

Literate Computing for Reproducible Infrastructure

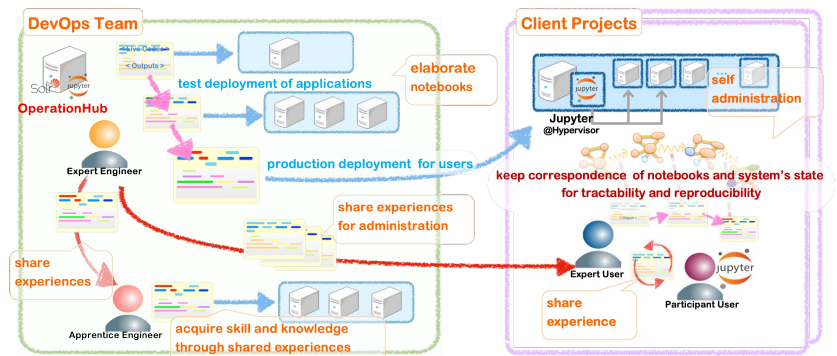


Literate Computing
for Reproducible Infrastructure

Reproducibility Extensions for Jupyter Notebook

Literate Computing for Reproducible Infrastructure (LC4RI) is our daily practice for managing IT infrastructure. The NII cloud operations team, a small DevOps group, operates over 350 nodes built on OpenStack and provides cloud computing and storage stacks as services. For reproducible research, it is essential to share infrastructure design and detailed IT workflows with participants, as well as to automate complex operations.

LC4RI uses Jupyter Notebooks to share reproducible IT workflows, enable collaboration within DevOps teams, and provide reproducible IT environments for research projects.



Reproducibility Extensions for Jupyter Notebook: Jupyter Notebook was designed initially as a non-linear explorative computing tool, typically for data-driven scientists. As a countermeasure to ensure robust, traceable, and reproducible IT operations, the "Semi-Linear Extension" constrains arbitrary cell executions into semi-linear orders and records throughout loggings. As for consolidating and flexible remarks, "sidestickers" enable collaborative annotation across notebooks.

"Semi-Linear Extension" for IT Operation

can continue execution on following not-yet-executed cells .

```
[1]: echo "This is 1st step" > foo; cat foo
```

```
[2]: echo "... 2nd step..." > foo && cat foo
```

```
In [3]: echooooo "... 3rd step... will fail" >> foo && cat foo
```

RuntimeError

Traceback (most recent call)

• : The light green bricks indicate successful completion.
• : The third light coral brick indicates some error.

❄️ : The snowflake indicates those bricks are frozen. Executions and edits are prohibited. The ❄️ will unfreeze bricks.
Success cells will be automatically frozen in order to prevent accidental duplicate operations. Error cells remain unfreeze so you can fix errors and re-execute the cell. You can continue execution on following not-yet-executed cells .

"sidestickers" Enable Flexible, Collaborative Annotation

sidestickers - Collaborative Annotation for Jupyter Notebook

Description: sidestickers is a notebook extension, which enables to attach sticky notes to each cell utilizing **Scrapbox.io**.

Jupyter notebook's narrative stories are efficient to share workflows and activities of researchers, educators, engineers, and other practitioners for reproducible computing. The notebooks are crucial tools both for describing and capturing a series of related events, results, or the like as narratives, in either prospective or retrospective cases. However, it is not sufficient enough because those narratives mainly focus on the subjects and stories within the notebook itself. We would like to have separate **channels for meta-, side-, and reflective-communications**, which are well-known as cases for sticky notes.

Place a sticky note

The toggle button on Toolbar "📌" allows/hide sidestickers.
For enabling sidestickers, please refer [Let's try sidestickers and README.md](#).

This cell already has a sticky note!

This cell does not have a sticky note yet.
If you have write permission in the associated Scrapbox, you can create a note page.

Note pages in "ep_weave" or "Scrapbox.io"

Can list multiple pages with a cell

Sticky Notes for Notebook files, they appear on the...

Let's try "sidestickers"

Sticky Notes for Notebook files, they appear on the right.

This is a page on Scrapbox.

sidestickers are attaching to the Python!

A note page

sidestickers enable to attach a sticky note to each cell

Throughout Loggings for Operation Trails

!! Summarizes massive output lines on GUI.

At each cell's execution, store all original output lines into individual files with codes, time, and id.

head

errors

tail

350+ nodes are managed via OperationHub

OperationHub is a tailored JupyterHub for DevOps instead of *sshd* on the management server. It provides segregated Jupyter containers for individual operators and mechanisms for Notebook sharing and reuse.

OperationHub

Search & Reuse

Operator A

Operator B

Container

Container

openstack.

350+ nodes and a dozen services

IT operation via Jupyter Notebook

Use individual credentials for operational traceability

Try on MyBinder

<https://mybinder.org/v2/gh/NII-cloud-operation/Jupyter-IT-docker/sc-demo?urlpath=tree>



National Institute of Informatics
<https://literate-computing.github.io/>

