ITINERIS

Open Science on Cloud using Jupyter Notebooks

Cloud workflows with NaaVRE

Concepts, tutorial and exercises

G. Pelouze, S. Koulouzis, Z. Zhao

(LifeWatch ERIC VLIC / MNS, University of Amsterdam)

25/06/2025

LifeWatch ERIC

University of Amsterdam

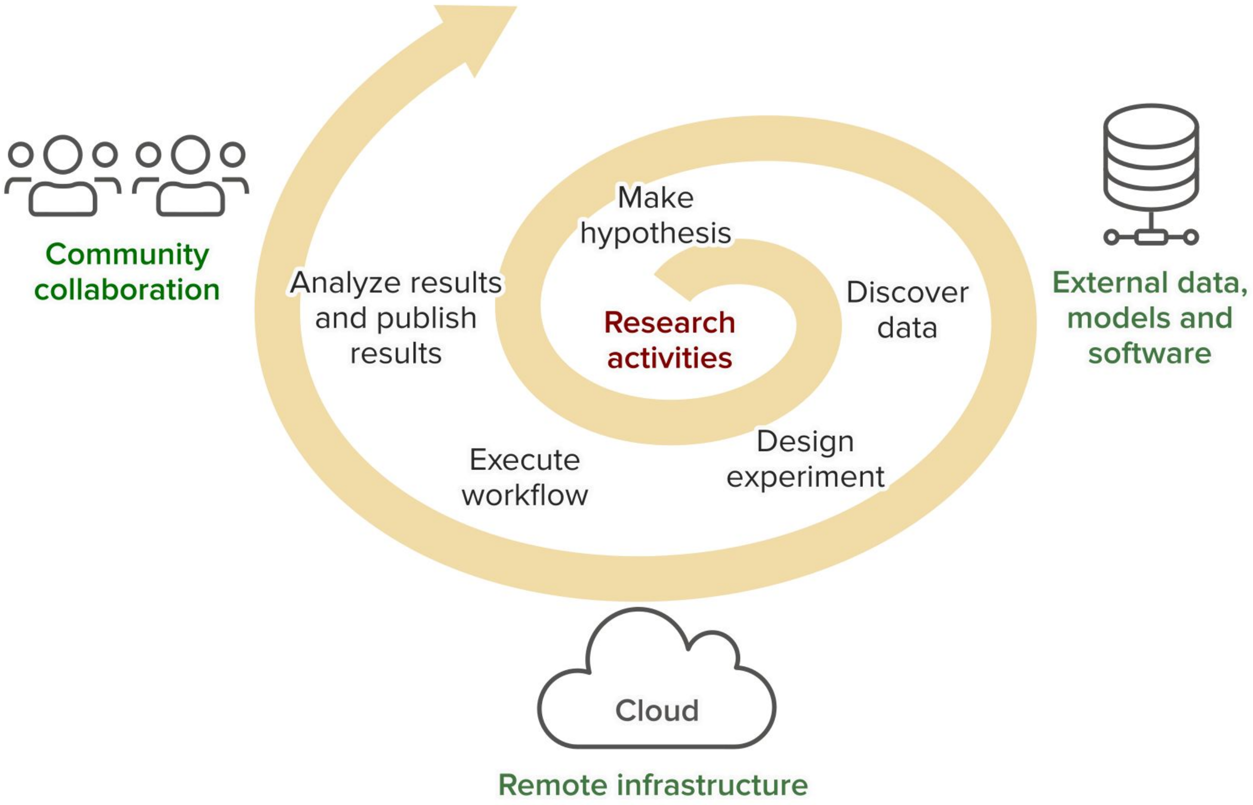
IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 "Education and Research" - Component 2: "From research to business" - Investment 3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani

Data-centric research lifecycle



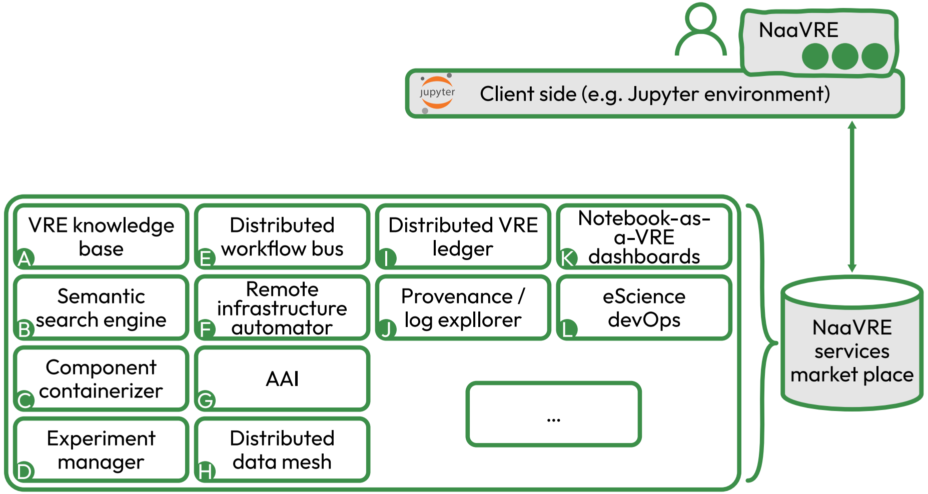
“Virtual Research Environments (VREs) provide user-centric support for discovering and selecting data and software services from different sources and composing and executing application workflows, …” (Jeffery et al., 2020)

How to support data-centric experiments

- **Computational notebooks**
  - Widely used by researchers
  - Documentation and visualization alongside code
  - Interactive code execution
  - → **developing and tweaking models**
- **Virtual research environment**
  - Collaboration
  - FAIR data, models and services
  - Remote infrastructure
  - Scaling up models (e.g. porting prototype to large-scale cloud environment)
  - → **sharing mature models & more**
- **Combined, they are NaaVRE**

NaaVRE

Notebook as a **Virtual Research Environment**



NaaVRE

Client side (e.g. Jupyter environment)

A VRE knowledge base

B Semantic search engine

C Component containerizer

D Experiment manager

E Distributed workflow bus

F Remote infrastructure automator

G AAI

H Distributed data mesh

I Distributed VRE ledger

J Provenance / log expllorer

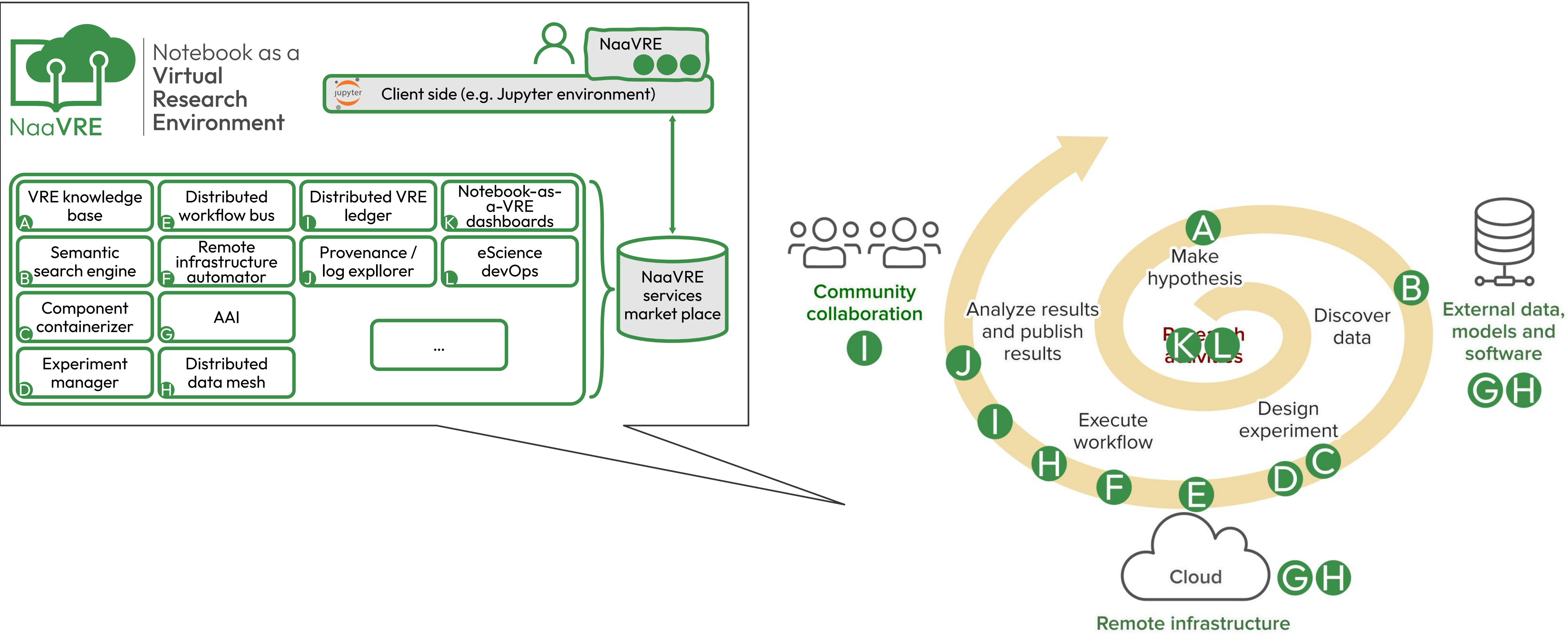
K Notebook-as-a-VRE dashboards

L eScience devOps

...

NaaVRE services market place

Research activities and Jupyter extensions



NaaVRE

Notebook as a **Virtual Research Environment**

NaaVRE

Client side (e.g. Jupyter environment)

VRE knowledge base A

Distributed workflow bus E

Distributed VRE ledger I

Notebook-as-a-VRE dashboards K

Semantic search engine B

Remote infrastructure automator F

Provenance / log expllorer J

eScience devOps L

Component containerizer C

AAI G

...

Experiment manager D

Distributed data mesh H

NaaVRE services market place

Community collaboration I

Make hypothesis A

Discover data B

Design experiment C D

Execute workflow E F H

Analyze results and publish results I J

Research activities K L

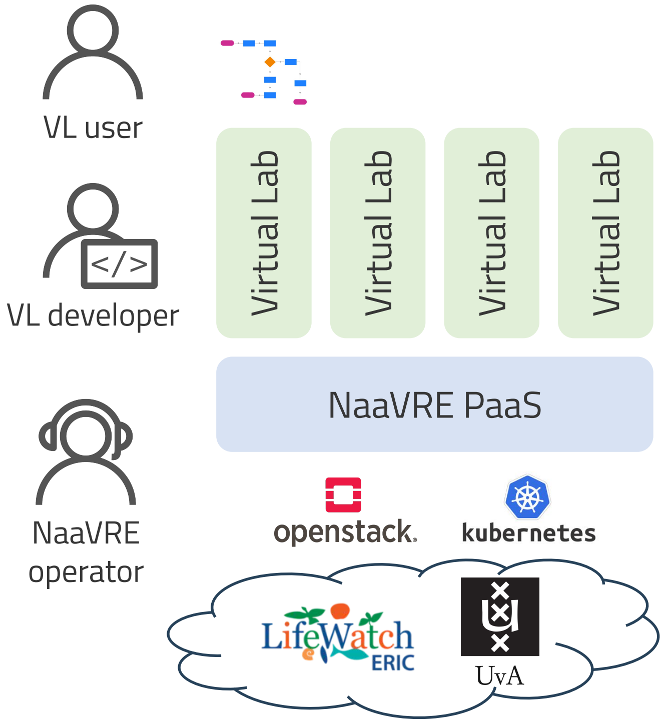
External data, models and software G H

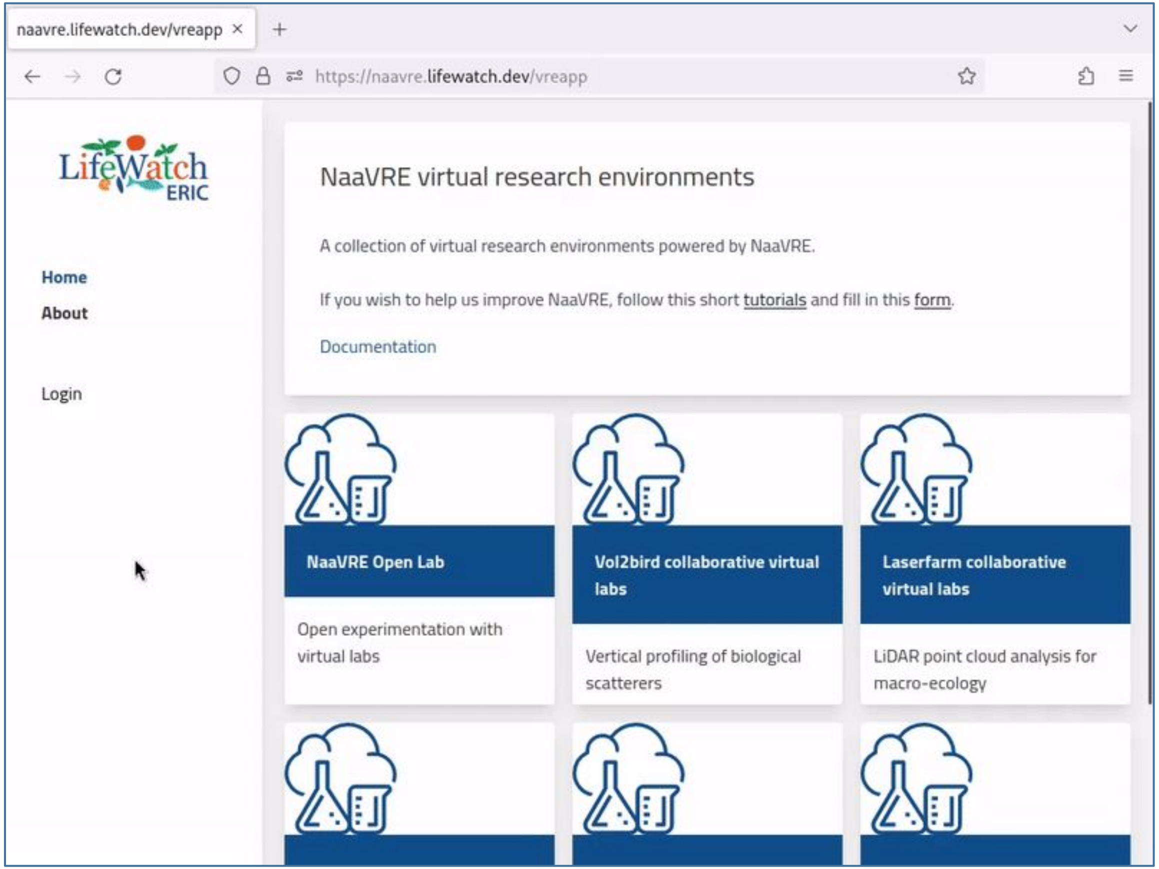
Cloud G H

Remote infrastructure

Zhao, Z., Koulouzis, S., Bianchi, R., Farshidi, S., Shi, Z., Xin, R., Wang, Y., Li, N., Shi, Y., Timmermans, J., Kissling, W.D.: Notebook-as-a-VRE (NaaVRE): From private notebooks to a collaborative cloud virtual research environment. Softw Pract Exp. spe.3098 (2022). https://doi.org/10.1002/spe.3098.

Notebook-as-a-VRE and Virtual Labs




naavre.lifewatch.dev/vreapp
https://naavre.lifewatch.dev/vreapp
LifeWatch ERIC
Home
About
Login

NaaVRE virtual research environments

A collection of virtual research environments powered by NaaVRE.

If you wish to help us improve NaaVRE, follow this short tutorials and fill in this form.

Documentation

NaaVRE Open Lab

Open experimentation with virtual labs

Vol2bird collaborative virtual labs

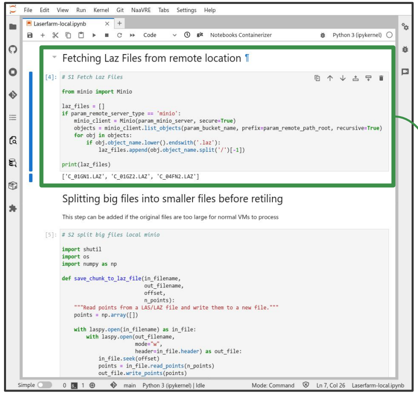
Vertical profiling of biological scatterers

Laserfarm collaborative virtual labs

LiDAR point cloud analysis for macro-ecology

From notebooks to cloud workflows

1. PROTOTYPE MODEL IN NOTEBOOK



Fetching Laz Files from remote location ¶

```
[4]: # S1 Fetch Laz Files

from minio import Minio

laz_files = []
if param_remote_server_type == 'minio':
    minio_client = Minio(param_minio_server, secure=True)
    objects = minio_client.list_objects(param_bucket_name, prefix=param_remote_path_root, recursive=True)
    for obj in objects:
        if obj.object_name.lower().endswith('.laz'):
            laz_files.append(obj.object_name.split('/')[-1])

print(laz_files)
```

```
['C_01GN1.LAZ', 'C_01GZ2.LAZ', 'C_04FN2.LAZ']
```

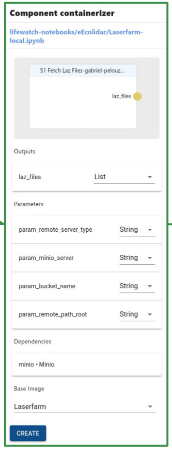
Splitting big files into smaller files before retiling

This step can be added if the original files are too large for normal VMs to process

```
[5]: # S2 split big files local minio

import shutil
import os
import numpy as np

def save_chunk_to_laz_file(in_filename,
                           out_filename,
                           offset,
                           n_points):
    """Read points from a LAS/LAZ file and write them to a new file."""
    points = np.array([])

    with laspy.open(in_filename) as in_file:
        with laspy.open(out_filename,
                        mode="w",
                        header=in_file.header) as out_file:
            in_file.seek(offset)
            points = in_file.read_points(n_points)
            out_file.write_points(points)
```

Simple 0 1 main Python 3 (ipykernel) | Idle Mode: Command Ln 7, Col 26 Laserfarm-local.ipynb

2. CONTAINERIZE CELLS



Component containerizer

lifewatch-notebooks/eEcolidar/Laserfarm-local.ipynb

Outputs

laz_files	List

Parameters

param_remote_server_type	String
param_minio_server	String
param_bucket_name	String
param_remote_path_root	String

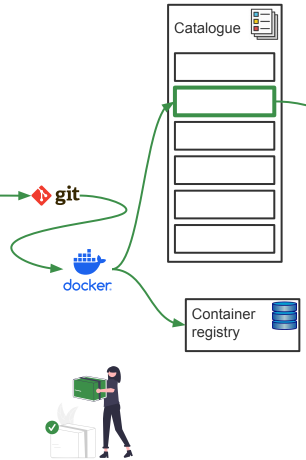
Dependencies

minio • Minio

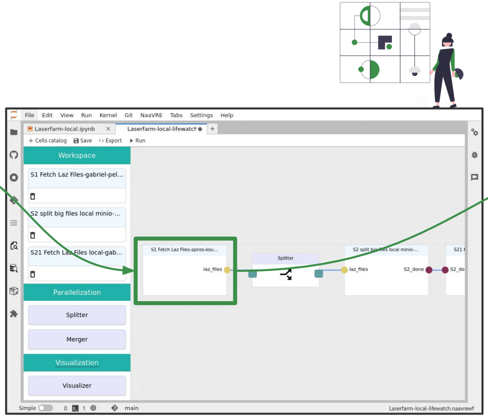
Base Image

Laserfarm

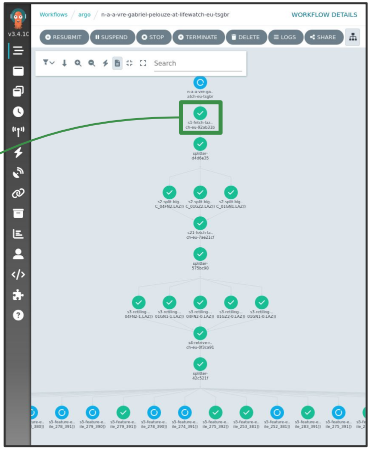
CREATE



3. COMPOSE A WORKFLOW


File Edit View Run Kernel Git NaaVRE Tabs Settings Help
Laserfarm-local.ipynb
Laserfarm-local-lifewatch
+ Cells catalog
Save
Export
Run
Workspace
S1 Fetch Laz Files-gabriel-pel...
S2 split big files local minio-...
S21 Fetch Laz Files local-gab...
Parallelization
Splitter
Merger
Visualization
Visualizer
S1 Fetch Laz Files-spiros-kou...
laz_files
Splitter
S2 split big files local minio-...
laz_files
S2_done
S21
S2_do
Simple 0 1 main
Laserfarm-local-lifewatch.naavrewf

4. RUN IT IN THE CLOUD

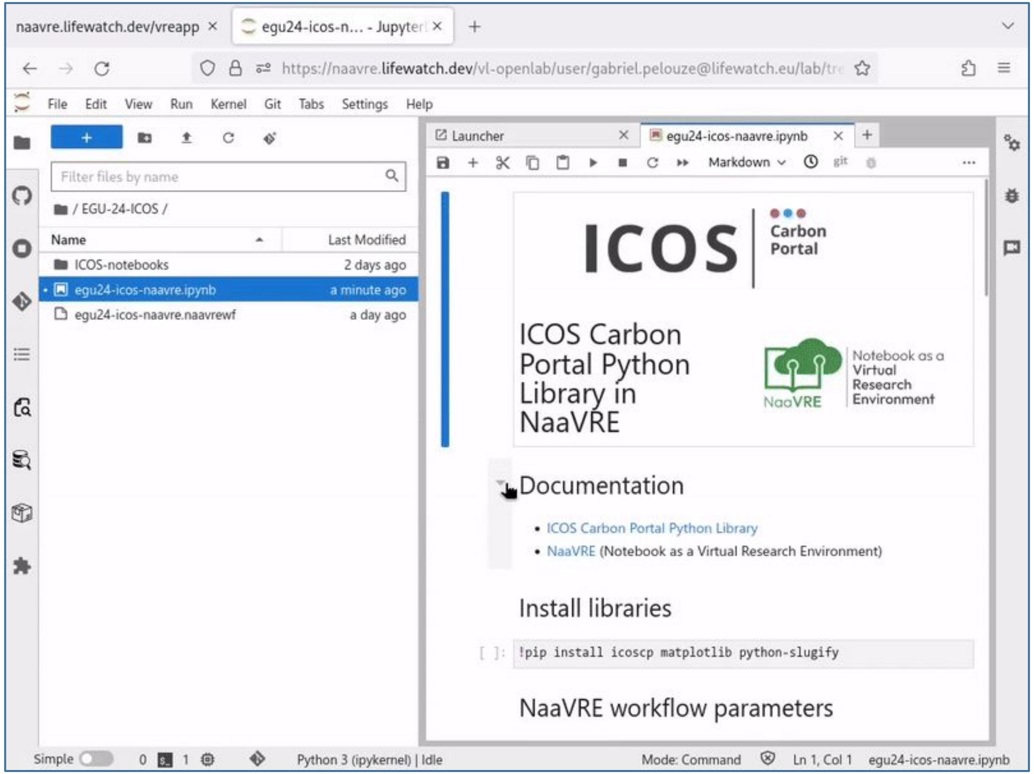

v3.4.10
Workflows / argo / n-a-a-vre-gabriel-pelouze-at-lifewatch-eu-tsgbr
WORKFLOW DETAILS
RESUBMIT
SUSPEND
STOP
TERMINATE
DELETE
LOGS
SHARE
Search
n-a-a-vre-ga.. atch-eu-tsgbr
s1-fetch-laz.. ch-eu-92ab31b
splitter-d4d6e35
s2-split-big.. C_04FN2.LAZ])
s2-split-big.. C_01GZ2.LAZ])
s2-split-big.. C_01GN1.LAZ])
s21-fetch-la.. ch-eu-7ae21cf
splitter-575bc98
s3-retiling-.. 04FN2-1.LAZ])
s3-retiling-.. 01GN1-1.LAZ])
s3-retiling-.. 04FN2-0.LAZ])
s3-retiling-.. 01GZ2-0.LAZ])
s3-retiling-.. 01GN1-0.LAZ])
s4-retrive-r.. ch-eu-0f3ca91
splitter-42c521f
s5-feature-e.. ile_279_390])
s5-feature-e.. ile_279_391])
s5-feature-e.. ile_278_390])
s5-feature-e.. ile_274_391])
s5-feature-e.. ile_275_392])
s5-feature-e.. ile_253_381])
s5-feature-e.. ile_252_381])
s5-feature-e.. ile_283_391])
s5-feature-e.. ile_275_391])

From notebooks to cloud workflows

1. PROTOTYPE MODEL IN NOTEBOOK
2. CONTAINERIZE CELLS

Encapsulate **user-selected cells** as standardized services and **containerize** them as **reusable components**.

Code cell → docker



naavre.lifewatch.dev/vreapp
egu24-icos-n... - Jupyter
https://naavre.lifewatch.dev/vl-openlab/user/gabriel.pelouze@lifewatch.eu/lab/tr
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
Launcher
egu24-icos-naavre.ipynb
Markdown
git
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
egu24-icos-naavre.ipynb
ICOS Carbon Portal
NaaVRE Notebook as a Virtual Research Environment

/ EGU-24-ICOS /

Name	Last Modified
ICOS-notebooks	2 days ago
egu24-icos-naavre.ipynb	a minute ago
egu24-icos-naavre.naavrewf	a day ago

ICOS Carbon Portal Python Library in NaaVRE

Documentation

- ICOS Carbon Portal Python Library
- NaaVRE (Notebook as a Virtual Research Environment)

Install libraries

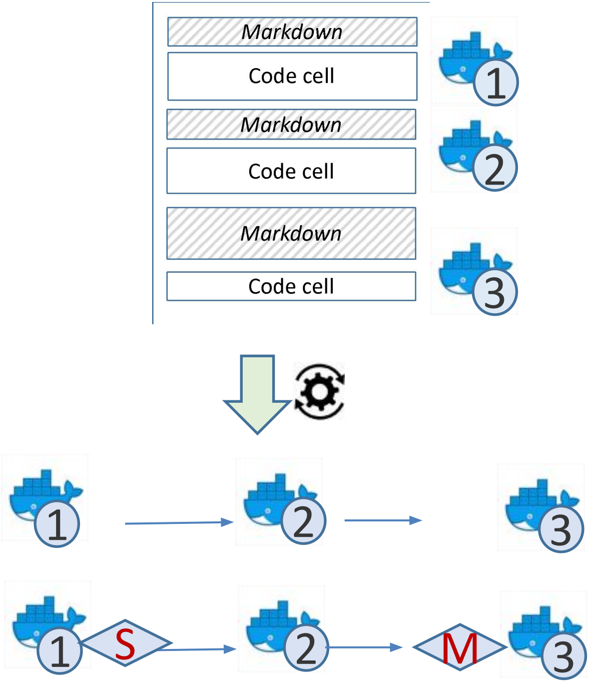
```
[ ]: !pip install icoscp matplotlib python-slugify
```

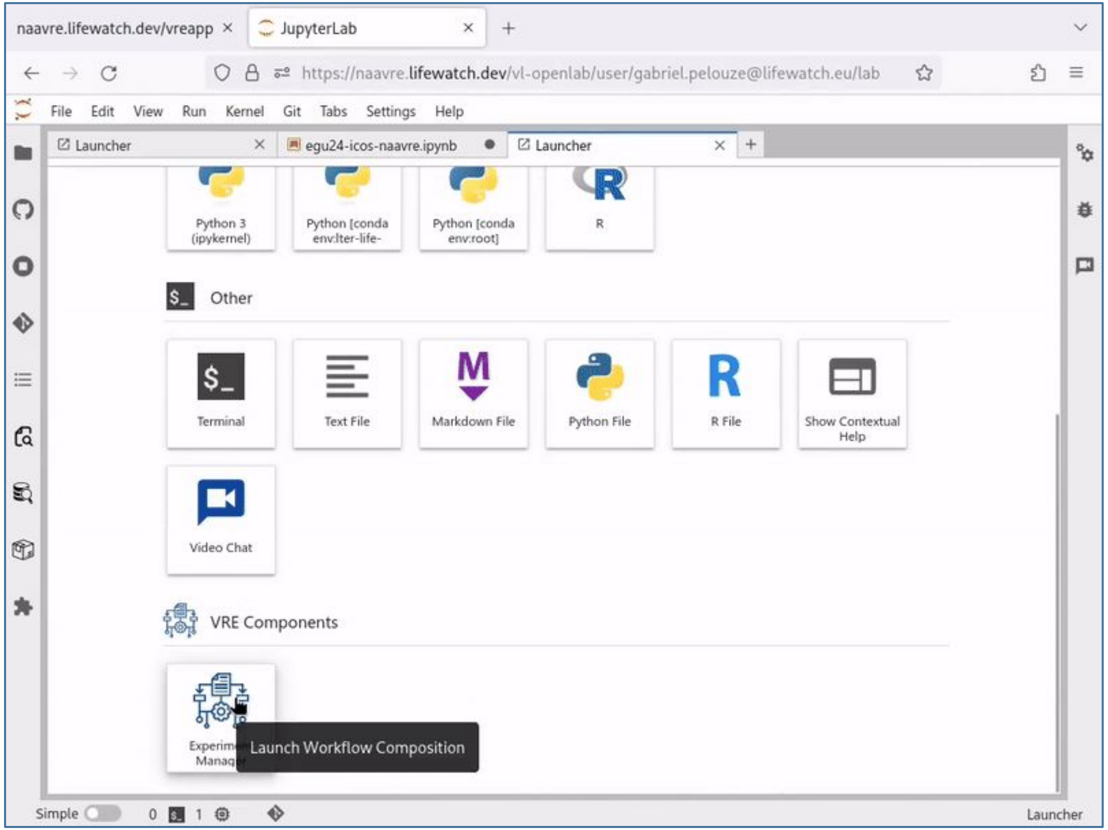
NaaVRE workflow parameters

From notebooks to cloud workflows

3. COMPOSE A WORKFLOW: Customize data flow and input

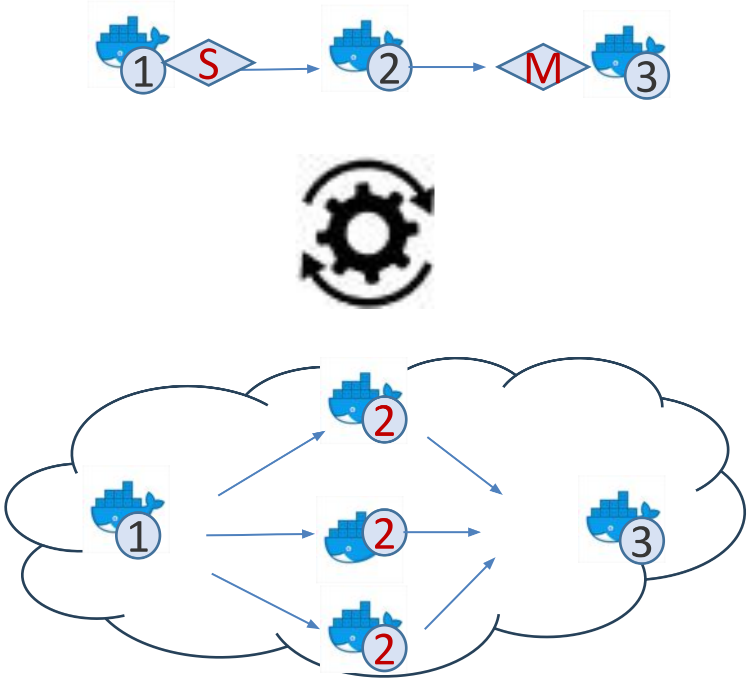
An experiment is a workflow of services, with customized input data and infrastructure

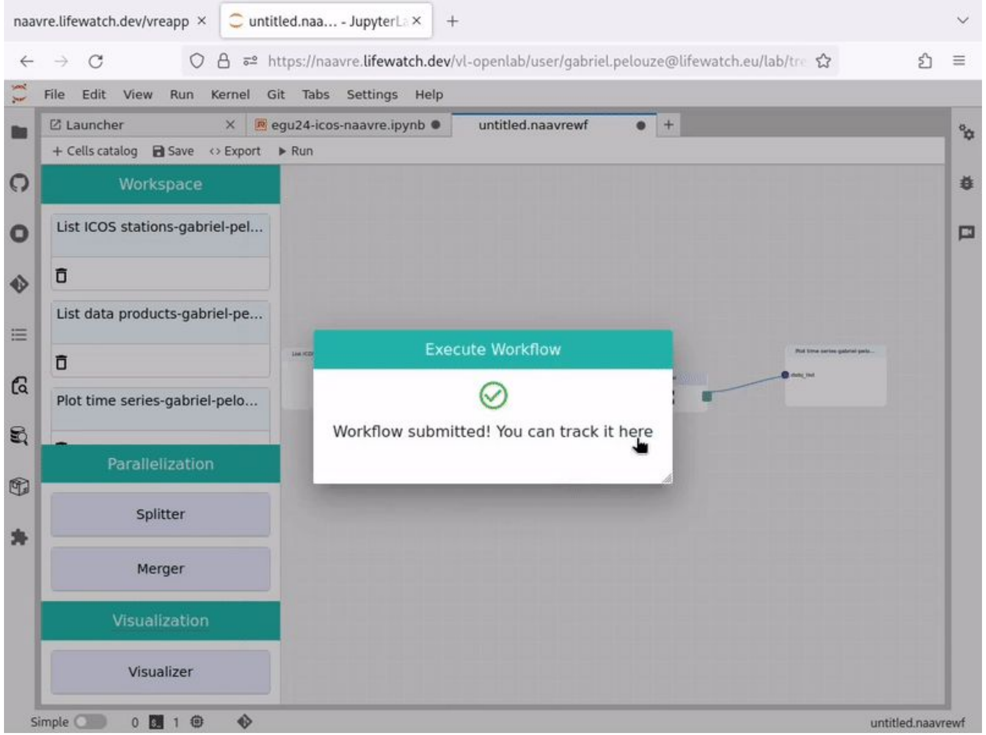




From notebooks to cloud workflows

4. RUN THE WORKFLOW: Automate the workflow components deployment and execution

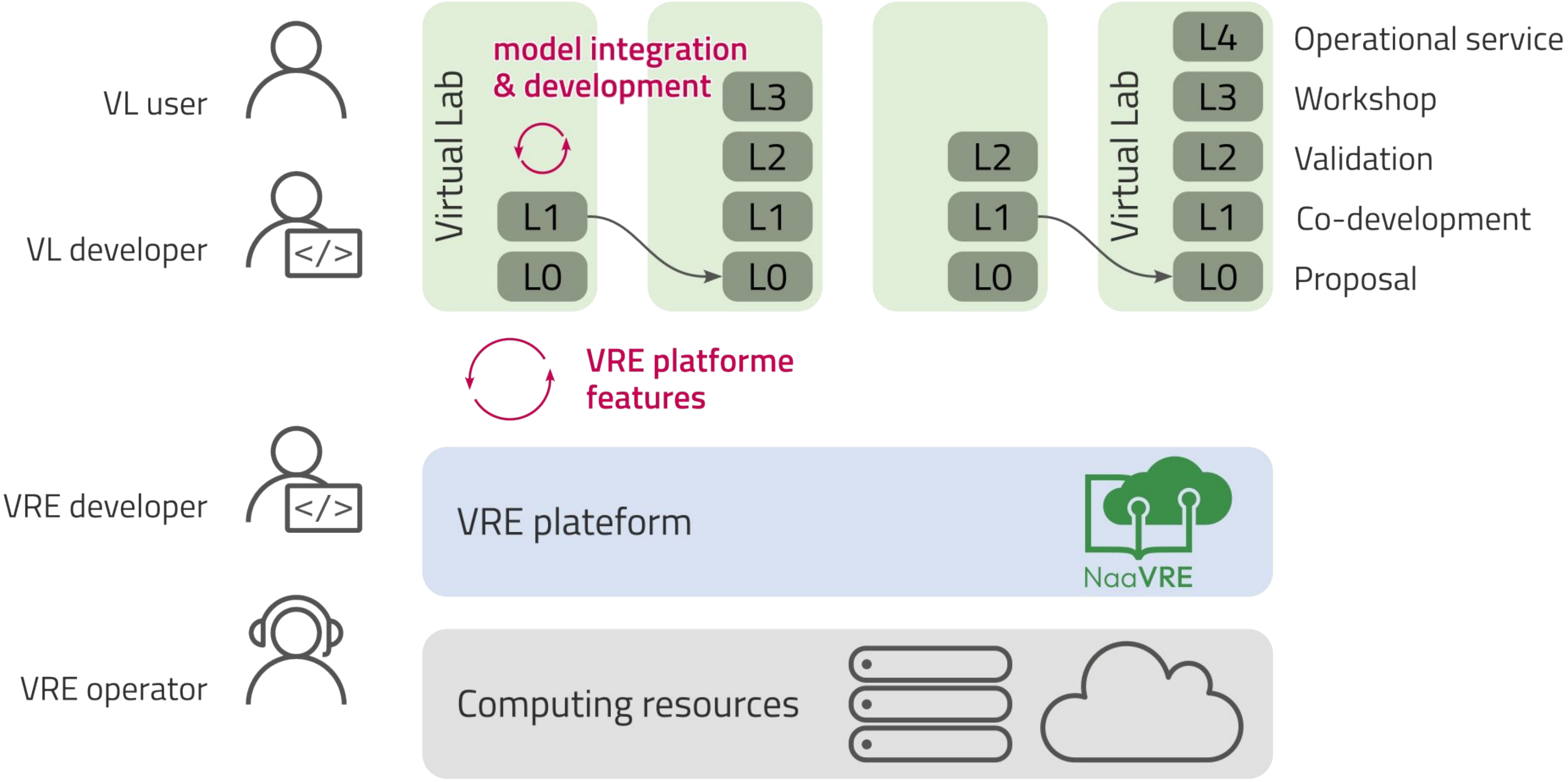



naavre.lifewatch.dev/vreapp
untitled.naa... - JupyterL
https://naavre.lifewatch.dev/vl-openlab/user/gabriel.pelouze@lifewatch.eu/lab/tre
File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
Launcher
egu24-icos-naavre.ipynb
untitled.naavrewf
+ Cells catalog
Save
<> Export
▶ Run
Workspace
List ICOS stations-gabriel-pel...
List data products-gabriel-pe...
Plot time series-gabriel-pelo...
Parallelization
Splitter
Merger
Visualization
Visualizer
Execute Workflow
Simple
0
1
untitled.naavrewf

Workflow submitted! You can track it here

Zhou, H., Hu, Y., Ouyang, X., Su, J., Koulouzis, S., Laat, C., Zhao, Z.: CloudsStorm: A framework for seamlessly programming and controlling virtual infrastructure functions during the DevOps lifecycle of cloud applications. Softw: Pract Exper. 49, 1421–1447 (2019). https://doi.org/10.1002/spe.2741

VRE and Virtual Lab co-development framework

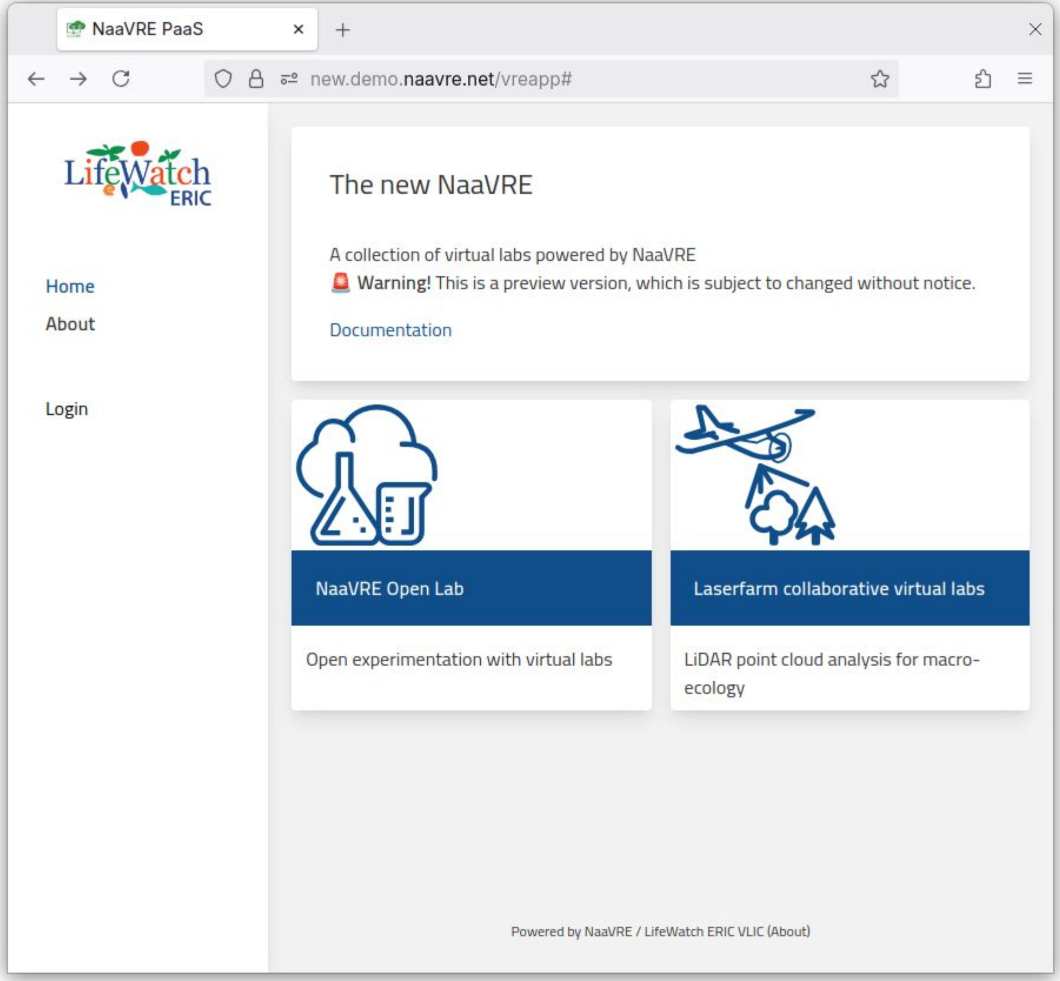


Hands-on NaaVRE

Starting point

Get started at

https://new.demo.naavre.net



The new NaaVRE

A collection of virtual labs powered by NaaVRE

🚨 **Warning!** This is a preview version, which is subject to changed without notice.

Documentation

NaaVRE Open Lab

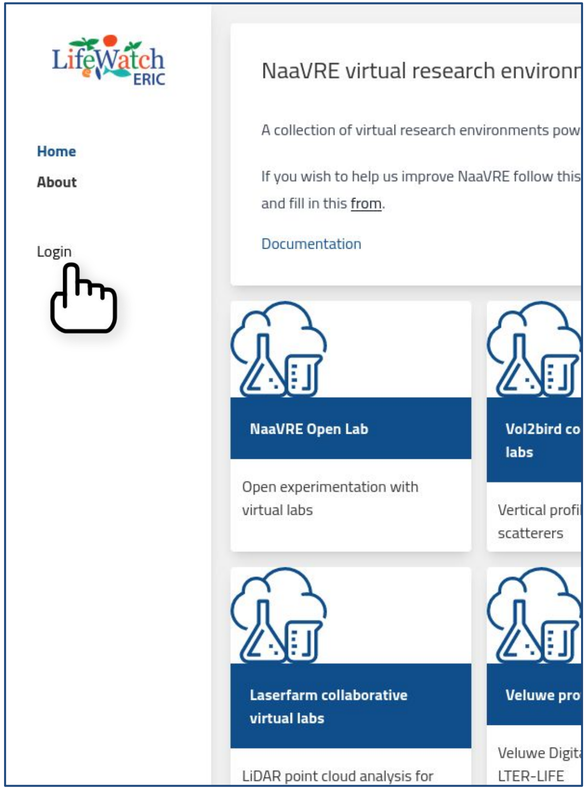
Open experimentation with virtual labs

Laserfarm collaborative virtual labs

LiDAR point cloud analysis for macro-ecology

Powered by NaaVRE / LifeWatch ERIC VLIC (About)

Sign in



Home

About

Login

NaaVRE virtual research environn

A collection of virtual research environments pow

If you wish to help us improve NaaVRE follow this and fill in this from.

Documentation

NaaVRE Open Lab

Open experimentation with virtual labs

Vol2bird co labs

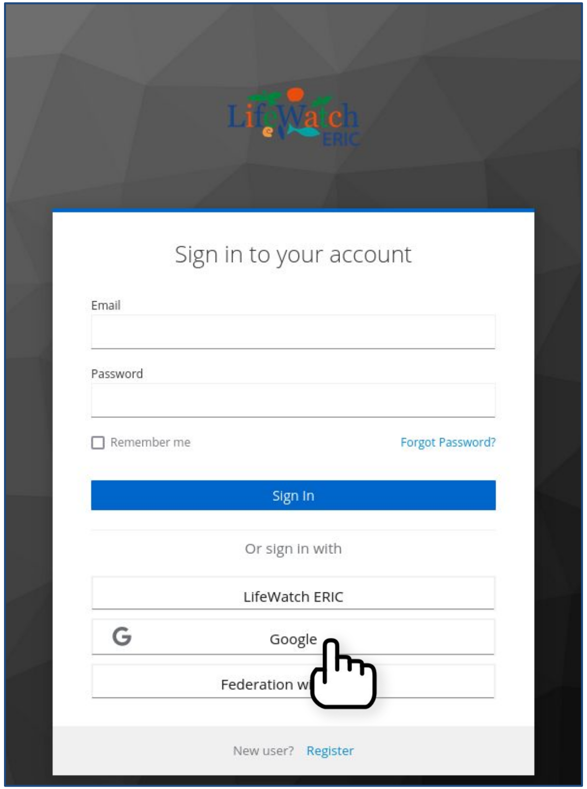
Vertical profi scatterers

Laserfarm collaborative virtual labs

LiDAR point cloud analysis for

Veluwe pro

Veluwe Digit LTER-LIFE



Sign in to your account

Email

Password

□ Remember me Forgot Password?

Sign In

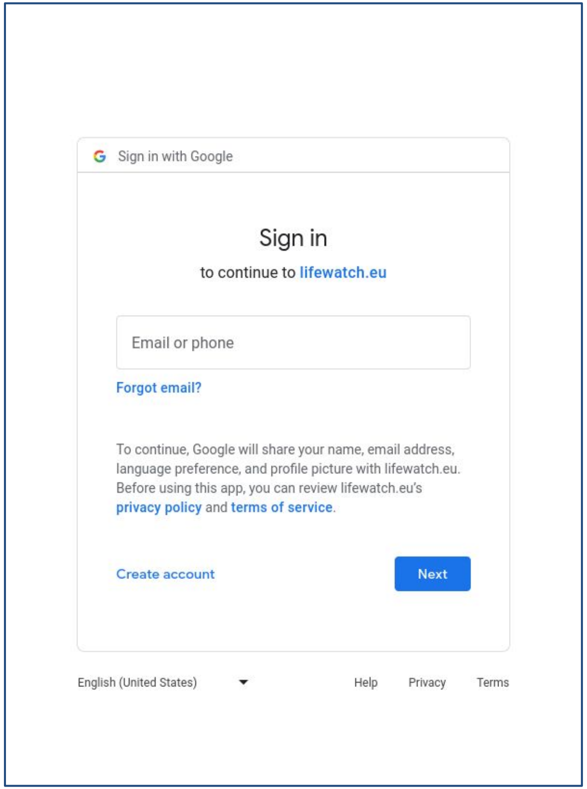
Or sign in with

LifeWatch ERIC

Google

Federation w

New user? Register



Sign in with Google

Sign in

to continue to lifewatch.eu

Email or phone

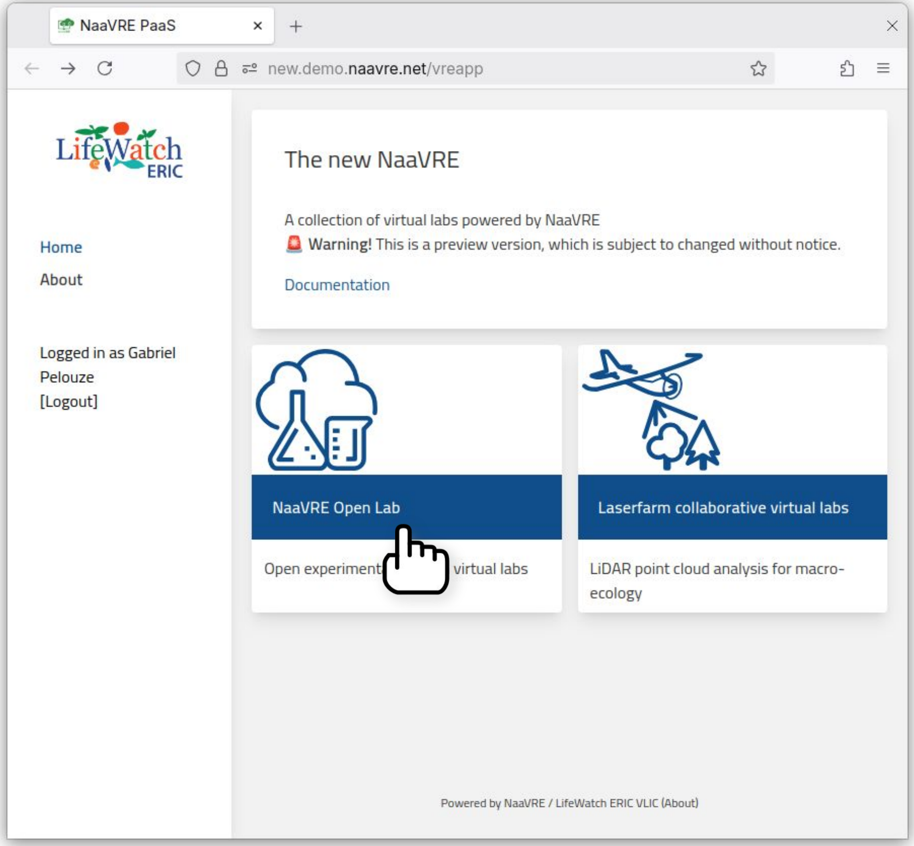
Forgot email?

To continue, Google will share your name, email address, language preference, and profile picture with lifewatch.eu. Before using this app, you can review lifewatch.eu's privacy policy and terms of service.

Create account Next

English (United States) Help Privacy Terms

Pick your collaborative virtual labs



NaaVRE PaaS
new.demo.naavre.net/vreapp
LifeWatch ERIC
Home
About
Logged in as Gabriel Pelouze
[Logout]
NaaVRE Open Lab
Laserfarm collaborative virtual labs
Documentation

The new NaaVRE

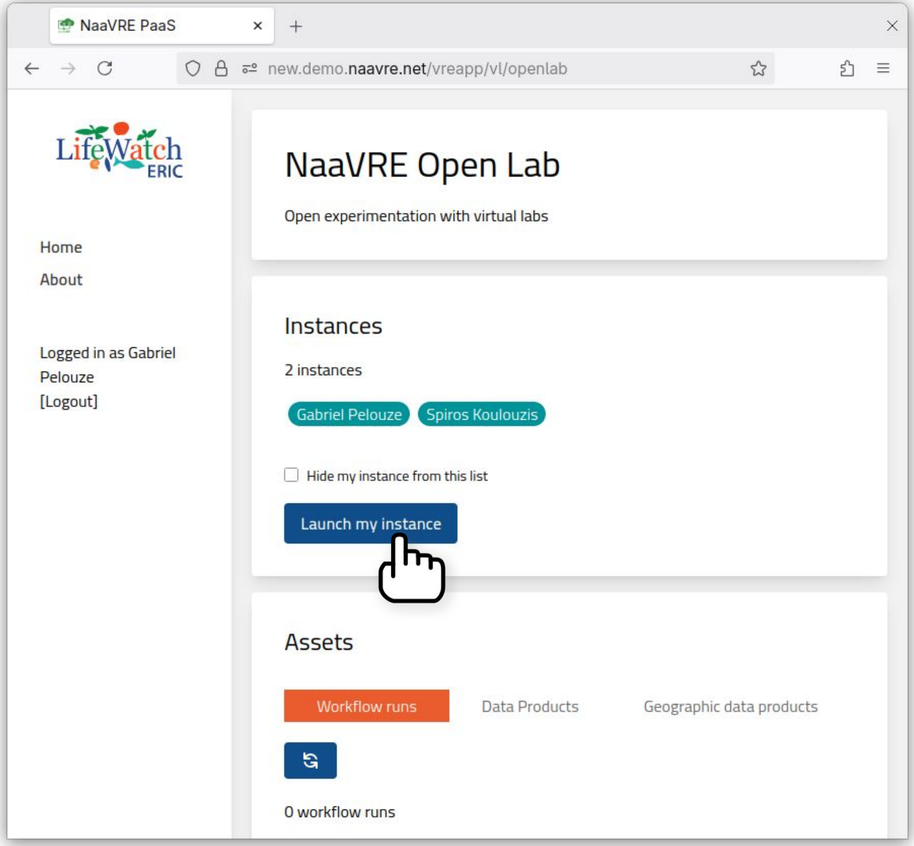
A collection of virtual labs powered by NaaVRE

🚨 **Warning!** This is a preview version, which is subject to changed without notice.

Open experiment… virtual labs

LiDAR point cloud analysis for macro-ecology

Powered by NaaVRE / LifeWatch ERIC VLIC (About)



NaaVRE Open Lab

Open experimentation with virtual labs

Instances

2 instances

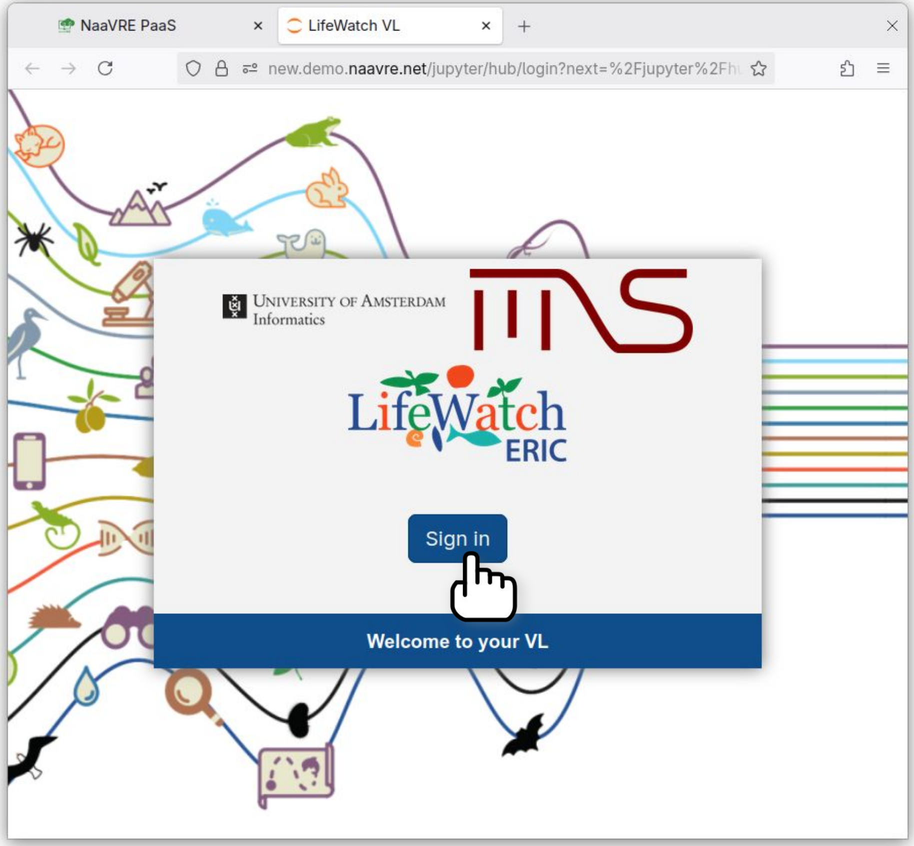
□ Hide my instance from this list

Assets

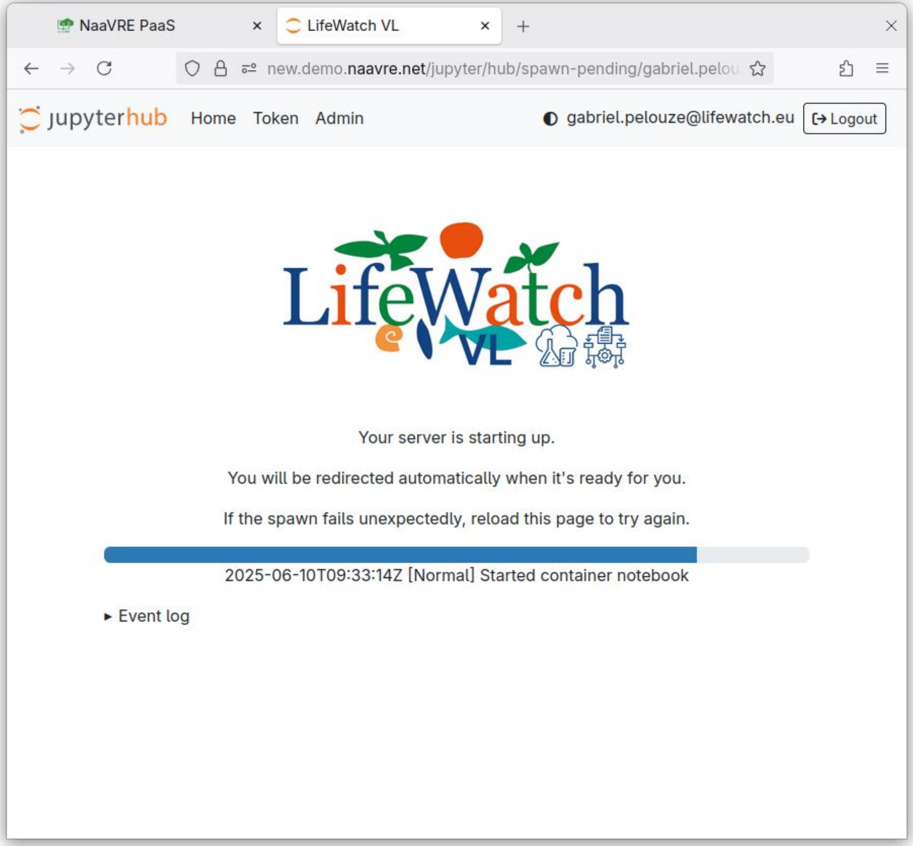
0 workflow runs

🚀 Starting your virtual lab

Pick your collaborative virtual labs



Welcome to your VL


NaaVRE PaaS
LifeWatch VL
new.demo.naavre.net/jupyter/hub/spawn-pending/gabriel.pelou
jupyterhub
Home
Token
Admin
gabriel.pelouze@lifewatch.eu
Logout
LifeWatch
VL

Your server is starting up.

You will be redirected automatically when it's ready for you.

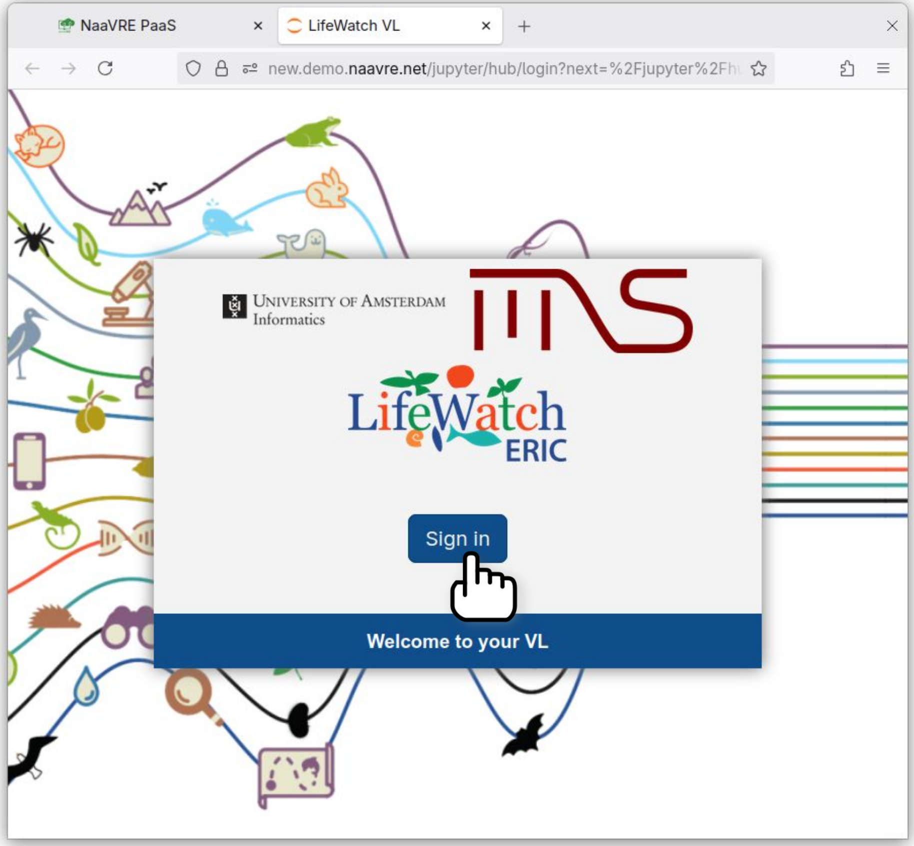
If the spawn fails unexpectedly, reload this page to try again.

```
2025-06-10T09:33:14Z [Normal] Started container notebook
```

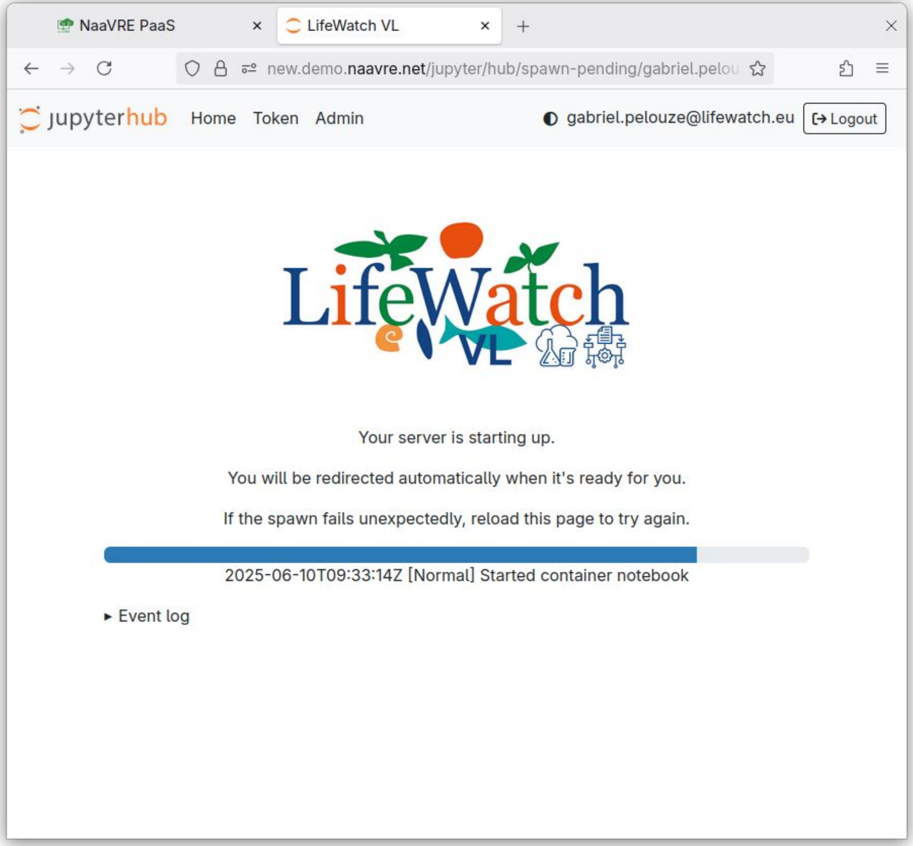
▸ Event log

🚀 Starting your virtual lab

Start your virtual lab



NaaVRE PaaS
LifeWatch VL
new.demo.naavre.net/jupyter/hub/login?next=%2Fjupyter%2Fh
University of Amsterdam Informatics
LifeWatch ERIC
Sign in
Welcome to your VL



NaaVRE PaaS
LifeWatch VL
new.demo.naavre.net/jupyter/hub/spawn-pending/gabriel.pelou
jupyterhub
Home
Token
Admin
gabriel.pelouze@lifewatch.eu
Logout
LifeWatch VL

Your server is starting up.

You will be redirected automatically when it's ready for you.

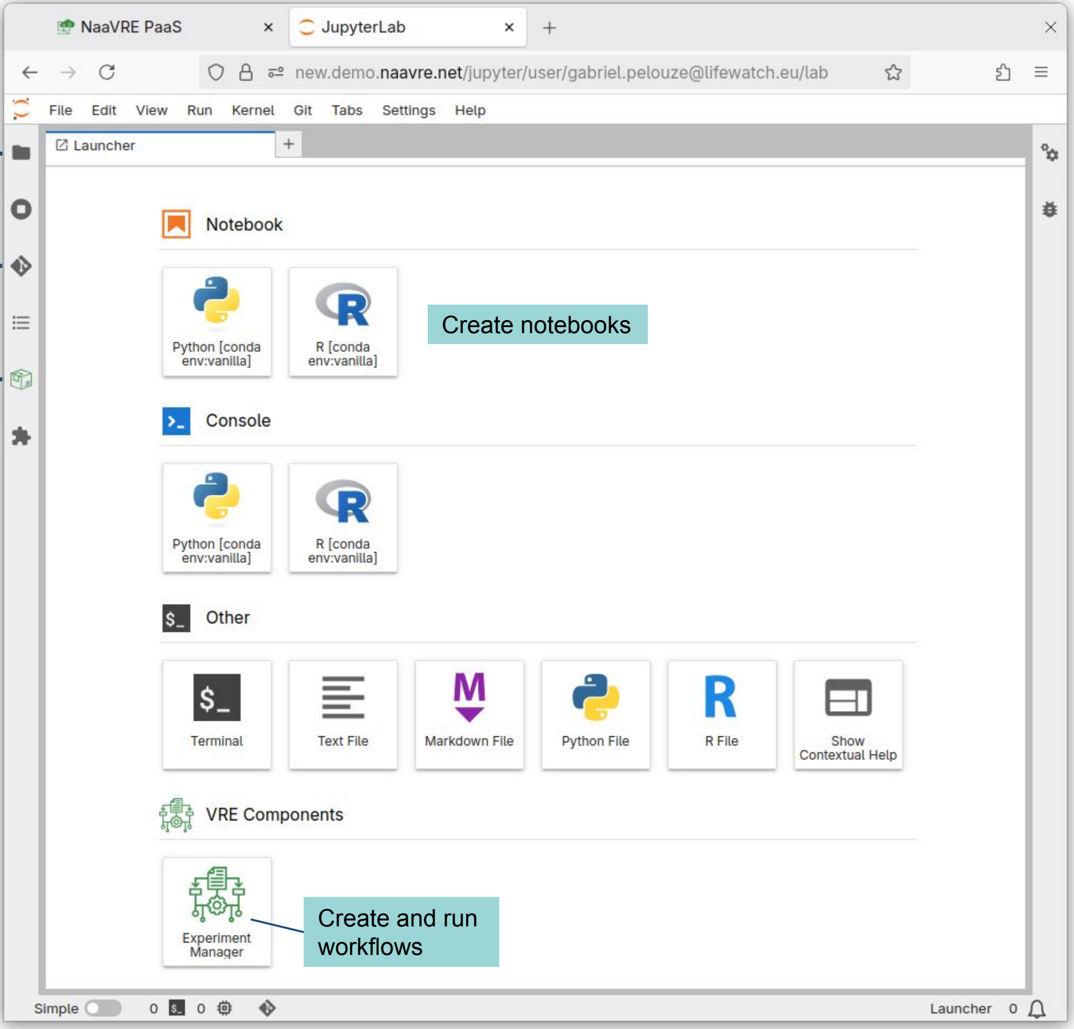
If the spawn fails unexpectedly, reload this page to try again.

```
2025-06-10T09:33:14Z [Normal] Started container notebook
```

▸ Event log

🚀 Starting your virtual lab

The NaaVRE interface



Manage local files

Manage your work with Git

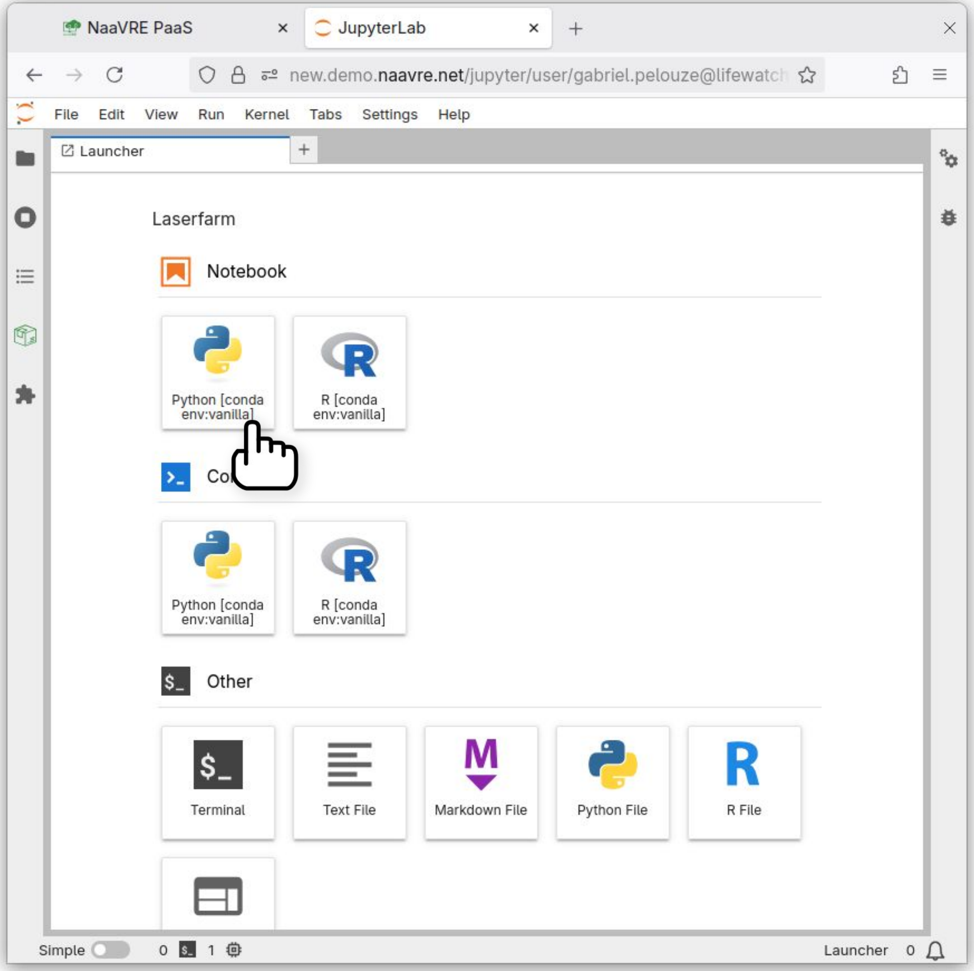
Containerize cells

Create notebooks

Create and run workflows

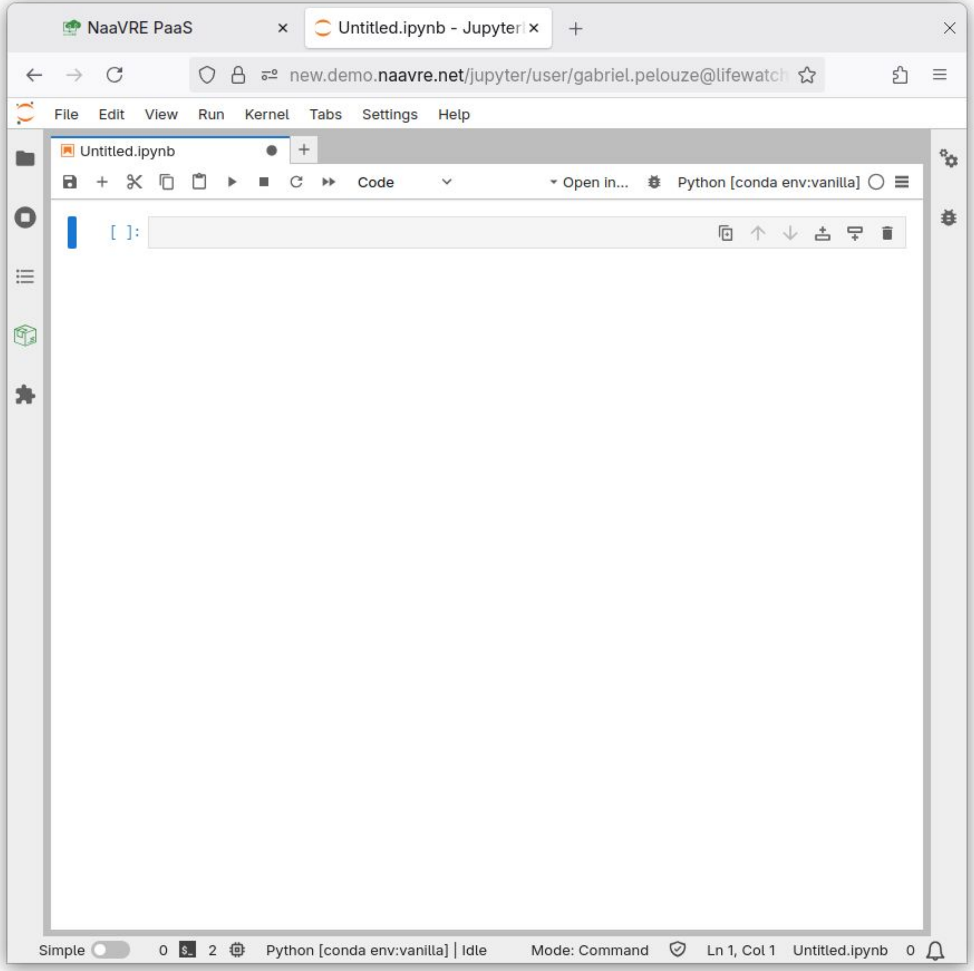
Create a notebook

1. Create a new notebook



Create a notebook

1. Create a new notebook



Create a notebook

1. Create a new notebook
2. Write some code

In Python

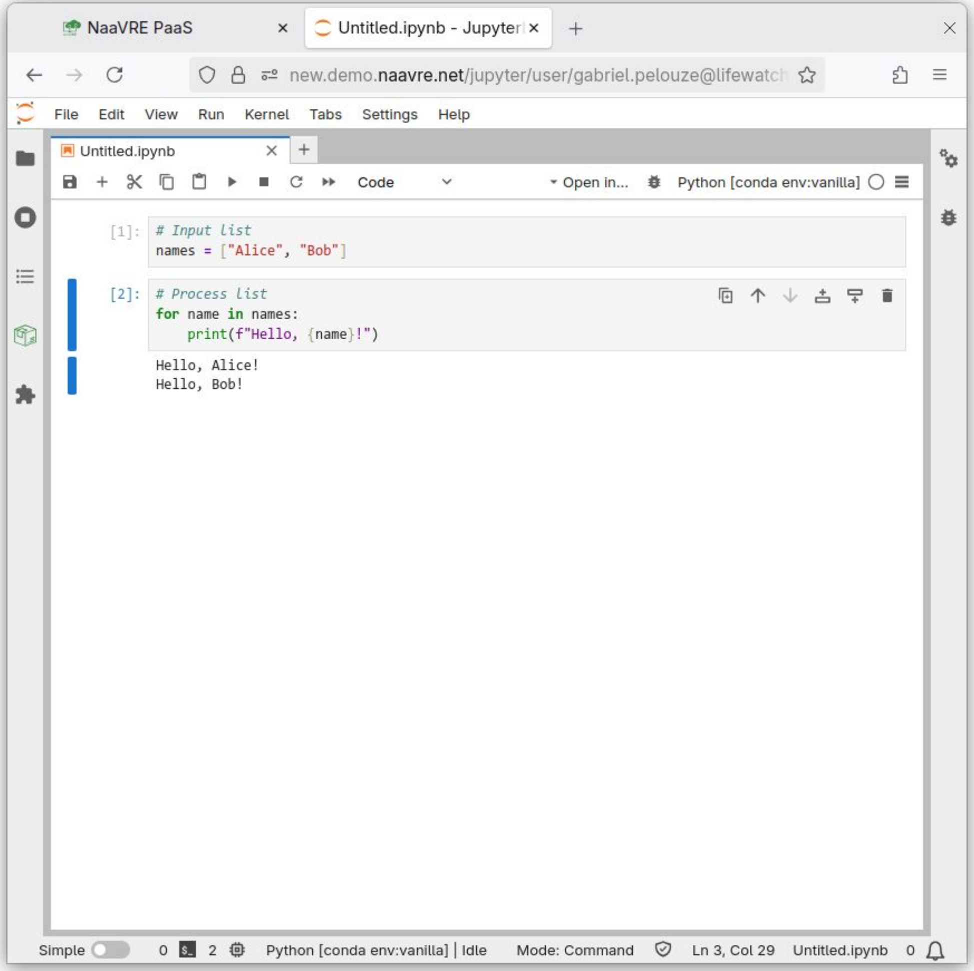
```python
# Input list
names = ["Alice", "Bob"]
```

```python
# Process list
for name in names:
    print(f"Hello, {name}!")
```

In R

```r
# Input list
names = list("Alice", "Bob")
```

```r
# Process list
for (name in names) {
    print(sprintf("Hello, %s!", name))
}
```

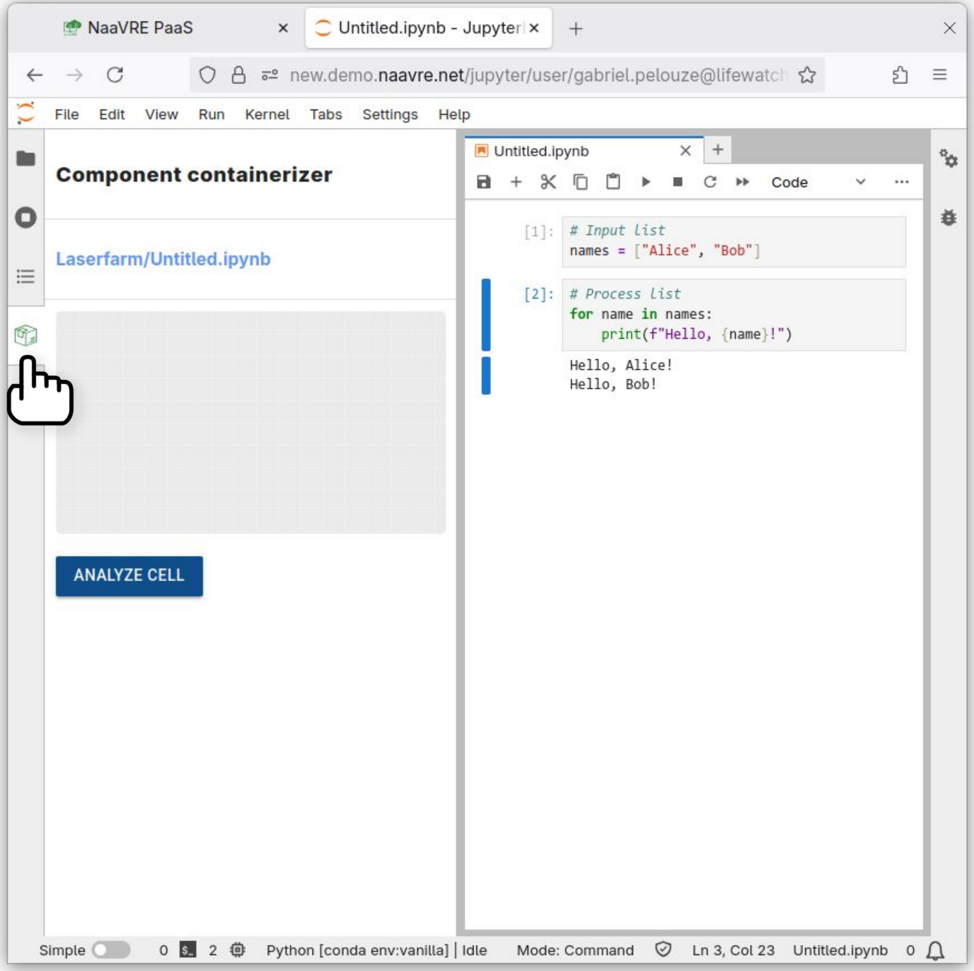

NaaVRE PaaS
Untitled.ipynb - Jupyter
new.demo.naavre.net/jupyter/user/gabriel.pelouze@lifewatch
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb
Code
Open in...
Python [conda env:vanilla]

```
[1]: # Input list
     names = ["Alice", "Bob"]

[2]: # Process list
     for name in names:
         print(f"Hello, {name}!")

     Hello, Alice!
     Hello, Bob!
```

Simple 0 2 Python [conda env:vanilla] | Idle Mode: Command Ln 3, Col 29 Untitled.ipynb 0

Create a notebook

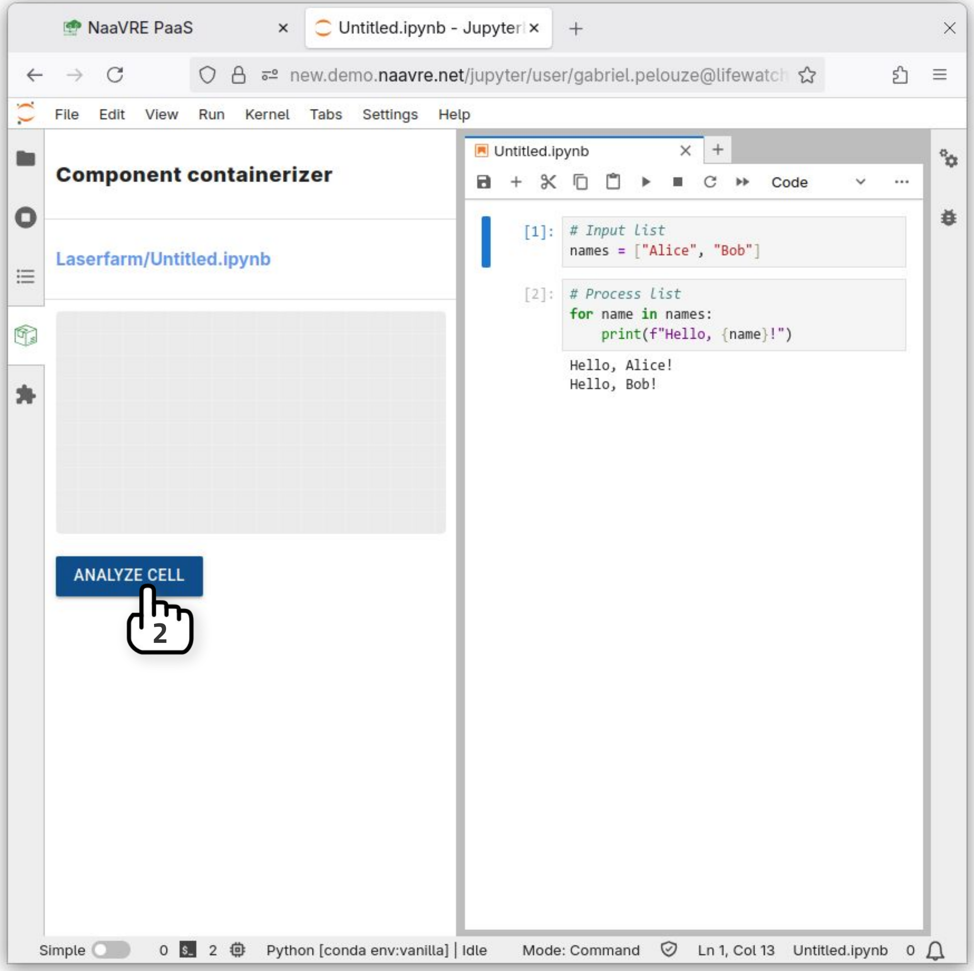
1. Create a new notebook
2. Write some code
3. Containerize the cells
   a. open the component containerizer



```
[1]: # Input list
     names = ["Alice", "Bob"]

[2]: # Process list
     for name in names:
         print(f"Hello, {name}!")

     Hello, Alice!
     Hello, Bob!
```

Create a notebook

1. Create a new notebook
2. Write some code
3. Containerize the cells
   a. open the component containerizer
   b. select the cell you want to containerize and click on “Analyze Cell”

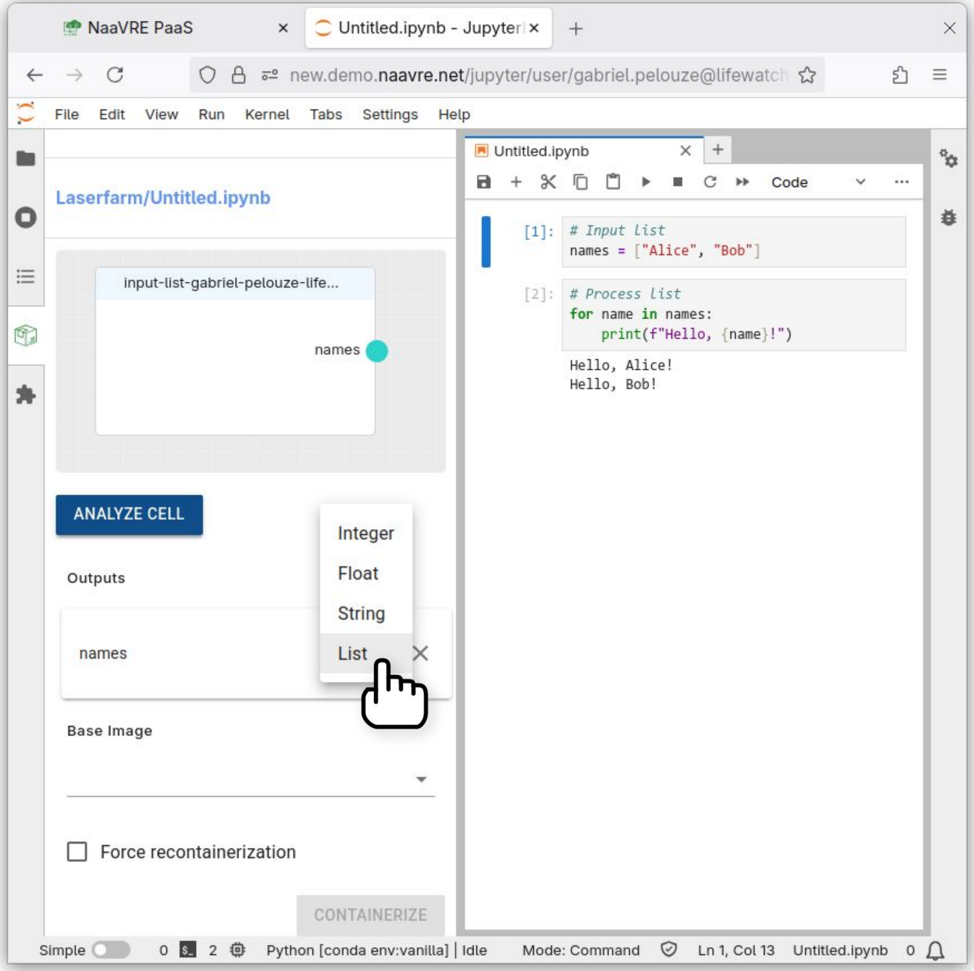


```
[1]: # Input list
     names = ["Alice", "Bob"]

[2]: # Process list
     for name in names:
         print(f"Hello, {name}!")

     Hello, Alice!
     Hello, Bob!
```

Create a notebook

1. Create a new notebook
2. Write some code
3. Containerize the cells
   a. open the component containerizer
   b. select the cell you want to containerize
   c. if needed, adjust the variable types

In R	List
In Python	List *(set automatically)*



□ Force recontainerization

```
[1]: # Input list
     names = ["Alice", "Bob"]
```

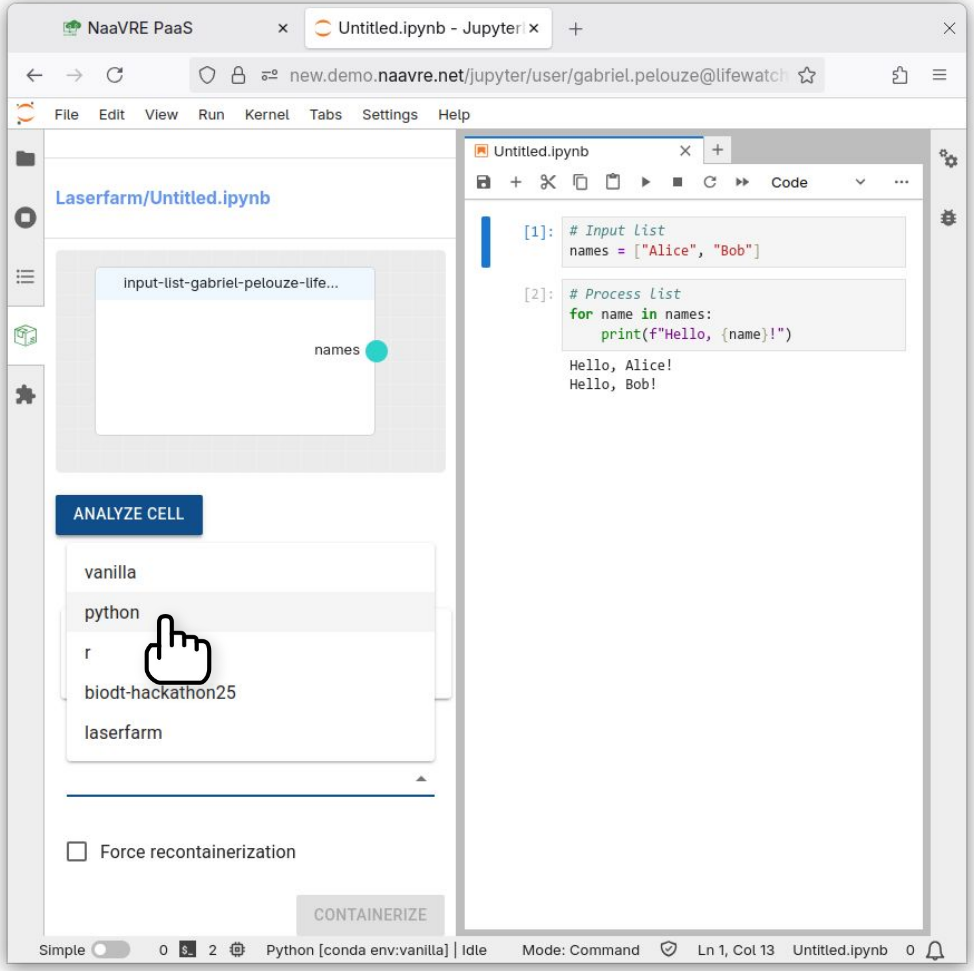
```
[2]: # Process list
     for name in names:
         print(f"Hello, {name}!")
```

```
Hello, Alice!
Hello, Bob!
```

Create a notebook

1. Create a new notebook
2. Write some code
3. Containerize the cells
   a. open the component containerizer
   b. select the cell you want to containerize
   c. if needed, adjust the variable types
   d. select the base image

In R	r
In Python	python



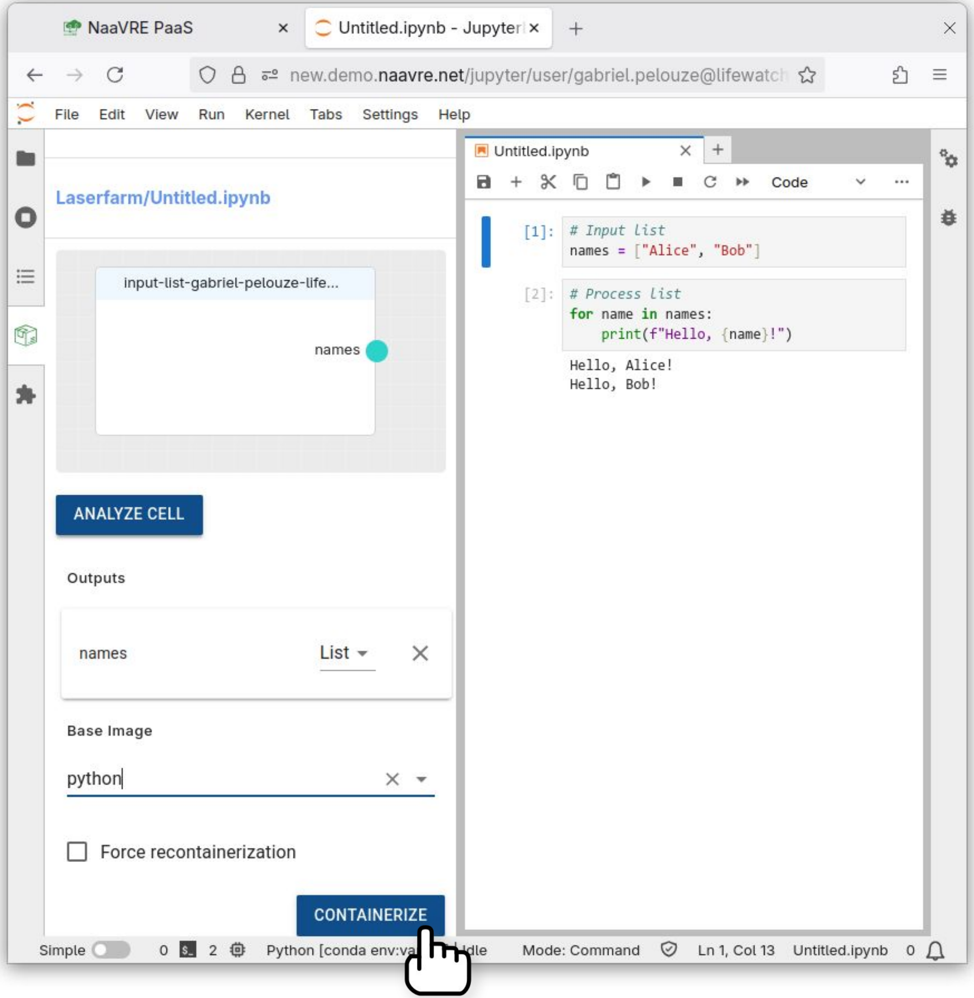
```
[1]: # Input list
names = ["Alice", "Bob"]
```

```
[2]: # Process list
for name in names:
    print(f"Hello, {name}!")
```

```
Hello, Alice!
Hello, Bob!
```

Create a notebook

1. Create a new notebook
2. Write some code
3. Containerize the cells
    a. open the component containerizer
    b. select the cell you want to containerize
    c. if needed, adjust the variable types
    d. select the base image
    e. containerize the cell



Outputs

names List

Base Image

python

□ Force recontainerization

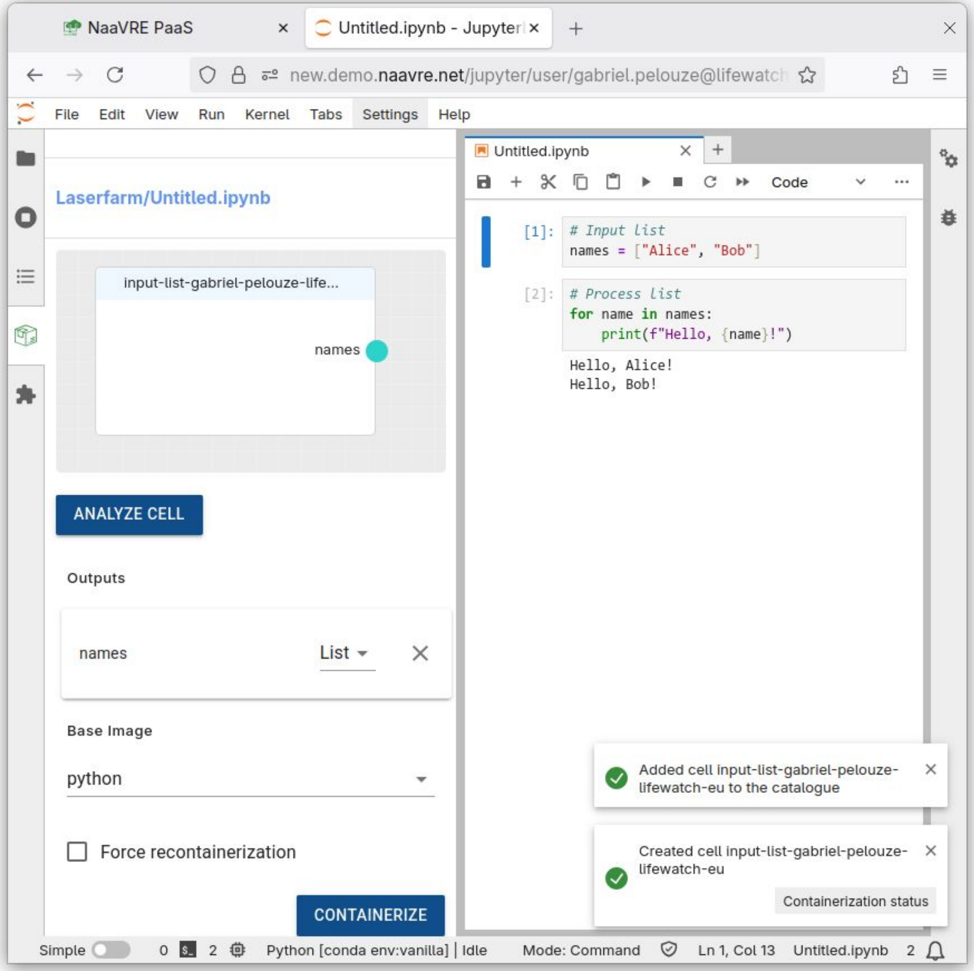
```
[1]: # Input list
names = ["Alice", "Bob"]
```

```
[2]: # Process list
for name in names:
    print(f"Hello, {name}!")
```

```
Hello, Alice!
Hello, Bob!
```

Create a notebook

1. Create a new notebook
2. Write some code
3. Containerize the cells
   a. open the component containerizer
   b. select the cell you want to containerize
   c. if needed, adjust the variable types
   d. select the base image
   e. containerize the cell
   f. well done!
   g. repeat steps b–e for other cells



Outputs

names List

Base Image

python

□ Force recontainerization

```
[1]: # Input list
names = ["Alice", "Bob"]

[2]: # Process list
for name in names:
    print(f"Hello, {name}!")

Hello, Alice!
Hello, Bob!
```

Added cell input-list-gabriel-pelouze-lifewatch-eu to the catalogue

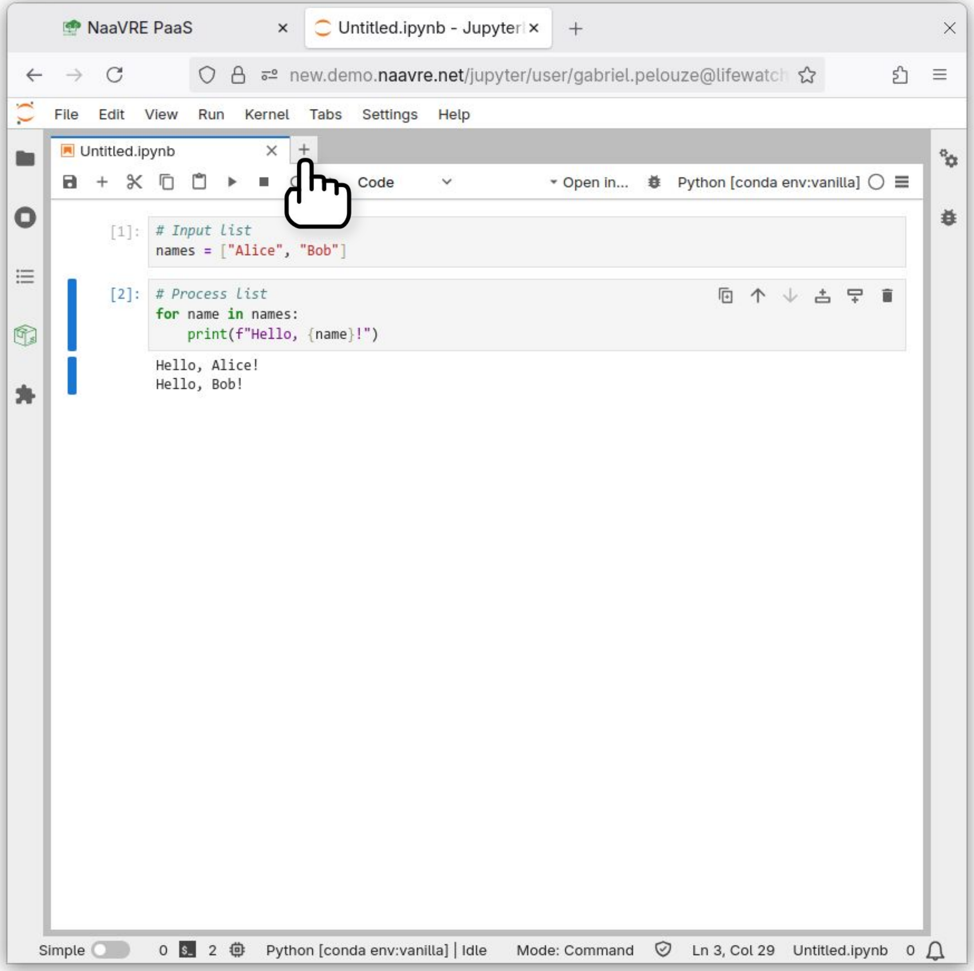
Created cell input-list-gabriel-pelouze-lifewatch-eu

Containerization status

Simple 0 2 Python [conda env:vanilla] | Idle Mode: Command Ln 1, Col 13 Untitled.ipynb 2

Create a workflow

1. Open the experiment manager



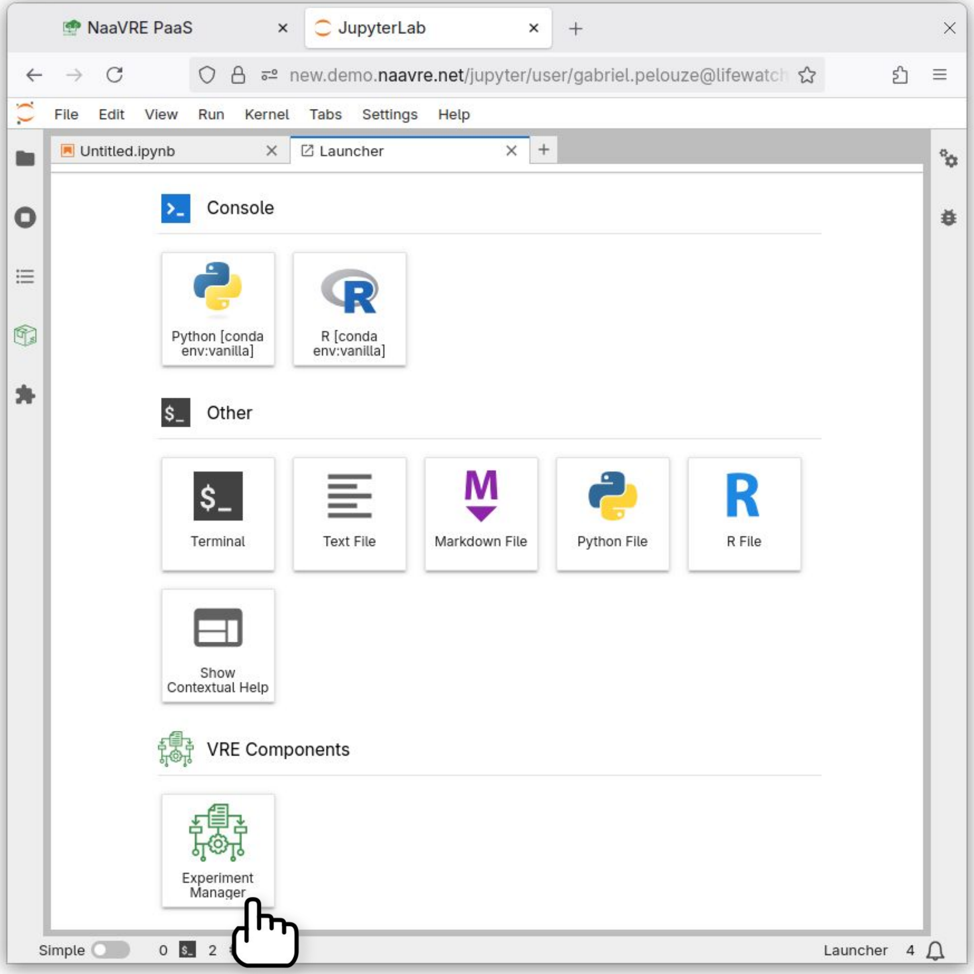
```
[1]: # Input list
     names = ["Alice", "Bob"]
```

```
[2]: # Process list
     for name in names:
         print(f"Hello, {name}!")
```

```
Hello, Alice!
Hello, Bob!
```

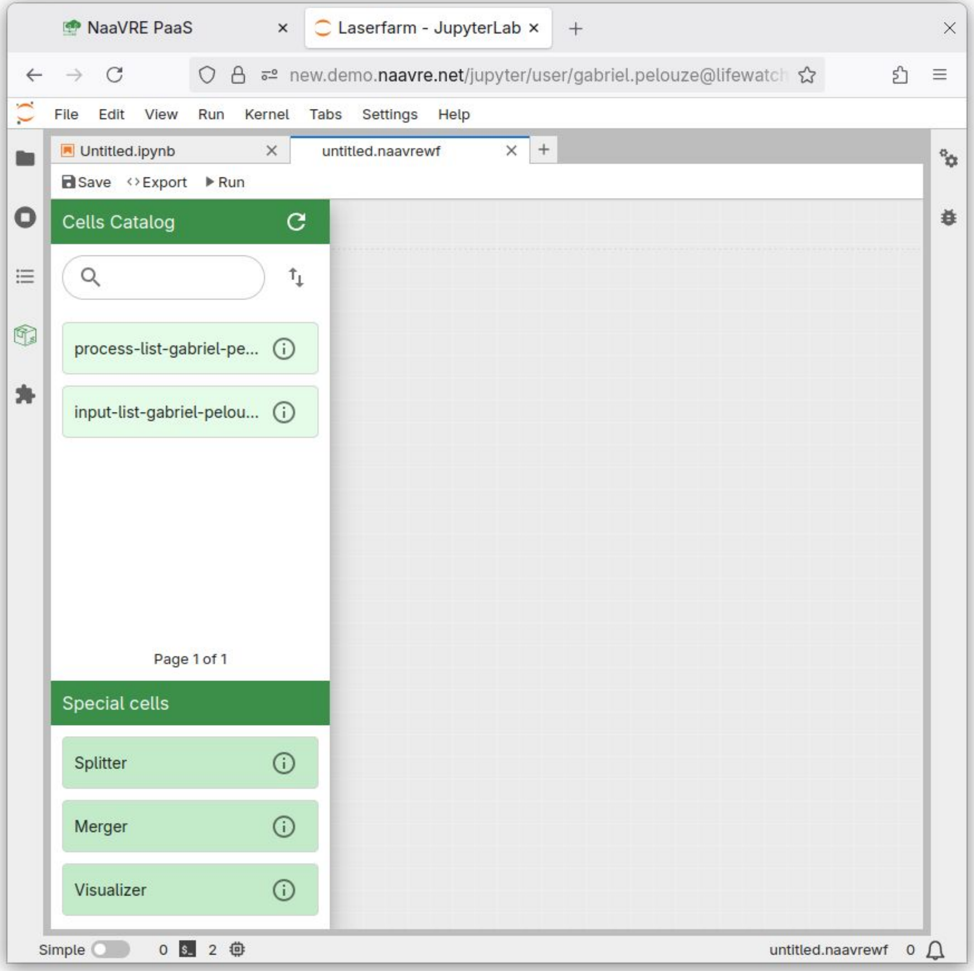
Create a workflow

1. Open the experiment manager



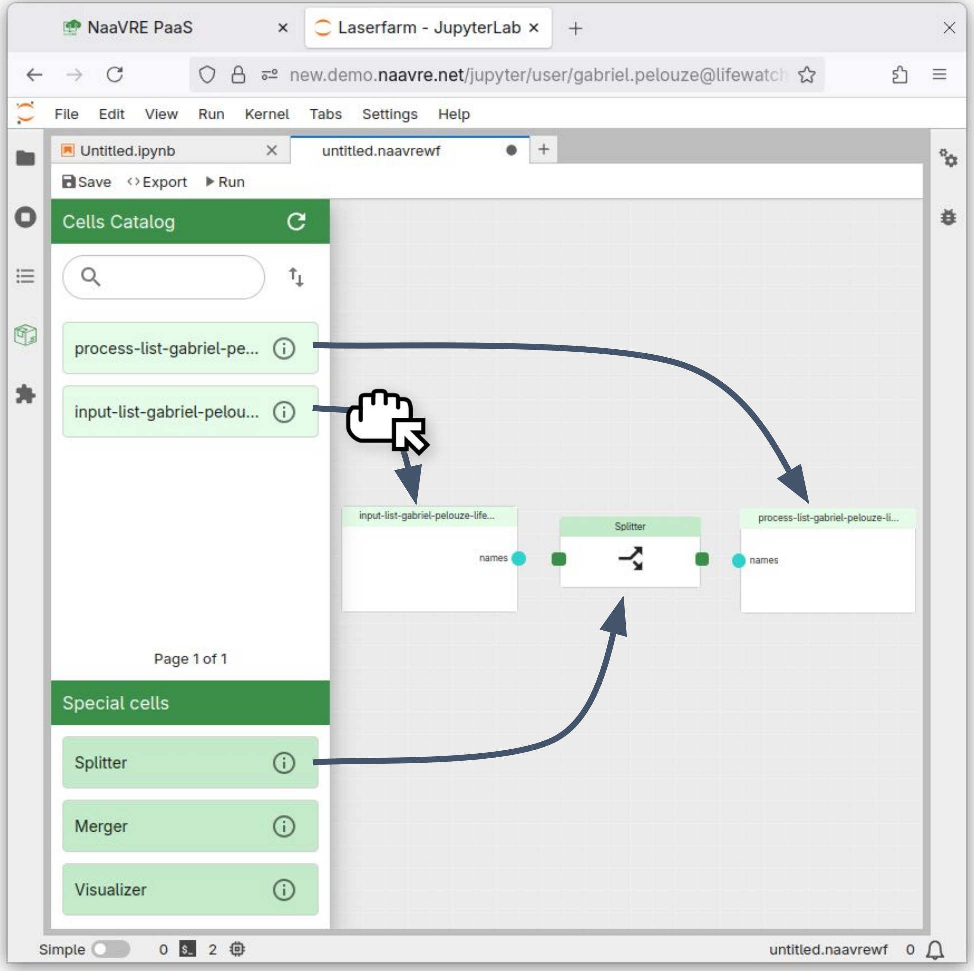
Create a workflow

1. Open the experiment manager



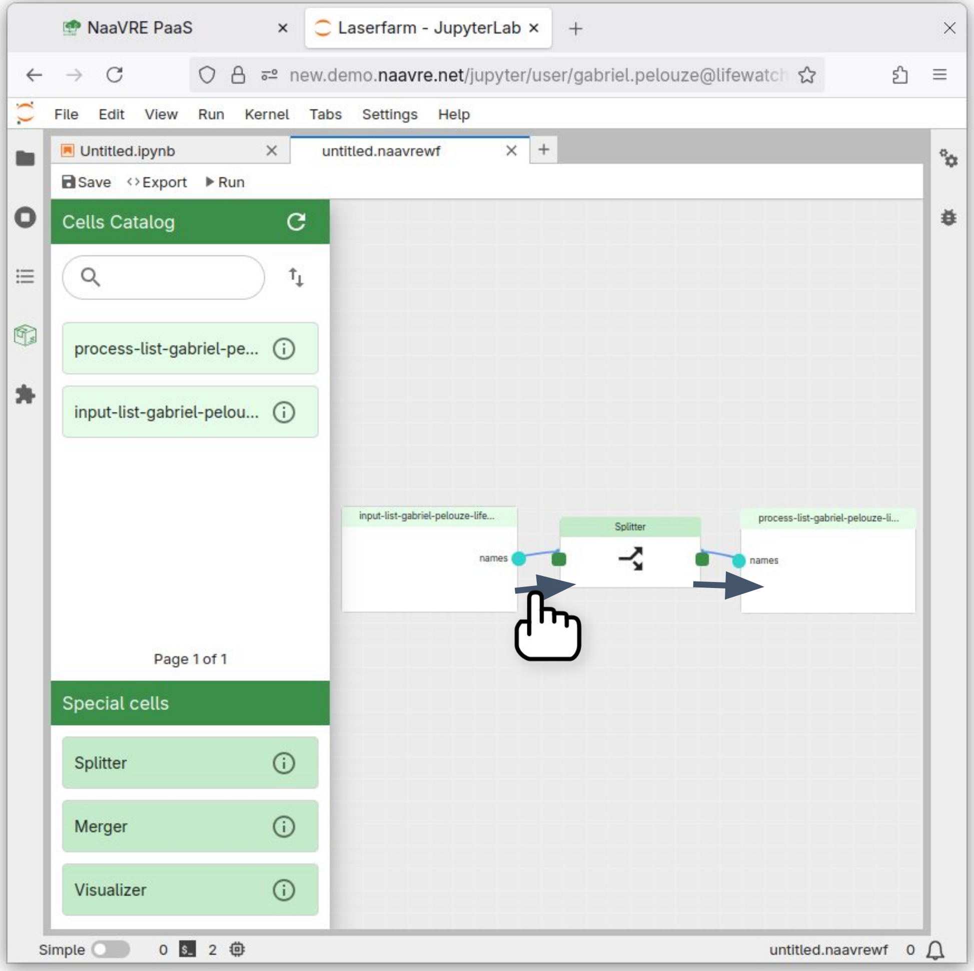
Create a workflow

1. Open the experiment manager
2. Drag and drop the cells onto the canvas



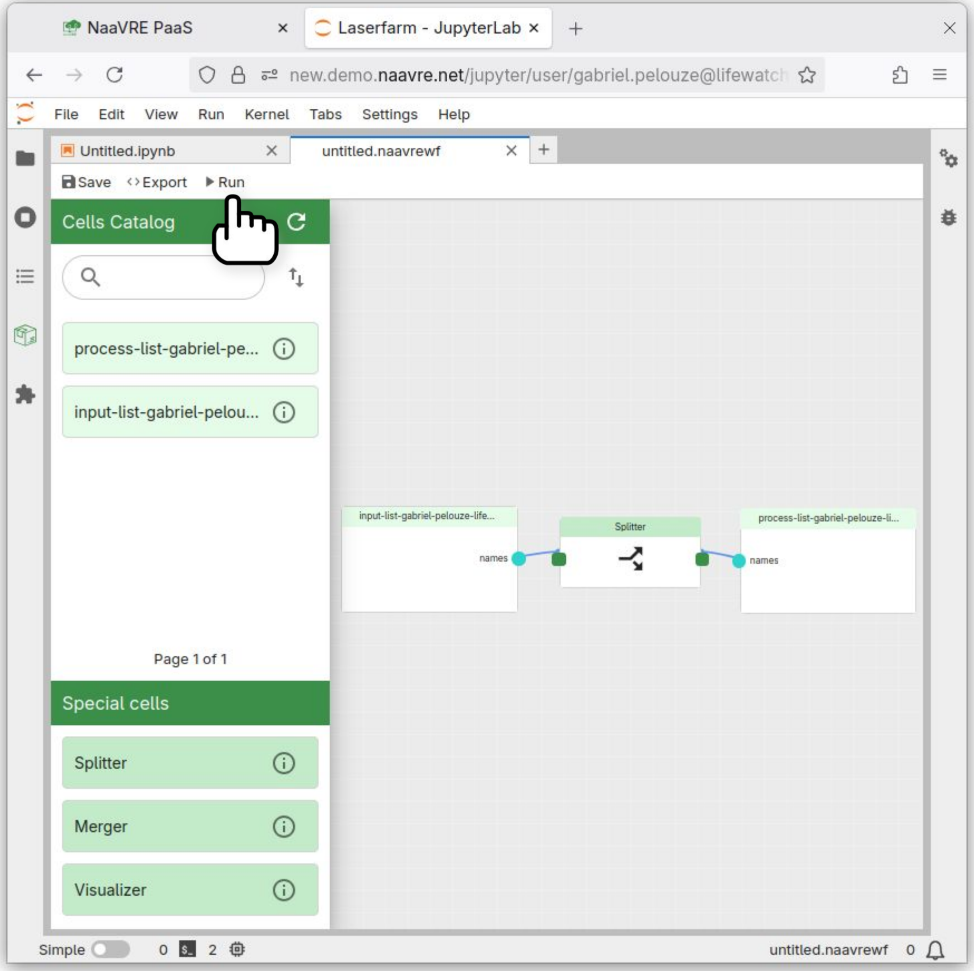
Create a workflow

1. Open the experiment manager
2. Drag and drop the cells onto the canvas
3. Connect the cells



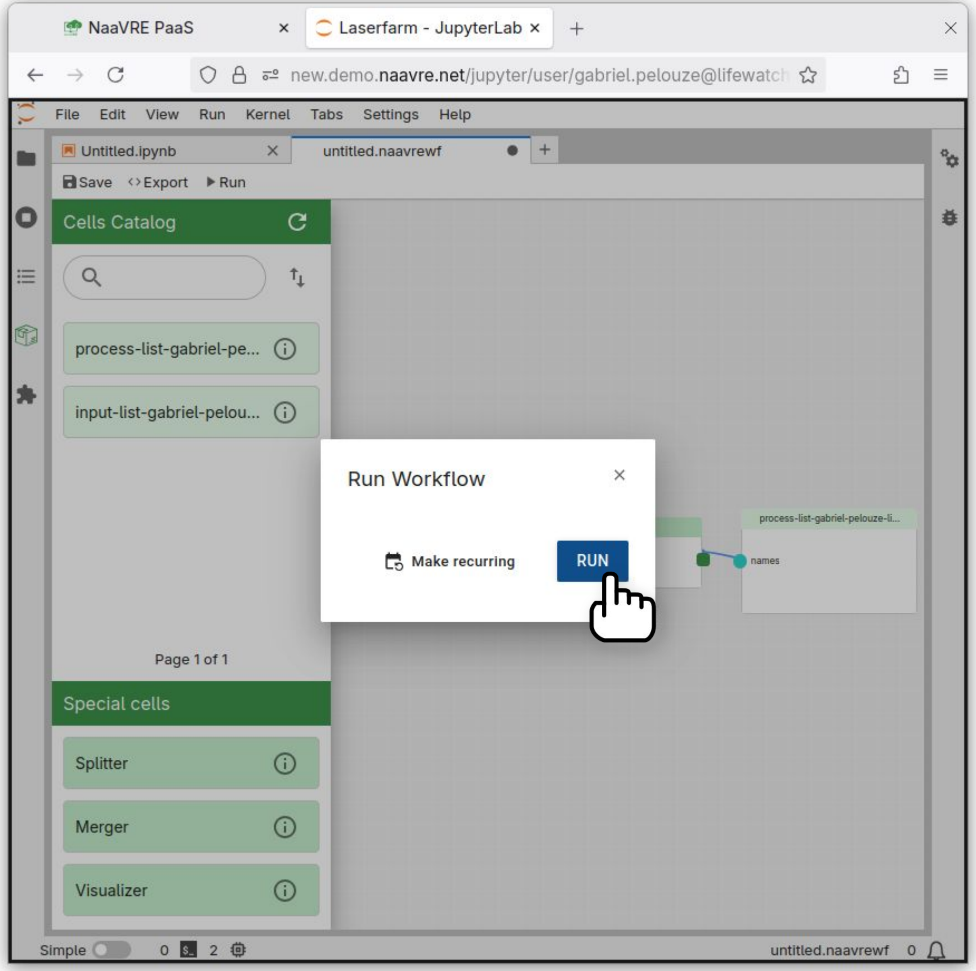
Create a workflow

1. Open the experiment manager
2. Drag and drop the cells onto the canvas
3. Connect the cells
4. Execute the workflow



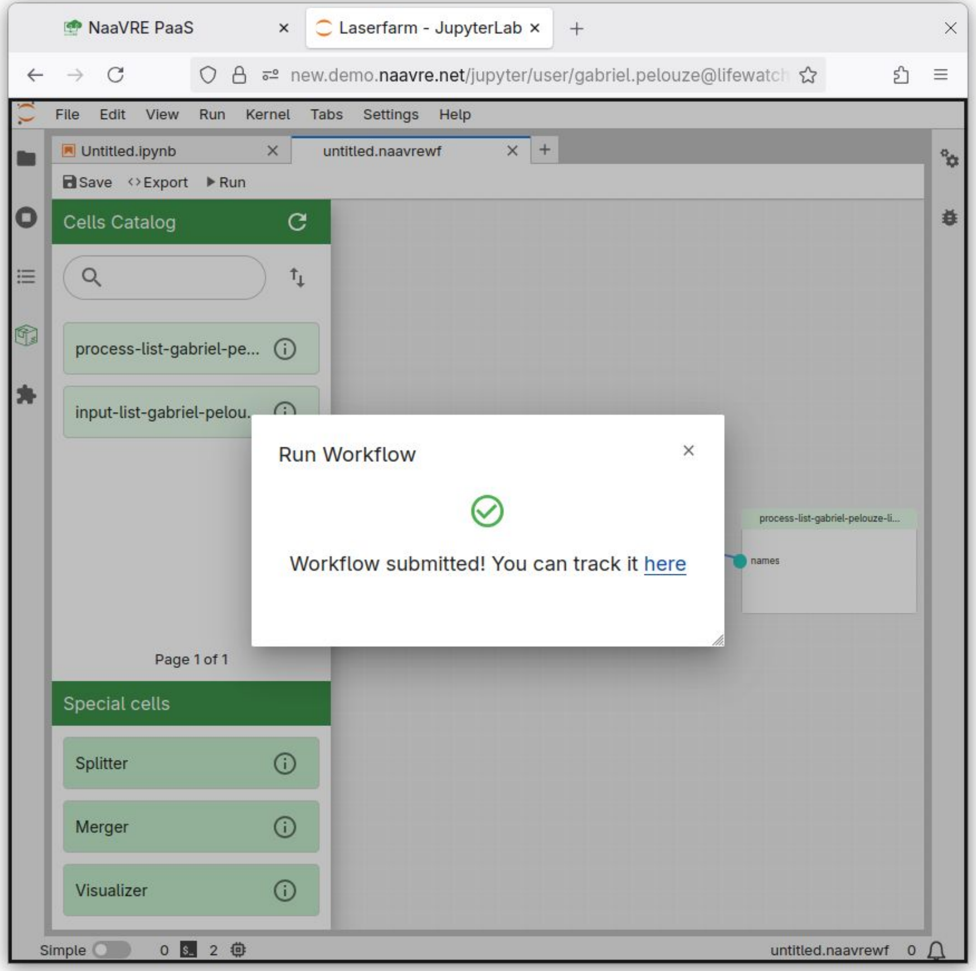
Create a workflow

1. Open the experiment manager
2. Drag and drop the cells onto the canvas
3. Connect the cells
4. Execute the workflow



Create a workflow

1. Open the experiment manager
2. Drag and drop the cells onto the canvas
3. Connect the cells
4. Execute the workflow
5. Well done!

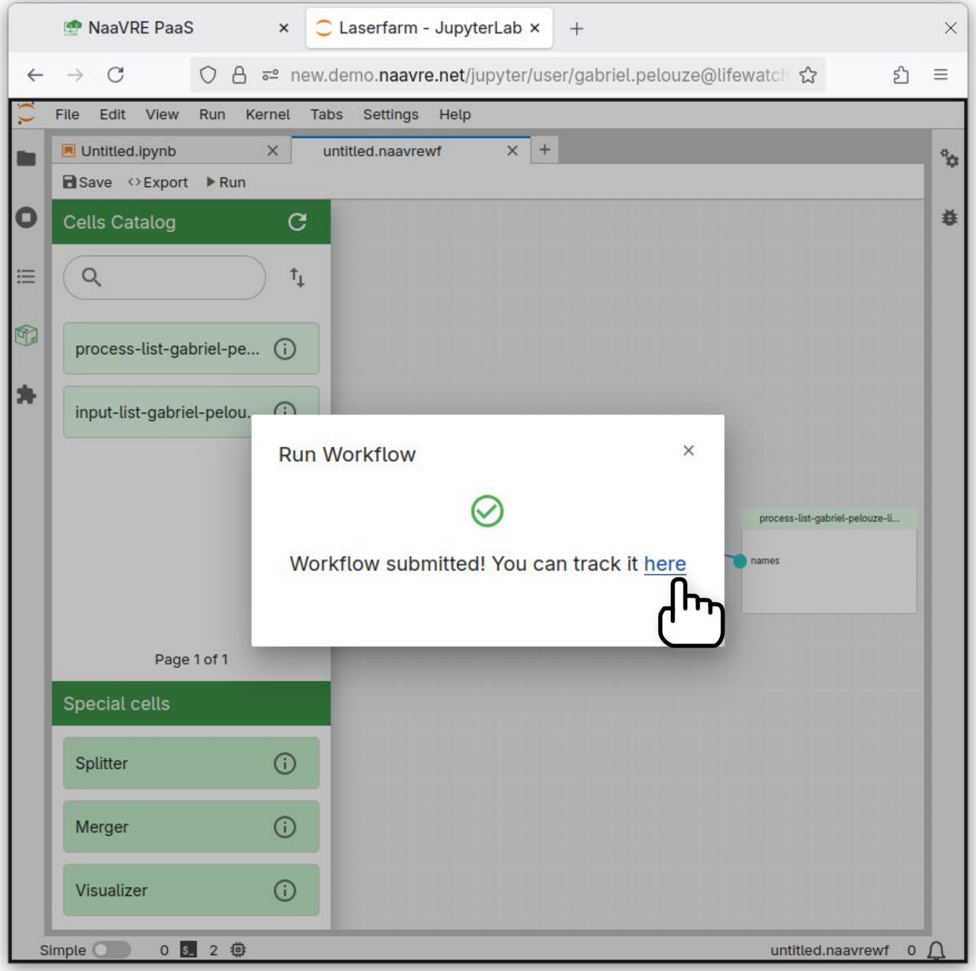


Run Workflow

Workflow submitted! You can track it here

Monitor a workflow

1. Click on the link to open the workflow manager, Argo
2. If needed, log in to argo



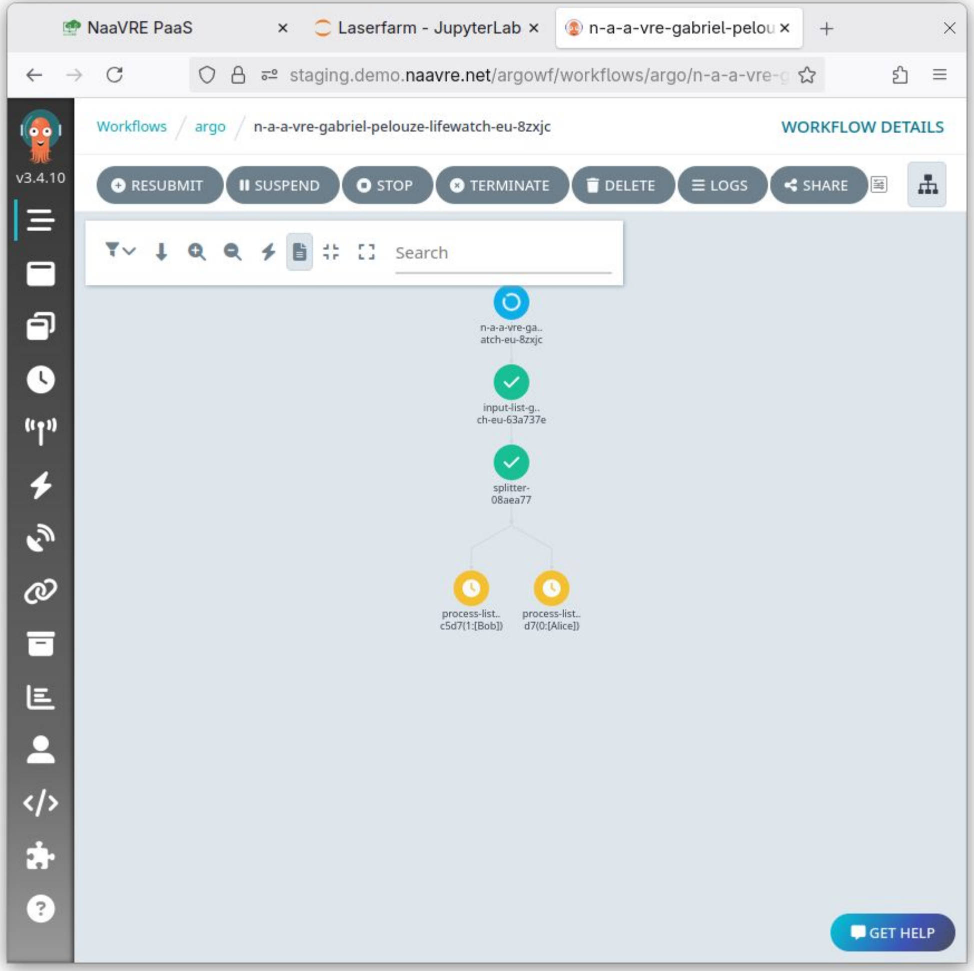
NaaVRE PaaS
Laserfarm - JupyterLab
new.demo.naavre.net/jupyter/user/gabriel.pelouze@lifewatch
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb
untitled.naavrewf
Save
Export
Run
Cells Catalog
process-list-gabriel-pe...
input-list-gabriel-pelou...
Page 1 of 1
Special cells
Splitter
Merger
Visualizer
process-list-gabriel-pelouze-li...
names
Simple
0
2
untitled.naavrewf
0

Run Workflow

Workflow submitted! You can track it here

Monitor a workflow

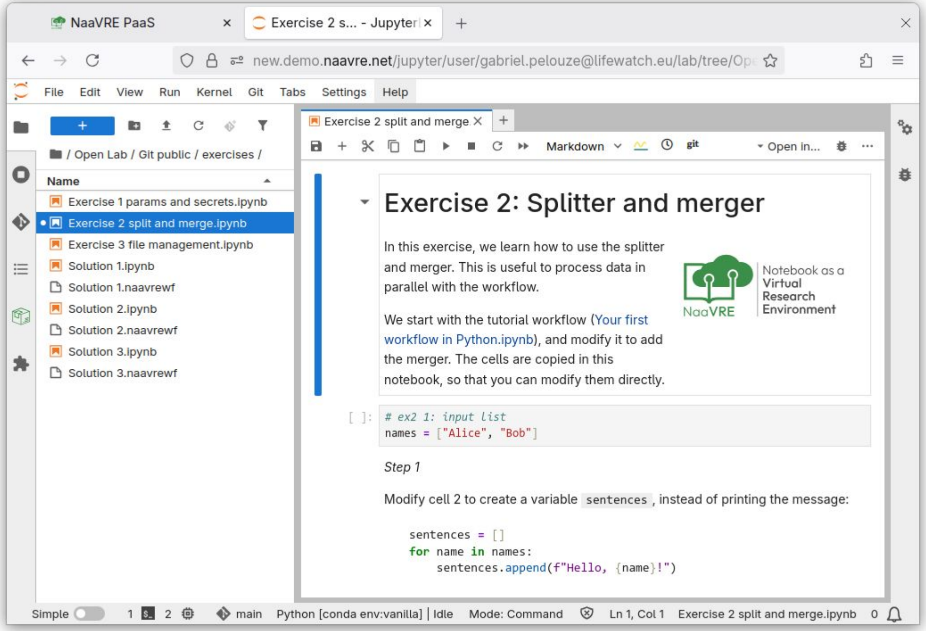
1. Click on the link to open the workflow manager, Argo
2. If needed, log in to argo
3. Check your workflow status


NaaVRE PaaS
Laserfarm - JupyterLab
n-a-a-vre-gabriel-pelou
staging.demo.naavre.net/argowf/workflows/argo/n-a-a-vre-g
v3.4.10
Workflows / argo / n-a-a-vre-gabriel-pelouze-lifewatch-eu-8zxjc
WORKFLOW DETAILS
RESUBMIT
SUSPEND
STOP
TERMINATE
DELETE
LOGS
SHARE
Search
n-a-a-vre-ga.. atch-eu-8zxjc
input-list-g.. ch-eu-63a737e
splitter-08aea77
process-list.. c5d7(1:[Bob])
process-list.. d7(0:[Alice])
GET HELP

Exercises

Start the **Open Lab** and the exercises in `Open Lab/Git public/exercises/` :

- Exercise 1: Workflow parameters and secrets
- Exercise 2: Splitter and merger to create parallel workflows
- Exercise 3: File management


NaaVRE PaaS
Exercise 2 s... - Jupyter
new.demo.naavre.net/jupyter/user/gabriel.pelouze@lifewatch.eu/lab/tree/Op
File Edit View Run Kernel Git Tabs Settings Help
/ Open Lab / Git public / exercises /
Name
Exercise 2 split and merge
Markdown
Open in...
NaaVRE
Notebook as a Virtual Research Environment
Simple 1 2 main Python [conda env:vanilla] | Idle Mode: Command Ln 1, Col 1 Exercise 2 split and merge.ipynb 0

Exercise 1 params and secrets.ipynb
Exercise 2 split and merge.ipynb
Exercise 3 file management.ipynb
Solution 1.ipynb
Solution 1.naavrewf
Solution 2.ipynb
Solution 2.naavrewf
Solution 3.ipynb
Solution 3.naavrewf

Exercise 2: Splitter and merger

In this exercise, we learn how to use the splitter and merger. This is useful to process data in parallel with the workflow.

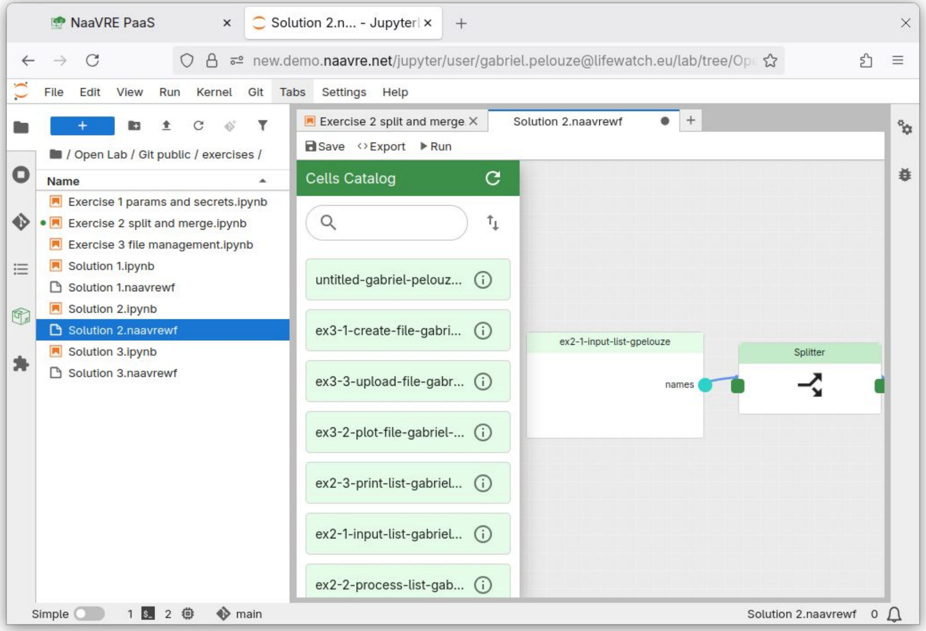
We start with the tutorial workflow (Your first workflow in Python.ipynb), and modify it to add the merger. The cells are copied in this notebook, so that you can modify them directly.

```
[ ]: # ex2 1: input list
     names = ["Alice", "Bob"]
```

Step 1

Modify cell 2 to create a variable `sentences`, instead of printing the message:

```
sentences = []
for name in names:
    sentences.append(f"Hello, {name}!")
```

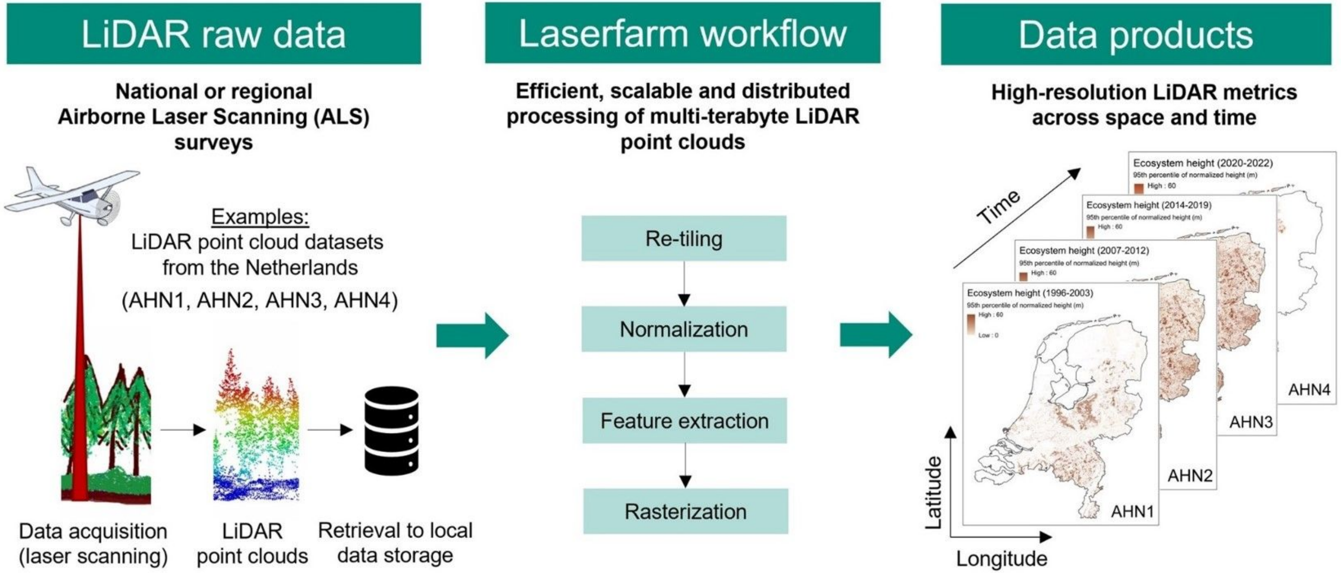

NaaVRE PaaS
Solution 2.n... - Jupyter
new.demo.naavre.net/jupyter/user/gabriel.pelouze@lifewatch.eu/lab/tree/Op
File Edit View Run Kernel Git Tabs Settings Help
/ Open Lab / Git public / exercises /
Name
Exercise 2 split and merge
Solution 2.naavrewf
Save
Export
Run
Cells Catalog
ex2-1-input-list-gpelouze
names
Splitter
Simple 1 2 main Solution 2.naavrewf 0

Exercise 1 params and secrets.ipynb
Exercise 2 split and merge.ipynb
Exercise 3 file management.ipynb
Solution 1.ipynb
Solution 1.naavrewf
Solution 2.ipynb
Solution 2.naavrewf
Solution 3.ipynb
Solution 3.naavrewf

untitled-gabriel-pelouz...
ex3-1-create-file-gabri...
ex3-3-upload-file-gabr...
ex3-2-plot-file-gabriel-...
ex2-3-print-list-gabriel...
ex2-1-input-list-gabriel...
ex2-2-process-list-gab...

Laserfarm Virtual Lab

Scenario: map vegetation characteristics using LiDAR data.



LiDAR raw data

National or regional Airborne Laser Scanning (ALS) surveys

Examples:
LiDAR point cloud datasets from the Netherlands
(AHN1, AHN2, AHN3, AHN4)

Data acquisition (laser scanning) → LiDAR point clouds → Retrieval to local data storage

Laserfarm workflow

Efficient, scalable and distributed processing of multi-terabyte LiDAR point clouds

Re-tiling → Normalization → Feature extraction → Rasterization

Data products

High-resolution LiDAR metrics across space and time

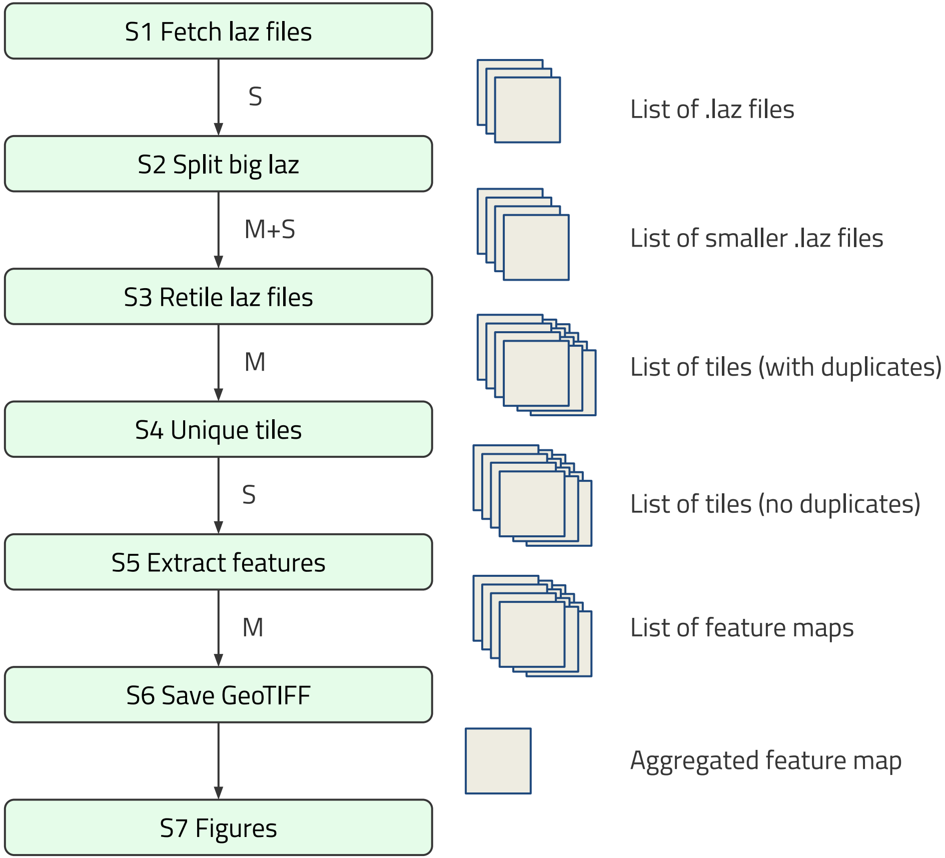
Time
Ecosystem height (2020-2022)
95th percentile of normalized height (m)
High : 60
AHN4
Ecosystem height (2014-2019)
95th percentile of normalized height (m)
High : 60
AHN3
Ecosystem height (2007-2012)
95th percentile of normalized height (m)
High : 60
AHN2
Ecosystem height (1996-2003)
95th percentile of normalized height (m)
High : 60
Low : 0
AHN1
Latitude
Longitude

Yifang Shi

Exercise

- Start the **Laserfarm** virtual lab
- Run the notebook `Laserfarm/Git public/exercises/Laserfarm.ipynb`
  Discuss the implications of processing large files in a notebook
- Containerize the cells and build a workflow
- Add splitters and mergers to parallelize the data processing. *Discuss with your classmates and tutors to figure out out where to add them*

Laserfarm workflow



S1 Fetch laz files

S

List of .laz files

S2 Split big laz

M+S

List of smaller .laz files

S3 Retile laz files

M

List of tiles (with duplicates)

S4 Unique tiles

S

List of tiles (no duplicates)

S5 Extract features

M

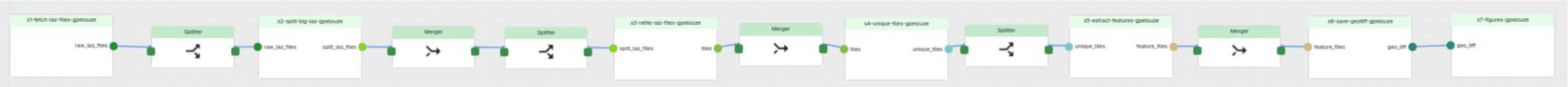
List of feature maps

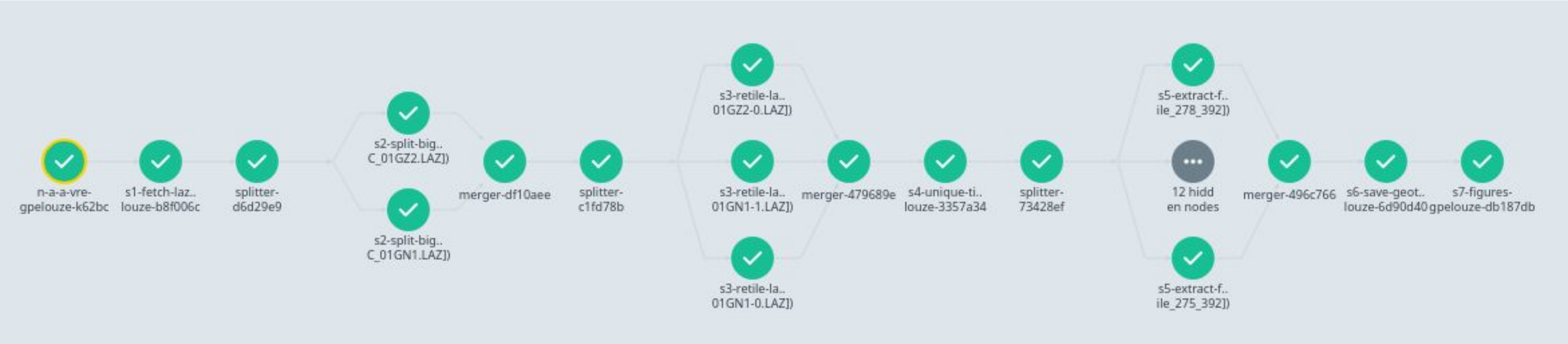
S6 Save GeoTIFF

Aggregated feature map

S7 Figures

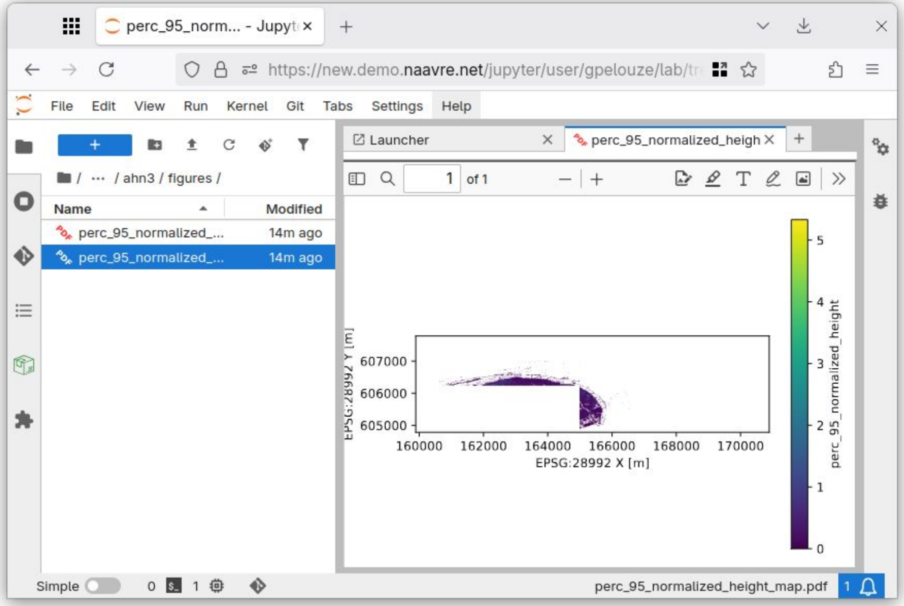
Laserfarm workflow





Laserfarm workflow results

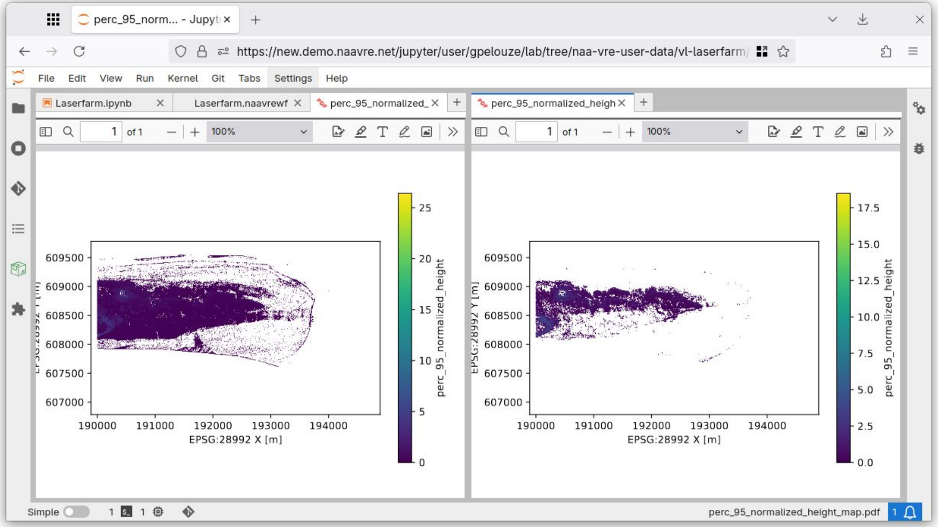
Getting the output: open the file browser and go to:
`naa-vre-user-data/vl-laserfarm/small_terschelling/ahn3/`



Other datasets: explore values of `param_minio_public_dataset_prefix`:

```
vl-laserfarm / small_terschelling / ahn3 /
               medium_ameland       ahn4
```

Other datasets: change value of `param_feature_name`
List: https://laserchicken.readthedocs.io/en/latest/#features



ITINERIS

THANKS!

LifeWatch ERIC

University of Amsterdam

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System (D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 "Education and Research" - Component 2: "From research to business" - Investment 3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani