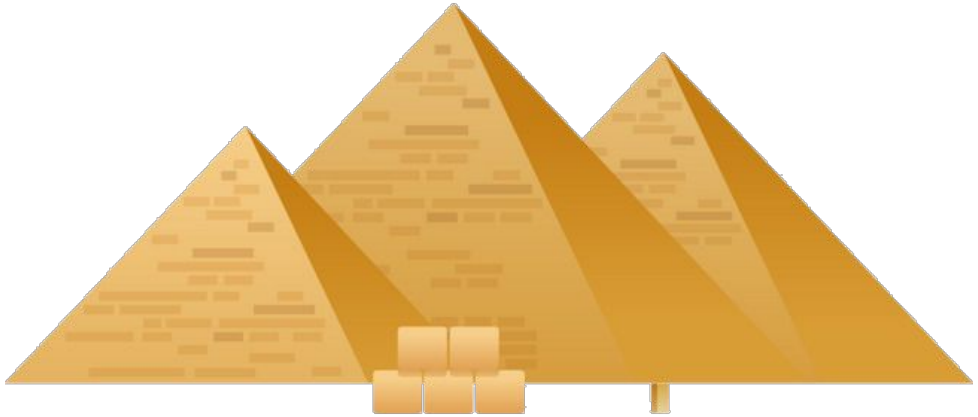


Thoth: the Python cloud resolver

Thoth team - thoth-station.ninja

Red Hat

Agenda

1. $ whoweare
2. Introducing the Python cloud resolver
3. Using Python cloud resolver
4. Declarative interface for the resolver - prescriptions
5. Containerized Python applications
6. Security - AIDevSecOps
7. Cross-index resolution
8. References

$ whoarewe

- Thoth - AIDevSecOps
  - Started (2018) as a research project in Red Hat AICoE team, Office of the CTO
  - thoth-station.ninja

- See our linked YouTube channel for more information
- Follow us on Twitter - @ThothStation

Our mission

- Help Python developers and data scientists create healthy applications

- Project has multiple parts:
  - AICoE-CI - a CI that builds container images
  - Thoth resolver - a recommendation engine for Python applications
    - AIDevSecOps
  - Dependency Monkey - a service that can validate software in a cluster
  - jupyterlab-requirements extension for managing dependencies
  - Bots maintaining GitHub repositories
  - A self hosted Python package index using Pulp available to all Red Hatters
  - Container image analysis and containerized Python applications

Introducing the Python cloud resolver

Python resolvers

- pip
  - the package installer for Python
- Pipenv
  - Python development workflow for humans
- Poetry
  - Python dependency management and packaging made easy
- Thoth
  - Resurrected ancient deities helping humans with software development

Python resolvers

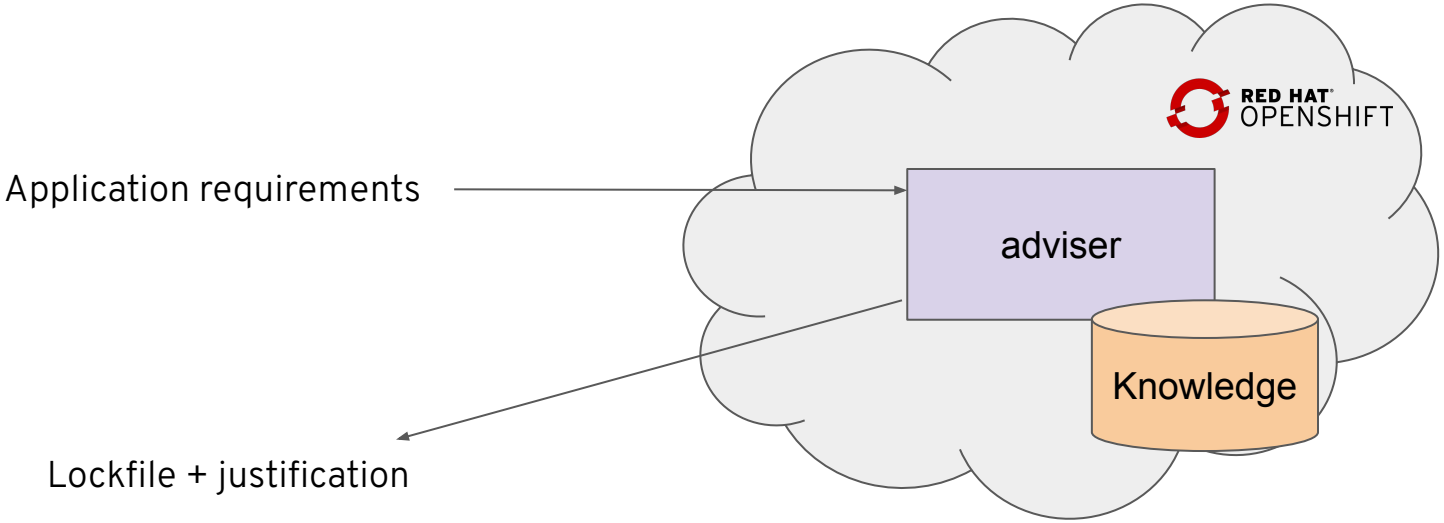
- pip
  - the package installer for Python
- Pipenv
  - Python development workflow for humans
- Poetry
  - Python dependency management and packaging made easy
- Thoth
  - Resurrected ancient deities helping humans with software development

Latest software is not always the greatest choice.

The Python resolver run in cloud

- Recommendation engine for Python applications
- Publicly available to the community
- Stochastic resolver implementing gradient-free reinforcement learning methods
  - Temporal difference learning is used in production
- See documentation for more information:
  - thoth-station.ninja/docs/developers/adviser

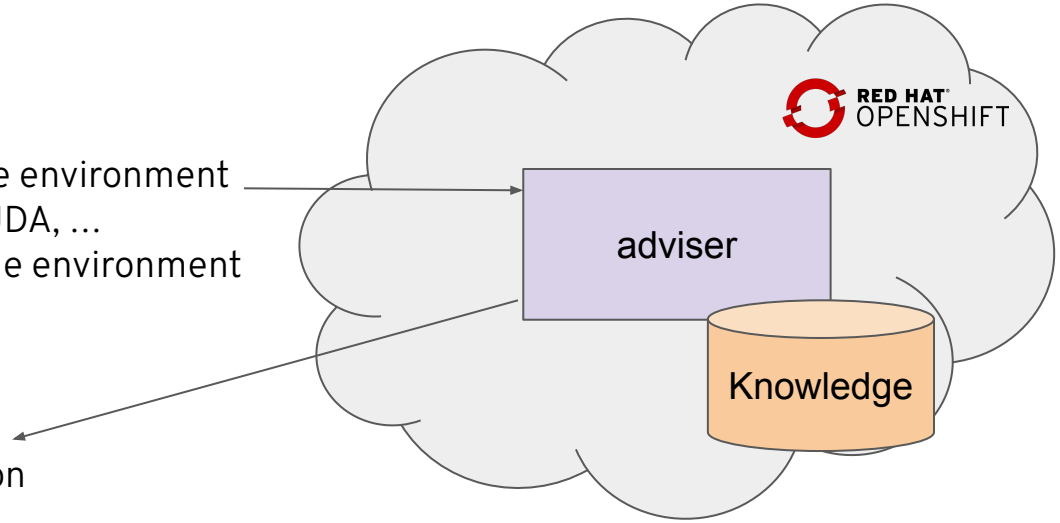
Python cloud resolver



Red Hat

Python cloud resolver

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

Red Hat

Using Python cloud resolver

📄 Managing security in Python applications with the Thoth cloud Python resolver

Red Hat

Thamos Command Line Interface

- One of the Thoth client tools, other tools:
  - jupyterlab-requirements
  - Kebechet bot

- Talks to Thoth's backend and helps with managing your environment

- Available on PyPI:

```
$ pip install thamos
$ thamos --help
$ thamos config
```

Runtime environments and recommendation types

- Concept of "*overlays*"
  - Each overlay states requirements for the application for the given runtime environment
  - Ex. runtime environment for inference and runtime environment for training
    - Can have different requirements and runtime environments

- Support for "*recommendation types*"
  - Different resolution considering intention with the application
  - thoth-station.ninja/recommendation-types
  - Ex. production environment should be secure, isolated environment where the training is done should be performant

Red Hat

```yaml
host: khemenu.thoth-station.ninja
requirements_format: pipenv

runtime_environments:
- name: "inference"
  recommendation_type: security
  operating_system:
    name: "rhel"
    version: "8"
  python_version: "3.8"

- name: "training"
  recommendation_type: performance
  operating_system:
    name: "rhel"
    version: "8"
  cuda_version: "11.1"
  python_version: "3.8"
```

```toml
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi-org-simple"

[packages]
flask = "*"
tensorflow = "~=2.4"

[dev-packages]

[requires]
python_version = "3.8"

[thoth]
disable_index_adjustment = false

[thoth.allow_prereleases]
```

```toml
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi-org-simple"

[[source]]
url = "https://thoth-station.ninja/simple"
verify_ssl = true
name = "aicoe"


[packages]
boto3 = "*"
tensorflow = {"version"="~=2.4", "index"="aicoe"}

[dev-packages]

[requires]
python_version = "3.8"

[thoth]
disable_index_adjustment = true

[thoth.allow_prereleases]
protobuf = true
```


Red Hat

🎥 *Demo: Thamos CLI*

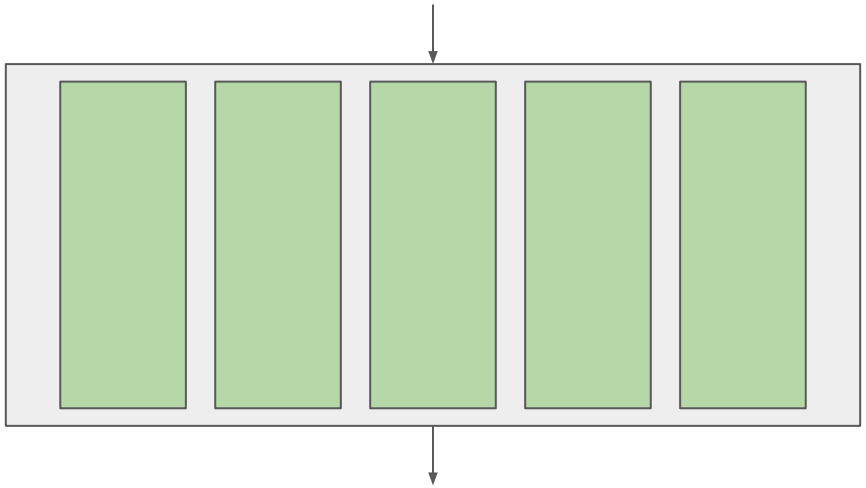
Red Hat

Declarative interface for the resolver to resolve Python packages following prescribed rules

📄 Thoth prescriptions for resolving Python dependencies

Resolution pipeline

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

Red Hat

Resolution pipeline

- The whole resolution process is seen as a pipeline
- The resolution pipeline is made out of pipeline units
- Each pipeline unit has its own type
- Pipeline units can be implemented in Python or declared in YAML files

Prescriptions - declarative interface to the cloud based resolver

- Provide a way to declarativelly state how the resolution process should look like
- Community driven open database used by the resolver to resolve high quality software
  - github.com/thoth-station/prescriptions
- A set of YAML files that are automatically consumed by the resolver in a deployment
- See documentation for more information:
  - thoth-station.ninja/docs/developers/adviser/prescription.html

Red Hat

Prescriptions

When using OpenShift or Kubernetes, one provides manifest files that state how the desired state of a cluster should look like. Prescriptions might be seen analogous to this - prescriptions provide a way to declaratively state how the desired dependency resolution should look like considering the prescribed rules. Then, it's up to the reinforcement learning algorithm implemented in Thoth's adviser to find a solution in the form of a lockfile respecting the prescribed rules, requirements for the application and other inputs to the Thoth's cloud resolver.

Prescriptions - Example

- Pillow in version 8.3.0 does not work with NumPy

  github.com/python-pillow/Pillow/issues/5571

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

```
>          frame_paletted = np.array(im, np.uint8)
E          TypeError: __array__() takes 1 positional argument but 2 were given

/lib/python3.9/site-packages/imageio/plugins/pillow.py:745: TypeError
```


Red Hat

```yaml
units:
 steps:
 - name: Pillow830TypeErrorStep
   type: step
   should_include:
     adviser_pipeline: true
   match:
     package_version:
       name: pillow
       version: ==8.3.0
       index_url: https://pypi.org/simple
     state:
       resolved_dependencies:
       - 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
```

```yaml
units:
steps:
- name: Pillow830TypeErrorStep
  type: step
  should_include:
    adviser_pipeline: true
  match:
    package_version:
      name: pillow
      version: ==8.3.0
      index_url: https://pypi.org/simple
    state:
      resolved_dependencies:
      - 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
```

```yaml
units:
 steps:
 - name: Pillow830TypeErrorStep
   type: step
   should_include:
     adviser_pipeline: true
   match:
     package_version:
       name: pillow
       version: ==8.3.0
       index_url: https://pypi.org/simple
     state:
       resolved_dependencies:
       - 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
```

```yaml
units:
 steps:
 - name: Pillow830TypeErrorStep
  type: step
  should_include:
   adviser_pipeline: true
  match:
   package_version:
    name: pillow
    version: ==8.3.0
    index_url: https://pypi.org/simple
   state:
    resolved_dependencies:
    - 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
```

```yaml
units:
 steps:
 - name: Pillow830TypeErrorStep
   type: step
   should_include:
     adviser_pipeline: true
   match:
     package_version:
       name: pillow
       version: ==8.3.0
       index_url: https://pypi.org/simple
     state:
       resolved_dependencies:
       - 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
```

```yaml
units:
 steps:
 - name: Pillow830TypeErrorStep
   type: step
   should_include:
     adviser_pipeline: true
   match:
     package_version:
       name: pillow
       version: ==8.3.0
       index_url: https://pypi.org/simple
     state:
       resolved_dependencies:
       - 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
```


Red Hat

🎥 *Demo: Prescriptions*

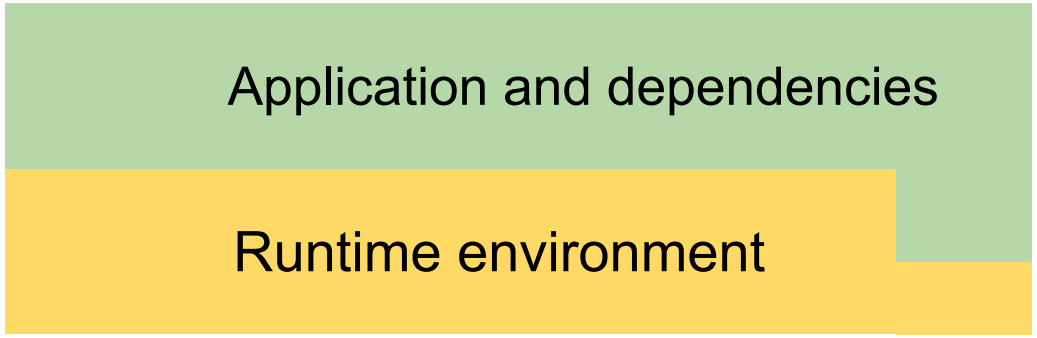
Red Hat

Containerized Python applications

[Build and extend containerized applications with Project Thoth]

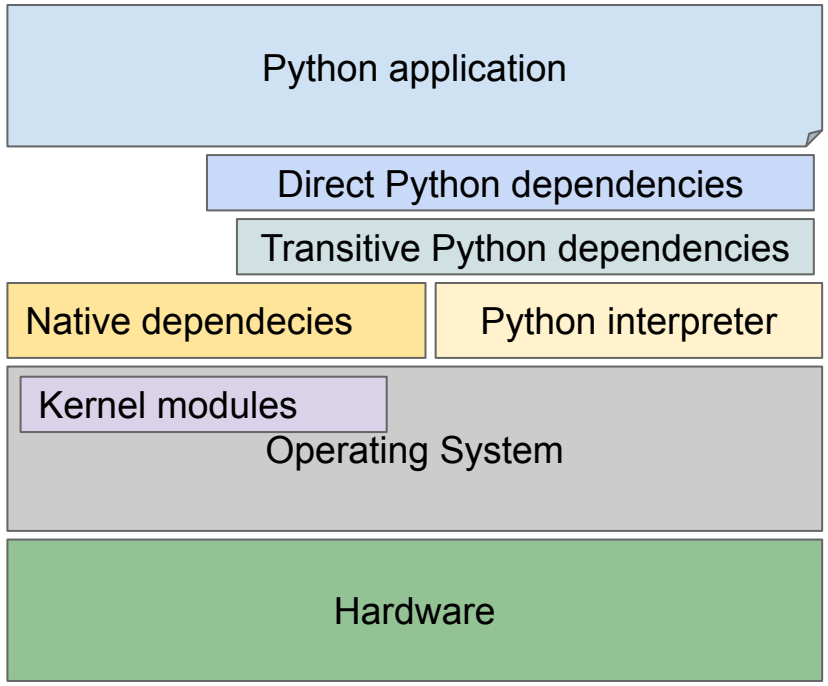
Fitting the containerized runtime environment

- RPM, Python packages, ABI, CUDA, cuDNN, OpenMKL, ...
- Resolver considers containerized environment
  - Still optional, the resolution might be “generic”, like pip, Pipenv, ...



Red Hat

Fitting the runtime environment



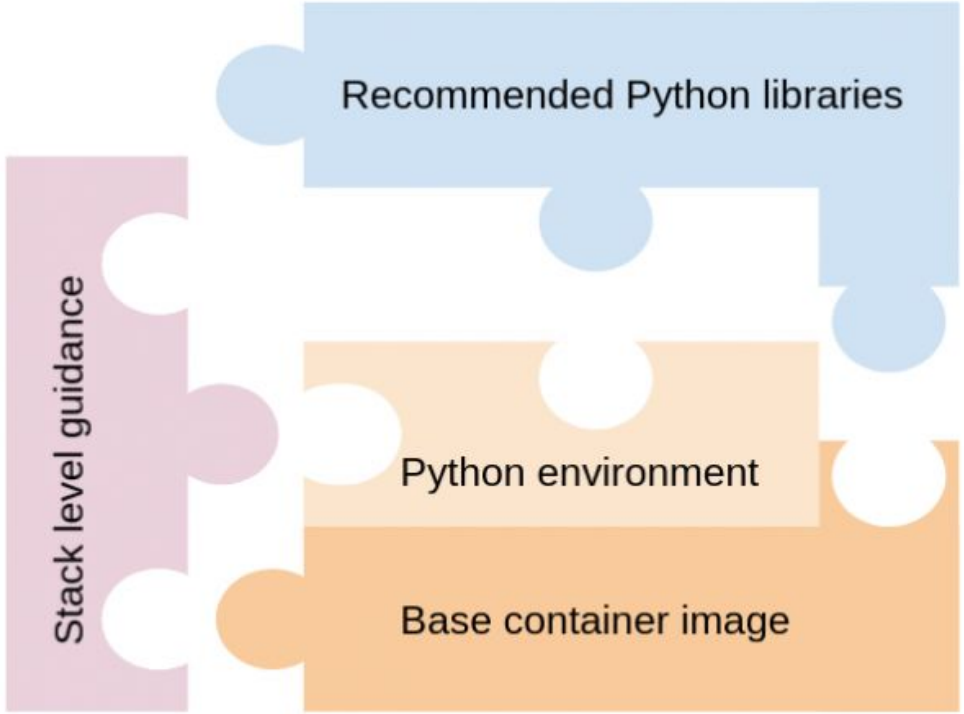
Red Hat

Fitting the containerized runtime environment

- Base container image
  - Python base container images
  - Predictable stacks
  - JupyterLab container images
  - ...

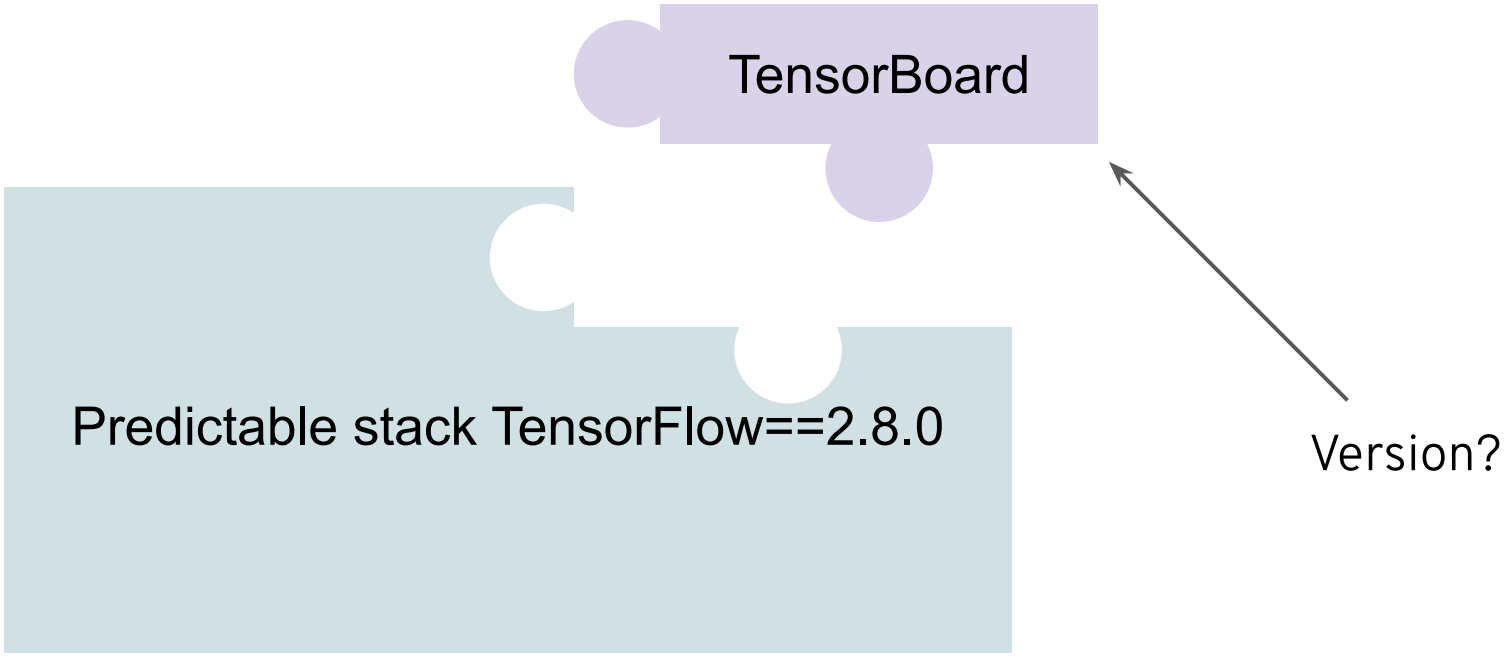
- Synthetize Python layer and base container image layer

Fitting the environment example



Red Hat

Fitting the environment example



Red Hat

thamos images --help

```
Usage: thamos images [OPTIONS]

  Check available Thoth container images.

  Examples:

    thamos images --output-format json

    thamos images --os-name fedora --os-version 35 --python-version 3.9

    thamos images --symbol GLIBC_FOO

Options:
  -o, --output-format [json|yaml|table]
                          Specify output format for the status report.
  -n, --os-name OS_NAME         Operating system name filter.
  -v, --os-version OS_VERSION     Operating system version filter.
  -p, --python-version PY_VERSION
                          Python interpreter version filter.
  --cuda-version CUDA_VERSION     CUDA version filter.
  --image-name IMAGE_NAME       Filter based on image name.
  --library-name LIBRARY_NAME     Filter based on library name.
  --symbol SYMBOL               Filter based on symbol.
  --package-name PACKAGE_NAME     Filter based on Python package name.
  --rpm-package-name RPM_PACKAGE_NAME
                          Filter based on RPM package name.
  --help                      Show this message and exit.
```


Red Hat

🎥 *Demo: Thamos and container images*

Red Hat

Security - AIDevSecOps

📄 [Secure your Python applications with Thoth recommendations]

Red Hat

Security - AIDevSecOps

- Docs: Thoth security advises
- Recommendations based on static source code analysis
  - See recommendations from the Python standard library (example)
- PyPA - advisory-db
  - A database of known vulnerabilities in Python ecosystem
  - github.com/pypa/advisory-db
- Security Scorecards by Open Source Security Foundation
  - openssf.org/blog/2020/11/06/security-scorecards-for-open-source-projects
  - Example: see scorecards_prefixed prescriptions for TensorFlow
- Container image analyses - vulnerabilities in the base container images used

... additional information about Python packages not strictly related to security

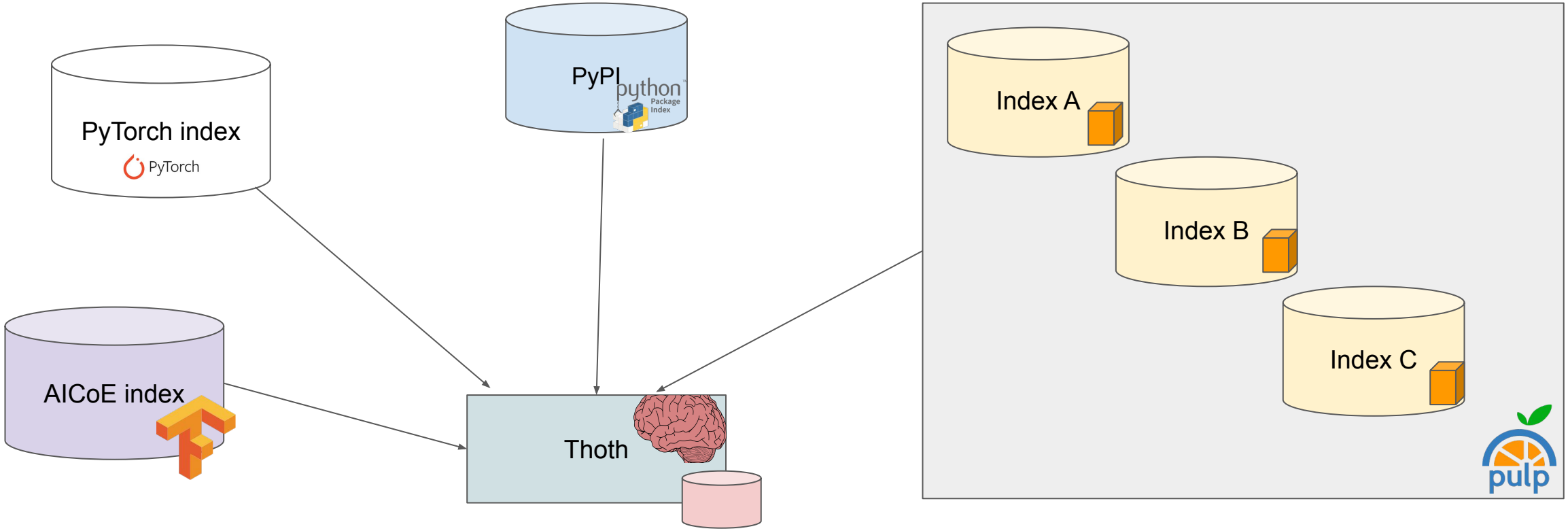
🎥 *Demo: Securing Python applications*

Red Hat

Cross-index resolution

📄 [Prevent Python dependency confusion attacks with Thoth]

Python package indexes & Thoth



[How to self-host a Python package index using Pulp]

Red Hat

Python package indexes

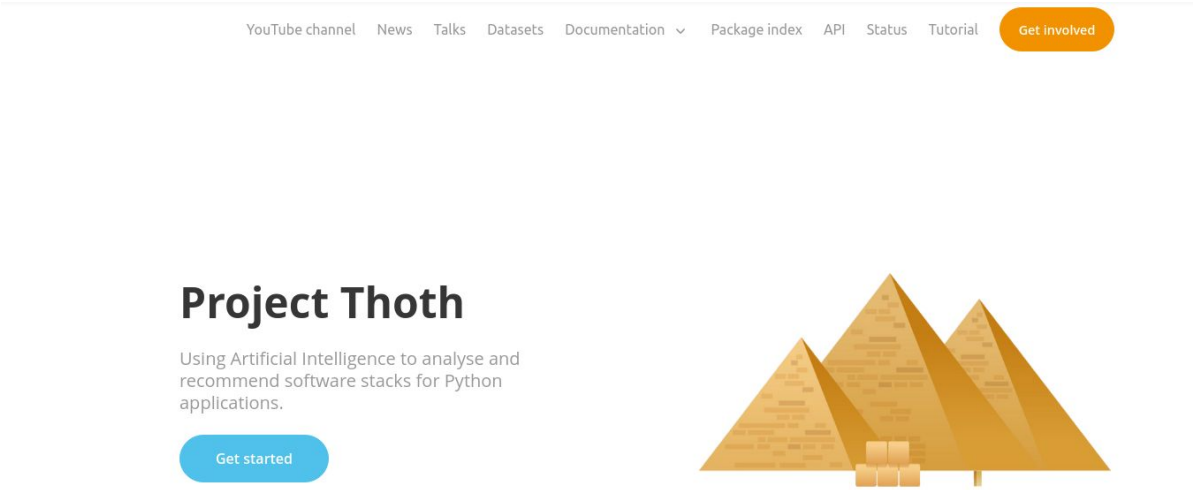
- Automatically monitored in a Thoth deployment
  - e.g. builds outside of manylinux standards:
    - index with CUDA specific builds
    - AVX2 optimized builds of TensorFlow

- Packages published on monitored Python package indexes are automatically analyzed
- Cross index resolution without dependency confusion
- One central point of knowledge in cloud based resolver

🎥 *Demo: Thamos and cross-index resolution*

References

References



thoth-station.ninja

References

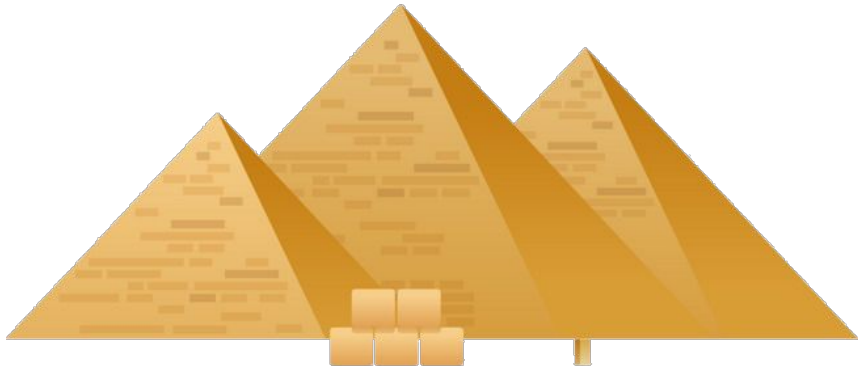
- Introspecting containerized Python applications in a cluster with Thoth Amun
- How to self-host a Python package index using Pulp
- Extracting dependencies from Python packages
- Extracting information from Python source code
- Prevent Python dependency confusion attacks with Thoth
- Build and extend containerized applications with Project Thoth
- Customize Python dependency resolution with machine learning
- Generating pseudorandom numbers in Python
- Secure your Python applications with Thoth recommendations
- Find and compare Python libraries with project2vec

References

- Thoth prescriptions for resolving Python dependencies
- Resolve Python dependencies with Thoth Dependency Monkey
- micropipenv: Installing Python dependencies in containerized applications
- Continuous learning in Project Thoth using Kafka and Argo
- Can we consider --editable a bad practice?
- Managing Python dependencies with the Thoth JupyterLab extension
- Use Kebechet machine learning to perform source code operations
- AI software stack inspection with Thoth and TensorFlow
- Microbenchmarks for AI applications using Red Hat OpenShift on PSI in project Thoth

🐦 References

- Thoth's website
  - thoth-station.ninja
- Source code:
  - github.com/thoth-station
- @ThothStation Twitter handle
- Thoth Station YouTube channel
- Talks and presentations



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

Red Hat