

Reproducibility and dependencies for Jupyter Notebooks

Operate First Data Science Community Meetup, 30th November 2021

Francesco Murdaca, Red Hat

Jupyter Notebooks (~2 min)

A quick intro to Jupyter Notebooks.

What problem are we trying to solve? (~6 min)

Dependency management for Jupyter Notebooks.

Project Thoth (~4 min)

Project Thoth overview

How does Thoth help to solve the problem? (~5 min)

How Thoth contributes to the solution of the problems stated.

Dependency Management Tutorial (~5 min)

Operate First, Project Meteor and dependency management tutorial

Conclusion (~3 min)

Jupyter Notebook

Operate First Data Science Community Meetup

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.



Language of choice

Jupyter support over 40 programming languages.



Notebook sharing

Sharing interactive code documents with others.



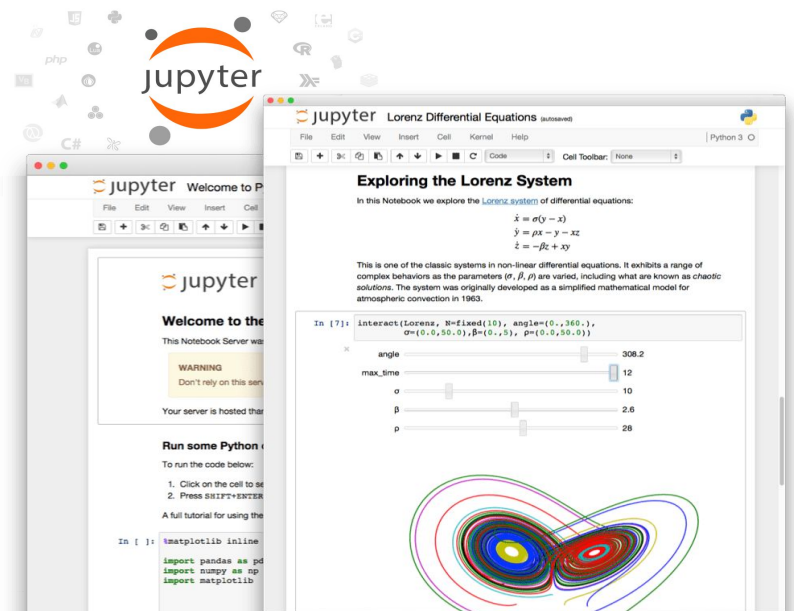
Interactive output

Rich, interactive output: HTML, images, videos, etc.

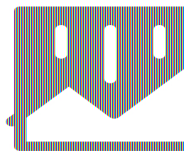


Big Data integration and analysis

Leverage big data tools and explore that data.



Heavily adopted



Data Scientists

Data analysis, modeling and visualization and analytical reports.



Educators and students

Assignments, interactive coding lessons, tutorials.



Developers

Rapid prototyping, POCs, testing and integration and example usage.

Trusted by many

Google

Microsoft

IBM

Bloomberg

O'REILLY®

ANACONDA.

SOUNDCLOUD

Quantopian

QuantStack
Scientific Computing

software
carpentry

NetApp®

NASA

Berkeley
UNIVERSITY OF CALIFORNIA

MICHIGAN STATE
UNIVERSITY

NYU

...

What problem are we trying to solve?

Operate First Data Science Community Meetup

What problem are we trying to solve?

```
pip install  
opencv-python
```


What problem are we trying to solve?

```
pip install  
opencv-python
```

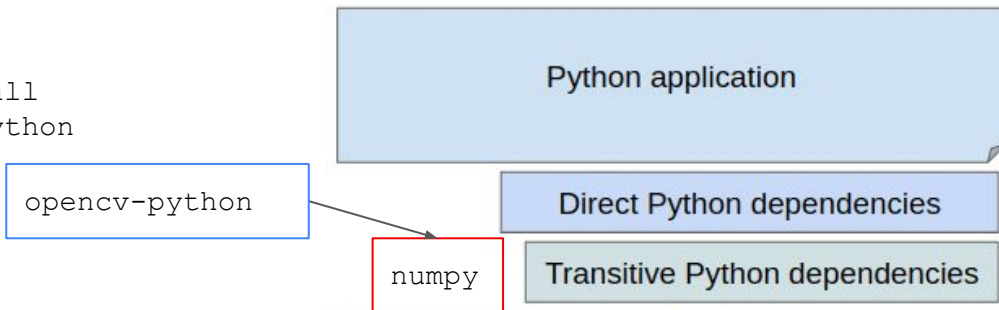
```
opencv-python
```

Python application

Direct Python dependencies

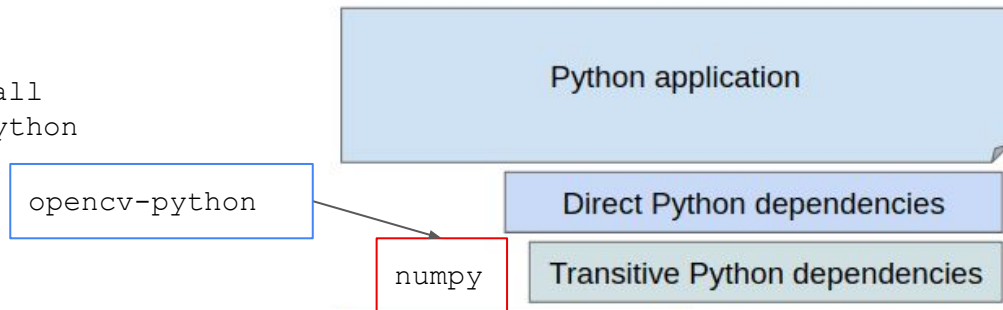
What problem are we trying to solve?

```
pip install  
opencv-python
```



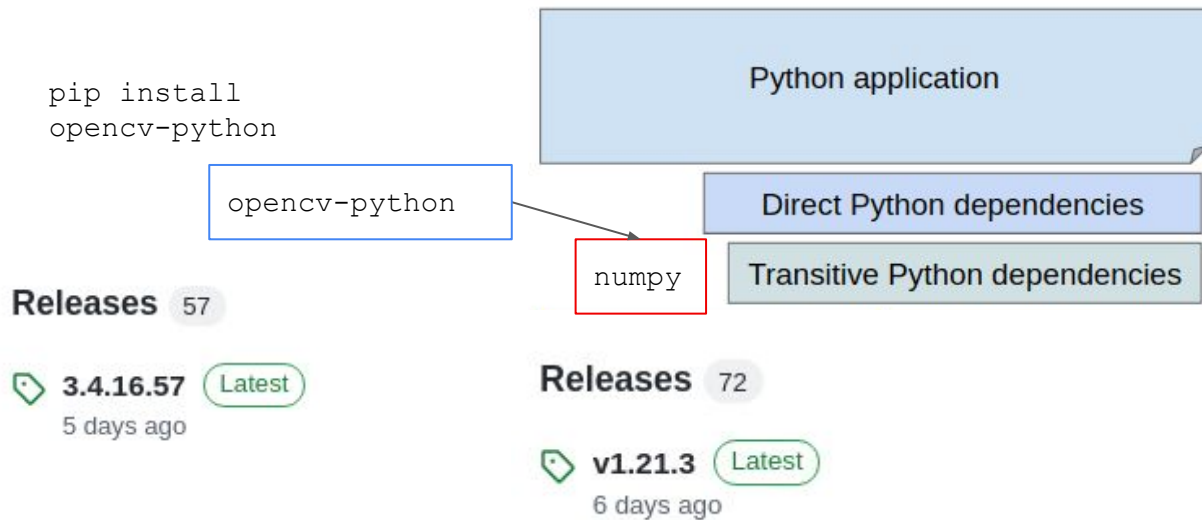
What problem are we trying to solve?

```
pip install  
opencv-python
```



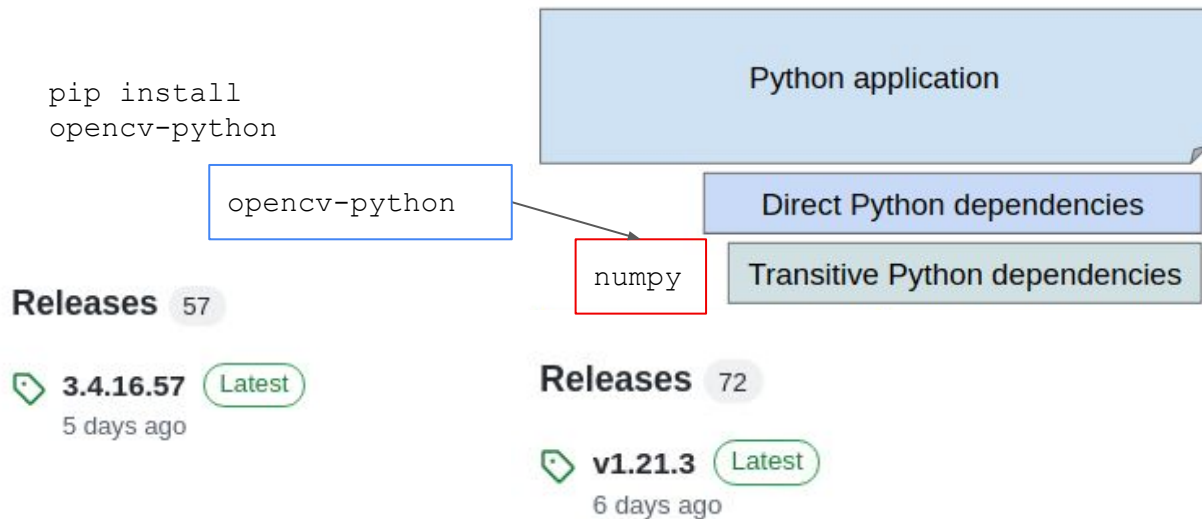
What about versions?

What problem are we trying to solve?



What about versions?

What problem are we trying to solve?



What about versions?

Releases Tags

Tags

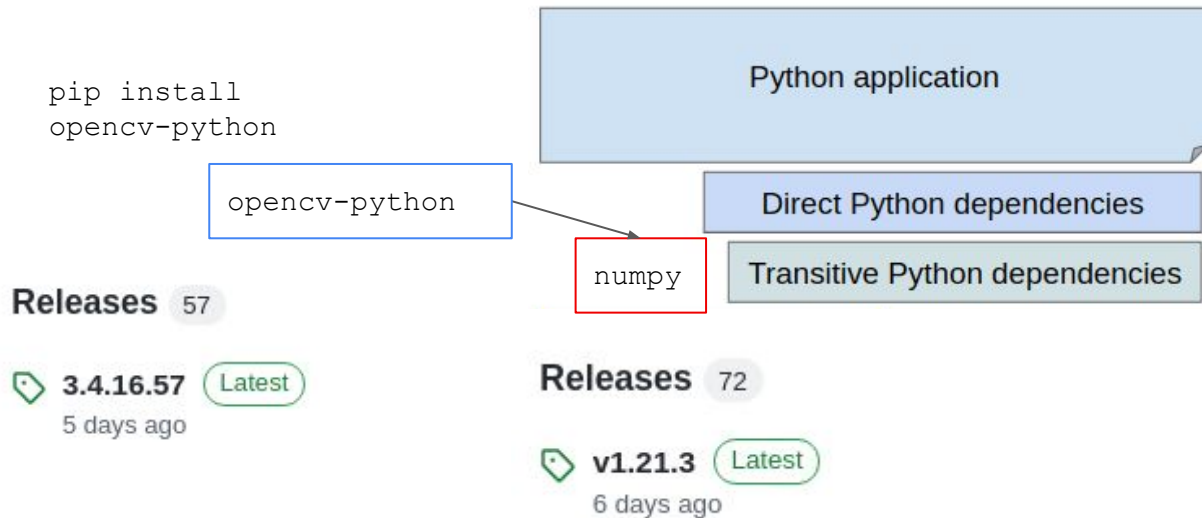
v1.21.3 ...
6 days ago - d4d0584

v1.21.2 ...
on Aug 15 - 2fe48d2

v1.21.1 ...
on Jul 18 - df6d260

v1.21.0 ...
on Jun 19 - b235f9e

What problem are we trying to solve?



What about versions?
What about hashes?

Releases Tags

Tags

v1.21.3 ...

6 days ago d4d0584

v1.21.2 ...

on Aug 15 2fe48d2

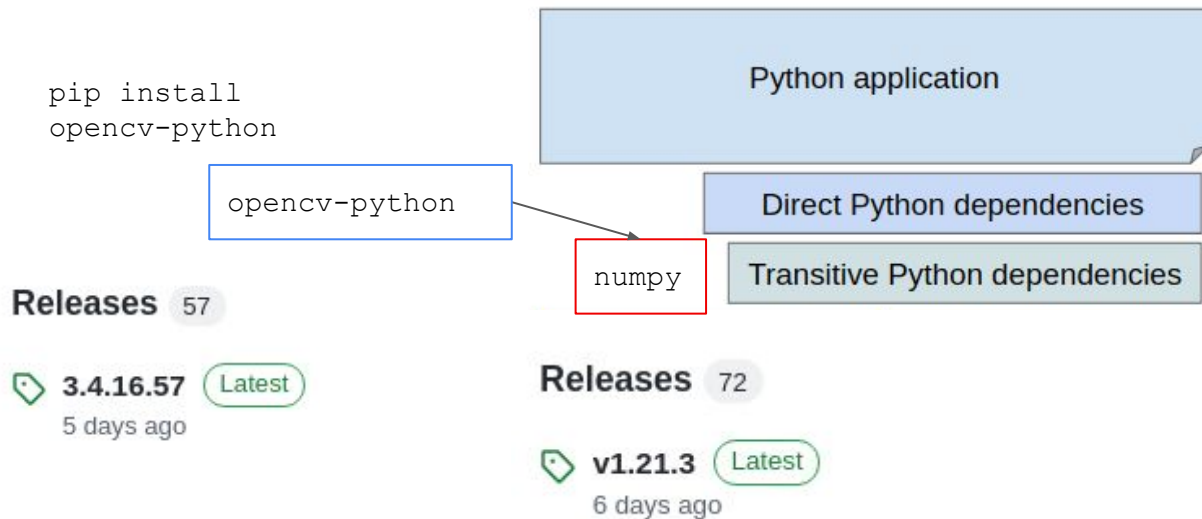
v1.21.1 ...

on Jul 18 df6d260

v1.21.0 ...

on Jun 19 b235f9e

What problem are we trying to solve?



What about versions?
What about hashes?
What about Python interpreter?

Releases Tags

Tags

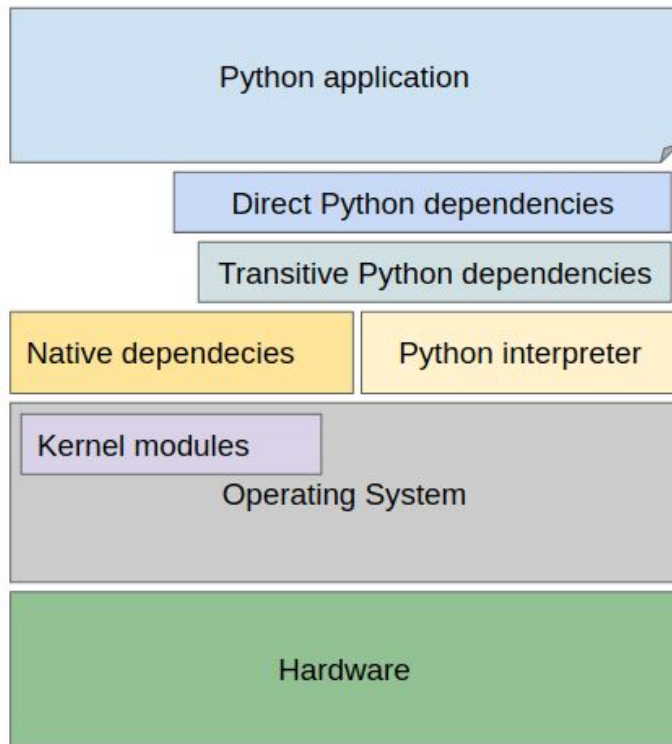
v1.21.3 ...
6 days ago - d4d0584

v1.21.2 ...
on Aug 15 - 2fe48d2

v1.21.1 ...
on Jul 18 - df6d260

v1.21.0 ...
on Jun 19 - b235f9e

What problem are we trying to solve?



What problem are we trying to solve?

Install dependencies

```
In [2]: ! pip install tensorflow  
! pip install boto3  
! pip install matplotlib
```

What problem are we trying to solve?

Install dependencies

```
In [2]: ! pip install tensorflow  
! pip install boto3  
! pip install matplotlib
```

This will not guarantee reproducibility!

What problem are we trying to solve?

```
1  voila
2  folium
3  numpy
4  pandas
5  ipywidgets
6  ipykernel
7  matplotlib
```

What problem are we trying to solve?

```
1  voila
2  folium
3  numpy
4  pandas
5  ipywidgets
6  ipykernel
7  matplotlib
```

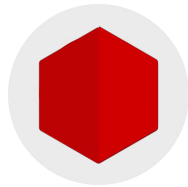
Having requirements.txt with no versions stated does not guarantee to have reproducible notebook!

Jupyter Notebooks are by default **NOT** stand-alone



Managing dependencies

Requirements
are **decoupled from a notebook***
into
manifest files, such as
requirements.txt or *Pipfile.lock*



Containerisation

A specialised tools or a custom
Dockerfile is needed so that all
notebooks requirements* are
present in the resulting image.



Sharing

The consumer must first **manually**
set up **environment** for them using
provided* **manifest** files.

*It is not uncommon that **NO manifest files are provided** and hence Notebook users must **find out dependencies themselves**.

Difficulties for both authors and consumers

Authors have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies to the environment

[optional] Create/Update custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.

[optional] Create/Update manifest files

Consumers have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies* to the environment

[optional] Create custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.

*It is not uncommon that **NO manifest files are provided** and hence Notebook users must **find out dependencies themselves**.

Difficulties for both authors and consumers

Authors have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies to the environment

[optional] Create/Update custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.

[optional] Create/Update manifest files

Consumers have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies* to the environment

[optional] Create custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.



*It is not uncommon that **NO manifest files are provided** and hence Notebook users must **find out dependencies themselves**.

Difficulties for both authors and consumers

Authors have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies to the environment

[optional] Create/Update custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.

[optional] Create/Update manifest files

Consumers have to...

Create an environment

Ideally a virtual environment for the notebook to run in.

Install dependencies* to the environment

[optional] Create custom kernel

It is a recommended approach (and the best practice) to create a custom kernel for each project.



Reproducibility!

*It is not uncommon that **NO manifest files are provided** and hence Notebook users must **find out dependencies themselves**.

Project Thoth

Operate First Data Science Community Meetup



Project Thoth

- Provide **a system and a user facing service**, that helps making well educated decisions by delivering a broad and deep knowledge set wrt frameworks relevant in the field of **AI applications**.
- **Deliver optimized, secured, well maintained and predictable images** for your AI applications
- Use **bots to automate mundane work** to offload humans work

Thoth Recommendation types

- Latest
- Stable
- Security
- Performance
- Testing

what we observe and store in our knowledge graph

Application Stack related:

- Buildtime and runtime environment
- Dependencies
- Performances

Software Package:

- Application Binary Interfaces (ABI)
- Security (CVE, analyzers)

Source Code Meta Information:

- Project features (TTR, TTCL, etc,..) from different software development platform (Github, GitLab, Pagure, etc...)

what we observe and use in resolution process

Prescriptions:

- Pipeline units described using a YAML file to heal Python Application
 - Pipeline unit type and its configuration used in the resolution process
- Opens a possibility to easily create pipeline units specific for recurrent issues or use-cases following a pattern
 - e.g. a pipeline unit specific for ODH use case when resolving for ODH base container images
 - ...

[Check the video](#)

How do we use this knowledge?

- Recommender system is called **Adviser** in Thoth.
- It uses Reinforcement Learning (RL).

[Check the video](#)

Thoth Integrations

- Command line tool thamos (developer laptop)
- Cyborg Kebechet (pull request/issues creator)
- Source-to-Image (container builder)
- Optimizing Deployment Pipelines
- **Jupyter Tools (data scientist browser)**

How does Thoth help solve the problem?

Operate First Data Science Community Meetup



jupyterlab-requirements

JupyterLab extension for dependency management and optimization

JupyterLab extension allows you to **manage dependencies and store everything in the Jupyter notebook metadata**:

- **requirements (Pipfile);**
- **requirements locked** with all versions and hashes of libraries (direct and transitive ones) **(Pipfile.lock);**
- **dependency resolution engine** used **(Thoth or Pipenv);**
- configuration file containing runtime environment (only for Thoth resolution engine).

How does Thoth helps solving the problem?

jupyterlab-requirements

JupyterLab extension for dependency management and optimization

```
ock --help
pykernel_launcher.py lock [-h] [--force] [--debug]
                        [--kernel-name KERNEL_NAME]
                        [--recommendation-type {[latest,stable,performance,security]}]
                        [--timeout TIMEOUT] [--os-name OS_NAME]
                        [--os-version OS_VERSION]
                        [--python-version PYTHON_VERSION] [--pipenv]

Requirements in notebook metadata [default: Thoth].

arguments:
  help      show this help message and exit
  e         Force request to Thoth.
  g         Debug/Verbose request to Thoth. WARNING: It has impact
            on the quality of the resolution process.
  el-name KERNEL_NAME
            Specify kernel name to be used when creating it.
  mmdation-type {[latest,stable,performance,security]}
            Specify recommendation type for thoth advise.
  out TIMEOUT
            Set timeout for Thoth request.
  ame OS_NAME
            Use OS name for request to Thoth.
  ersion OS_VERSION
            Use OS version for request to Thoth.
  on-version PYTHON_VERSION
            Use Python version for request to Thoth.
  nv         Use pipenv resolution engine.
```

Jupyter magic commands

available directly in your notebook cells
when you start a notebook

```
horus lock --help
is lock [OPTIONS] PATH

Requirements in notebook metadata.

horus lock [YOUR_NOTEBOOK].ipynb

name TEXT
    Name of kernel.
    Lock dependencies using Pipenv.
: INTEGER
    Set Timeout for Thoth advise request.
    [default: 180]
    Force Thoth advise request.
    Debug/verbose Thoth advise request. WARNING:
    It has impact on the quality of the
    resolution process.
mdation-type [latest|stable|performance|security]
    Recommendation type for Thoth advise
    request. [default: latest; required]
: TEXT
    OS name for Thoth advise request.
:ion TEXT
    OS version for Thoth advise request.
:version TEXT
    Python version for Thoth advise request.
    Show this message and exit.
```

CLI

that you can run from terminal or
integrated in pipelines

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
pandas		X	Install

OPTIONS

Kernel name: jupyterlab-requirements
Path root project: /home/mundaca/work/aiocel
Recommendation type: latest
Thoth timeout [s]: 180
Thoth -debug: False

Cancel

UI

accessible through **Manage Dependencies** button that appears in
the notebook when it is opened in
JupyterLab

jupyterlab-requirements

Jupyter magic commands

```
[2]: %horus lock --help
usage: ipykernel_launcher.py lock [-h] [--force] [--debug]
                                   [--kernel-name KERNEL_NAME]
                                   [--recommendation-type [{latest,stable,performance,security}]]
                                   [--timeout TIMEOUT] [--os-name OS_NAME]
                                   [--os-version OS_VERSION]
                                   [--python-version PYTHON_VERSION] [--pipenv]

Lock requirements in notebook metadata [default Thoth].

optional arguments:
  -h, --help            show this help message and exit
  --force               Force request to Thoth.
  --debug               Debug/Verbose request to Thoth. WARNING: It has impact
                        on the quality of the resolution process.
  --kernel-name KERNEL_NAME
                        Specify kernel name to be used when creating it.
  --recommendation-type [{latest,stable,performance,security}]
                        Specify recommendation type for thoth advise.
  --timeout TIMEOUT     Set timeout for Thoth request.
  --os-name OS_NAME     Use OS name for request to Thoth.
  --os-version OS_VERSION
                        Use OS version for request to Thoth.
  --python-version PYTHON_VERSION
                        Use Python version for request to Thoth.
  --pipenv              Use pipenv resolution engine.
```

- Run commands from notebook cells:
 - Handle dependencies from cells (add/remove)
 - Handle kernels from cells (set-kernel)
 - Lock dependencies from cells (lock)
- Different resolution engines (Thoth, Pipenv)

jupyterlab-requirements

Command Line Interface (CLI)

```
fmurdaca@pc-7 ~ --/work/aicoe/jupyterlab-requirements master • horus lock --help
```

```
Usage: horus lock [OPTIONS] PATH
```

Lock requirements in notebook metadata.

Examples: horus lock [YOUR_NOTEBOOK].ipynb

Options:

--kernel-name TEXT	Name of kernel.
--pipenv	Lock dependencies using Pipenv.
--timeout INTEGER	Set timeout for Thoth advise request. [default: 180]
--force	Force Thoth advise request.
--debug	Debug/verbose Thoth advise request. WARNING: It has impact on the quality of the resolution process.
--recommendation-type [latest stable performance security]	Recommendation type for Thoth advise request. [default: latest; required]
--os-name TEXT	OS name for Thoth advise request.
--os-version TEXT	OS version for Thoth advise request.
--python-version TEXT	Python version for Thoth advise request.
--help	Show this message and exit.

- Run commands from terminal
 - Handle dependencies from cells (add/remove)
 - Handle kernels from cells (set-kernel)
 - Lock dependencies from cells (lock)
- Handle jupyter notebook dependencies in CI/CD pipelines
- Different resolution engines (Thoth, Pipenv)

How does Thoth helps solving the problem?

jupyterlab-requirements

User Interface (UI)

- Interactive UI to handle dependencies

Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
pandas	*		

Install

OPTIONS

Kernel name:

Path root project:

Recommendation type:

Thoth timeout [s]:

Thoth --force:

Thoth --debug:

Cancel

jupyterlab-requirements Python package

```
pip install jupyterlab-requirements
```

```
jupyter lab
```

Dependency management tutorial for Jupyter Notebooks

Operate First Data Science Community Meetup

Dependency management tutorial

- start working on a new notebook
- create dependencies for your existing notebook
- convert notebook that uses pip commands in cells
- use a reproducible notebook

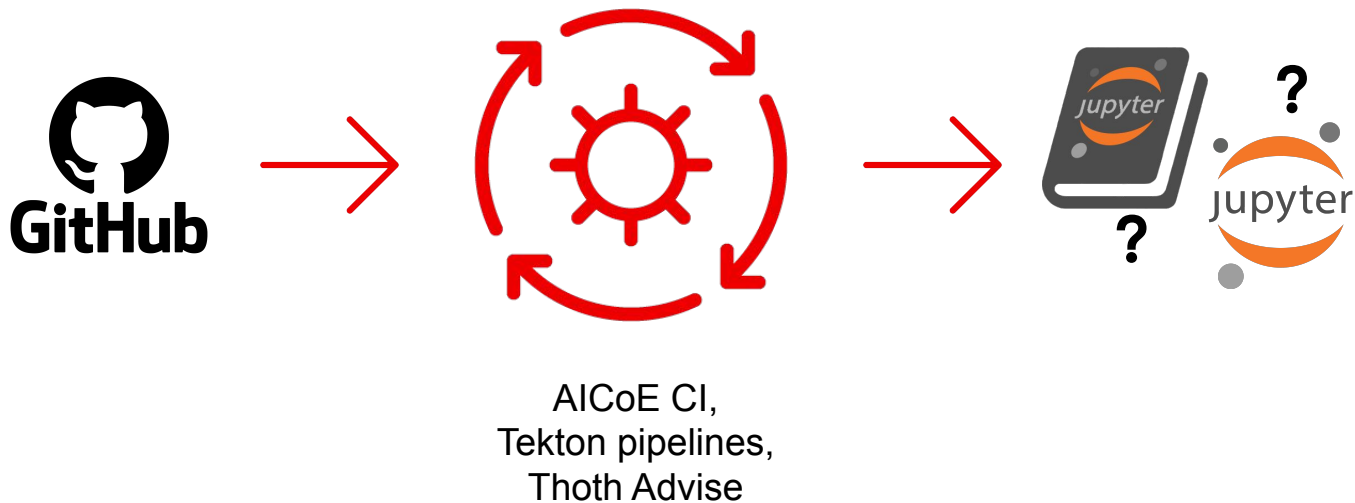
Operate First



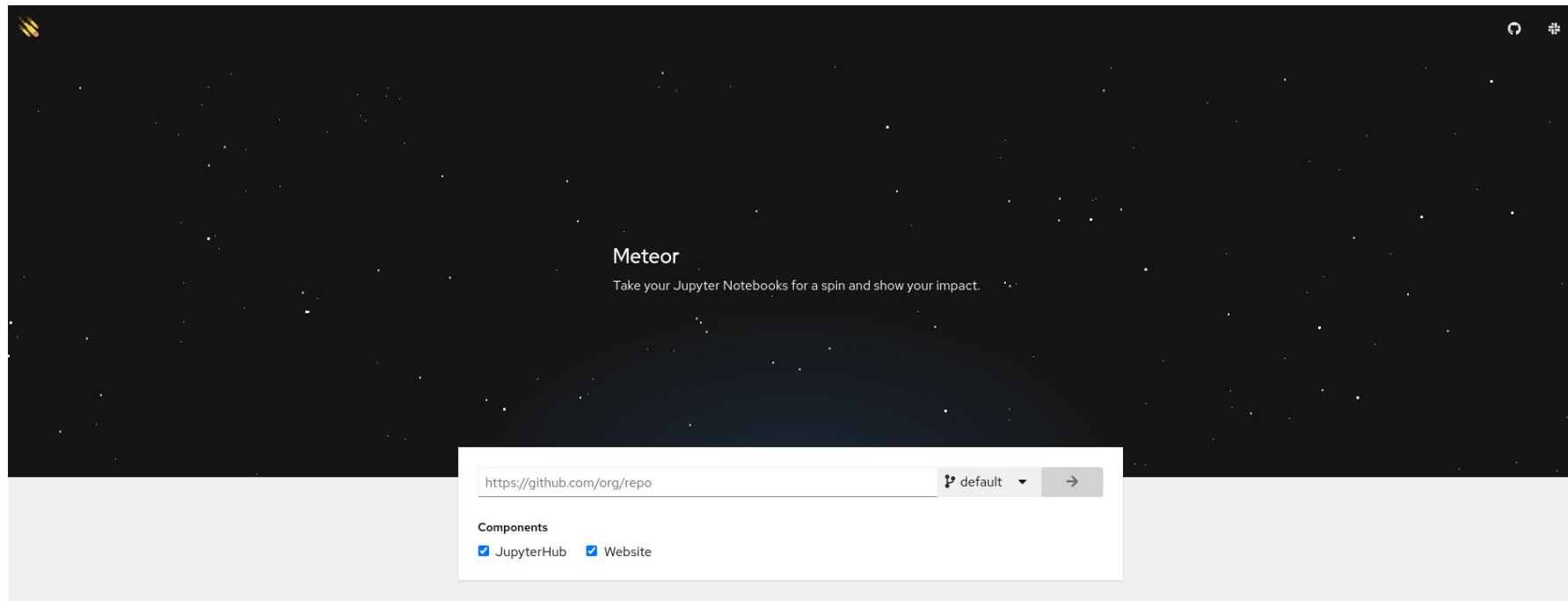
<https://operate-first.cloud>

<https://github.com/operate-first>

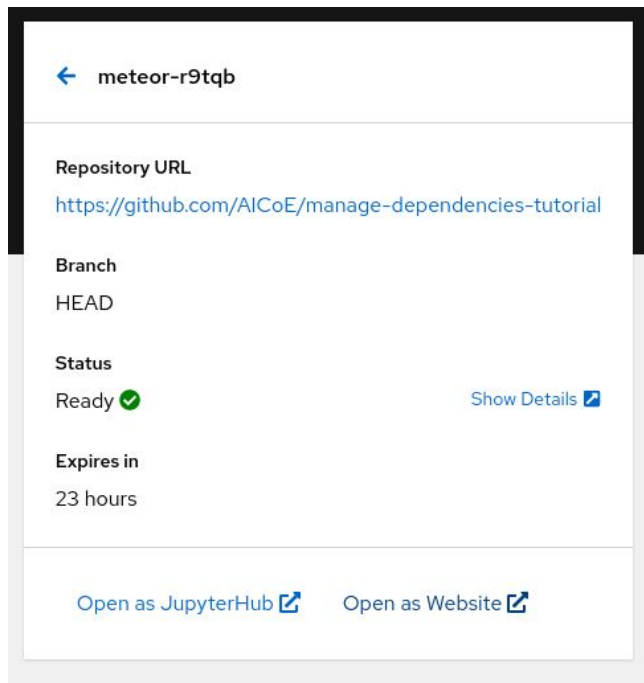
Project Meteor



Project Meteor



Project Meteor





Manage Dependencies Tutorial

Search this book...

Thoth Tutorial - manage your dependencies in Jupyter notebooks.

BEHIND THE SCENES

Pre-requisites

Setup initial environment

MANAGE DEPENDENCIES IN NOTEBOOKS

Reproducibility of Jupyter Notebooks ▾

CONTRIBUTE AND SAVE CHANGES

Push your changes on your GitHub repo

Project Meteor - Jupyter Book



Contents

What you will learn with this tutorial?

Where you will run this tutorial?

Why does the tutorial repository have this structure?

Tutorial pre-requisites

Tutorial Steps

References

Thoth Tutorial - manage your dependencies in Jupyter notebooks.

This tutorial is used to show how to manage dependencies for Jupyter Notebooks using Python to allow reproducibility and shareability.

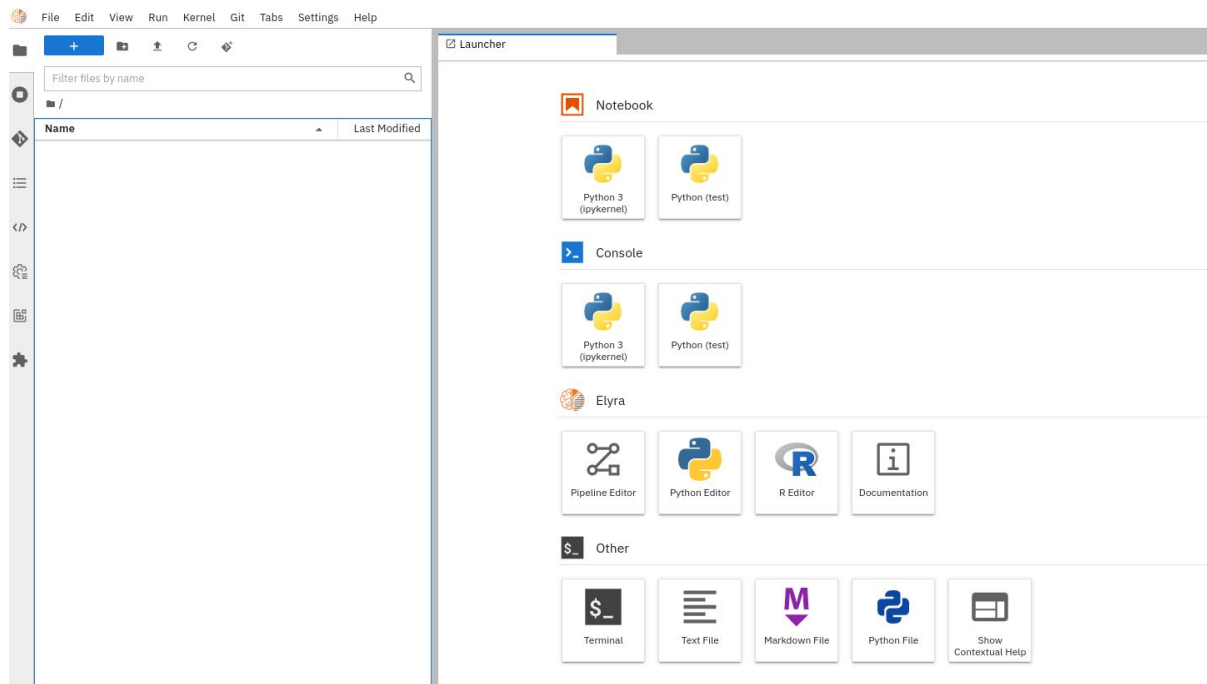
Even though many developers (including data scientists) focus on their core problems when working on their experiments, there is one aspect that can make these projects not reusable. One of the first steps during the development of a project is the [selection of libraries or dependencies](#). When someone runs `pip install <package-name>`, they might not be aware that along with the library that is going to be installed, a direct dependency, many other dependencies will be installed on your machine, so called transitive dependencies. Any change in one of those dependencies can break your experiment. It's fundamental to have a way to state all the dependencies used, including the operating system, python interpreter and hardware that was used to run a certain experiment.

Dependency management is one of the most important requirements for reproducibility. Having dependencies clearly stated allows portability of notebooks, so they can be shared safely with others, reused in other projects or simply reproduced. If you want to know more about this issue in the data science domain, have a look at this [article](#) or this [video](#).

[Project Thoth](#) keeps dependencies up to date by giving recommendations through developer's daily tools. Thanks to this service, developers (including data scientists) do not have to worry about managing the dependencies after they are selected, since conflicts can be handled by Thoth bots and automated pipelines. Having this AI support can benefit AI projects, offering improvements such as performance improvements due to optimized dependencies and additional security since insecure libraries cannot be introduced. If you want to know more, have a look at [Thoth's website](#).

Within the different Thoth integrations, in this tutorial we are going to focus on the JupyterLab extension for dependency management, which is called [jupyterlab-requirements](#).

Project Meteor - JupyterLab environment



Conclusions

Operate First Data Science Community Meetup

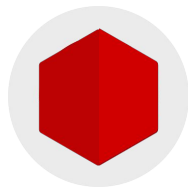
A notable improvements...



Managing dependencies

Requirements are **decoupled from a notebook**^{*} into manifest files, such as *requirements.txt* or *Pipfile.lock*

Requirements are **locked** and **embedded** directly into the notebook. No additional files are needed.



Containerisation

A specialised tools or a custom Dockerfile is needed so that all notebooks requirements^{*} are present in the resulting image.

Jupyter Notebooks with embedded dependencies can be built directly using **Jupyter Notebook S2I** **without any additional files**.



Sharing

The consumer must first **manually** set up **environment** for them using provided^{*} **manifest** files.

Jupyter Notebooks can be shared as **stand-alone units** without any additional files. Environment is prepared in a **single click**.

With the focus on reproducibility



Resolved Jupyter Notebook dependencies

When the notebook is distributed, unless specified otherwise, the **very same versions** are used which guarantees compatibility and reliable results.*



Project Thoth

- Website <https://thoth-station.ninja/>
- Twitter <https://twitter.com/thothstation>
- GitHub <https://github.com/thoth-station>
- Thoth Station YouTube https://www.youtube.com/channel/UCIUlDuq_hQ6vlzmqM59B2Lw/videos
- Tutorial <https://github.com/AlCoE/manage-dependencies-tutorial>

Thank you

Red Hat is the world's leading provider of
enterprise open source software solutions.
Award-winning support, training, and consulting
services make
Red Hat a trusted adviser to the Fortune 500.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat