Human Brain Project

EBRAINS

# Software Delivery and Installation for the Lab

24 November 2021 | EBRAINS CodeJam #12

Athanasios Karmas

Technical Coordination Team HBP SGA3

ATHENA Research & Innovation Information Technologies

# Outline

- Challenges
- Requirements / Users
- Spack package manager
- High Level Overview
- End-Users
- Component Owners
- First Release information
- Next Steps
- Demo session
- Thank you

# Challenges

- Collaboratory 1 and Collaboratory 2
  - Previously, no methodology for updating tools
- Difficult to update/add software
  - it would break dependencies of all existing notebooks
  - library dependency conflicts are difficult to solve by image builders
  - tool installation and testing was often insufficiently documented
- Old development environment (compilers, interpreters)

# Requirements / Actors

- Smaller base container image
    - easier to maintain
    - decoupled system and application software
    - separately testable
- Decentralized software development
- Modular software stack
- Central toolbox: as many of the tools as possible in the same activation environment to work out of the box
- Modern development environment

➢ End-User

➢ Developer

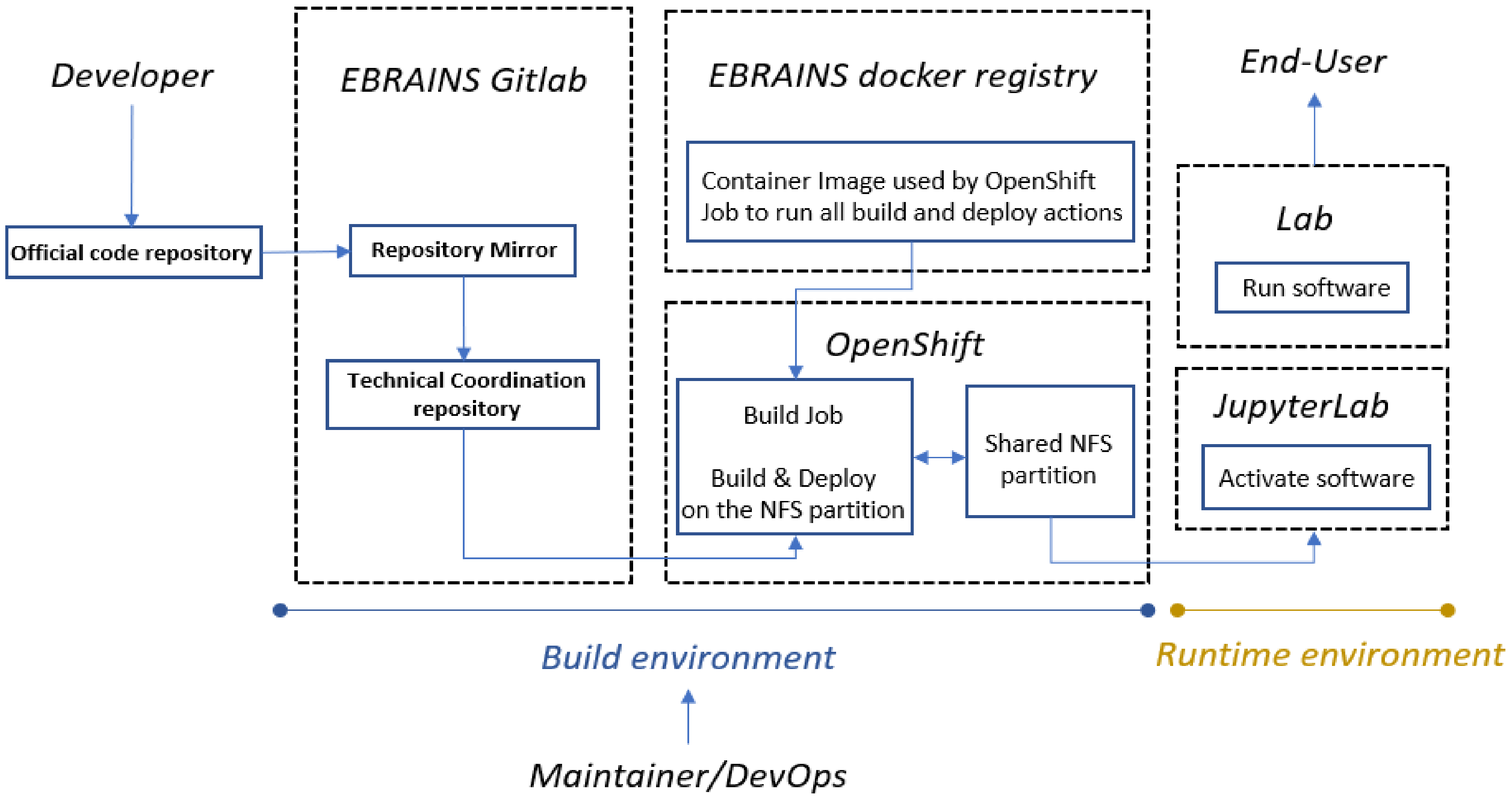
➢ DevOps/Maintainer

# Spack package manager

- Install scientific software
- Build packages with multiple versions, configurations, compilers
- Builds can coexist
- Not tied to a particular language
- Manage shared installations and modules
- Unique packages/dependencies configuration have their own prefix
- Key concepts
  - Spack package
  - Spack spec

# High Level Overview

Build and deliver EBRAINS tools with Spack for the Lab



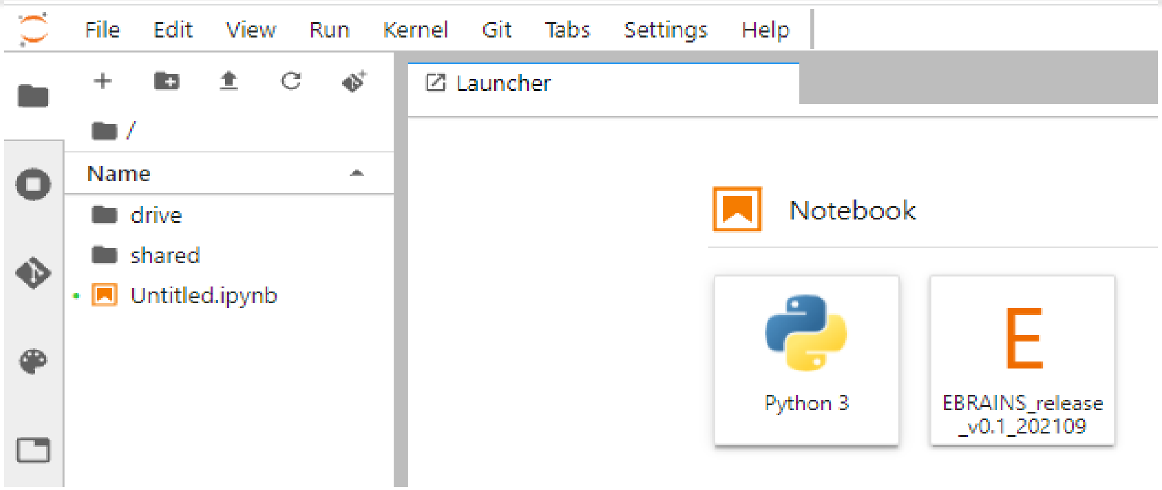
# High Level Overview

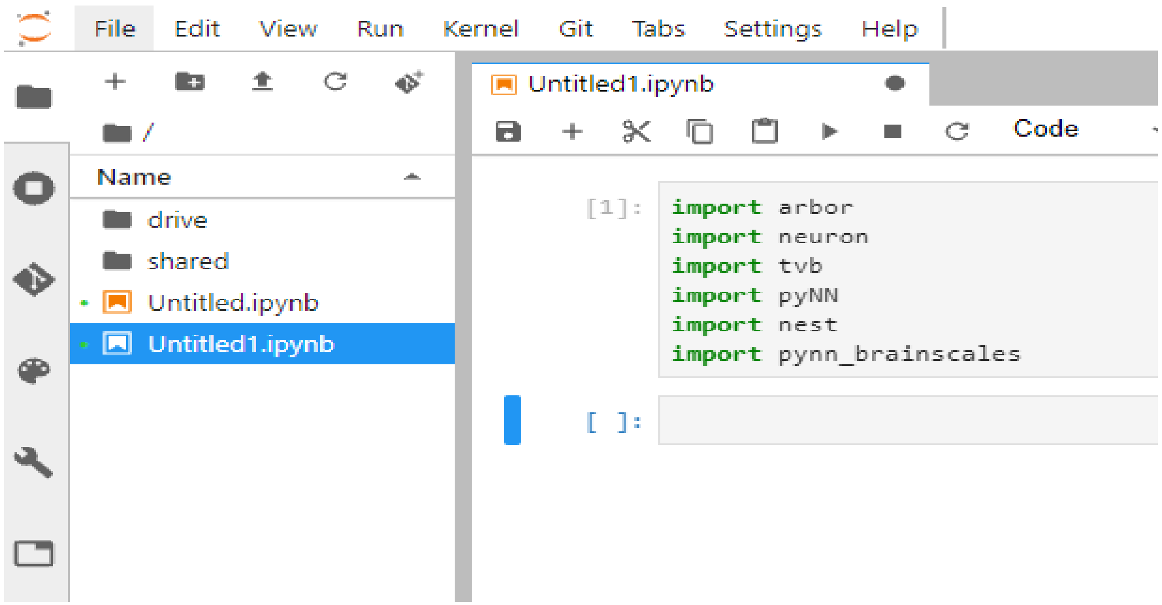
- Describe software stack, dependencies and build instructions with Spack
    - Each component must provide a Spack package
- Build and install components on a shared NFS partition (shared by the build environment and the runtime environment)
    - The process is coordinated and performed centrally by TC in Gitlab CI
- All deployed software is available to users of the Lab
- Common modern version of core compilers
- Dependencies tracking

# End-Users

Straightforward and simple steps:

- Connect to production  Lab
- Select docker image
- Start your server
- Start a new Notebook with the kernel with prefix “EBRAINS_release”
- Load the software you would like to work through standard python “import” commands


File Edit View Run Kernel Git Tabs Settings Help
Launcher
/
Name
drive
shared
Untitled.ipynb
Notebook
Python 3
EBRAINS_release _v0.1_202109



```
[1]: import arbor
     import neuron
     import tvb
     import pyNN
     import nest
     import pynn_brainscales

[ ]:
```

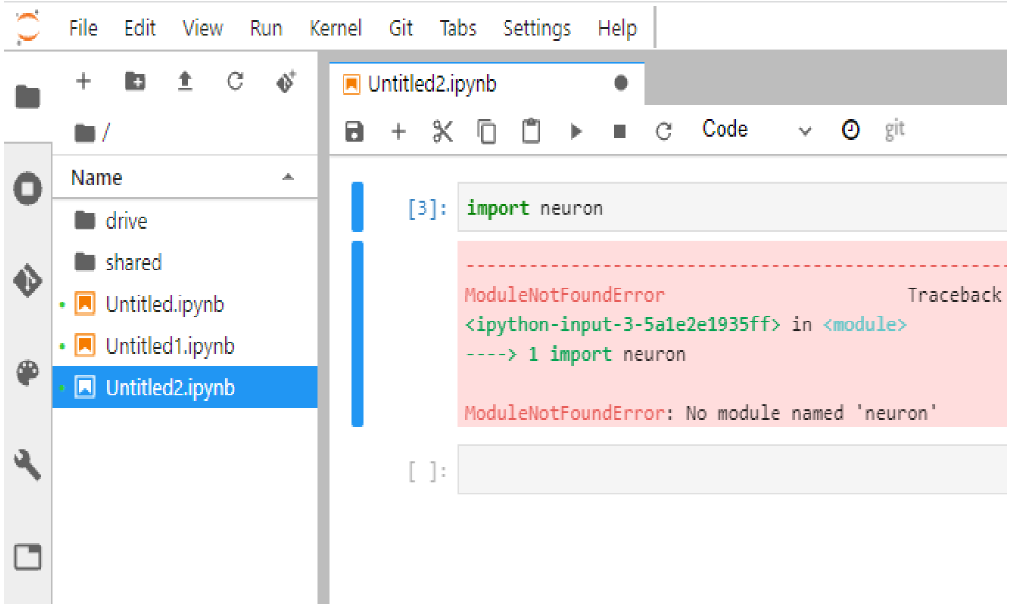
# End-Users

## Before

- Less tools / dependencies conflict
- Manual steps required (pip install)

## After

- Multiple tools available simultaneously
- Several versions of a tool (& dependencies)
- Seamless – only import statements

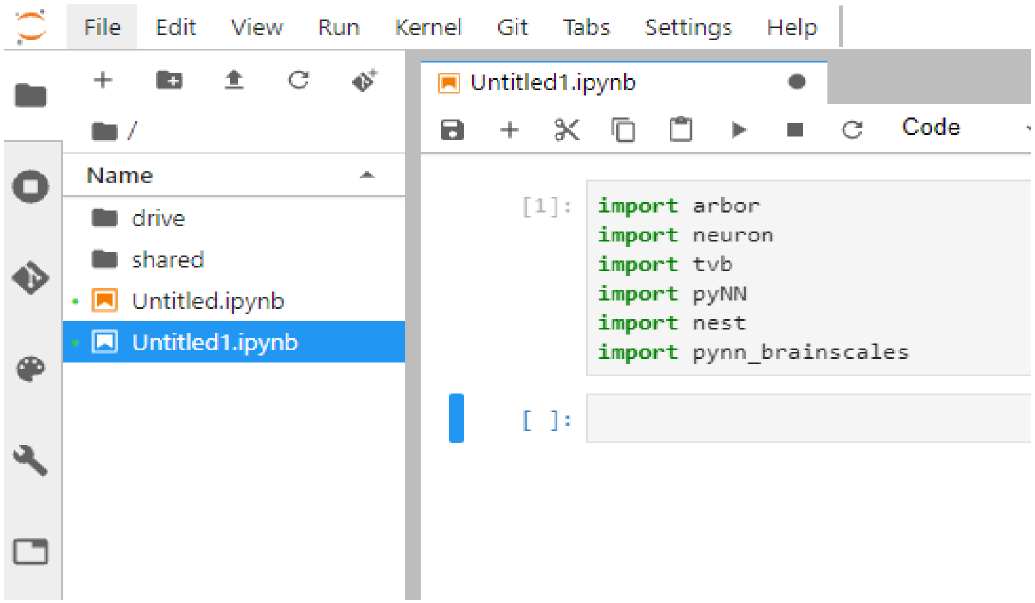


```
[3]: import neuron

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback
<ipython-input-3-5a1e2e1935ff> in <module>
----> 1 import neuron

ModuleNotFoundError: No module named 'neuron'

[ ]:
```



```
[1]: import arbor
     import neuron
     import tvb
     import pyNN
     import nest
     import pynn_brainscales

[ ]:
```

# Component Owners

How to get software built and delivered ?

- ➢ Prepare a Spack package and spec
- ➢ Make them available to the central build and delivery process

1. TC pulls them automatically from the component’s mirror (WIP)
2. Merge Request to the TC repository (package and spec)
3. Package in the Spack upstream.

Spec ⇨ Pulled automatically from the component’s mirror (1) or
Merge Request to the TC repository (2)

- ➢ Documentation available in TC Collab

# First Release information

- Tested in lab-int
- Tools
  - Arbor
  - BrainScaleS
  - Nest
  - Neuron
  - PyNN
  - TVB
- Tools coexist in the same environment
- GCC 10.3.0 and python 3.8.11

# Next steps

- Onboarding of more tools
- More seamless integration
- Improvements in the pipelines
- Load and unload tools with ease at runtime
- EBRAINS produces releases of tools every few months
- Intermediate tool deployments can be shared with advanced users (at their own risk)
- Join TC Weeklies (every Wednesday@11:00 CET/CEST)

# Demo session outline

How to get software built and delivered ?

- Prepare a Spack package and spec for a Python module
- Make contributions available to the Technical Coordination build flow pipeline
- Check and validate the pipeline

Human Brain Project EBRAINS

# Thank You!

www.humanbrainproject.eu www.ebrains.eu