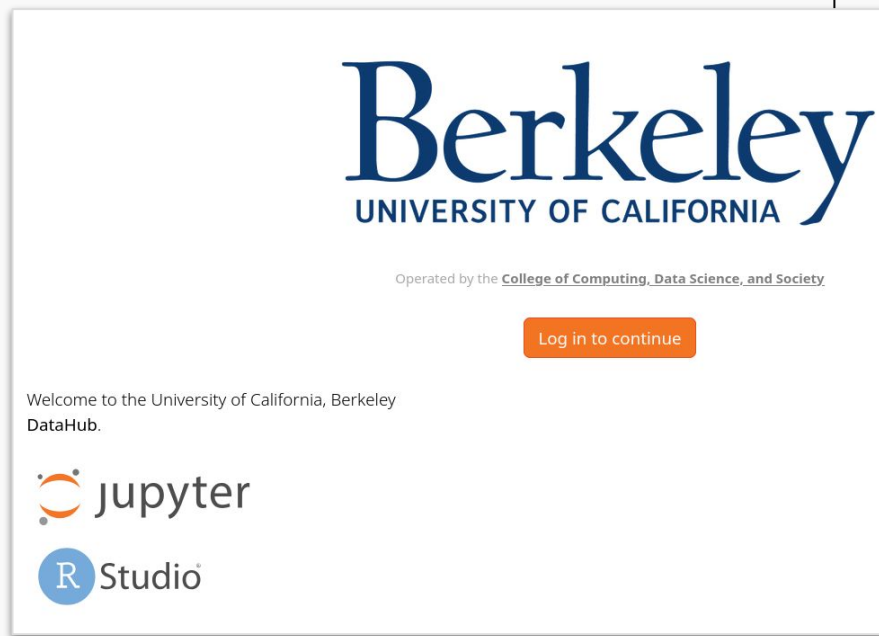
- Download Options ↓
- Download ZIP (35.7 MB)
- Explore Options ⚙
- Binder

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

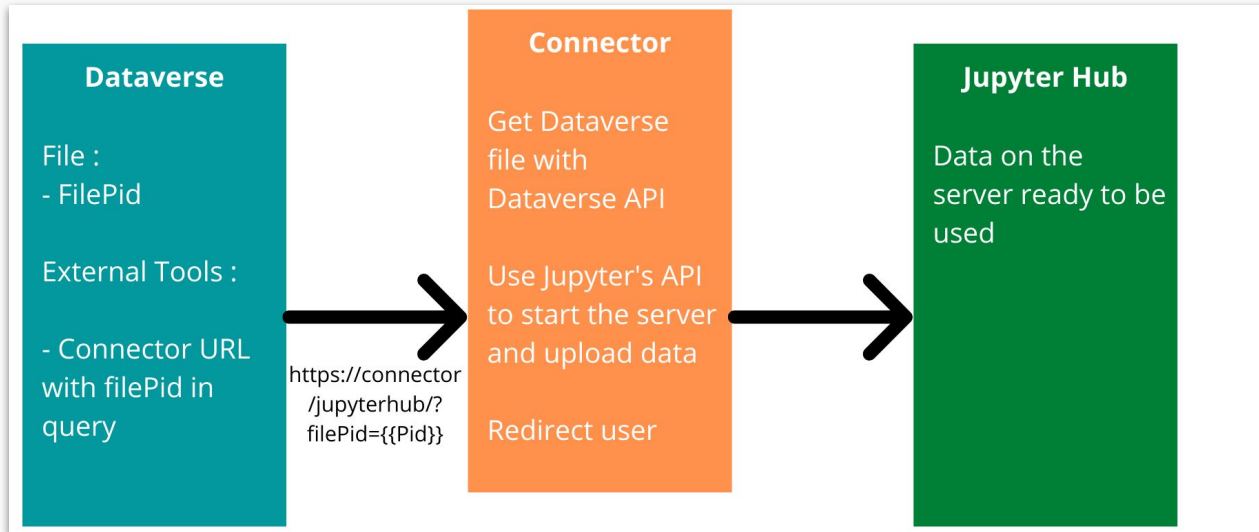
- Specific handling of Dockerfiles,
- replication-package metadata,
- New APIs,
- UI components for computational access

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

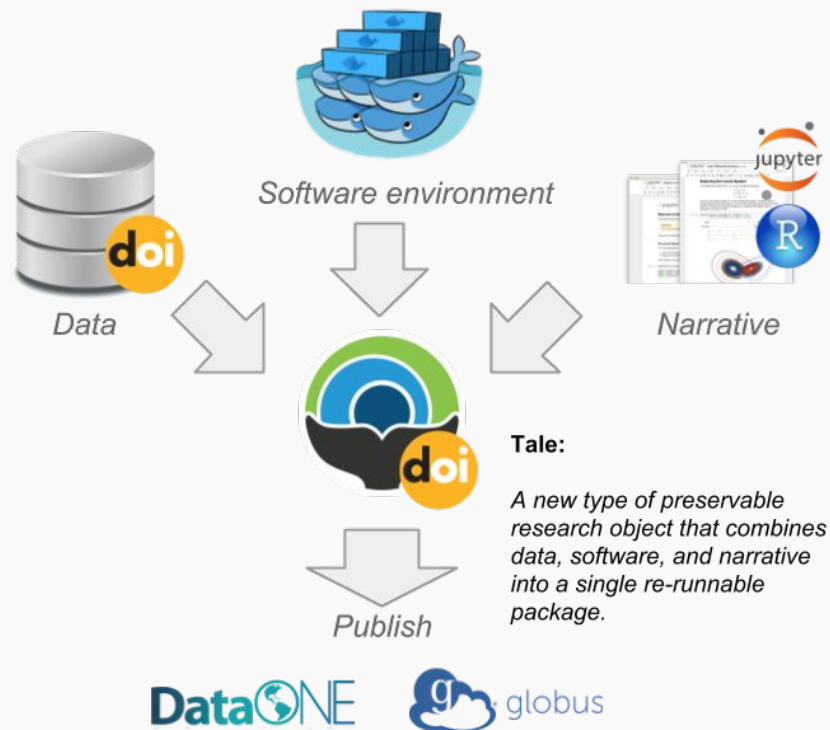
European Open Science Cloud (EOSC)

The screenshot shows the INRAE data portal interface. At the top, there is a navigation bar with 'Search', 'About', 'User Guide', 'Support', 'English', 'Sign Up', and 'Log In'. Below this is a teal banner with the text 'Un écosystème au service du partage et de l'ouverture des données de la recherche' and 'FÉDÉRER, ACCOMPAGNER, PARTAGER, OUVRIR, RÉUTILISER'. The main content area displays 'Portail Data INRAE' and 'Génération datapaper'. The dataset title is '8429 Ano fichiers PROD.txt'. Below the title, there is a 'File Citation' section and a 'Dataset Citation' section. A red arrow points to the 'Jupyter Hub' option in the 'Access File' dropdown menu.

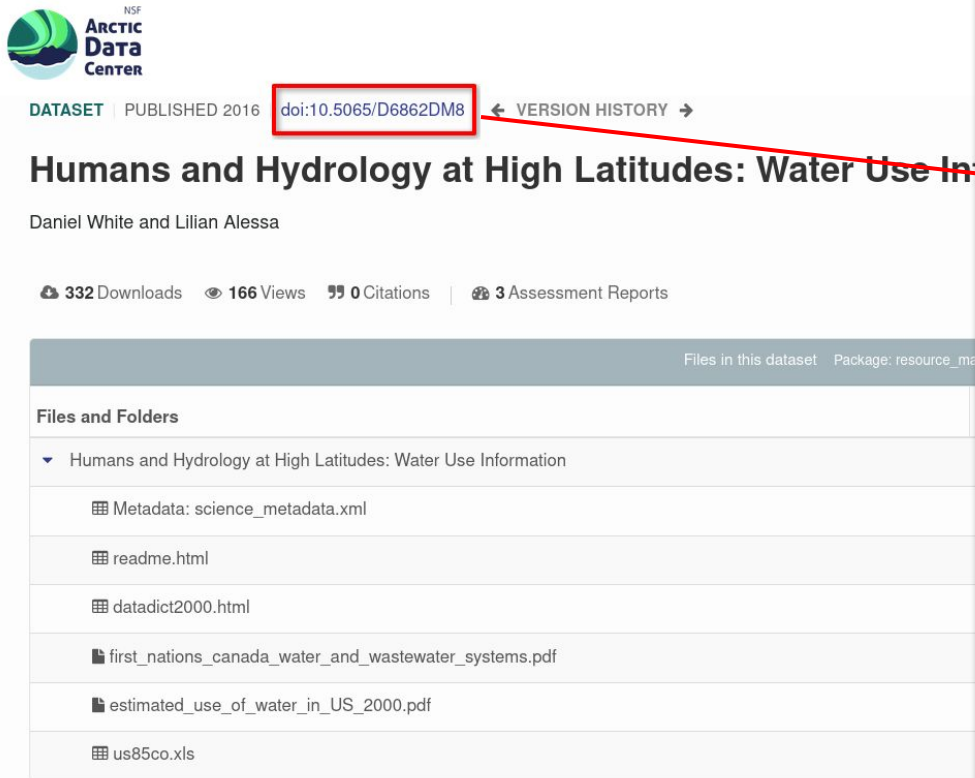
The screenshot shows the EOSC-Pillar Jupyter Hub connector interface. At the top, it says 'EOSC-Pillar Jupyter Hub connector'. Below this is the Jupyter Hub logo. The interface displays the following information: 'filePid : doi:10.82233/QUY8RK', 'datasetPid : doi:10.70112/V26AIH', 'RAM : 2000 M' with a slider, 'Cores : 1 2 3 4' with a slider, and a 'Save' button.

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



NSF
Arctic Data Center

DATASET PUBLISHED 2016 doi:10.5065/D6862DM8 ← VERSION HISTORY →

Humans and Hydrology at High Latitudes: Water Use Information

Daniel White and Lilian Alessa

332 Downloads | 166 Views | 0 Citations | 3 Assessment Reports

Files in this dataset Package: resource_materials

Files and Folders

- Humans and Hydrology at High Latitudes: Water Use Information
 - Metadata: science_metadata.xml
 - readme.html
 - datadict2000.html
 - first_nations_canada_water_and_wastewater_systems.pdf
 - estimated_use_of_water_in_US_2000.pdf
 - us85co.xls

Create New Tale from DOI

DOI or dataset URL

10.5065/D6862DM8

Title

Humans and Hydrology at High Latitudes: Water Use Information

Compute Environment

JupyterLab

Input Data

Data Source: doi:10.5065/D6862DM8

- READ ONLY** *recommended* — Treat as source dataset for analysis [Why would I do this?](#)
- READ/WRITE** — Enable data editing

Cancel

Create New Tale

EaaSI: essential underlying infrastructure

- EaaS 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, EaaS 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 EaaS 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, EaaS) 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
