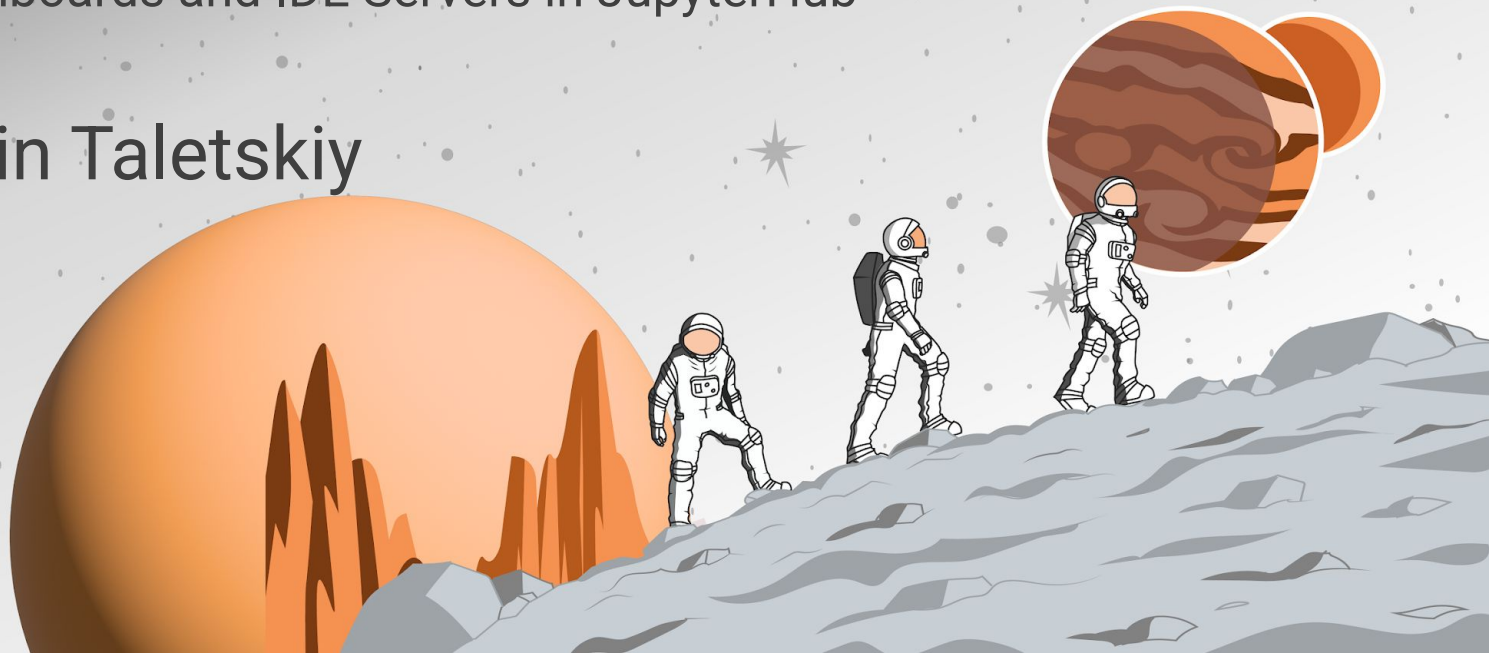
jupytercon

JupyterHub Satellites

Running Dashboards and IDE Servers in JupyterHub

Konstantin Taletskiy



#JupyterCon

About Me

ILLINOIS TECH

physics, NVIDIA CUDA, CUDA, Python and Jupyter user

AXLE

jupyterhub Data Science platform for NIH, Jupyter contributor

ANACONDA

open-source,



#JupyterCon

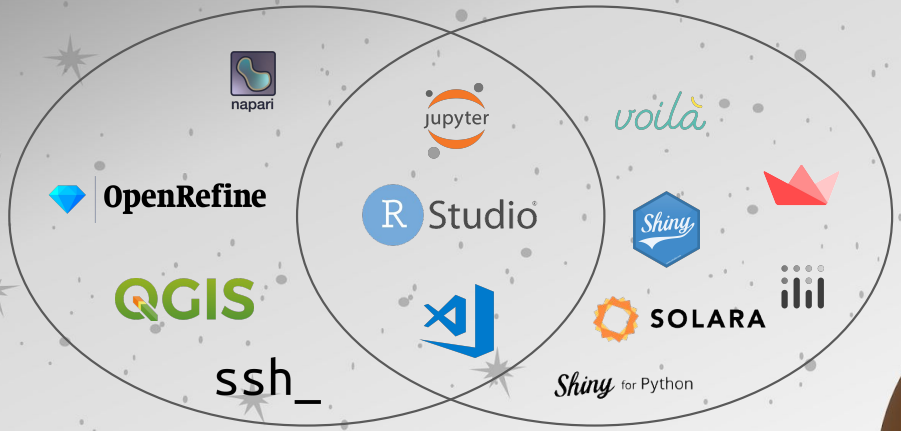
Related talk



Not Just for Notebooks: JupyterHub in 2025 - Yuvi P, 2i2c

Wednesday November 5, 2025 12:10PM

Pacific Room

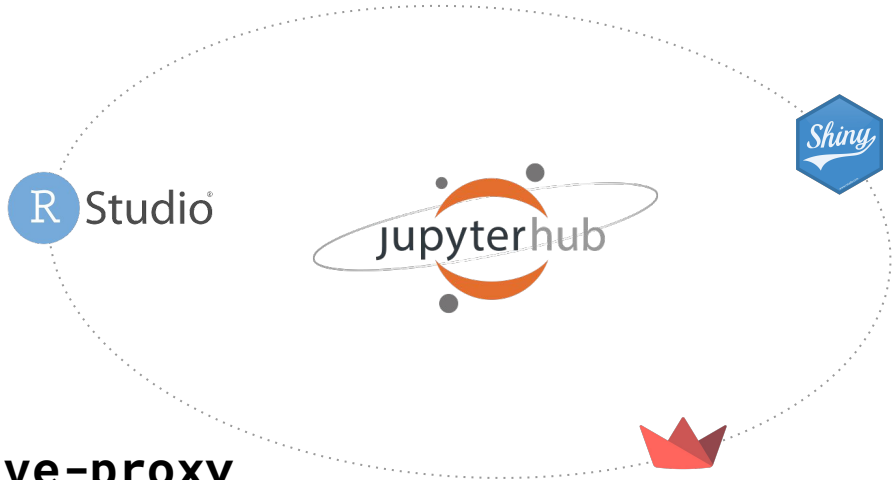




#JupyterCon

Why Satellites?

1. Users needed more than notebooks → RStudio, Shiny, Streamlit, IDEs
2. We wanted one place for auth, spawning, quotas, data access
3. Many apps were low-usage → don't keep them running all the time



→ **`jhsingle-native-proxy`**

Case Study: RStudio for Training

jupytercon

Context

- **RStudio** was a top request from NIH **training** programs

Solution

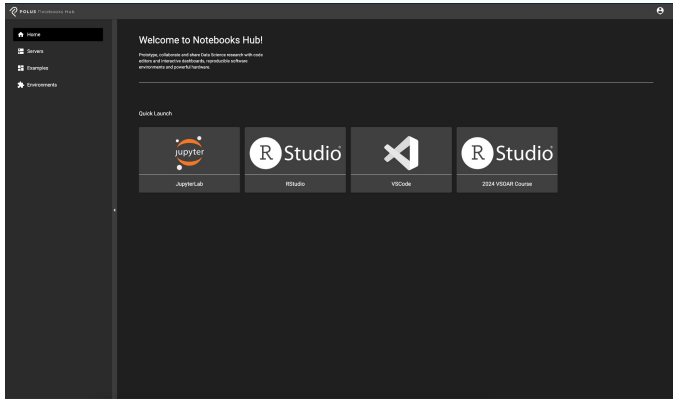
- Hosted RStudio through **JupyterHub** for NIH coding camps

Impact

- **Same R environment** for every student
- No local installs
- Launcher was obvious to first-time R users

RStudio®

"RStudio launcher was specific to our needs and easy to find"



Dashboards: Exploring Data Without Coding

jupytercon



SOLARA

plotly

voilà

Shiny

Shiny for Python

Streamlit

Case Study: Migrating R Shiny Apps

jupytercon

Context

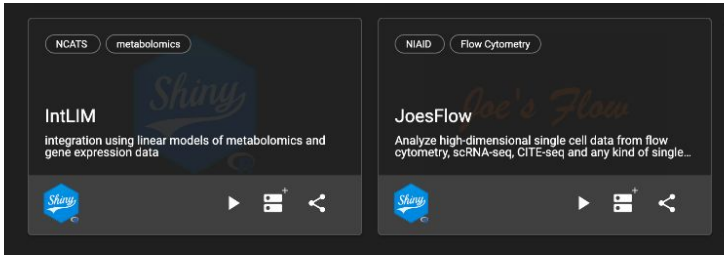
- Valuable Shiny apps (SEQUIN, Joe's Flow, IntLIM)
- …but usage was **sporadic** (few times a month)
- Still needed dedicated servers + support staff

Solution

- Run Shiny apps **on demand** through **JupyterHub**

Impact

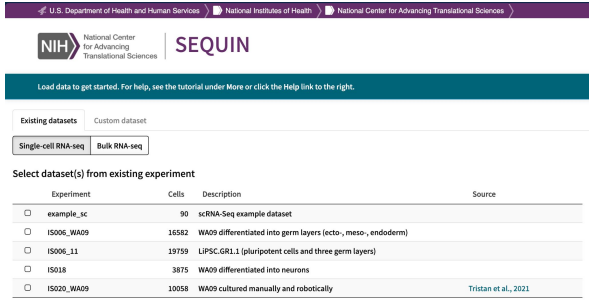
- Infra cost ↓ (no always-on app servers)
- Same auth / quotas as notebooks
- Rely on the existing **JupyterHub support team** → fewer infra to maintain



Case Study: Migrating R Shiny Apps

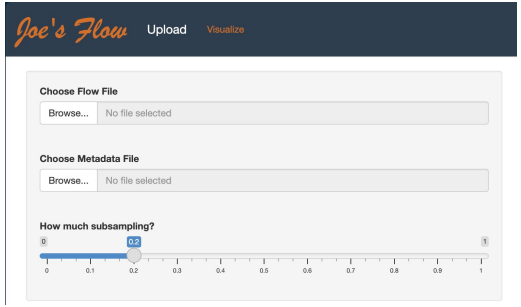
jupytercon

SEQUIN



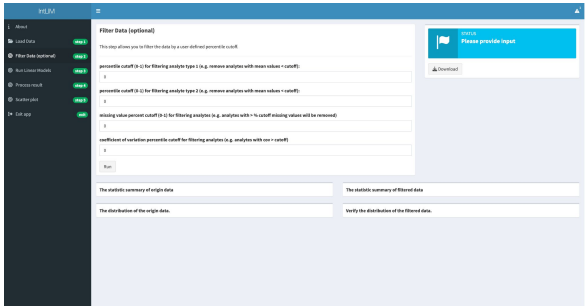
Analyzing and visualizing RNA-seq data

Joe's Flow



Multiomic data explorer

IntLIM



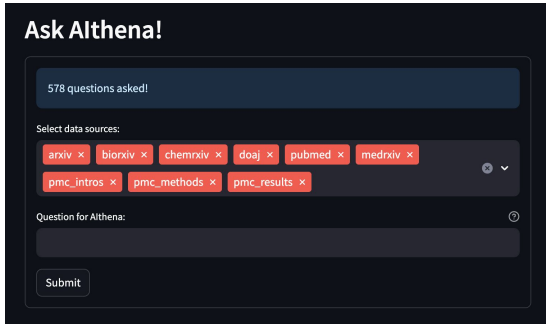
Metabolomics and gene expression data

Case Study: AIThena Chatbot Success

jupytercon

Context

- Scientific Q&A over papers with citations
- RAG API already existed; no bandwidth for UI/auth



Solution

- Streamlit UI deployed through **JupyterHub**
- RAG API ran as a JupyterHub service



Impact

- 300+ researchers used it
- Selected for the NAIRR pilot program
- Fast iteration – no new web stack or extra ops

https://github.com/PolusAI/aithena

You need three things to wrap any web app to run it in JupyterHub:

1. Command to run in CLI, i.e. `streamlit run –<flags>`
2. Expose web application on a port, i.e. `3000`
3. Work with a base URL, i.e. `/user/<email>`

Formula

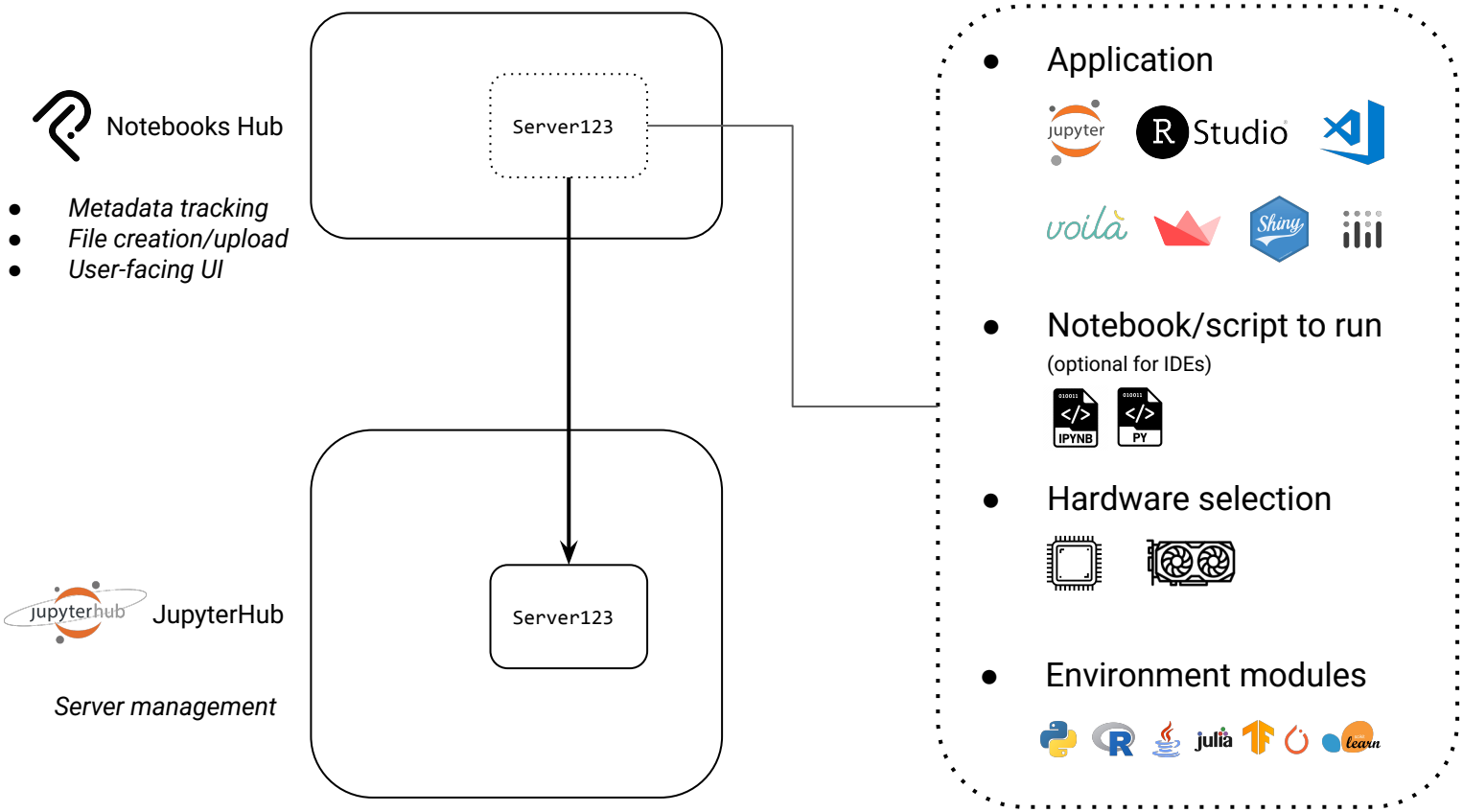
```
jhsingle-native-proxy --port {hub-port} --destport {app-port} \
  -- {command} {app-flags}
```

Integrated vs Standalone Approaches

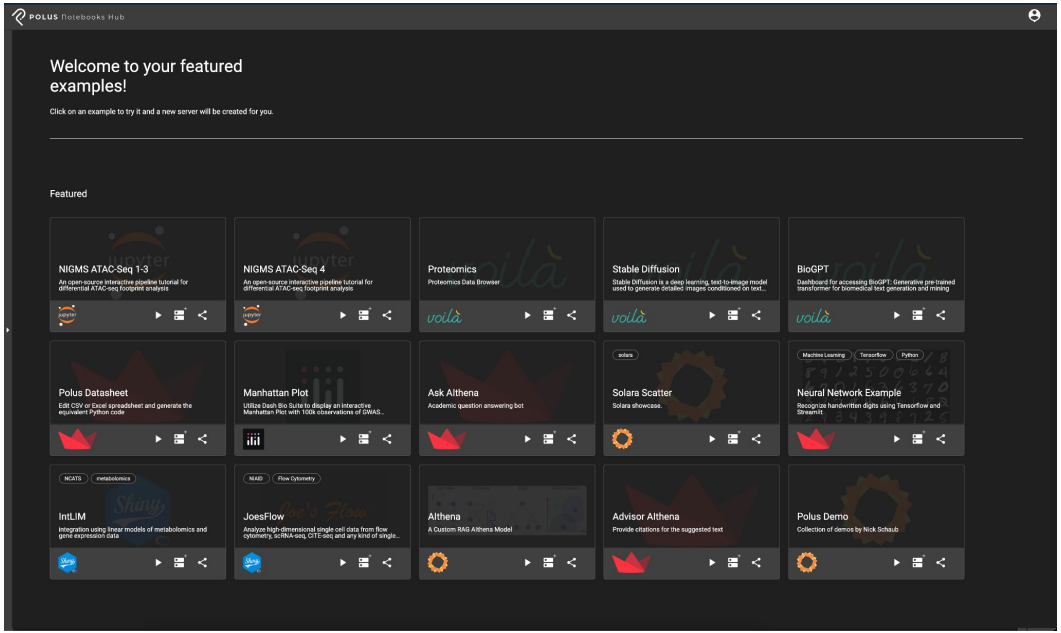
jupytercon

	Integrated	**Standalone**
Where it runs	in JupyterLab **user server**	as a separate **user server**
Best for	JupyterLab-first users	consumers of dashboards
✅ Pros	simple; shares auth/quotas & FS	isolation; scales independently; survives logout
⚠️ Trade-offs	shares resources with notebooks; tied to session	more plumbing; separate config/state; resource utilization
Setup	`jupyter-server-proxy`	`jhsingle-native-proxy` `jupyter standaloneproxy`

The Metadata Layer: Unlocking Reproducibility



Metadata Layer in Action



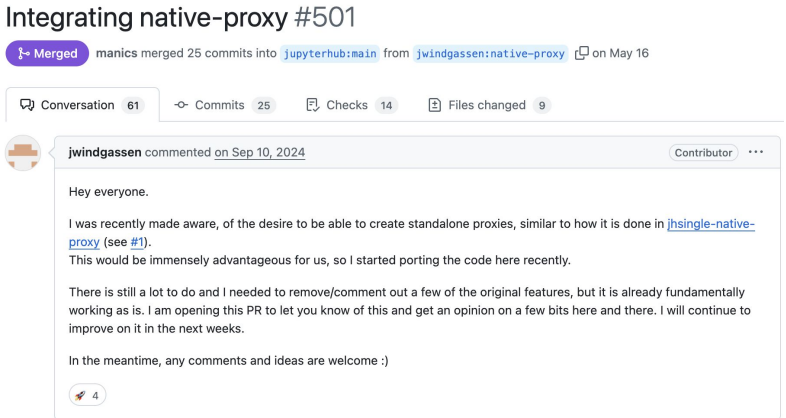
- Examples: curated dashboards — zero setup for non-experts.
- Quickstart: pre-configured IDEs — one click to work.
- Dashboard: see all your servers (status/owner/last used), Start/Resume/Stop, and Create New custom servers (image/size/hw).

Reunion with jupyter-server-proxy

jupytercon

standalone-proxy support was added to `jupyter-server-proxy`. This establishes a common standard and builds on the work `jhsingle-native-proxy` was pioneering

`jhsingle-native-proxy → jupyter standaloneproxy`



Integrating native-proxy #501

Merged **manics** merged 25 commits into `jupyterhub:main` from `jwindgassen:native-proxy` on May 16

Conversation 61 · Commits 25 · Checks 14 · Files changed 9

jwindgassen commented on Sep 10, 2024 — Contributor

Hey everyone.

I was recently made aware, of the desire to be able to create standalone proxies, similar to how it is done in jhsingle-native-proxy (see #1).
This would be immensely advantageous for us, so I started porting the code here recently.

There is still a lot to do and I needed to remove/comment out a few of the original features, but it is already fundamentally working as is. I am opening this PR to let you know of this and get an opinion on a few bits here and there. I will continue to improve on it in the next weeks.

In the meantime, any comments and ideas are welcome :)

🚀 4

https://github.com/jupyterhub/jupyter-server-proxy/pull/501

Thank you!

jupytercon

Work done at Axle

Learn more:
https://polusai.github.io/notebooks-hub

 John McAuliffe Software Engineer

 Swazoo Claybon Software Engineer

 Eduardo Luiggi Infrastructure Data Scientist

 Sunny Yu Director of Data Platforms

Access/collaboration: sunny.yu@axleinfo.com

#JupyterCon

Currently at Anaconda

See our Booth: #T6

Talk by **Kyle Kelley** *Runtime Agents: Unleashing Event Sourced Collaboration for Jupyter* (Wednesday November 5 11:35am)

marketplace labextensions.dev

