How to make your Python Jupyter Notebook Standalone and Reproducible to allow others to replicate your experiments

PyCon US 2022

Maya Costantini

Francesco Murdaca

$whoami

@MayaCostantini

mayaCostantini

Associate Software Engineer, Red Hat’s Office of the CTO

Passionate about Python & Open Source contributor

Paris, France

Thoth

Jupyter Notebooks

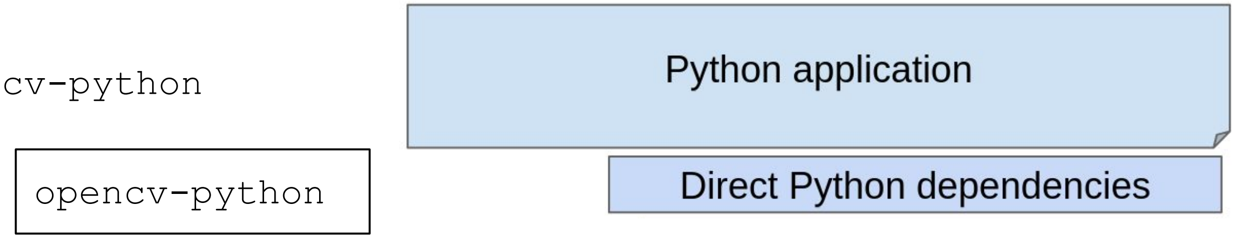
An Open Source web application to create documents that contain live code, equations, visualizations and narrative text

- Support for over 40 programming languages
- Share interactive code
- Rich, interactive output: HTML, images, videos, etc
- Leverage Big Data tools

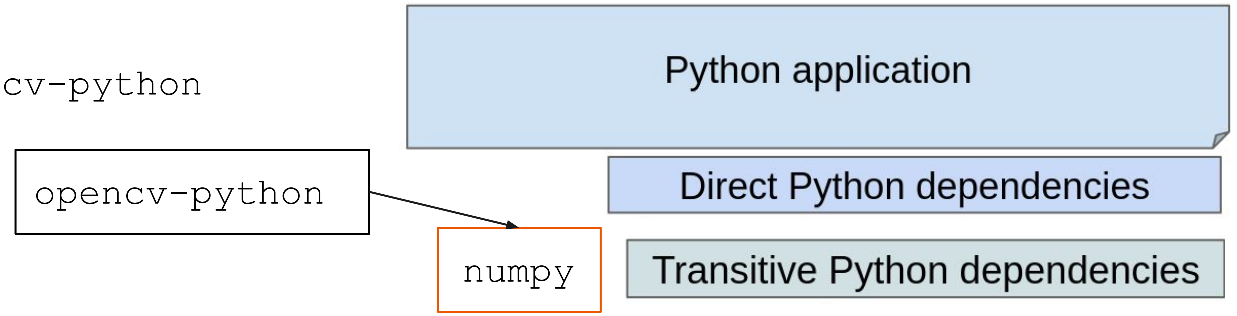
Source: https://jupyter.org

What problems are we trying to solve?

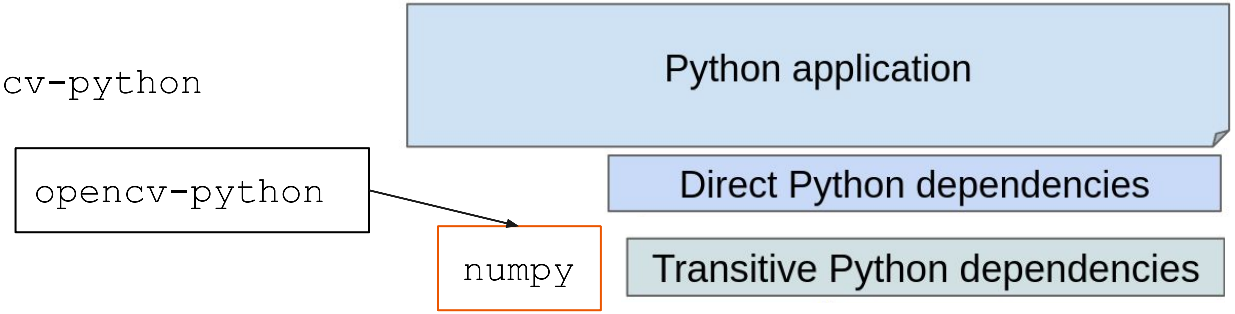
```
pip install opencv-python
pip install opencv-python
```



```
pip install opencv-python
```

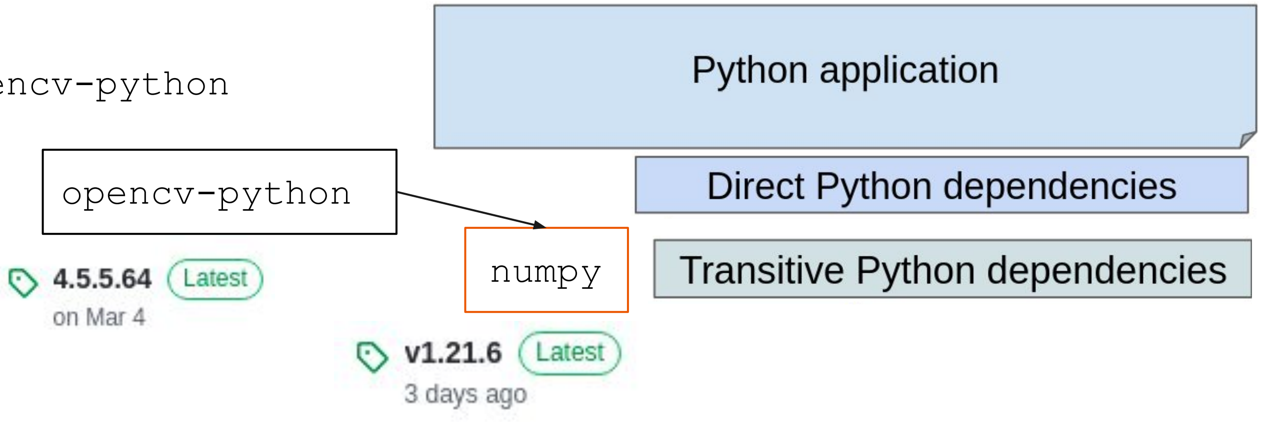


```
pip install opencv-python
```



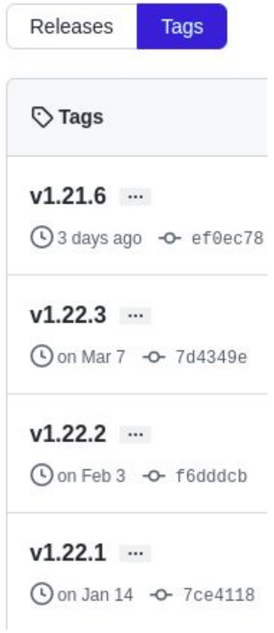
What about versions?

```
pip install opencv-python
```



What about versions?

```
pip install opencv-python
```


Releases Tags

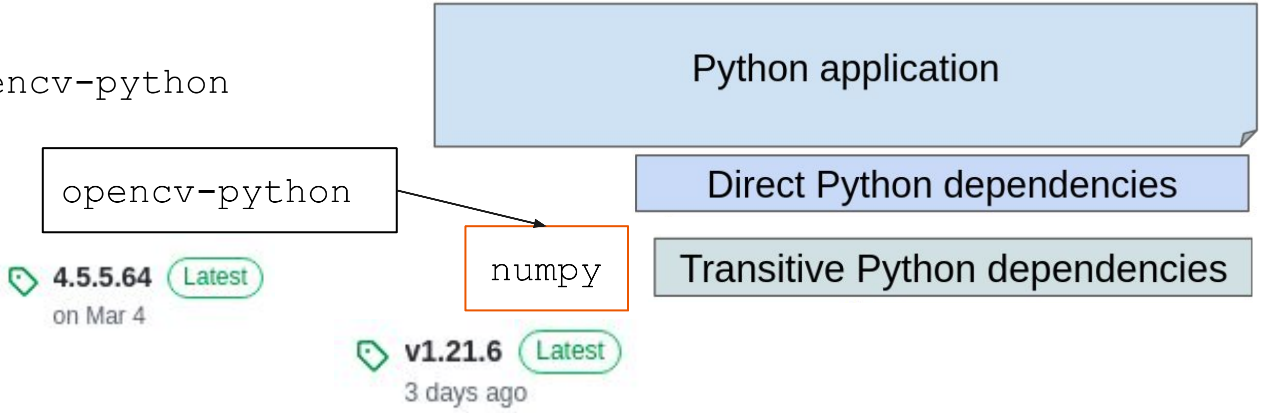
Tags

v1.21.6 ··· 3 days ago ef0ec78

v1.22.3 ··· on Mar 7 7d4349e

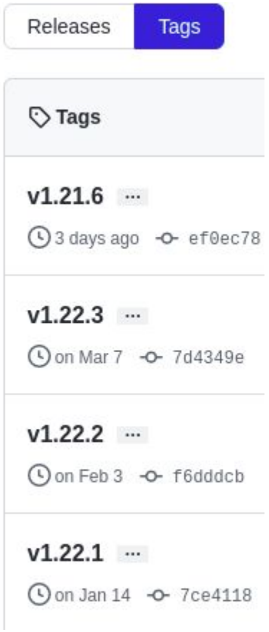
v1.22.2 ··· on Feb 3 f6dddcb

v1.22.1 ··· on Jan 14 7ce4118



What about versions?

```
pip install opencv-python
```



Releases | Tags

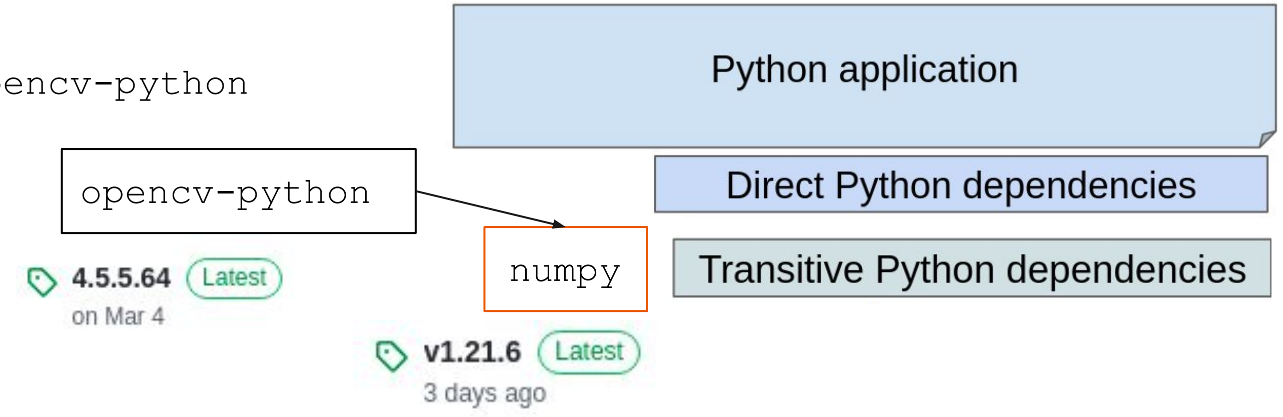
Tags

v1.21.6 ···
3 days ago ef0ec78

v1.22.3 ···
on Mar 7 7d4349e

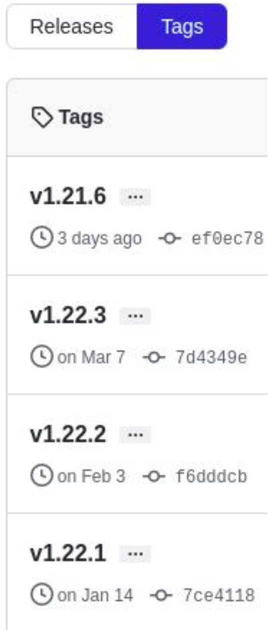
v1.22.2 ···
on Feb 3 f6dddcb

v1.22.1 ···
on Jan 14 7ce4118



What about versions?
What about hashes?

```
pip install opencv-python
```



Releases Tags

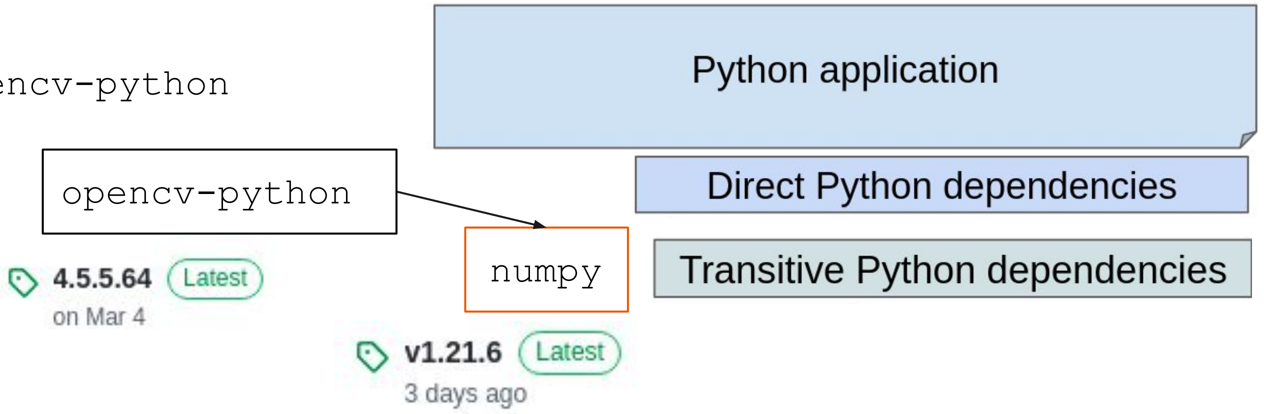
Tags

v1.21.6 ···
3 days ago ef0ec78

v1.22.3 ···
on Mar 7 7d4349e

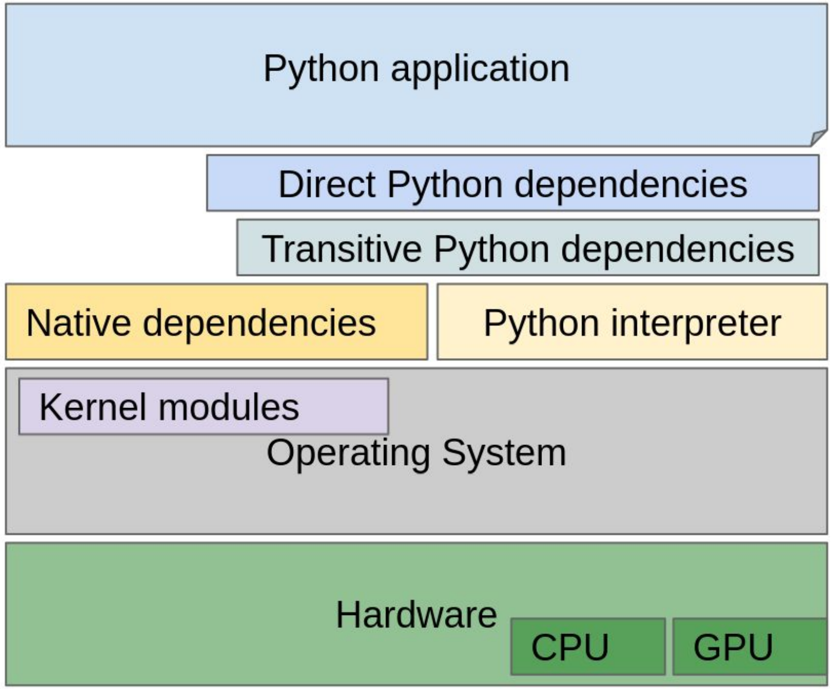
v1.22.2 ···
on Feb 3 f6dddcb

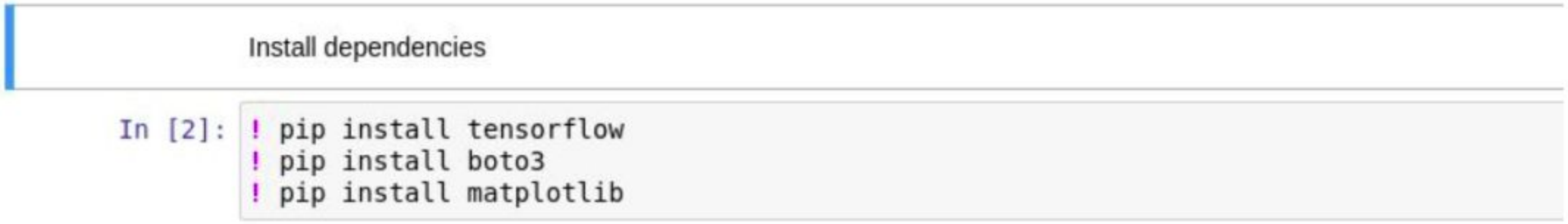
v1.22.1 ···
on Jan 14 7ce4118



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

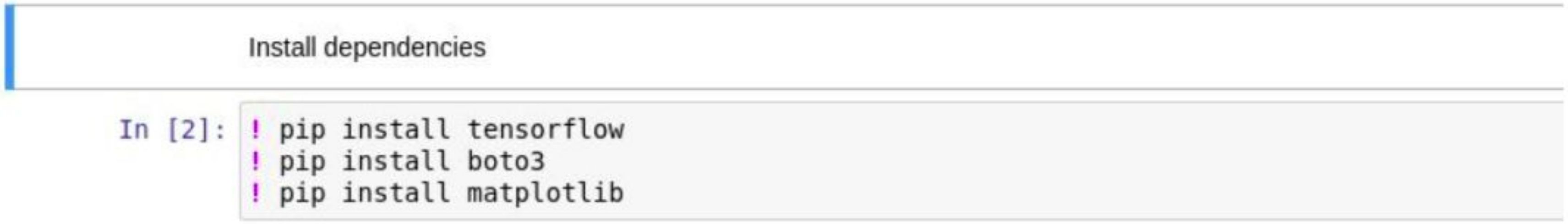
```
pip install opencv-python
```





Install dependencies

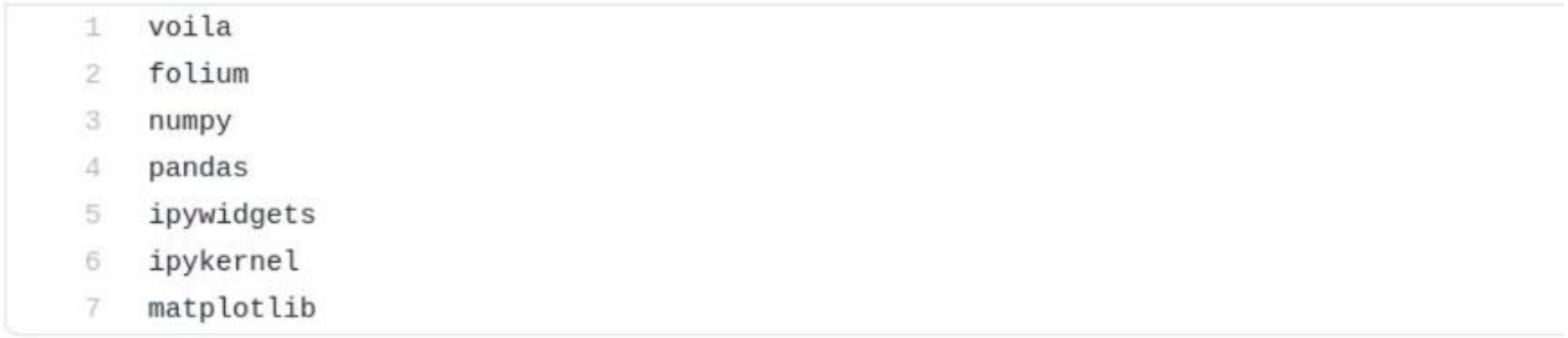
```
In [2]: ! pip install tensorflow
        ! pip install boto3
        ! pip install matplotlib
```



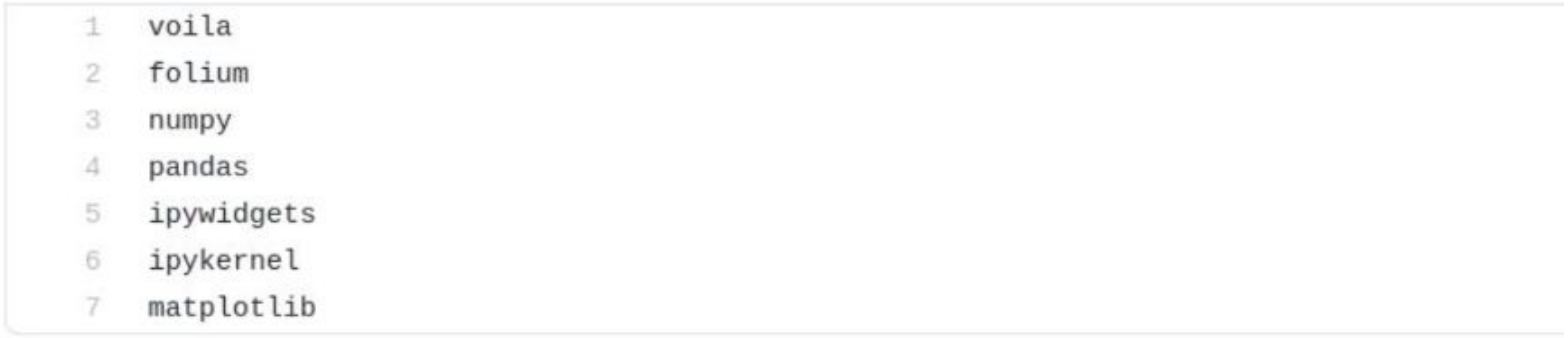
Install dependencies

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

This does not guarantee any reproducibility!



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



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

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

Jupyter Notebooks are by default **NOT** standalone

It is not uncommon that **no manifest files are provided** and hence notebook users must **find out dependencies themselves**

Managing dependencies

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

Containerization

A specialized tool or a **custom Dockerfile** is needed so that all notebook requirements are present in the resulting image

Sharing

The consumer must first **set up manually an environment** using provided manifest files

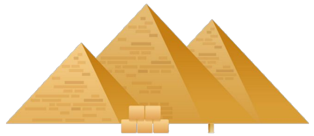
Difficulties for both authors and consumers

Authors have to...	Consumers have to...
Create an environment	Create an environment
Install dependencies in the environment	Install dependencies in the environment
Create/update custom kernel [optional]	Create/update custom kernel [optional]
Create/update manifest files [optional]	

⚠ **Reproducibility**

How can Thoth help you manage dependencies in your Jupyter Notebook?

Project Thoth



thoth-station.ninja @ThothStation github.com/thoth-station

Help Python developers and Data Scientists create healthy applications

Solving Python dependencies using Machine Learning in the cloud

Team of 10 engineers, ~50 contributors

Open Source project, contributions are welcome!

Red Hat

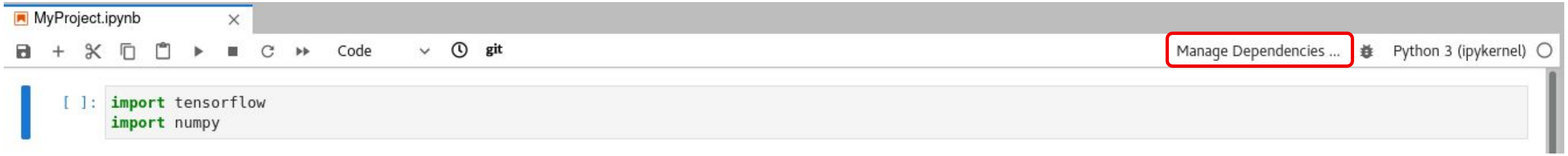
An interactive, extensible web interface for Project Jupyter

Thoth’s extension for JupyterLab : `jupyterlab-requirements`

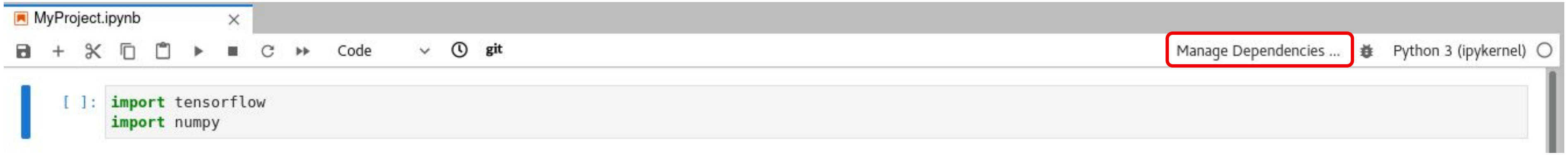
Manage your dependencies and store everything in the **Jupyter**

Notebook metadata:

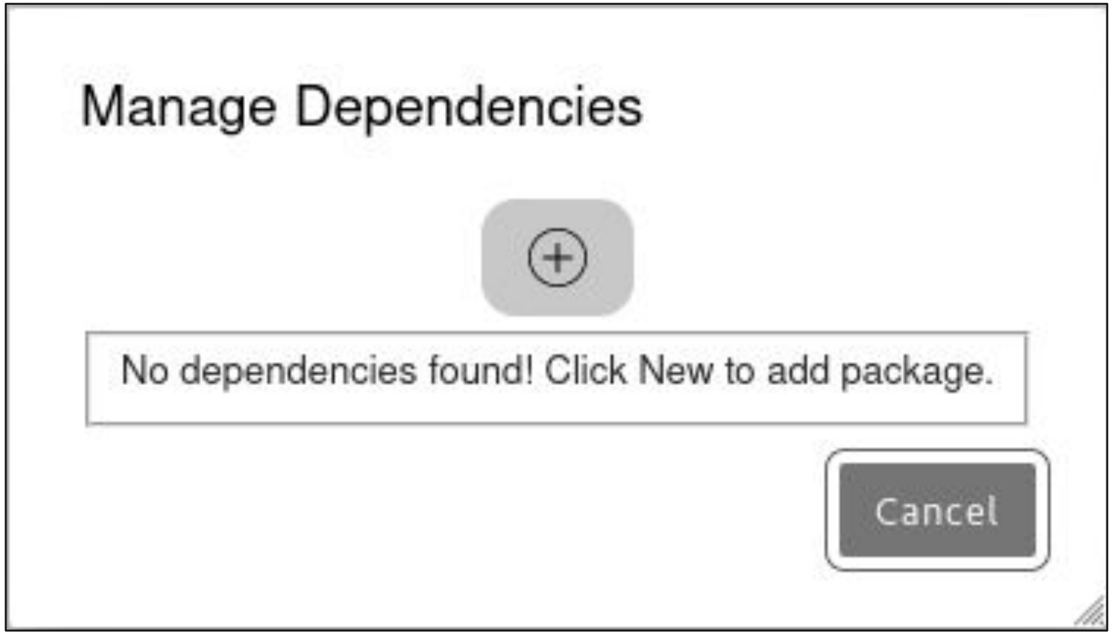
- Manage a notebook requirements without leaving it
- Provide a **unique** and **optimized** environment for each notebook
- Solve dependencies with **Thoth’s resolution engine**



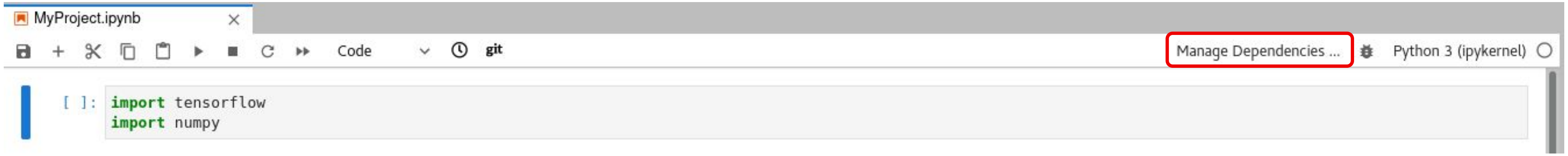
```
import tensorflow
import numpy
```



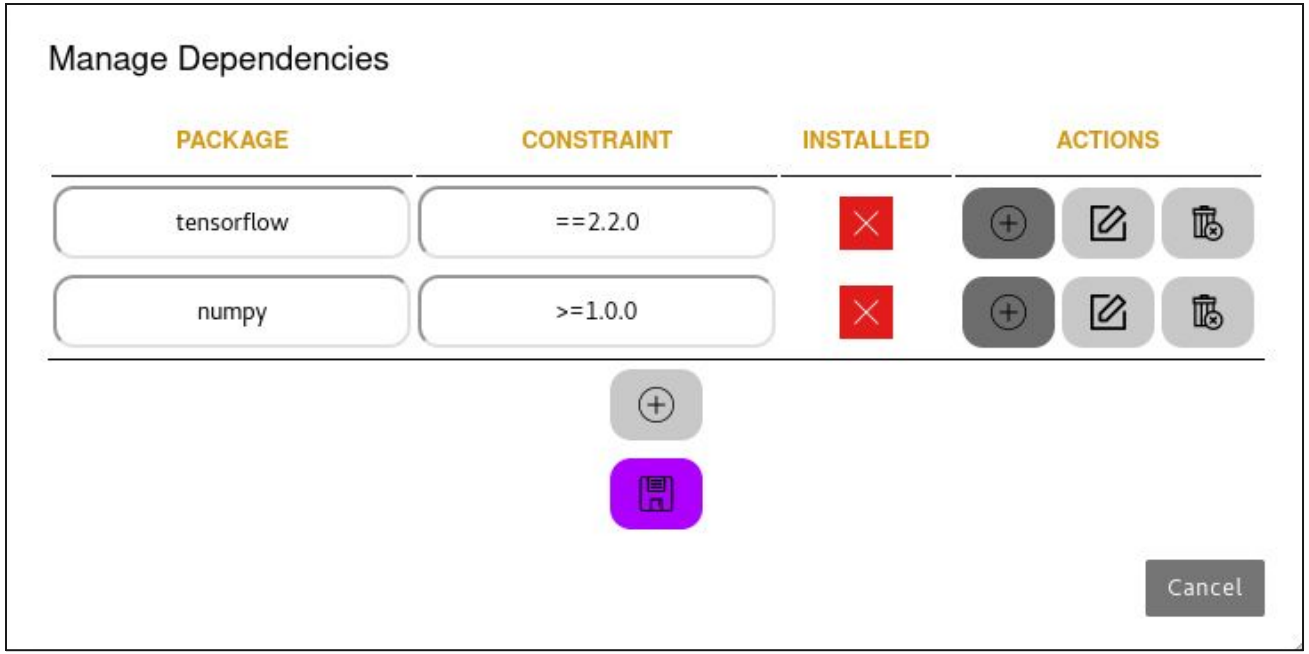
```
[ ]: import tensorflow
     import numpy
```



No dependencies found! Click New to add package.

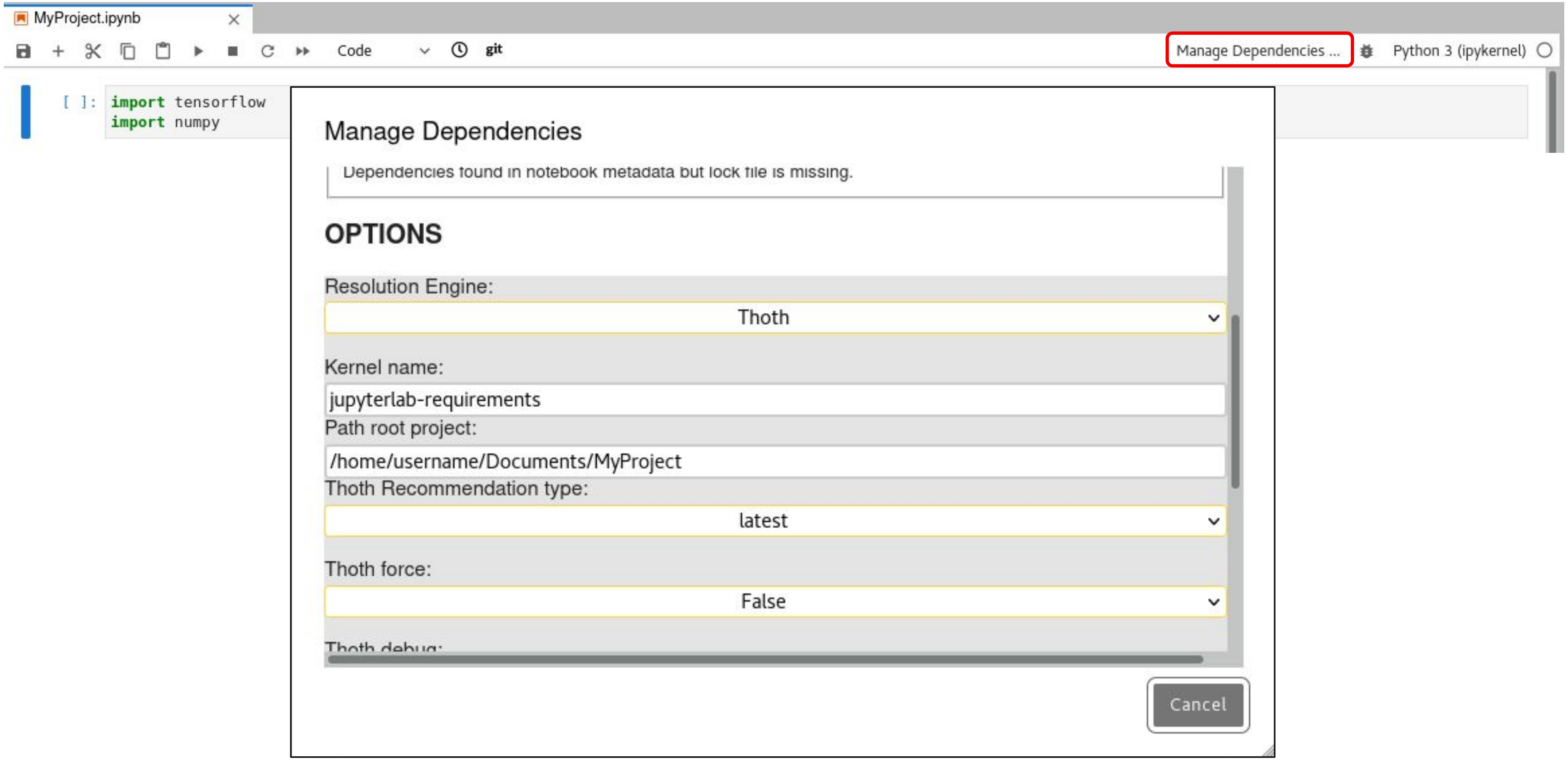


```
[ ]: import tensorflow
     import numpy
```



Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	❌	
numpy	>=1.0.0	❌	



```
[ ]: import tensorflow
     import numpy
```

Manage Dependencies

Dependencies found in notebook metadata but lock file is missing.

OPTIONS

Resolution Engine: Thoth

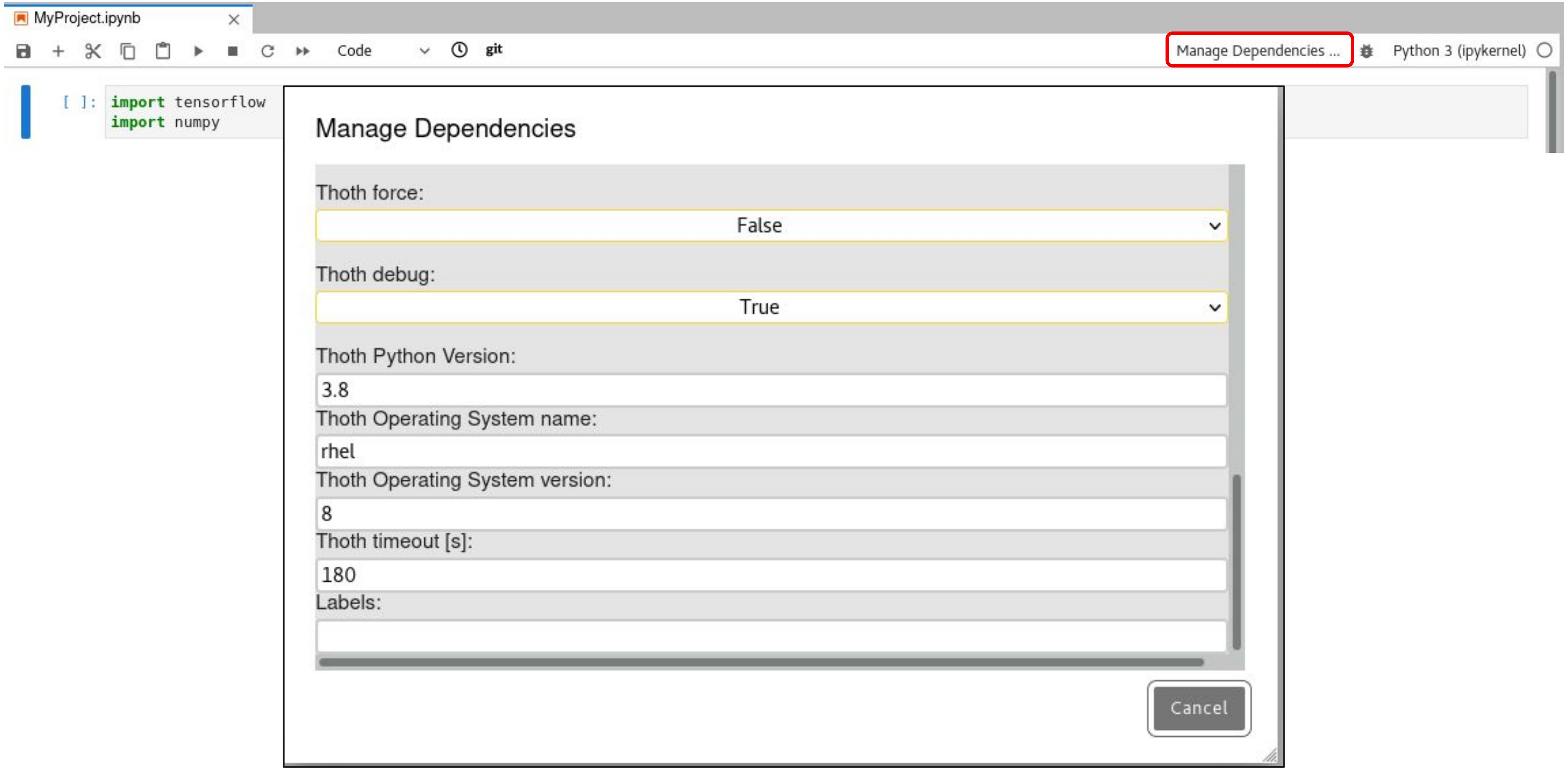
Kernel name: jupyterlab-requirements

Path root project: /home/username/Documents/MyProject

Thoth Recommendation type: latest

Thoth force: False

Thoth debug:



```
[ ]: import tensorflow
     import numpy
```

Manage Dependencies

Thoth force: False

Thoth debug: True

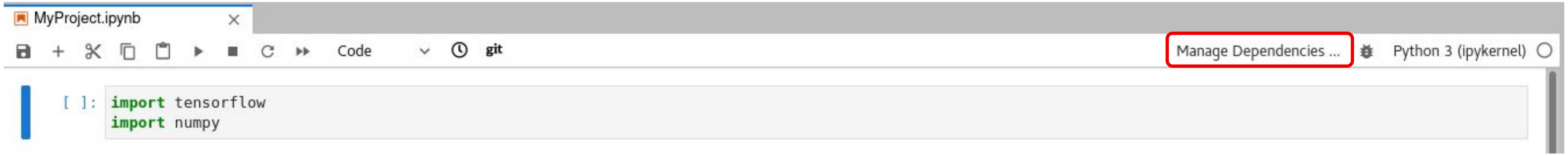
Thoth Python Version: 3.8

Thoth Operating System name: rhel

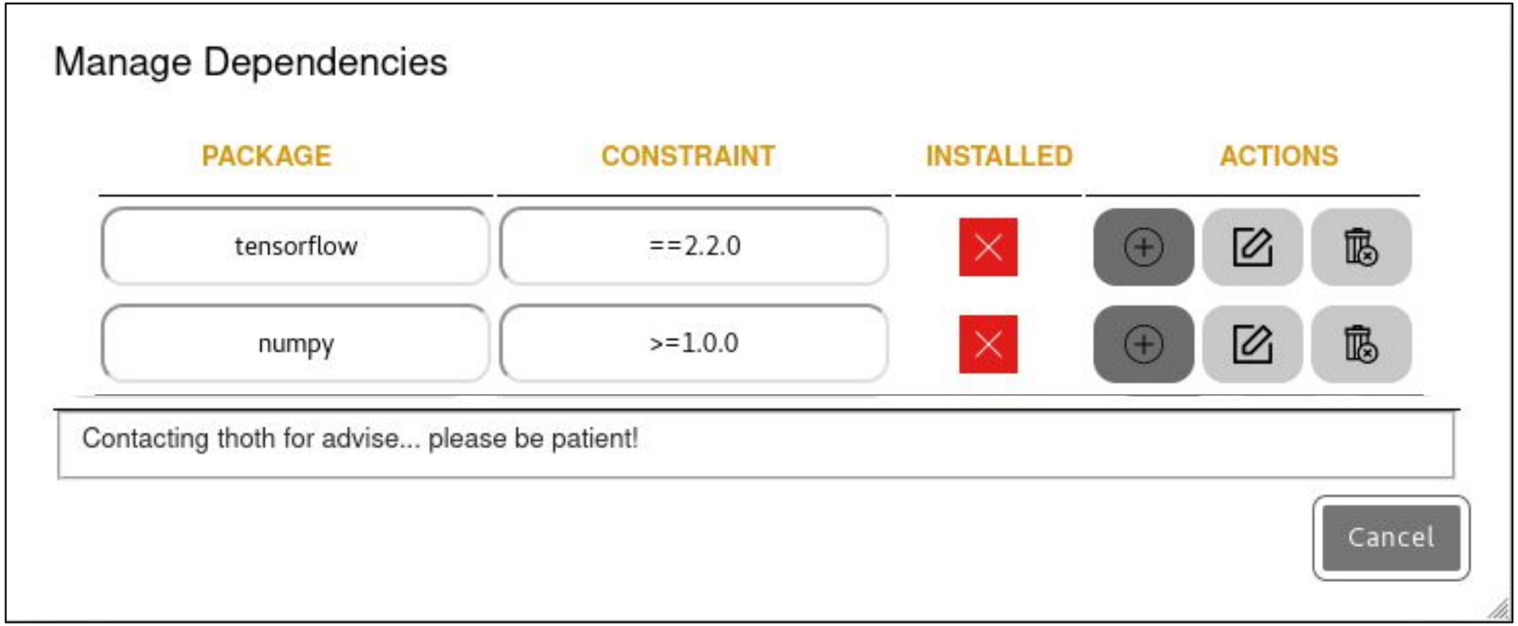
Thoth Operating System version: 8

Thoth timeout [s]: 180

Labels:



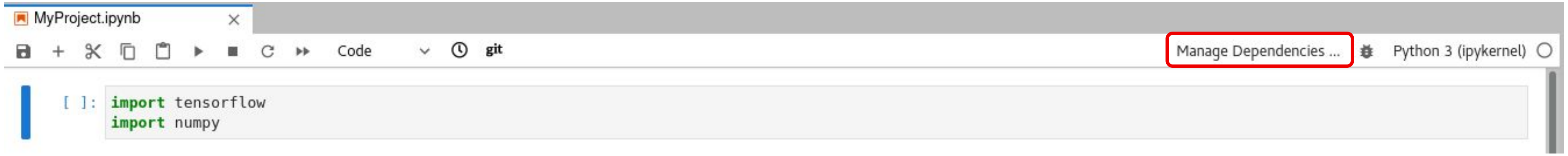
```
[ ]: import tensorflow
     import numpy
```



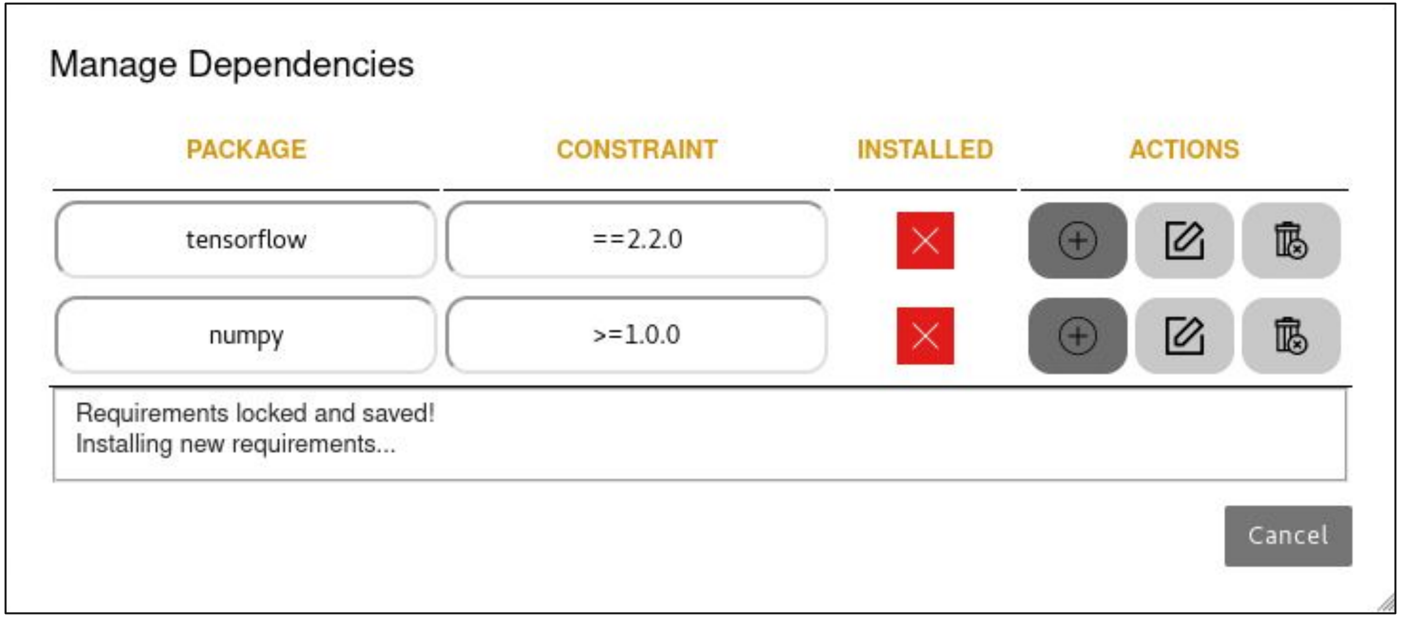
Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	❌	⊕ ✎ 🗑
numpy	>=1.0.0	❌	⊕ ✎ 🗑

Contacting thoth for advise... please be patient!

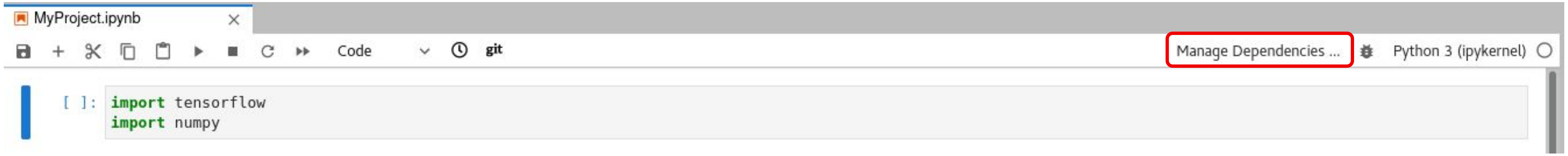


```
[ ]: import tensorflow
     import numpy
```

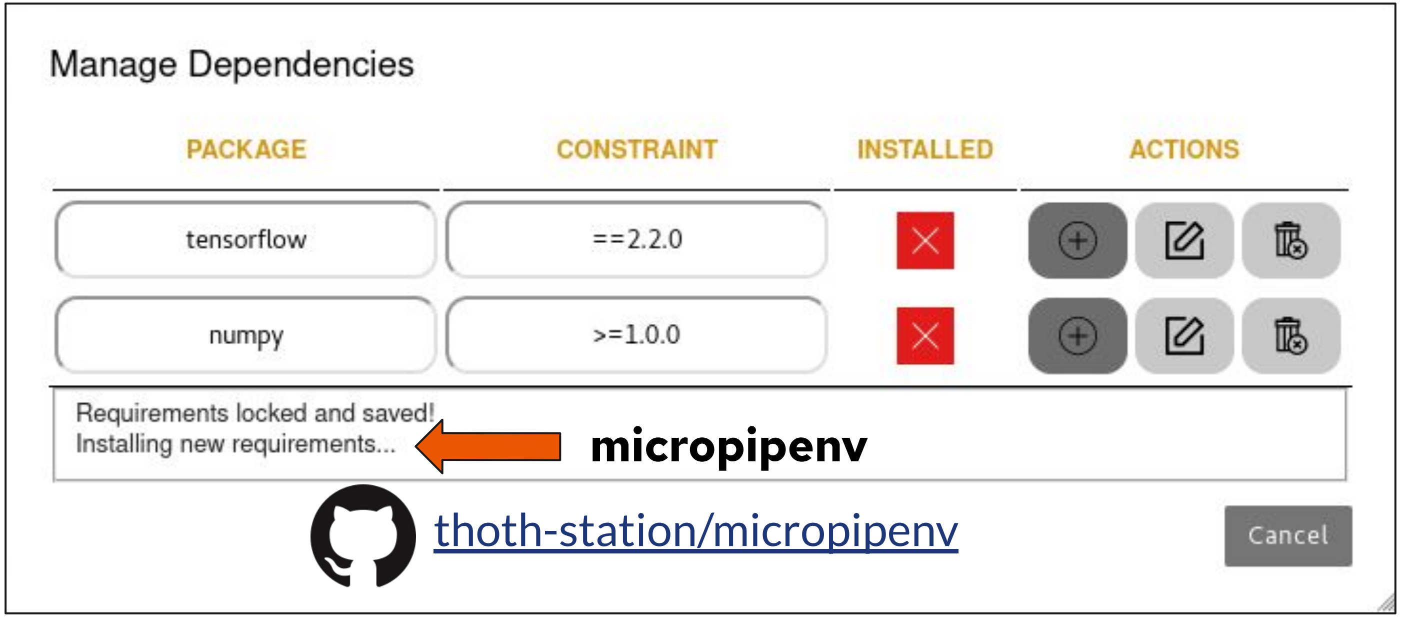


PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	✗	
numpy	>=1.0.0	✗	

Requirements locked and saved!
Installing new requirements...



```
[ ]: import tensorflow
     import numpy
```



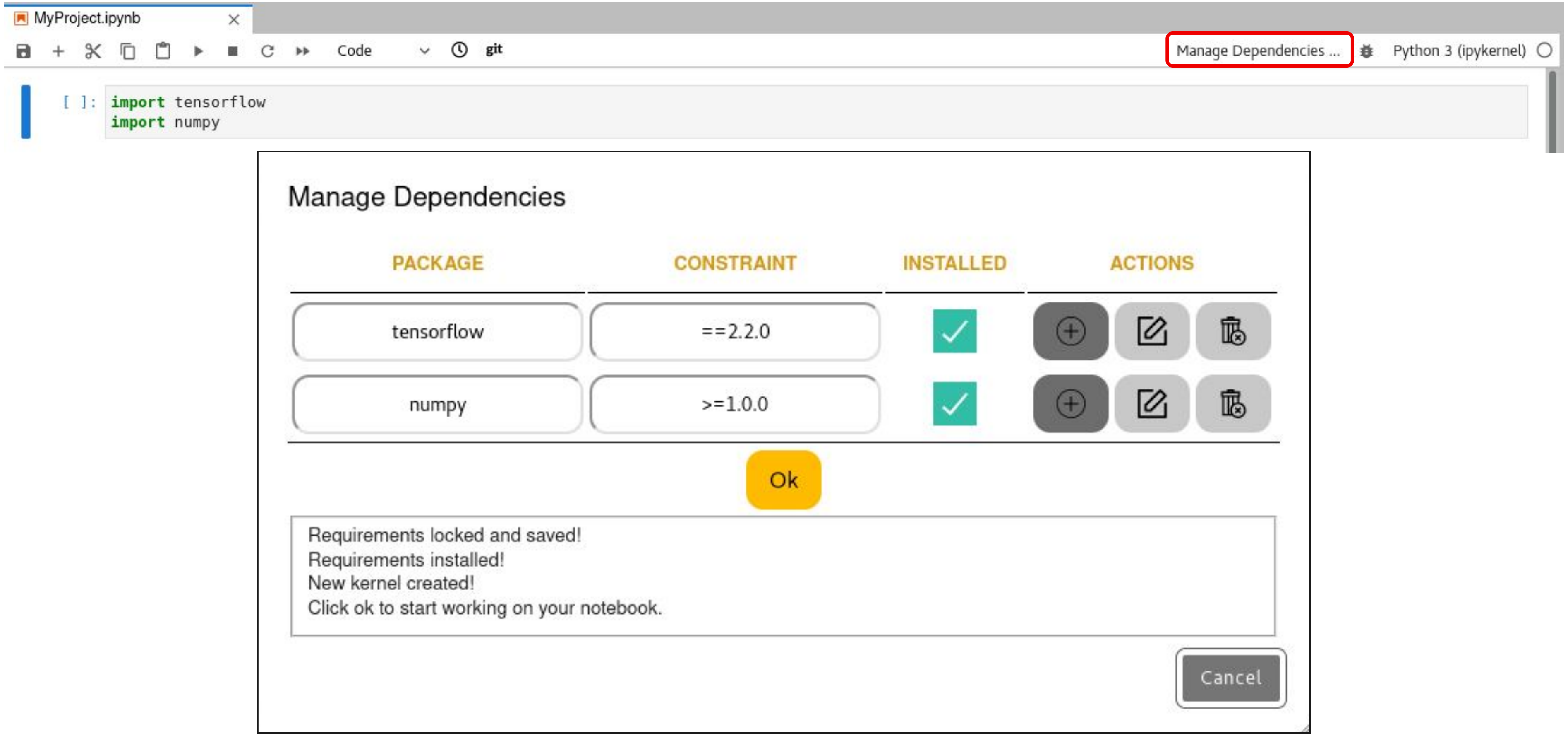
Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	🟥	
numpy	>=1.0.0	🟥	

```
Requirements locked and saved!
Installing new requirements...
```

micropipenv

thoth-station/micropipenv

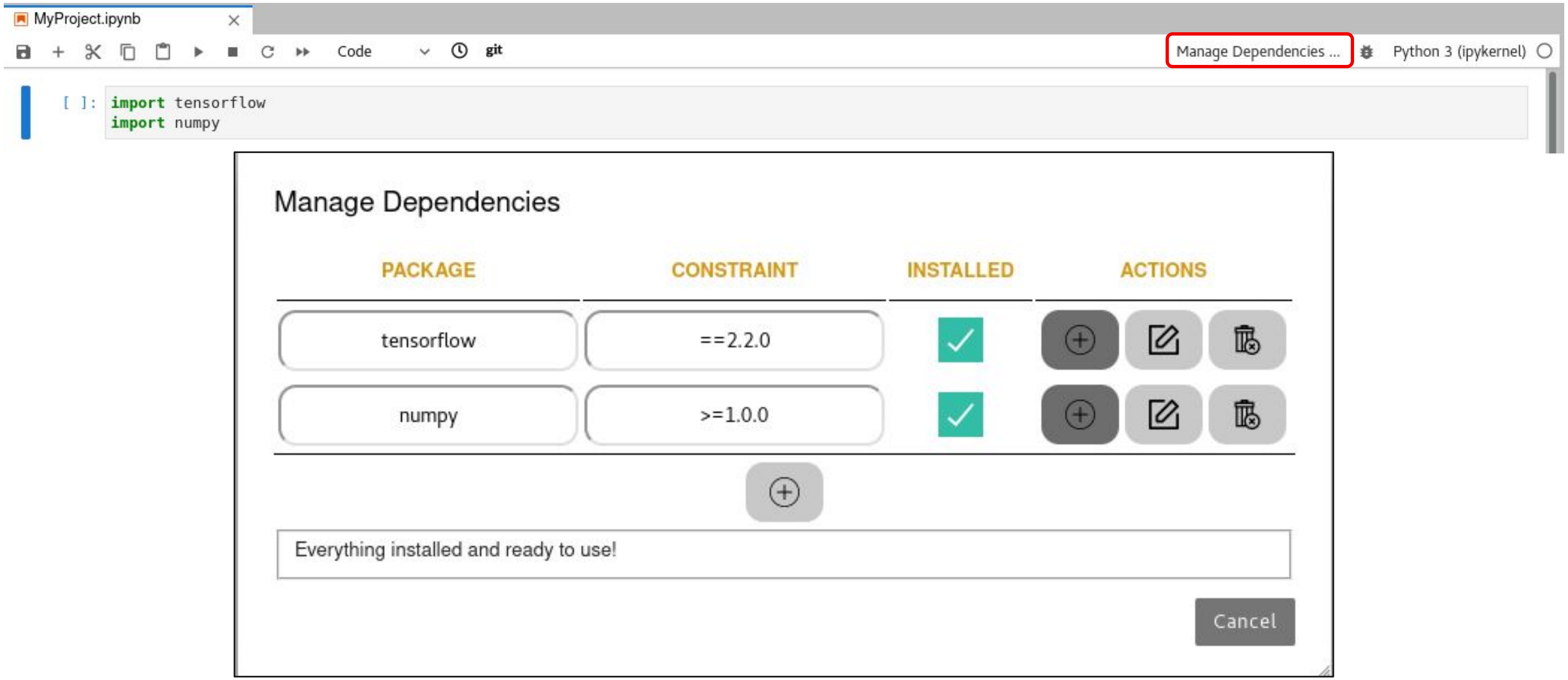


```
[ ]: import tensorflow
     import numpy
```

Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	✓	
numpy	>=1.0.0	✓	

```
Requirements locked and saved!
Requirements installed!
New kernel created!
Click ok to start working on your notebook.
```

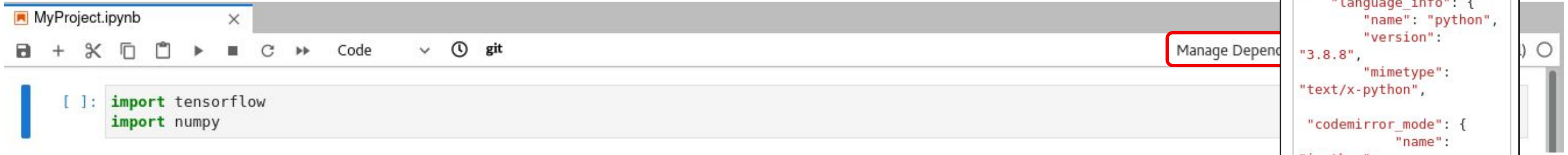


```
[ ]: import tensorflow
     import numpy
```

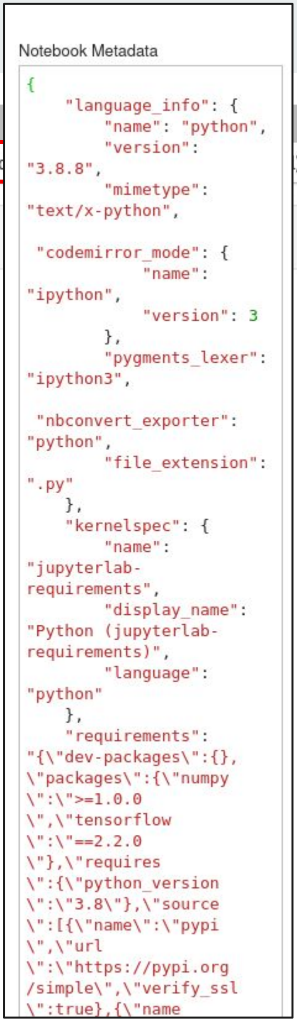
Manage Dependencies

PACKAGE	CONSTRAINT	INSTALLED	ACTIONS
tensorflow	==2.2.0	✓	
numpy	>=1.0.0	✓	

Everything installed and ready to use!

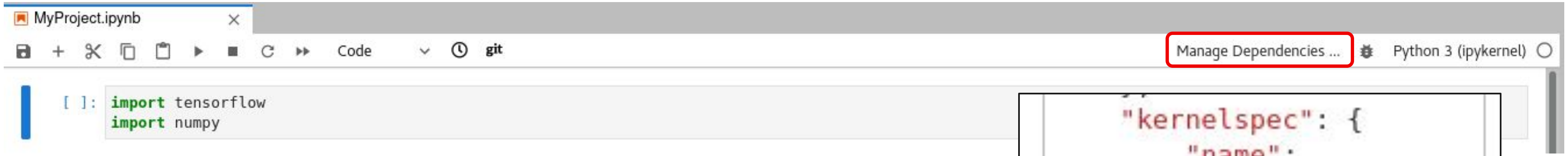


```
[ ]: import tensorflow
     import numpy
```



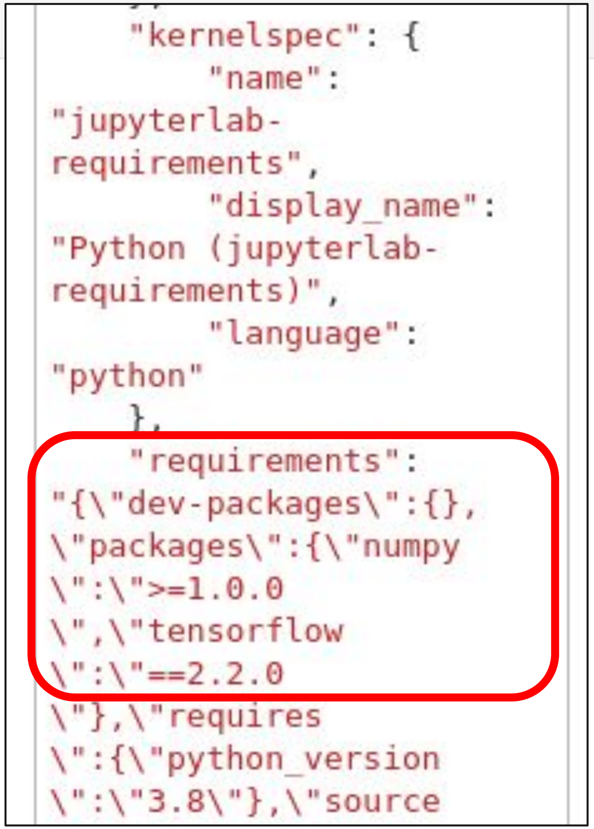
```
{
    "language_info": {
        "name": "python",
        "version":
"3.8.8",
        "mimetype":
"text/x-python",

 "codemirror_mode": {
            "name":
"ipython",
            "version": 3
        },
        "pygments_lexer":
"ipython3",

 "nbconvert_exporter":
"python",
        "file_extension":
".py"
    },
    "kernelspec": {
        "name":
"jupyterlab-
requirements",
        "display_name":
"Python (jupyterlab-
requirements)",
        "language":
"python"
    },
    "requirements":
"{\"dev-packages\":{},
\"packages\":{\"numpy
\":\">=1.0.0
\",\"tensorflow
\":\"==2.2.0
\"},\"requires
\":{\"python_version
\":\"3.8\"},\"source
\":[{\"name\":\"pypi
\",\"url
\":\"https://pypi.org
/simple\",\"verify_ssl
\":true},{\"name
```

dependencies are locked in the notebook metadata



```python
[ ]: import tensorflow
     import numpy
```

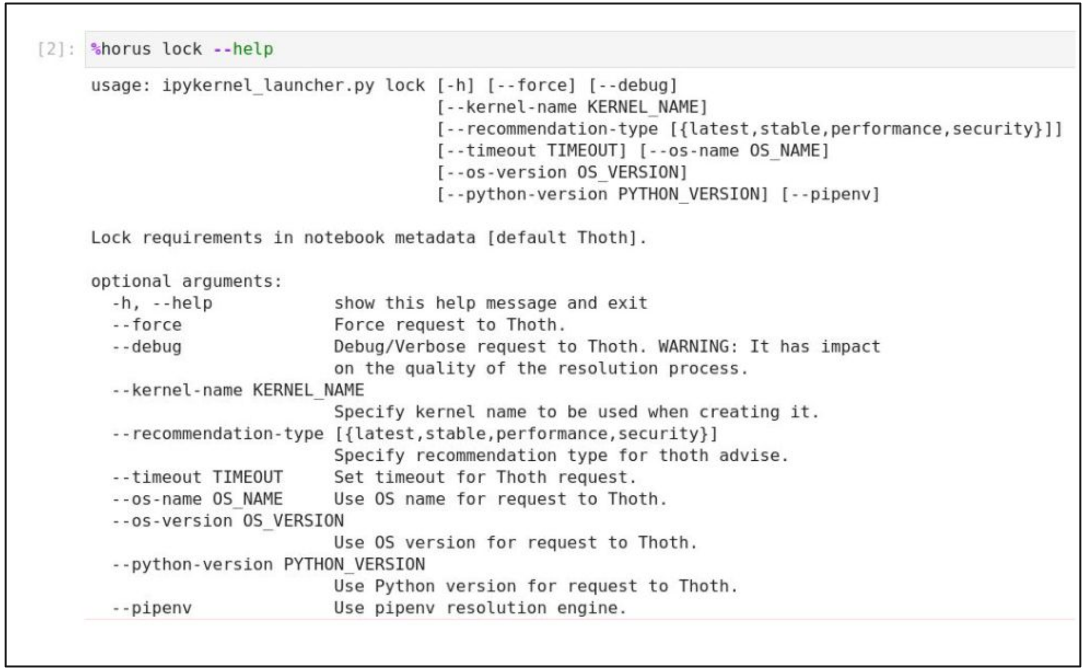
dependencies are locked in the notebook metadata



```
    "kernelspec": {
        "name":
"jupyterlab-
requirements",
        "display_name":
"Python (jupyterlab-
requirements)",
        "language":
"python"
    },
    "requirements":
"{\"dev-packages\":{},
\"packages\":{\"numpy
\":\">=1.0.0
\",\"tensorflow
\":\"==2.2.0
\"},\"requires
\":{\"python_version
\":\"3.8\"},\"source
```

%horus magic commands

Speed up development by managing dependencies **directly in notebook cells**



```
[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.
```

%horus magic commands

`%horus check`: Check notebook metadata about dependencies

`%horus convert`: Convert pip commands to horus commands to allow reproducibility

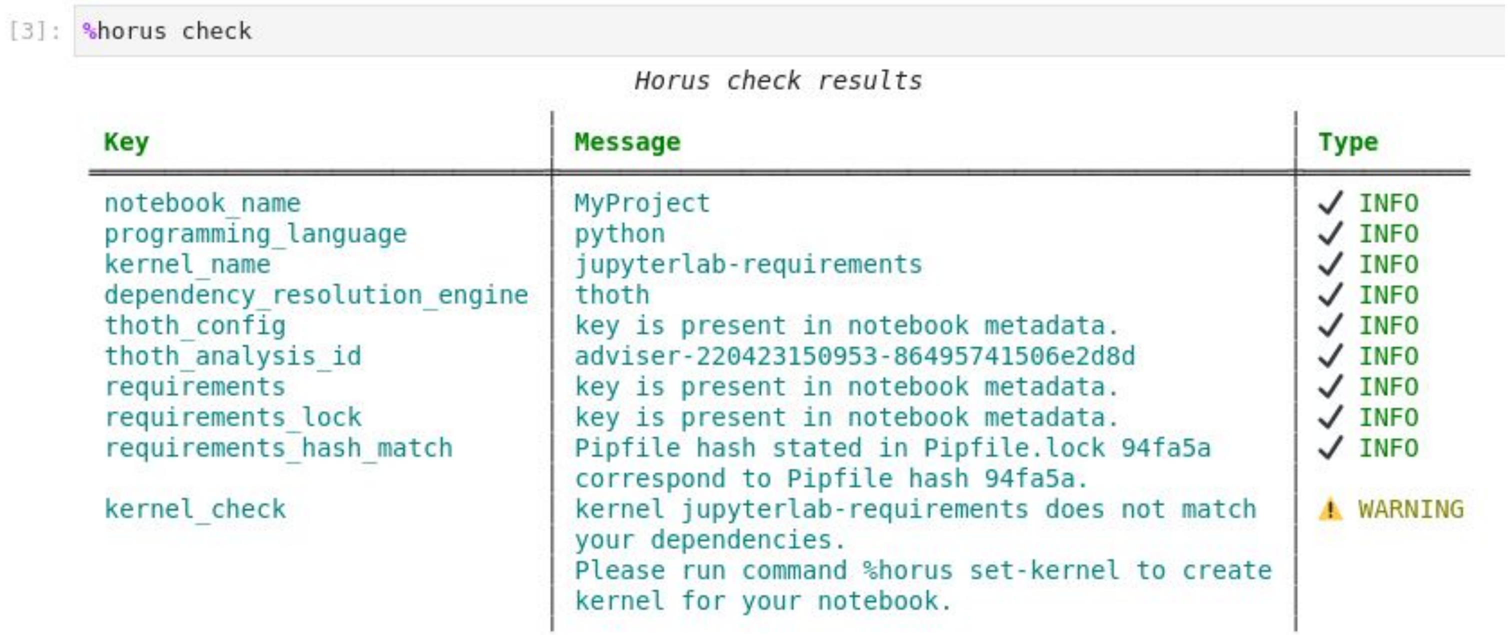
`%horus discover`: Discover dependencies and create Pipfile

`%horus requirements --add`: Add requirements to Pipfile

...

%horus magic commands

`%horus check`: Check notebook metadata about dependencies



```
[3]: %horus check
```

Horus check results

Key	Message	Type
notebook_name	MyProject	✓ INFO
programming_language	python	✓ INFO
kernel_name	jupyterlab-requirements	✓ INFO
dependency_resolution_engine	thoth	✓ INFO
thoth_config	key is present in notebook metadata.	✓ INFO
thoth_analysis_id	adviser-220423150953-86495741506e2d8d	✓ INFO
requirements	key is present in notebook metadata.	✓ INFO
requirements_lock	key is present in notebook metadata.	✓ INFO
requirements_hash_match	Pipfile hash stated in Pipfile.lock 94fa5a correspond to Pipfile hash 94fa5a.	✓ INFO
kernel_check	kernel jupyterlab-requirements does not match your dependencies. Please run command %horus set-kernel to create kernel for your notebook.	⚠ WARNING

Install and run jupyterlab-requirements

```
pip install jupyterlab-requirements

jupyter lab
```

What about containerized notebooks?

Customize your Jupyter Notebook container images

- **Source-to-Image (S2I)**: provides **ready-to-run images** by **injecting source code** directly into a container image

- Thoth S2I produces recommendations **targeting your specific hardware configuration** to run your application inside the cluster

- **S2I** (Source-to-Image) **Minimal Notebook builder**

How does Thoth work?

Thoth's resolver

"Recommend the greatest, not the latest"

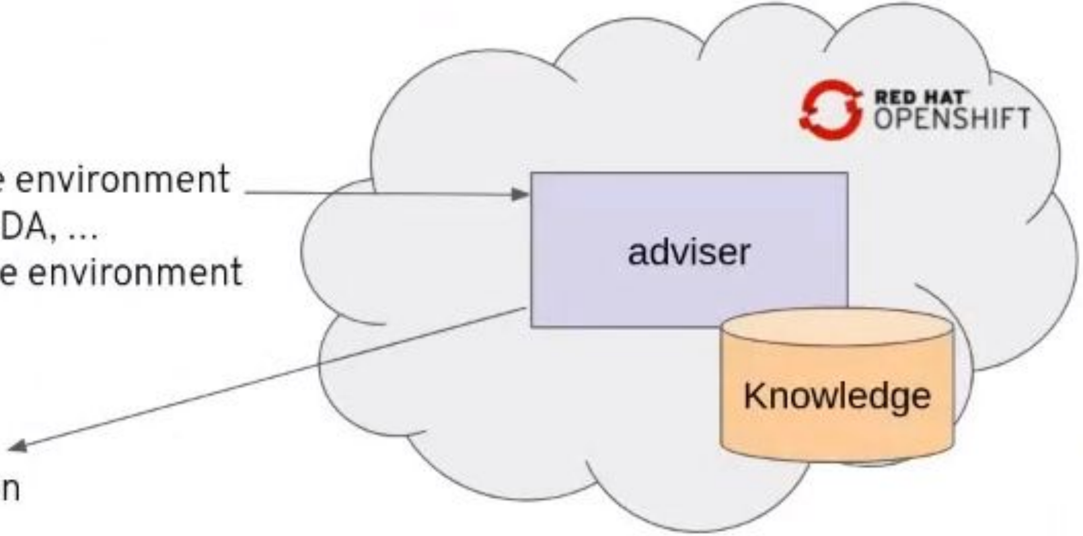
Thoth’s resolver

- **Recommendation types:**
  - latest
  - security
  - performance
  - stable
  - testing

- Uses **Reinforcement Learning** to recommend dependencies

- **Runs in the cloud**

Requirements
Constraints
Information about software in runtime environment
 - OS, Python version, base image, CUDA, ...
Information about hardware in runtime environment
 - CPU, GPU, ...
Static source-code analysis
Recommendation type



Lockfile + justification

What we observe in our knowledge graph

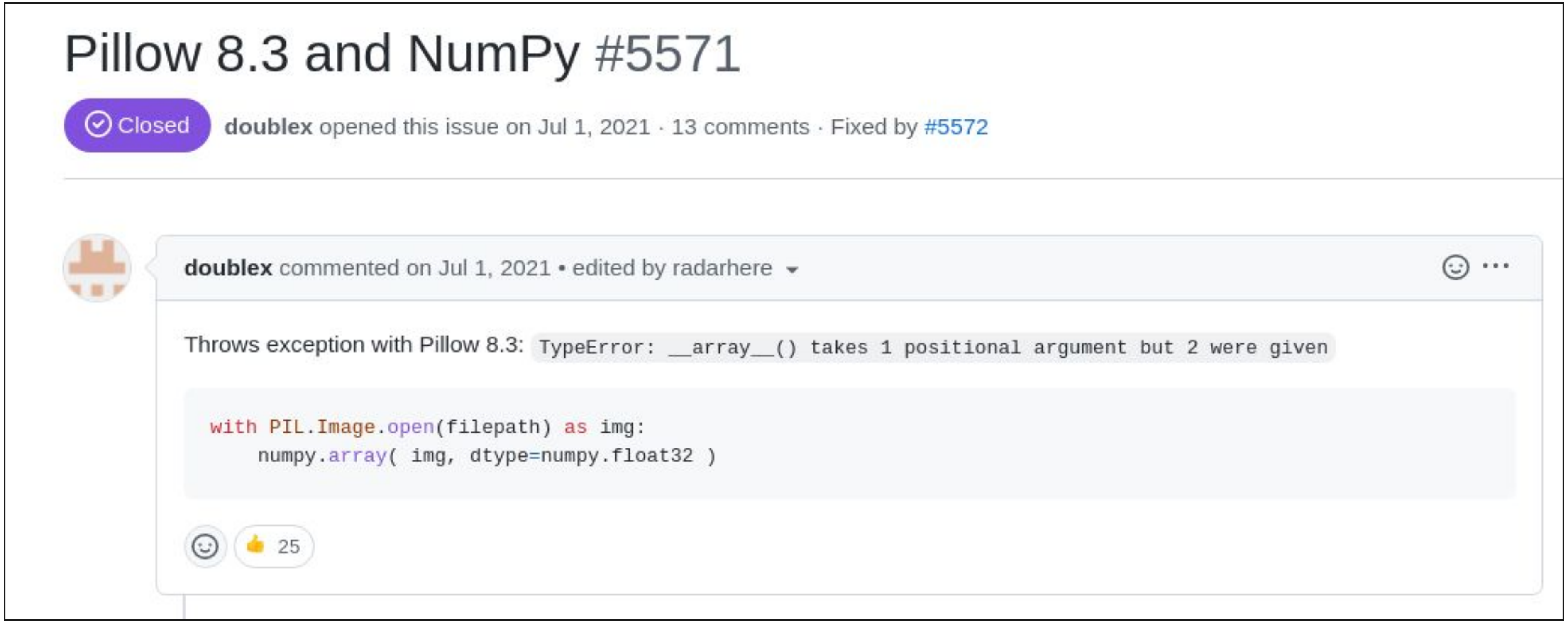
- **Application Stack**
  - Buildtime and runtime environment
  - Dependencies
  - Performances

- **Software Packages**
  - Application Binary Interfaces (ABI)
  - Security: CVE, analyzers...

- **Source code meta information**

Heal your application with prescriptions

- Declaratively state how the resolution process should look like
- YAML files automatically consumed by the resolver
- Contribute! [thoth-station/prescriptions]()



Pillow 8.3 and NumPy #5571

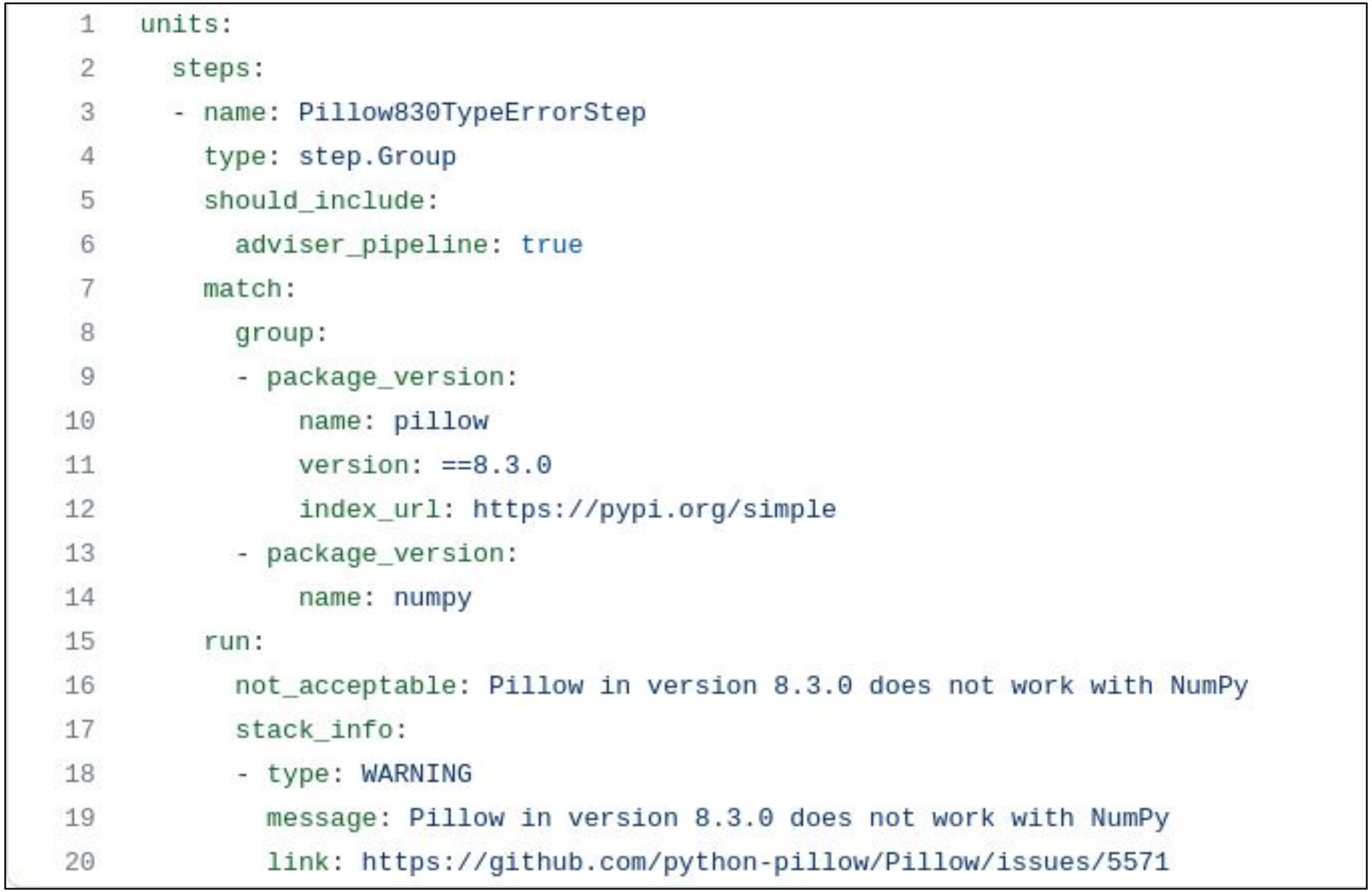
Closed doublex opened this issue on Jul 1, 2021 · 13 comments · Fixed by #5572

doublex commented on Jul 1, 2021 • edited by radarhere

Throws exception with Pillow 8.3: `TypeError: __array__() takes 1 positional argument but 2 were given`

```
with PIL.Image.open(filepath) as img:
    numpy.array( img, dtype=numpy.float32 )
```

👍 25



```yaml
units:
  steps:
  - name: Pillow830TypeErrorStep
    type: step.Group
    should_include:
      adviser_pipeline: true
    match:
      group:
      - package_version:
          name: pillow
          version: ==8.3.0
          index_url: https://pypi.org/simple
      - package_version:
          name: numpy
    run:
      not_acceptable: Pillow in version 8.3.0 does not work with NumPy
      stack_info:
      - type: WARNING
        message: Pillow in version 8.3.0 does not work with NumPy
        link: https://github.com/python-pillow/Pillow/issues/5571
```

Aggregating knowledge about packages

Evaluate **dependencies reliability**:

- Package popularity
- Information about maintainers
- CVE, Security Scorecards
- Releases frequency
- Artifacts size
- …

Conclusions

Notable improvements...

Managing dependencies

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

Containerization

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

Sharing

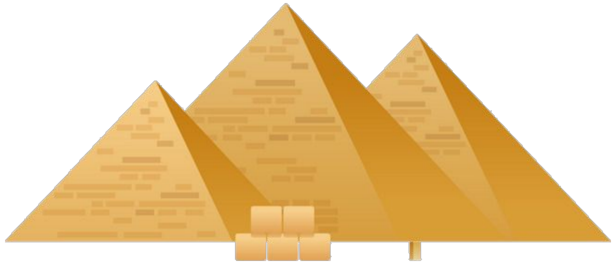
Jupyter Notebooks can be shared as **stand-alone units** without any additional files. Environment is prepared in a **few clicks**

...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

Thank you!



thoth-station.ninja

@ThothStation

github.com/thoth-station