

# Streamline data science workflow development using Jupyter Notebook And Airflow

Neha Singla, Apple  
Sathish Kumar Thangaraj, Apple



# Agenda

- Jupyter Notebooks
- Data Science Workflow
- Data Science Workflow- Pain Points
- Streamlining Data Science Workflow
- Demo
- Jupyter Workflow with Airflow Architecture

# Jupyter Notebooks

- Jupyter Ecosystem
- Multiple languages
- Prototyping
- Data Exploration
- Iterative Experiments

The screenshot displays a Jupyter Notebook interface with a dark theme. On the left, a file browser shows a directory structure with files like '01 - Introduction.ipynb', '02 - Local maps.ipynb', etc. The main notebook area is titled '01 - Introduction.ipynb' and contains the following content:

### Creating a Map

First, let's create a map instance:

```
[3]: unfolded_map = create_map()
```

In environments that support Jupyter Widgets, such as Jupyter Notebook, JupyterLab, and Google Colab, simply put the map variable as the last or only line in a cell:

```
[4]: unfolded_map
```

In Jupyter Lab we also have the option of displaying a map as a separate side pane using the `Sidecar` package. In other environments than Jupyter Lab, using `sidecar` will probably not work.

```
[5]: sc = Sidecar(title='Unfolded Map', anchor='split-right')
with sc:
    display(unfolded_map)
```

### Adding data

We can now add a dataframe as a dataset to the map:

```
[9]: unfolded_map.add_dataset({
    'data': pd.DataFrame({
        'City': ['Buenos Aires', 'Brasilia', 'Santiago', '
        'Country': ['Argentina', 'Brazil', 'Chile', 'Colom
        'Latitude': [-34.58, -15.78, -33.45, 4.60, 10.48],
        'Longitude': [-58.66, -47.91, -70.66, -74.08, -66.
    })
})
```

[9]: LocalDataset(id='7d7296c0-fb78-4f08-b8dc-7bc2dd734f49',

On the right, a side pane titled 'Unfolded Map' shows a map of South America with several cities marked by colored dots. The map includes labels for countries like Mexico, Cuba, Colombia, Ecuador, Peru, Brazil, Bolivia, Paraguay, and Argentina. The interface also shows a menu with options like 'Share', 'Docs', and 'Help'.

# Data Science Workflow

- Data sources



- Languages, Libraries and frameworks



- Compute

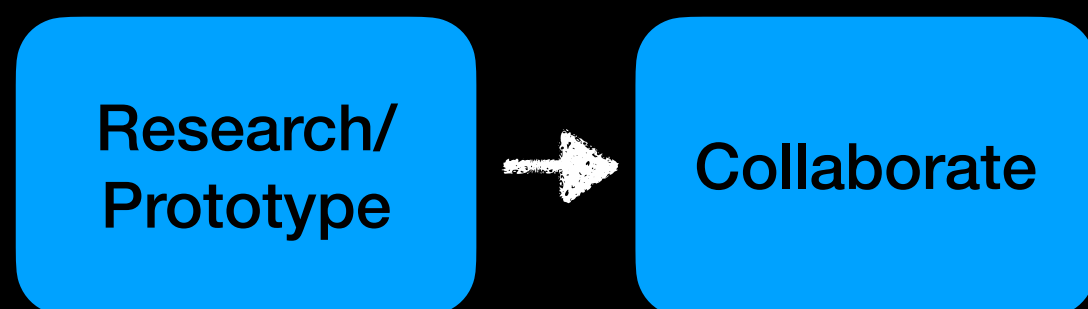


- Machine Learning

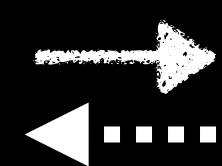


- Share and collaborate

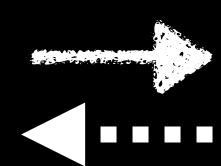
# Data Science Workflow Continued...



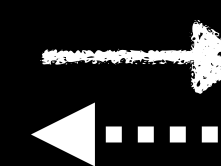
Reproduce/  
Fix



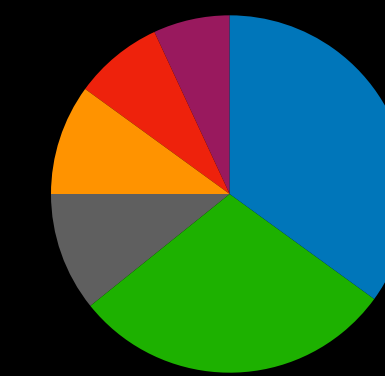
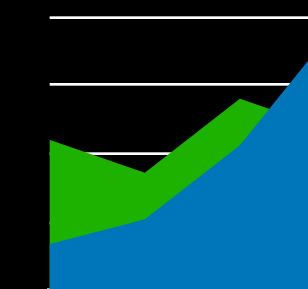
Work with data  
engineer for prod  
rollout



Platform  
Engineer/ SRE



Share Artifacts/  
Results



# Data Science Workflow- Pain Points

- Knowledge buried in multiple notebook files
- Extract code and work with data engineer to run in production
- Reproducibility
- Running large scale interactive experiments
- Debug failed pipelines



# Jupyter - Beyond Interactive use cases

- Long running sessions
- Jobs running at a regular interval
- Notebook workflows

# Meet **Workflows** on Jupyter Notebooks.

Workflows > gv2fol4m2vus

## Yelp-review-data-ml-workflow

Deploy

Runs **Tasks**

```
graph LR; A[Spark-Ingest-Data] --> B[Spark-Feature-Gen]; B --> C[fine-tune-LLM1]; B --> D[fine-tune-LLM]; C --> E[evaluate-LLMs]; D --> E;
```

**Workflow details**

- Name: Yelp-review-data-ml-workflow
- Created at: 4/18/2024, 3:08:58 PM
- Updated at: 4/18/2024, 3:11:29 PM
- Last deployed: 4/18/2024, 3:13:49 PM
- Namespace: [REDACTED]
- Status: **ACTIVE**
- Deployment status: **UPDATED**



Demo

# Why Airflow ?

- General purpose orchestration system
- Extendibility
- Monitoring capabilities
- Huge community and support

# Papermill Operator

- Parameterize Jupyter Notebooks
- Execute Jupyter Notebooks

# Papermill Operator Limitations

- Tightly coupled with python runtime
- Scaling
- Multi-tenancy

# Papermill Operator- Enhancements

<https://github.com/apache/airflow/pull/34840>

- Support multiple runtimes
- Language Agnostic
- Kernels can run in any cloud

The screenshot shows a GitHub pull request page for the Apache Airflow repository. The title of the pull request is "Extend papermill operator to support remote kernels #34840". It is marked as "Merged" and was merged by bolkedebruin on Nov 13, 2023. The pull request includes 40 conversations, 1 commit, 71 checks, and 14 files changed. A comment from akshaychitneni, dated Oct 9, 2023, describes the changes: "This PR adds support to run papermill operator that can connect to kernels managed externally by other systems. This would be useful to run the operator in cloud environments and would also be helpful to run spark or scala notebooks. It extends papermill to support new engine using the entry\_points as described here. It adds unittest and also a system test to run in CI environments. Validated using below steps in breeze environment:" followed by a code block containing shell commands: 

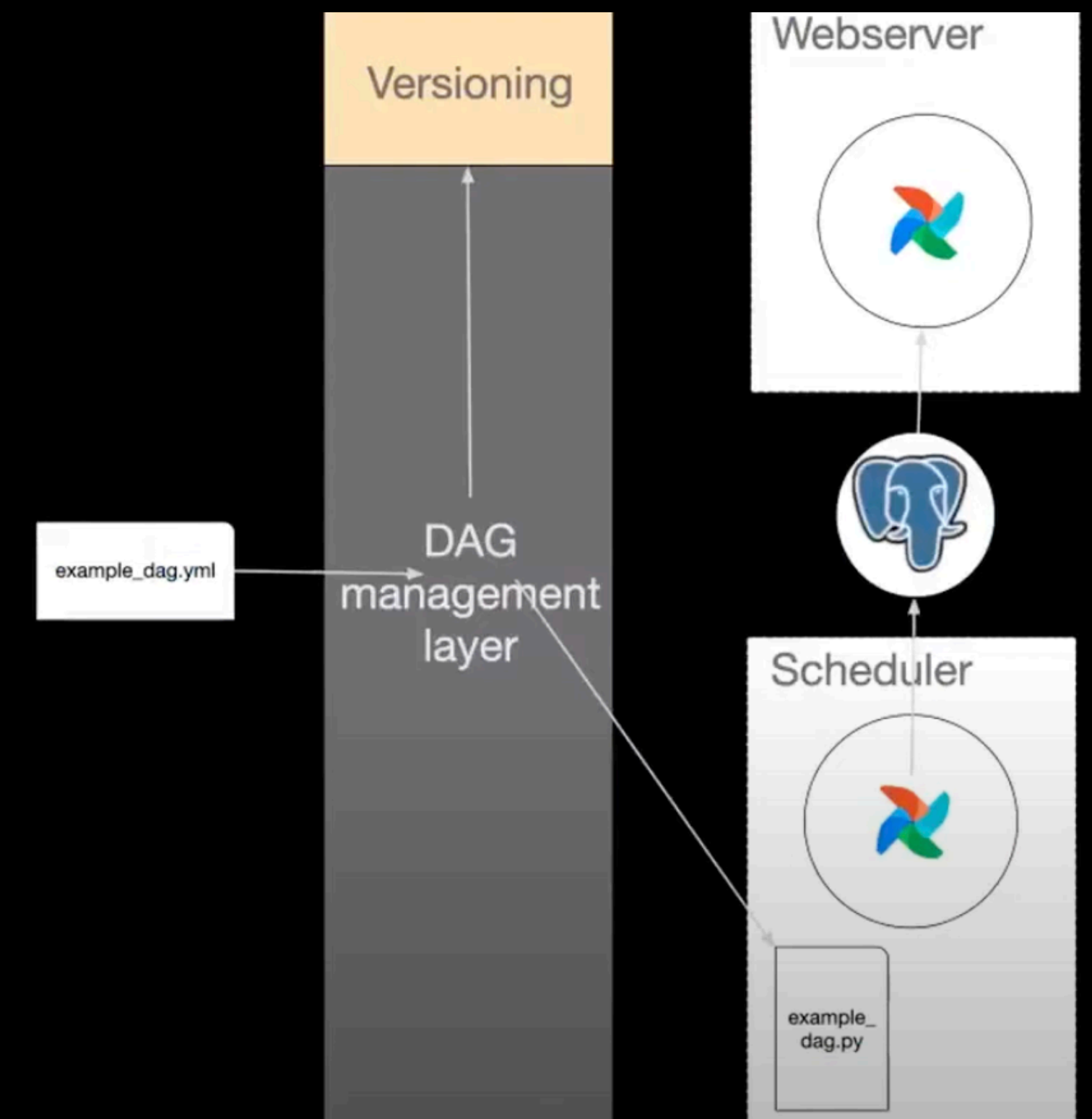
```
* breeze ci-image build --upgrade-to-newer-dependencies
* breeze start-airflow
* pytest --system papermill tests/system/providers/papermill/example_papermill_remote_verify.py
```

 A bot comment from boring-cyborg, dated Oct 9, 2023, congratulates the contributor and provides a link to the Contribution Guide. The right sidebar shows the pull request's metadata, including reviewers (bolkedebruin, Taragolis), assignees (none), labels (area:providers, area:system-tests, changelog:skip, provider:papermill), projects (none), and milestones (Airflow 2.8.0).

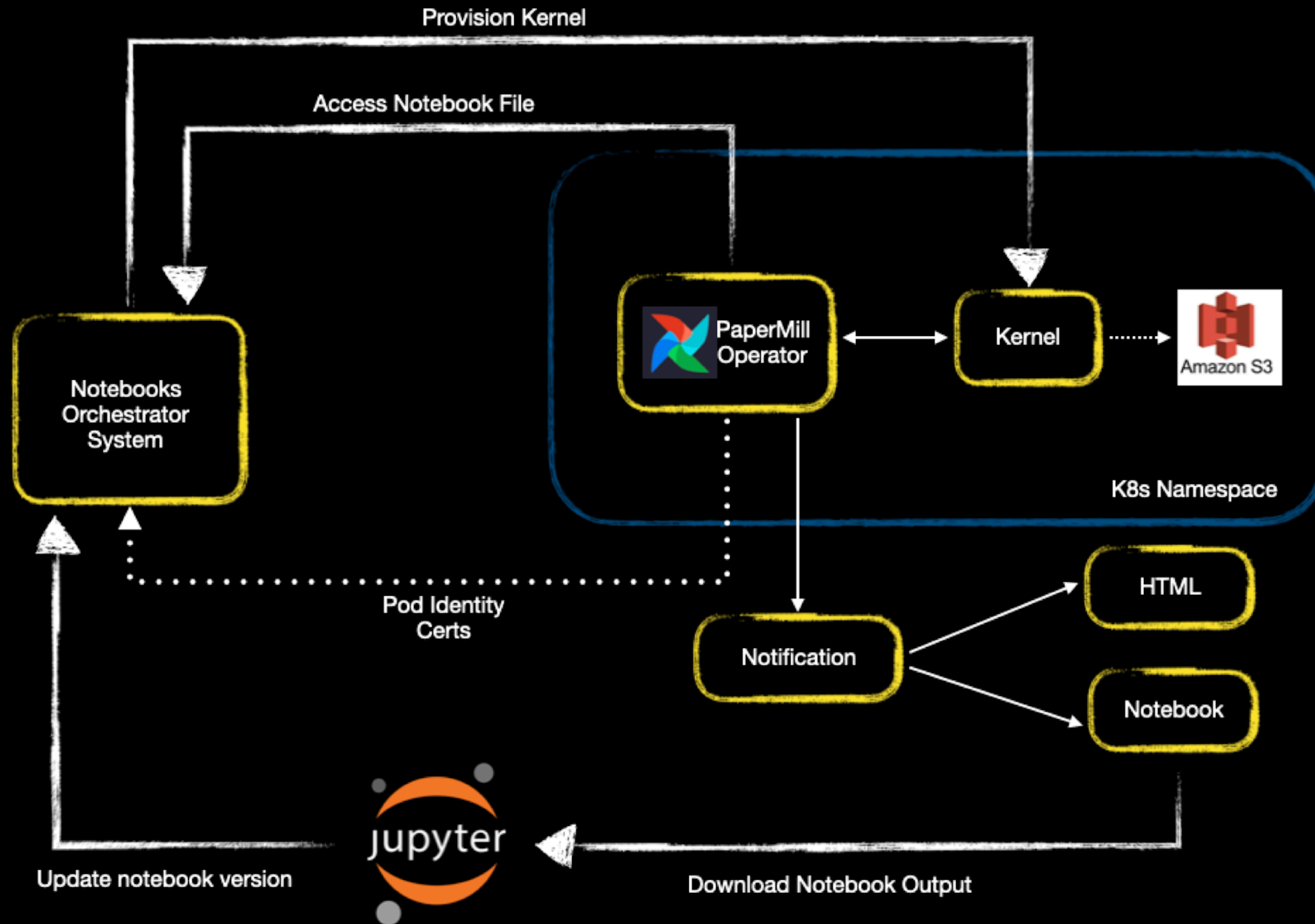


# Multi-tenant Pipeline Orchestration using Airflow

- Pipeline Domain Specific Language (DSL)
- Directed Acyclic Graph (DAG) Management
  - Version control
  - Deployment to K8s Namespace/Cluster
- Multi-tenancy
- Airflow Operator marketplace

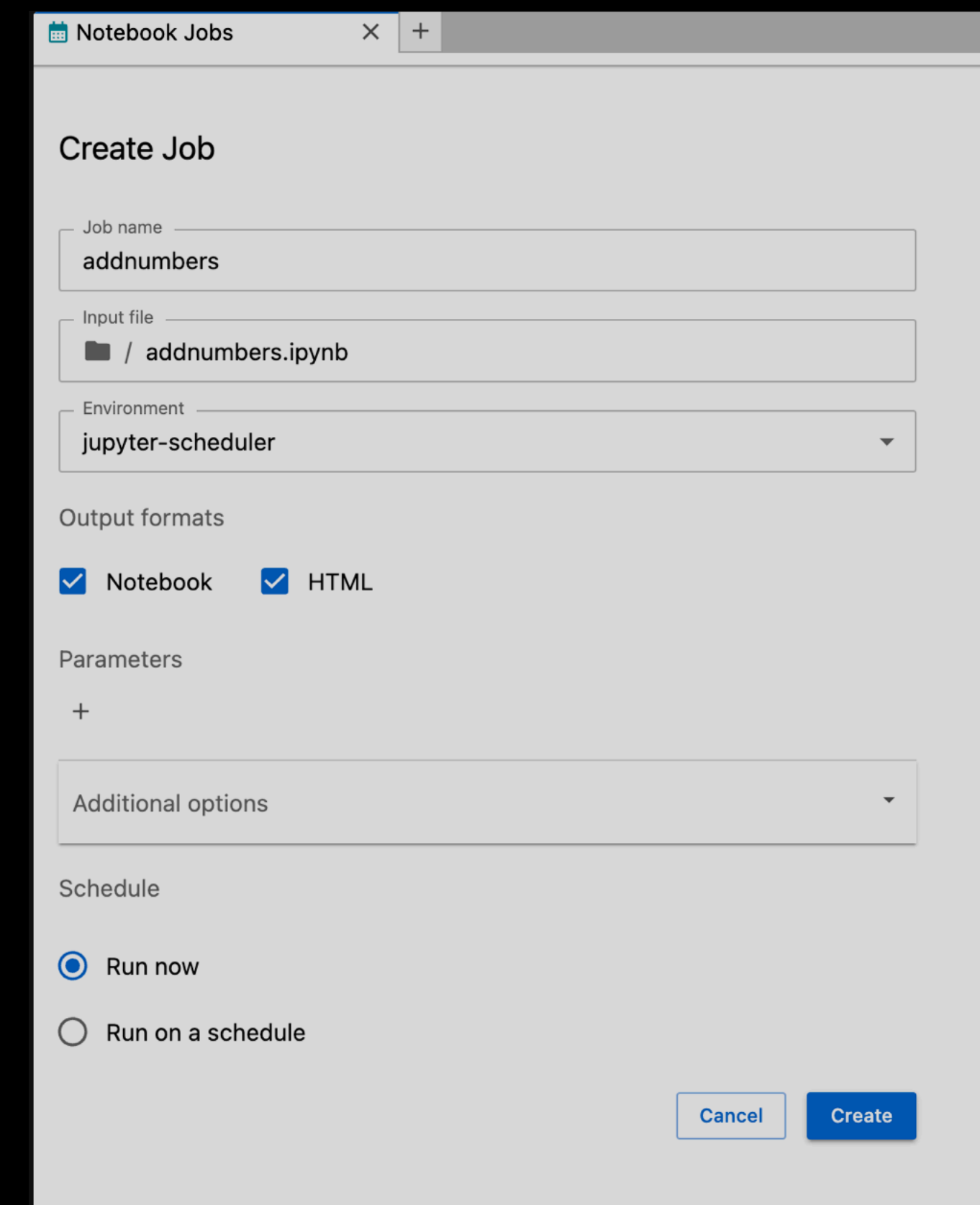
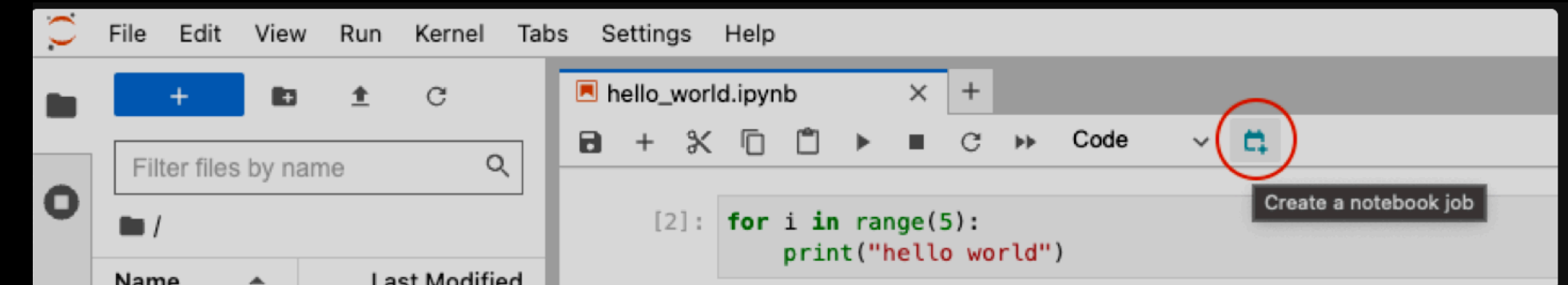


# Notebook Execution Using Airflow

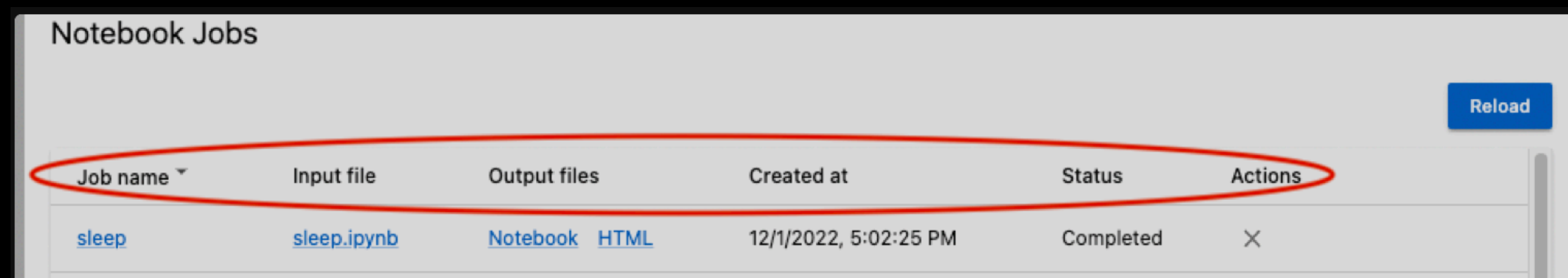


# Jupyter Scheduler

- Run jobs in background from Jupyter workspace
  - Run Once
  - Run on a Schedule
- Manage job definitions and jobs
- View historical runs
- Download job run output in multiple formats
- Notebooks as template

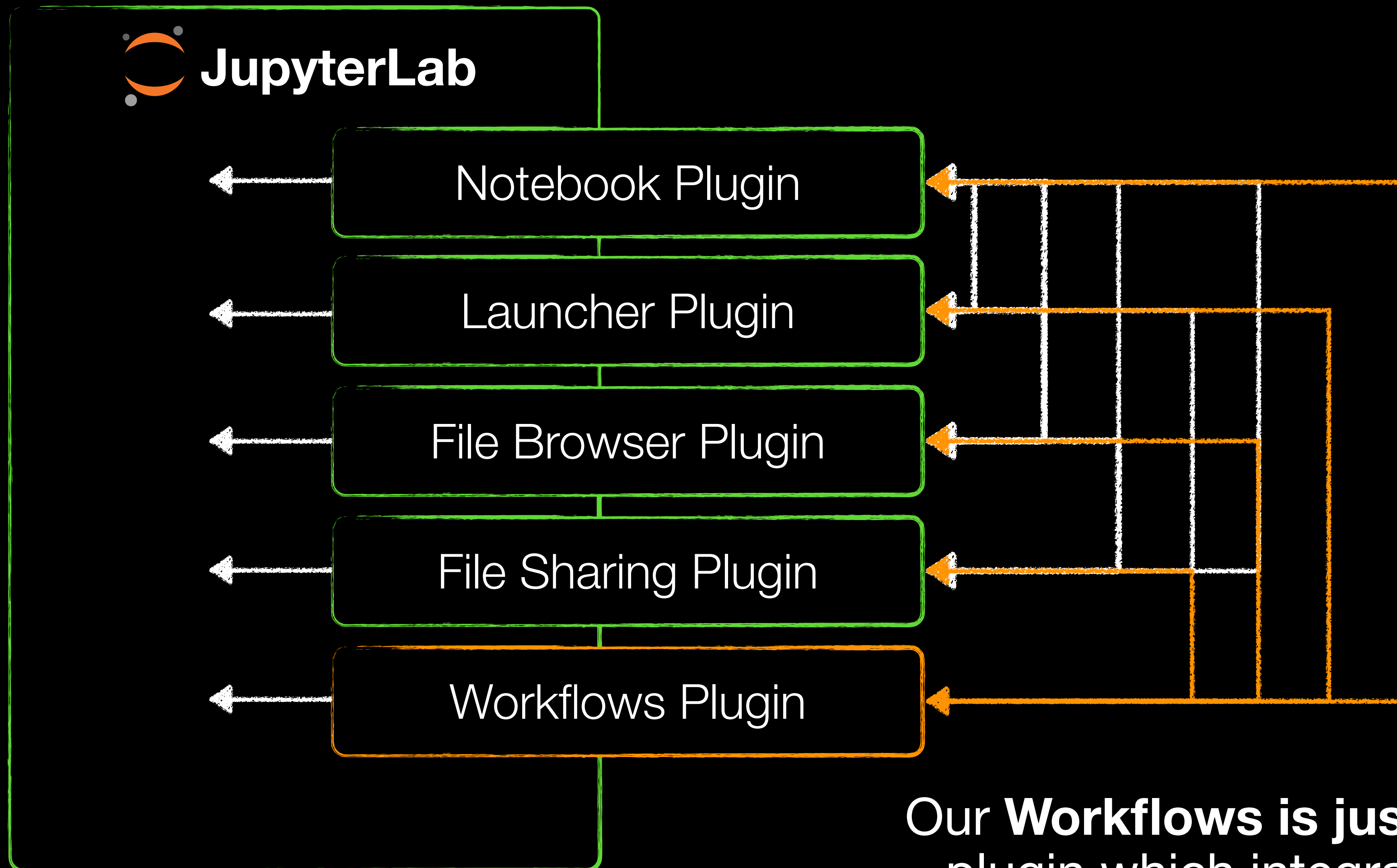
A screenshot of the 'Create Job' form in the Jupyter Scheduler interface. The form is titled 'Create Job' and contains several sections:

- Job name:** A text input field containing 'addnumbers'.
- Input file:** A dropdown menu showing '/ addnumbers.ipynb'.
- Environment:** A dropdown menu showing 'jupyter-scheduler'.
- Output formats:** Two checkboxes, 'Notebook' and 'HTML', both of which are checked.
- Parameters:** A section with a '+' sign and an empty text input field.
- Additional options:** A dropdown menu.
- Schedule:** Two radio buttons: 'Run now' (which is selected) and 'Run on a schedule'.

At the bottom right of the form are 'Cancel' and 'Create' buttons.

Job name	Input file	Output files	Created at	Status	Actions
<a href="#">sleep</a>	<a href="#">sleep.ipynb</a>	<a href="#">Notebook</a> <a href="#">HTML</a>	12/1/2022, 5:02:25 PM	Completed	<a href="#">X</a>

# Dynamic DAG Creation in Jupyter



Our **Workflows** is just another plugin which integrates with many other core plugins.

# Future Work

- Open Source
  - Inviting Collaborators
  - Jupyter Meetings
    - <https://jupyter-server.readthedocs.io/en/latest/contributors/team-meetings.html>
- Features
  - Event Driven Notebook workflows
  - Workflow Sharing



**Thank You**

Questions?