Reproducibility and dependencies for Jupyter Notebooks

Open Data Science Conference (ODSC) West 2021. 16th November 2021

Francesco Murdaca
Senior Data Scientist, AI CoE, Red Hat

Agenda

Jupyter Notebook

ODSC West 2021

Jupyter Notebooks

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

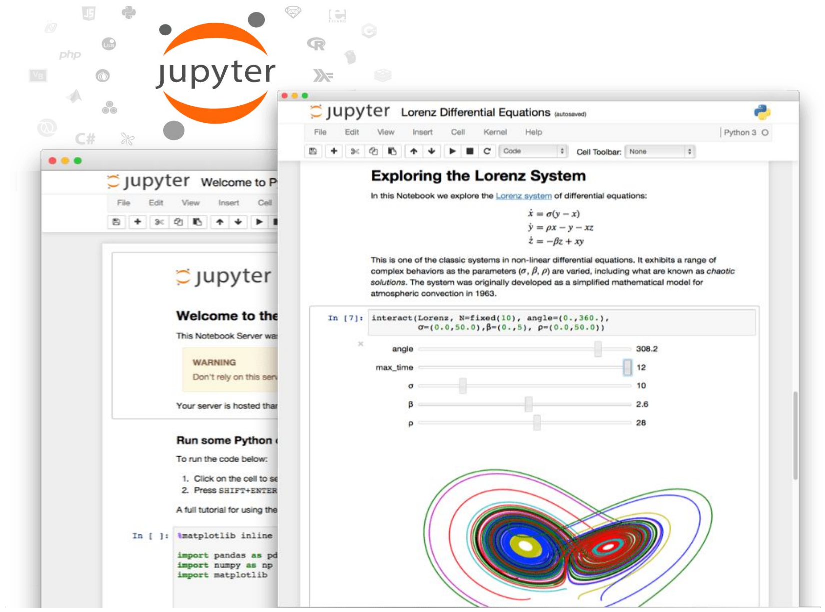
Sharing interactive code documents with others.

Interactive output

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

Big Data integration and analysis

Leverage big data tools and explore that data.


jupyter
php
C#
jupyter Welcome to P
File Edit View Insert Cell
jupyter Lorenz Differential Equations (autosaved)
File Edit View Insert Cell Kernel Help
Python 3
Code
Cell Toolbar: None

jupyter

Welcome to the

This Notebook Server wa

WARNING
Don't rely on this ser

Your server is hosted thar

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

```
angle     308.2
max_time  12
σ         10
β         2.6
ρ         28
```

Source https://jupyter.org/

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.

Red Hat

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

...

Red Hat

What problems are we trying to solve?

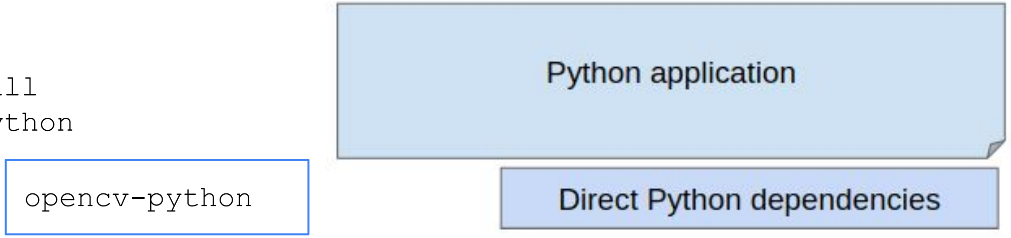
ODSC West 2021

What problems are we trying to solve?

```
pip install
opencv-python
```

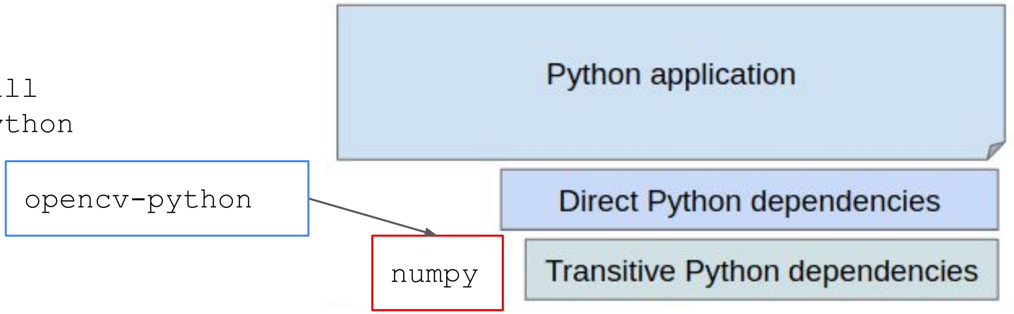
What problems are we trying to solve?

```
pip install
opencv-python
```



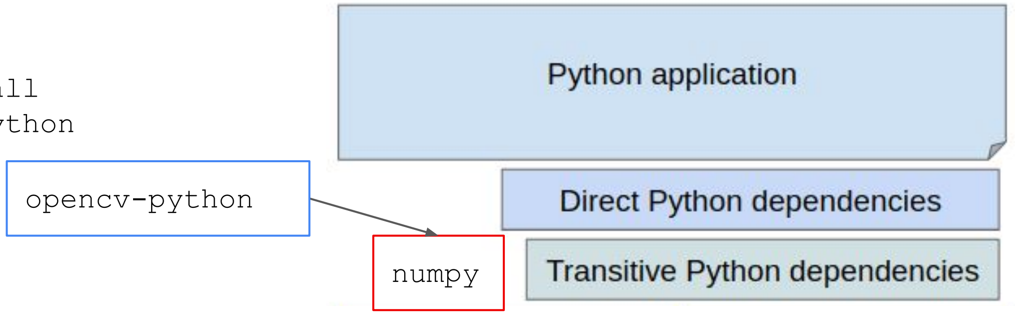
What problems are we trying to solve?

```
pip install
opencv-python
```



What problems are we trying to solve?

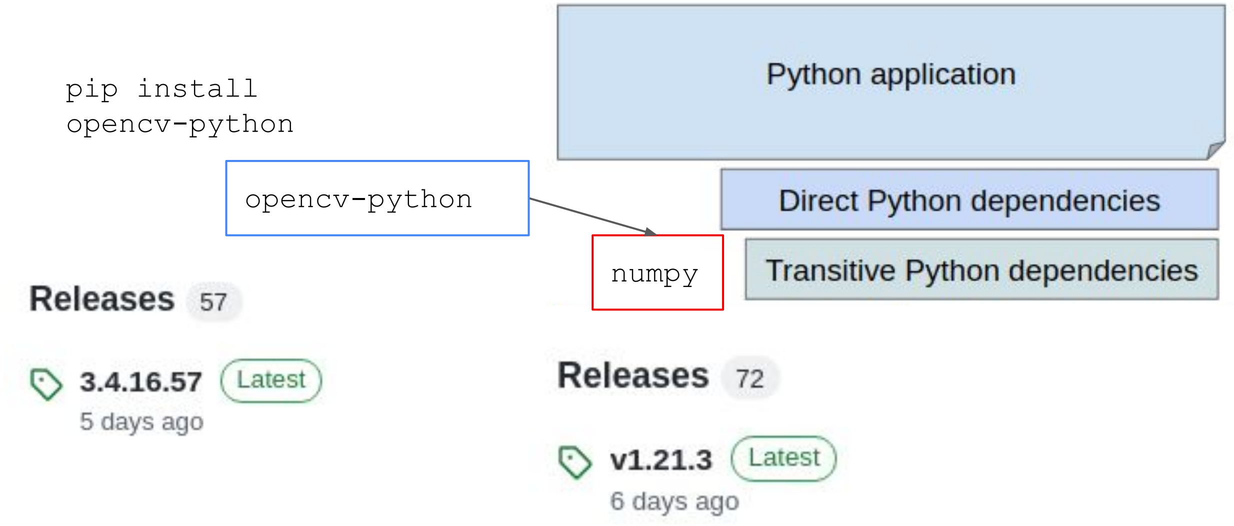
```
pip install
opencv-python
```



What about versions?

What problems are we trying to solve?

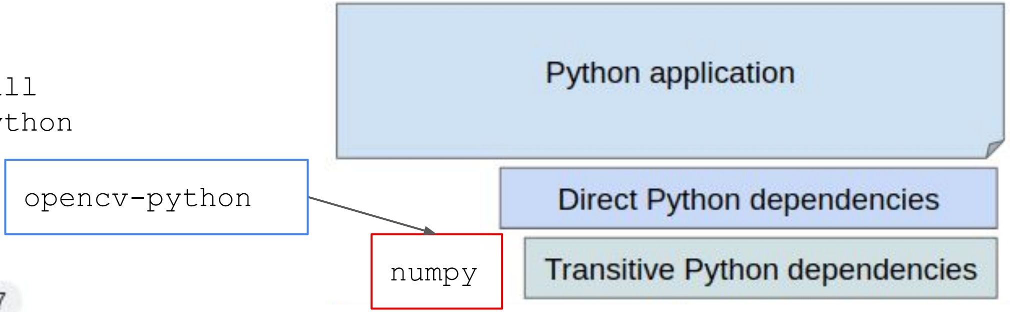
```
pip install
opencv-python
```

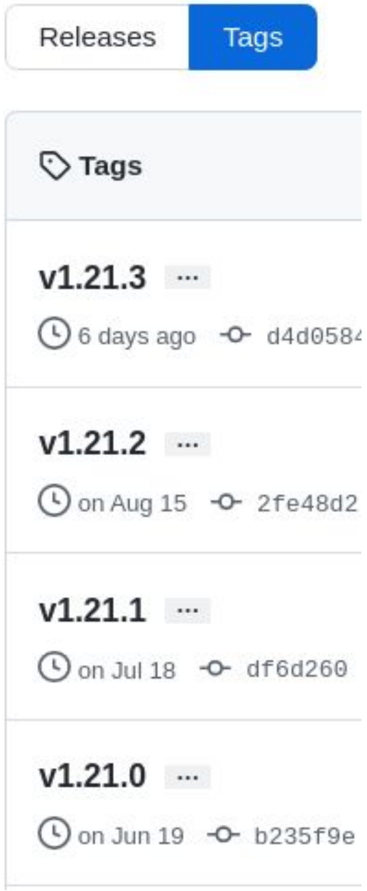


What about versions?

What problems are we trying to solve?

```
pip install
opencv-python
```

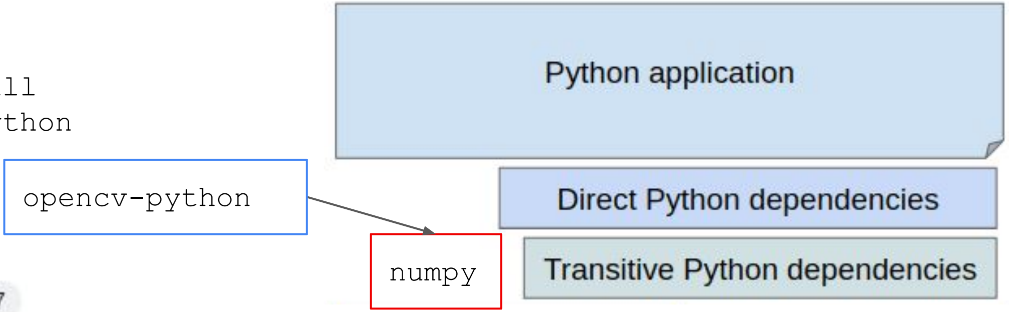




What about versions?

What problems are we trying to solve?

```
pip install
opencv-python
```

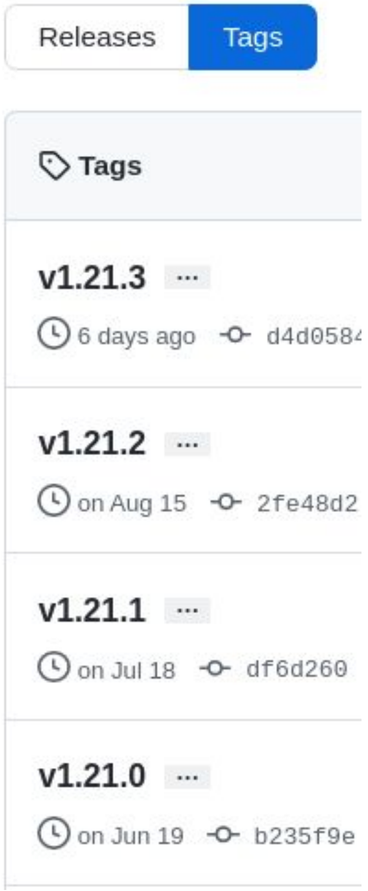


Releases 57

3.4.16.57 Latest
5 days ago

Releases 72

v1.21.3 Latest
6 days ago


Releases | Tags

Tags

v1.21.3 ···
6 days ago d4d0584

v1.21.2 ···
on Aug 15 2fe48d2

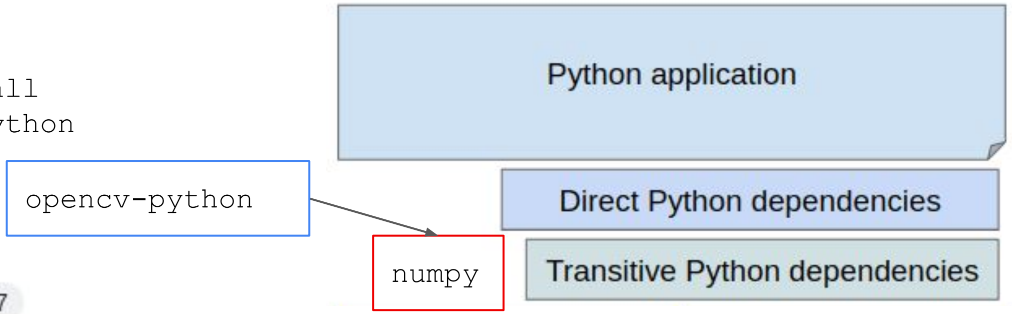
v1.21.1 ···
on Jul 18 df6d260

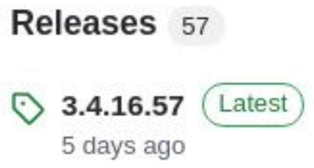
v1.21.0 ···
on Jun 19 b235f9e

What about versions?
What about hashes?

What problems are we trying to solve?

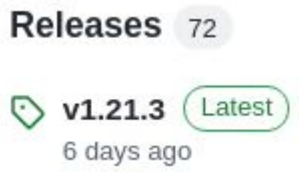
```
pip install
opencv-python
```




Releases 57

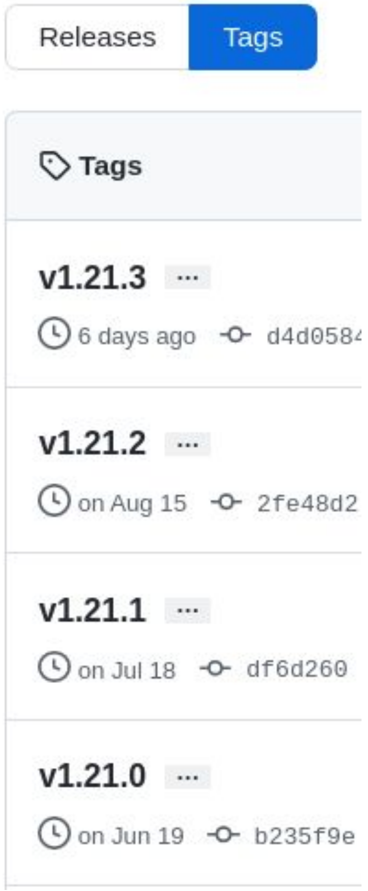
3.4.16.57 Latest

5 days ago


Releases 72

v1.21.3 Latest

6 days ago



Tags

v1.21.3 ···
6 days ago d4d058

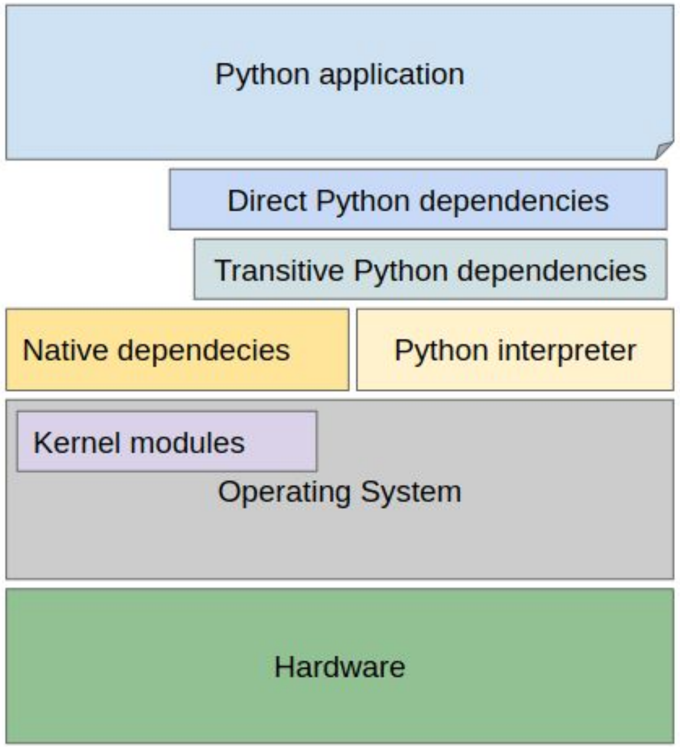
v1.21.2 ···
on Aug 15 2fe48d2

v1.21.1 ···
on Jul 18 df6d260

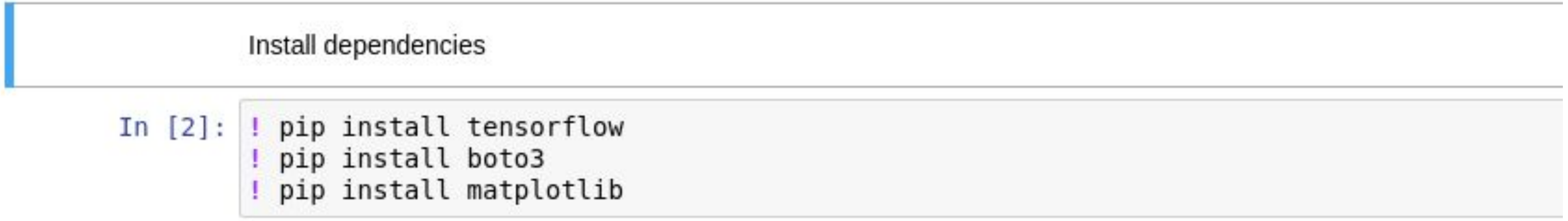
v1.21.0 ···
on Jun 19 b235f9e

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

What problems are we trying to solve?



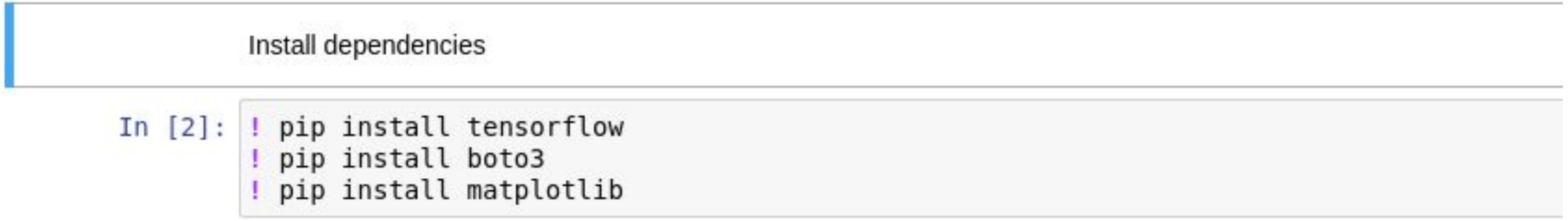
What problems are we trying to solve?



Install dependencies

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

What problems are we trying to solve?

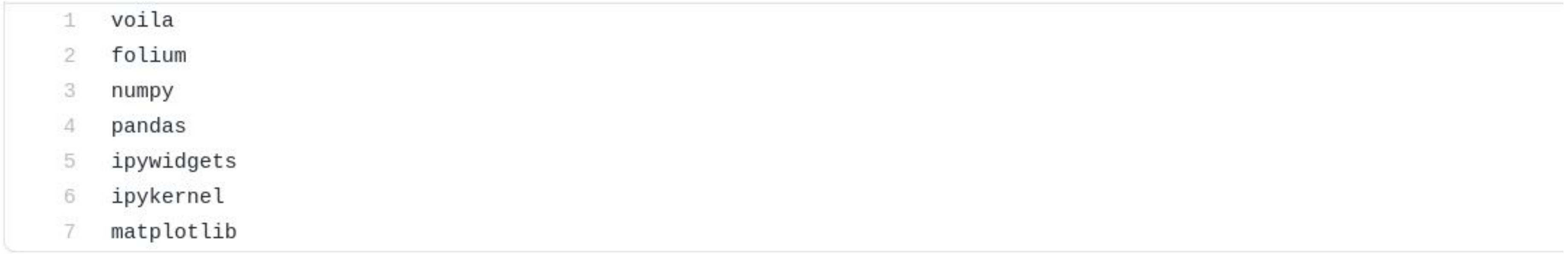

Install dependencies

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

This will not guarantee reproducibility!

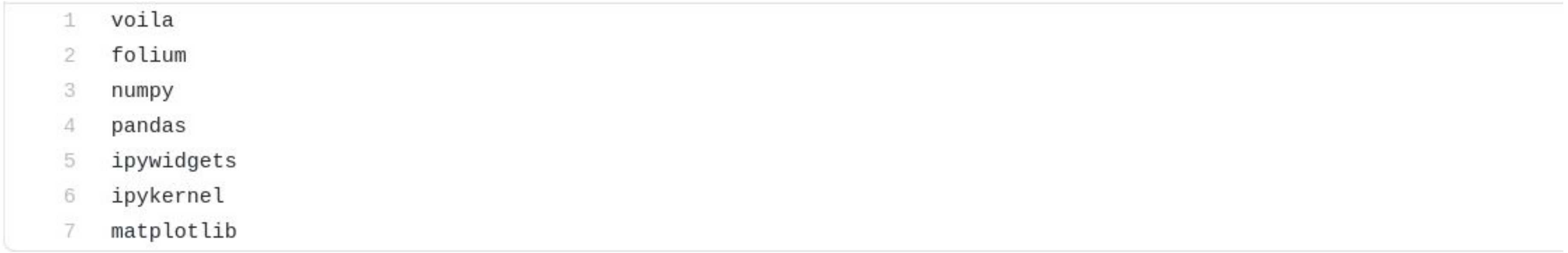
Red Hat

What problems are we trying to solve?



```
voila
folium
numpy
pandas
ipywidgets
ipykernel
matplotlib
```

What problems are we trying to solve?



```
voila
folium
numpy
pandas
ipywidgets
ipykernel
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**.

Source: https://www.linkedin.com/pulse/jupyter-notebooks-production-marek-%C4%8Derm%C3%A1k/

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.**

Source: https://www.linkedin.com/pulse/jupyter-notebooks-production-marek-%C4%8Derm%C3%A1k/

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.**

Source: https://www.linkedin.com/pulse/jupyter-notebooks-production-marek-%C4%8Derm%C3%A1k/

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

Reproducibility!

[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**.

Source: https://www.linkedin.com/pulse/jupyter-notebooks-production-marek-%C4%8Derm%C3%A1k/

Project Thoth

ODSC West 2021

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

Source: https://thoth-station.ninja/

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, TTCI, etc,..) from different software development platform (Github, GitLab, Pagure, etc...)

Source: https://thoth-station.ninja/

Thoth Recommendation types

- Latest
- Stable
- Security
- Performance
- Testing

Source: https://thoth-station.ninja/

How do we use this knowledge?

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

Check the video

Source: https://thoth-station.ninja/

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)**

Source: https://thoth-station.ninja/

How does Thoth help solve the problem?

ODSC West 2021

jupyterlab-requirements

JupyterLab extension for dependency management and optimization

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

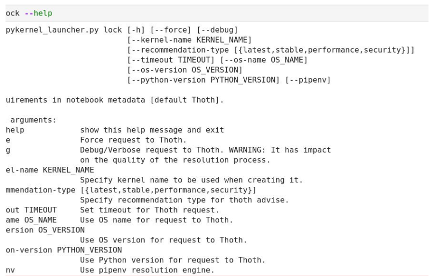
- **requirements (Pipfile)**;
- **requirements locked** with all versions and hashes of libraries (direct and transitive ones) **(Pipfile.lock)**;
- **dependency resolution engine** used **(Thoth or Pipenv)**;
- configuration file containing runtime environment (only for Thoth resolution engine).

Source:https://pypi.org/project/jupyterlab-requirements/

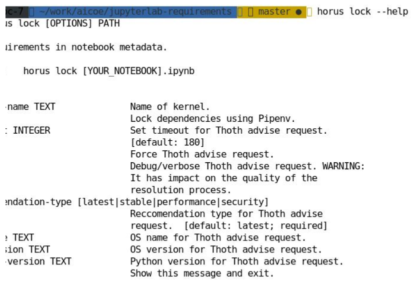
How does Thoth helps solving the problem?

jupyterlab-requirements

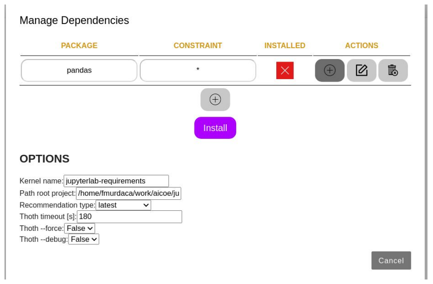
JupyterLab extension for dependency management and optimization



```
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]

uirements 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.
mmendation-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.
```



```
c-7  ~/work/aicoe/jupyterlab-requirements   master ●  horus lock --help
us lock [OPTIONS] PATH

uirements 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.
endation-type [latest|stable|performance|security]
                          Reccomendation type for Thoth advise
                          request.  [default: latest; required]
e TEXT                    OS name for Thoth advise request.
sion TEXT                 OS version for Thoth advise request.
version TEXT              Python version for Thoth advise request.
                          Show this message and exit.
```



Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
pandas	*	✗	

OPTIONS

Kernel name: jupyterlab-requirements
Path root project: /home/fmurdaca/work/aicoe/ju
Recommendation type: latest
Thoth timeout [s]: 180
Thoth --force: False
Thoth --debug: False

Jupyter magic commands

available directly in your notebook cells when you start a notebook

CLI

that you can run from terminal or integrated in pipelines

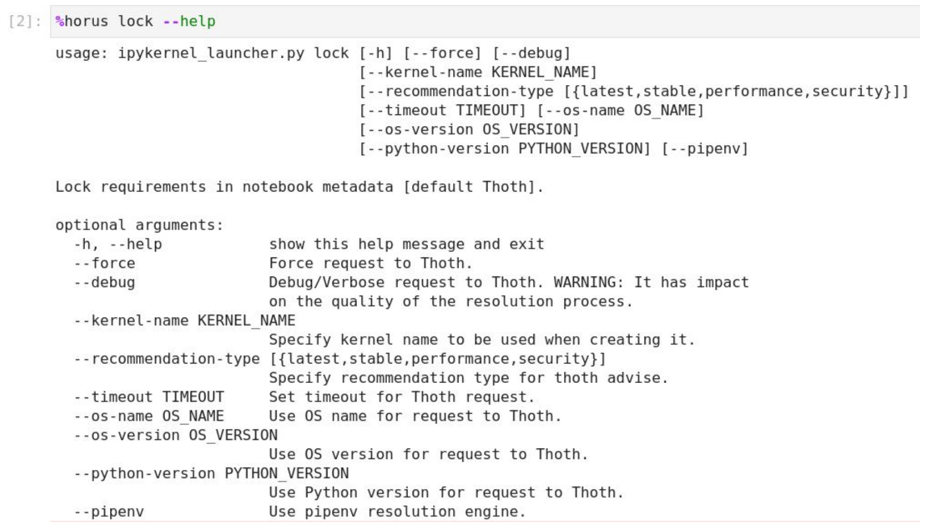
UI

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

Source:https://pypi.org/project/jupyterlab-requirements/

jupyterlab-requirements

Jupyter magic commands



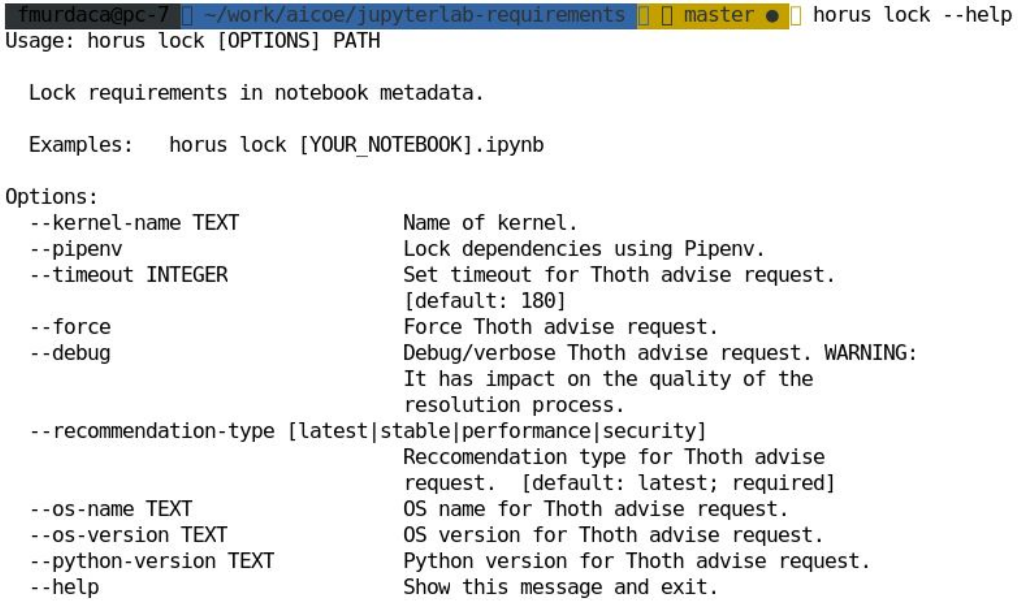
```
[2]: %horus lock --help

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)

Source:https://pypi.org/project/jupyterlab-requirements/

jupyterlab-requirements

Command Line Interface (CLI)



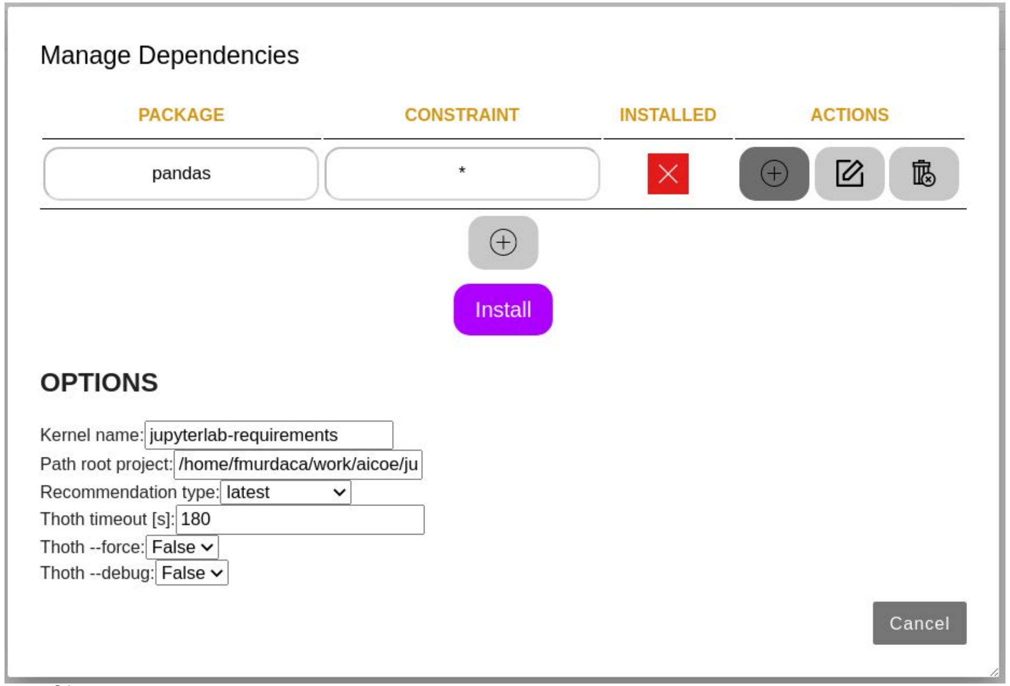
```
 fmurdaca@pc-7  ~/work/aicoe/jupyterlab-requirements   master ●  horus lock --help
Usage: horus lock [OPTIONS] PATH

  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]
                                  Reccomendation 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)

Source:https://pypi.org/project/jupyterlab-requirements/

jupyterlab-requirements

User Interface (UI)



Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
pandas	*	✗	

OPTIONS

Kernel name: jupyterlab-requirements
Path root project: /home/fmurdaca/work/aicoe/ju
Recommendation type: latest
Thoth timeout [s]: 180
Thoth --force: False
Thoth --debug: False

- Interactive UI to handle dependencies

Source:https://pypi.org/project/jupyterlab-requirements/

jupyterlab-requirements Python package

```
pip install jupyterlab-requirements

jupyter lab
```

Source: https://pypi.org/project/jupyterlab-requirements/

Dependency management tutorial for Jupyter Notebooks

ODSC West 2021

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

Source: https://github.com/AICoE/manage-dependencies-tutorial

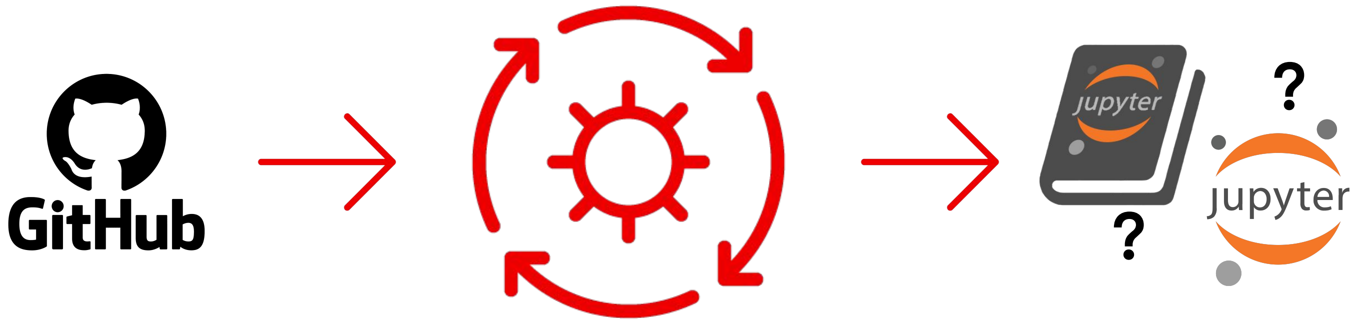
Operate First

https://operate-first.cloud

https://github.com/operate-first

Source: https://www.operate-first.cloud/

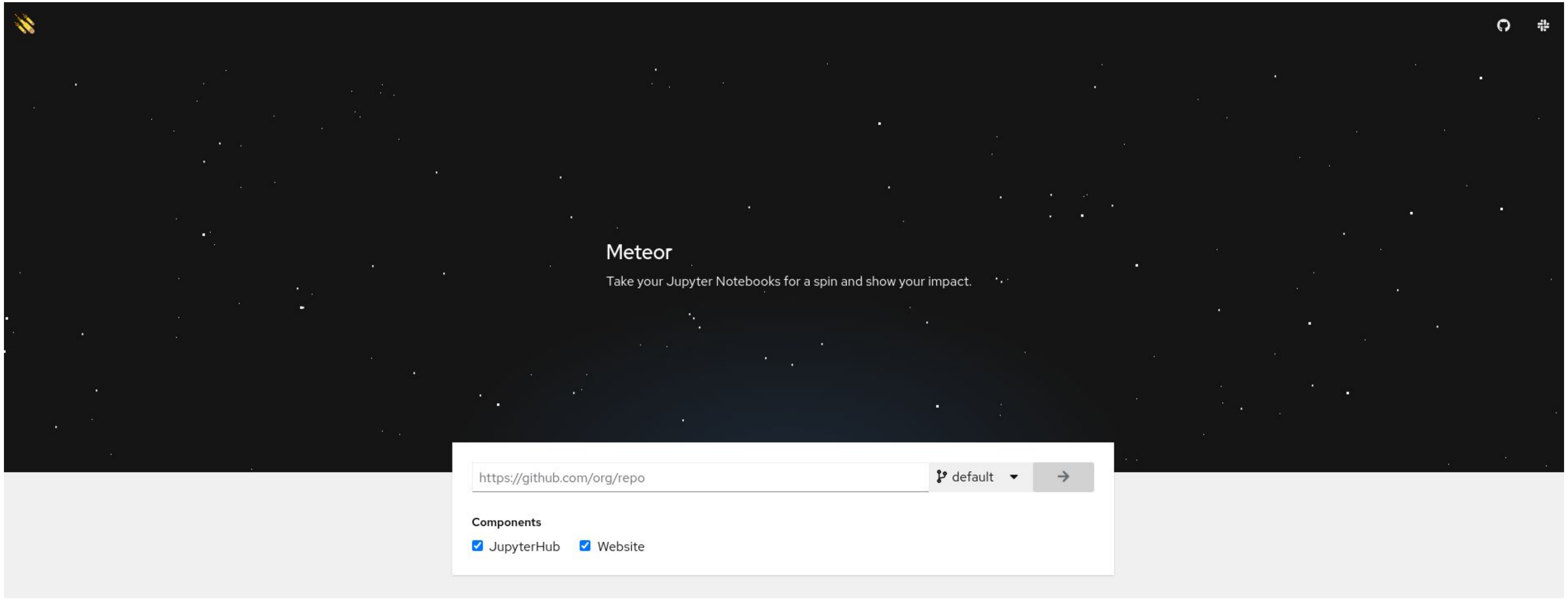
Project Meteor



AICoE CI,
Tekton pipelines,
Thoth Advise

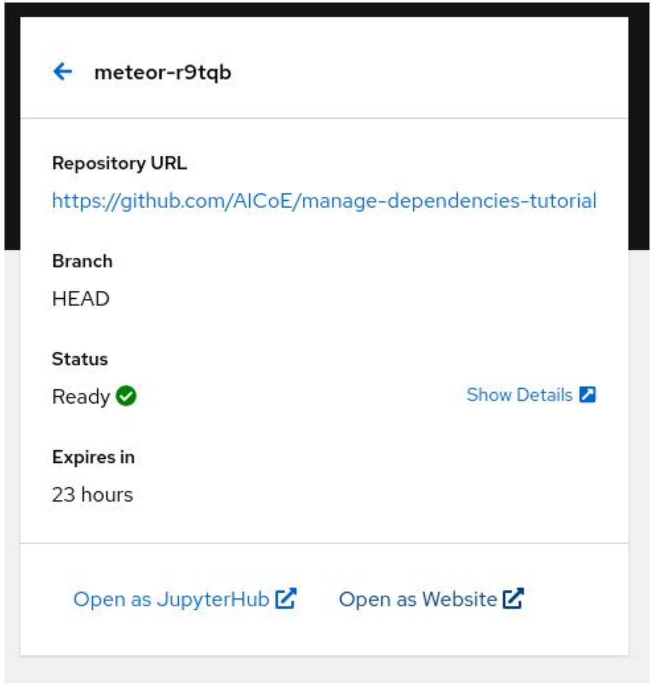
Source: https://github.com/AICoE/meteor

Project Meteor


Meteor
Take your Jupyter Notebooks for a spin and show your impact.
https://github.com/org/repo
default
Components
JupyterHub
Website

Source: https://shower.meteor.zone/

Project Meteor



meteor-r9tqb

Repository URL
https://github.com/AICoE/manage-dependencies-tutorial

Branch
HEAD

Status
Ready ✓

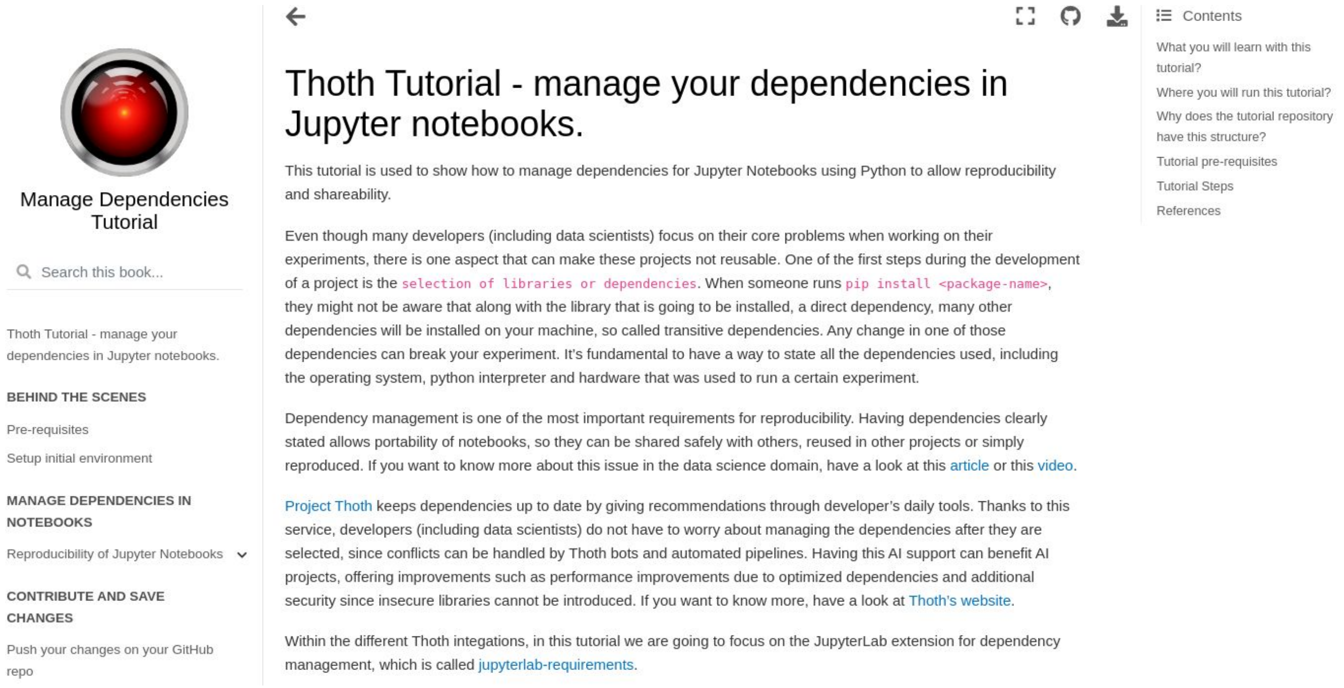
Show Details

Expires in
23 hours

Open as JupyterHub
Open as Website

Source: https://github.com/AICoE/meteor

Project Meteor - Jupyter Book



Manage Dependencies Tutorial

Search this book...

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.

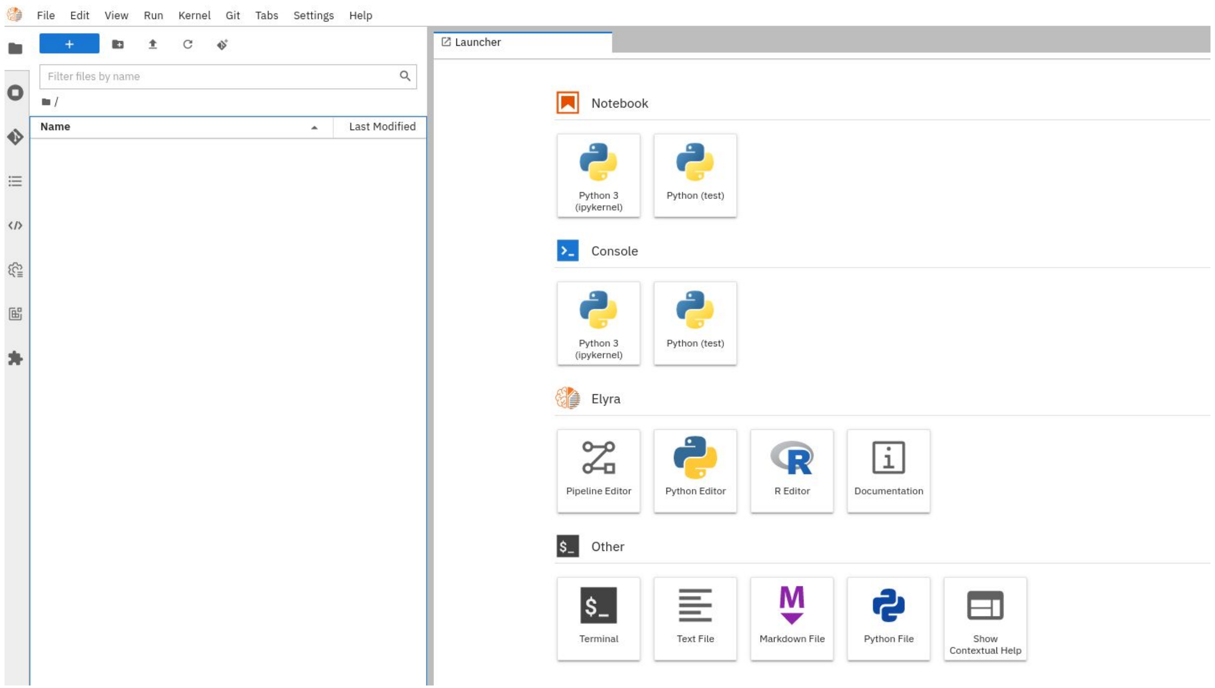
Within the different Thoth integations, in this tutorial we are going to focus on the JupyterLab extension for dependency management, which is called jupyterlab-requirements.

Contents

Source: https://github.com/AICoE/meteor

Red Hat

Project Meteor - JupyterLab environment



Source: https://github.com/AICoE/meteor

Conclusions

ODSC West 2021

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**.

Red Hat

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.*

*Arguably, reproducibility is not only about same versions, other factors have to be considered as well, like hardware and architecture, or random variables.

Conclusions - Contacts

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/UClUIDuq_hQ6vlzmqM59B2Lw/videos**
- **Tutorial** **https://github.com/AICoE/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