ActiveState

Retaining Your Software Developers

Keeping Software Devs Happy by Automating Open Source Pains

Pandemic-Created Pressures

Increased developer workload/stress as a result of:

- The need to enable a remote workforce “overnight”
  - Accelerated digital migration plans

“Work from anywhere” trend:

- Developers have more employment opportunities than ever before
  - The need to onboard new developers quicker

Introductions

Shane Warden

Director of Engineering

Loreli Cadapan

VP Product Management

The Open Source Opportunity

- Time consuming tasks
  - ie, Building dependencies from source
- Frustrations
  - ie, Dependency Hell
- Avoidable Risks
  - ie, Supply chain security

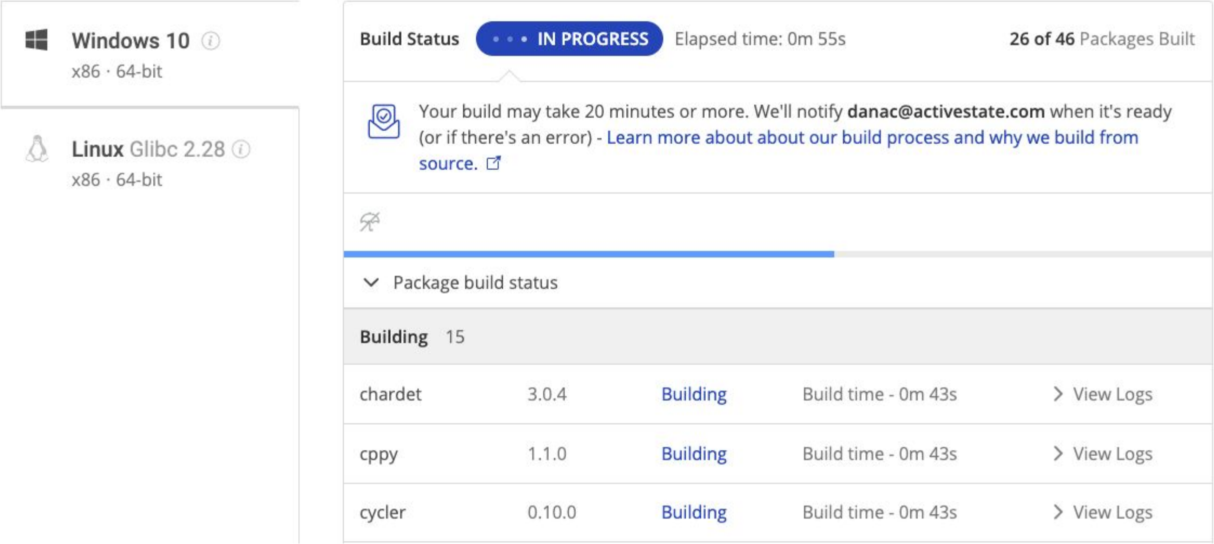
Eliminating these can reduce developer workload and stress

ActiveState

Help Developers Save Time

Automate Builds from Source

- Automatically build open source from source code + native libraries



Windows 10
x86 · 64-bit

Linux Glibc 2.28
x86 · 64-bit

Build Status • • • IN PROGRESS Elapsed time: 0m 55s **26 of 46** Packages Built

Your build may take 20 minutes or more. We'll notify **danac@activestate.com** when it's ready (or if there's an error) - Learn more about about our build process and why we build from source.

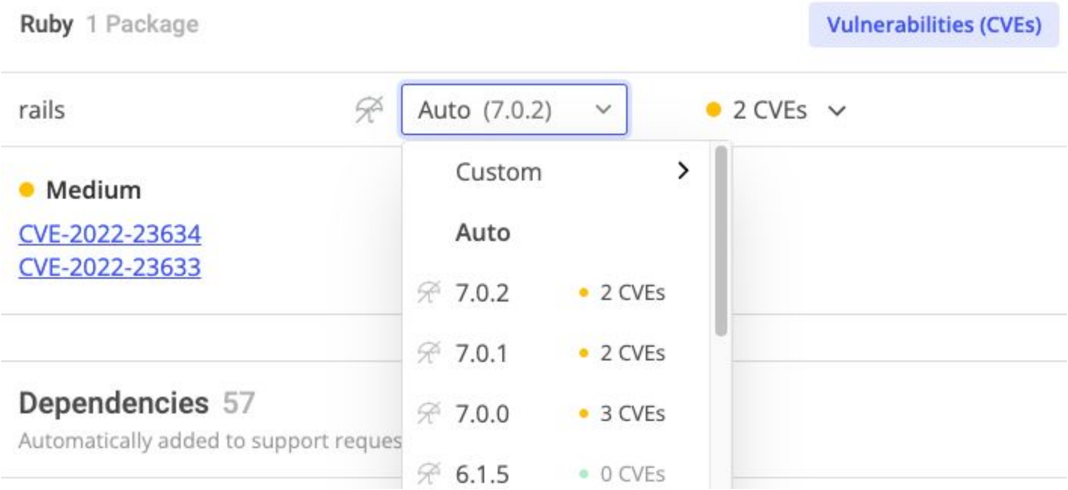
Package build status

Building 15

chardet	3.0.4	Building	Build time - 0m 43s	› View Logs
cppy	1.1.0	Building	Build time - 0m 43s	› View Logs
cycler	0.10.0	Building	Build time - 0m 43s	› View Logs

Automate Vulnerability Management

- Automatically monitor, notify and simplify remediation of vulnerabilities



Ruby 1 Package

Vulnerabilities (CVEs)

rails Auto (7.0.2) ● 2 CVEs

● Medium

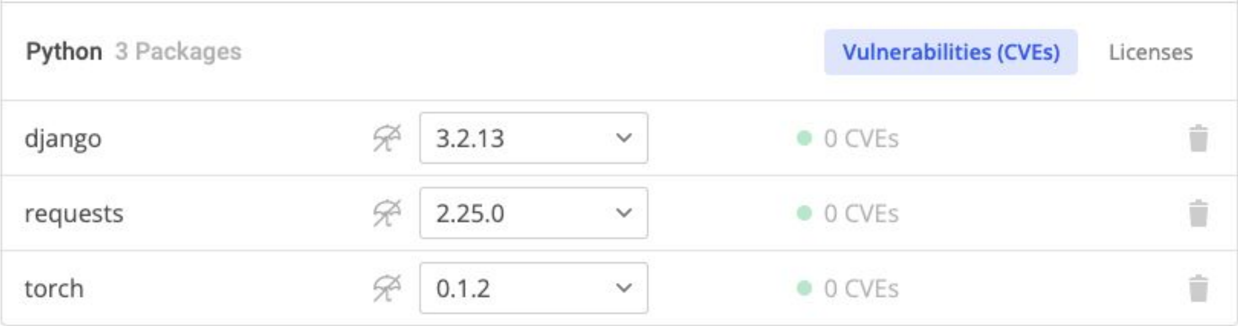
CVE-2022-23634
CVE-2022-23633

Dependencies 57

Automatically added to support reques

Eliminate Locating Dependencies

- Automatically create a registry of known, good dependencies that work together



Python 3 Packages

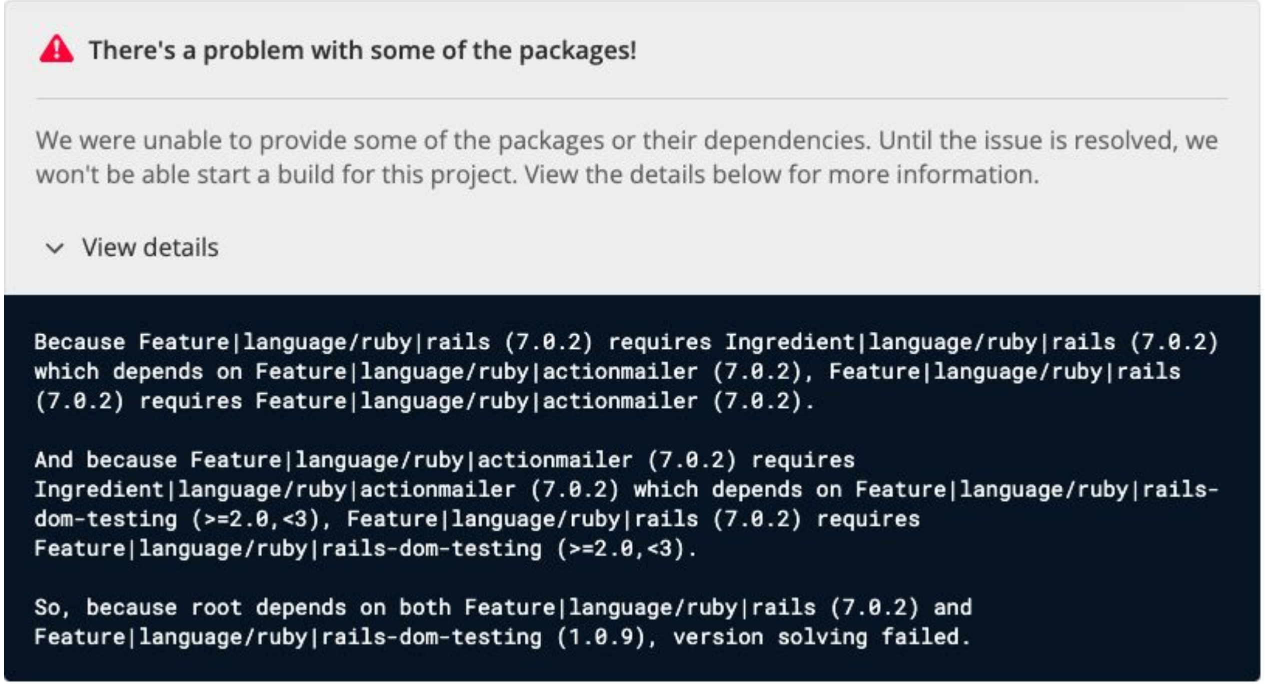
Vulnerabilities (CVEs) Licenses

django	3.2.13	● 0 CVEs
requests	2.25.0	● 0 CVEs
torch	0.1.2	● 0 CVEs

ActiveState

Reduce Developer Frustration

Eliminate Dependency Hell



There's a problem with some of the packages!

We were unable to provide some of the packages or their dependencies. Until the issue is resolved, we won't be able start a build for this project. View the details below for more information.

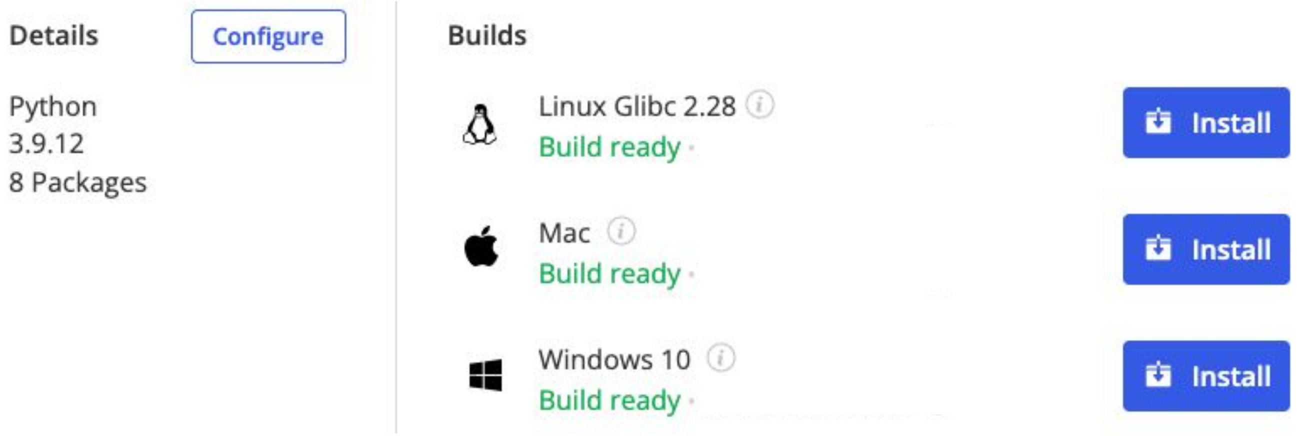
```
Because Feature|language/ruby|rails (7.0.2) requires Ingredient|language/ruby|rails (7.0.2)
which depends on Feature|language/ruby|actionmailer (7.0.2), Feature|language/ruby|rails
(7.0.2) requires Feature|language/ruby|actionmailer (7.0.2).

And because Feature|language/ruby|actionmailer (7.0.2) requires
Ingredient|language/ruby|actionmailer (7.0.2) which depends on Feature|language/ruby|rails-
dom-testing (>=2.0,<3), Feature|language/ruby|rails (7.0.2) requires
Feature|language/ruby|rails-dom-testing (>=2.0,<3).

So, because root depends on both Feature|language/ruby|rails (7.0.2) and
Feature|language/ruby|rails-dom-testing (1.0.9), version solving failed.
```

Automate Environment Reproducibility

- Eliminate “works on my machine” issues & ensure environment reproducibility



Details

Python
3.9.12
8 Packages

Builds

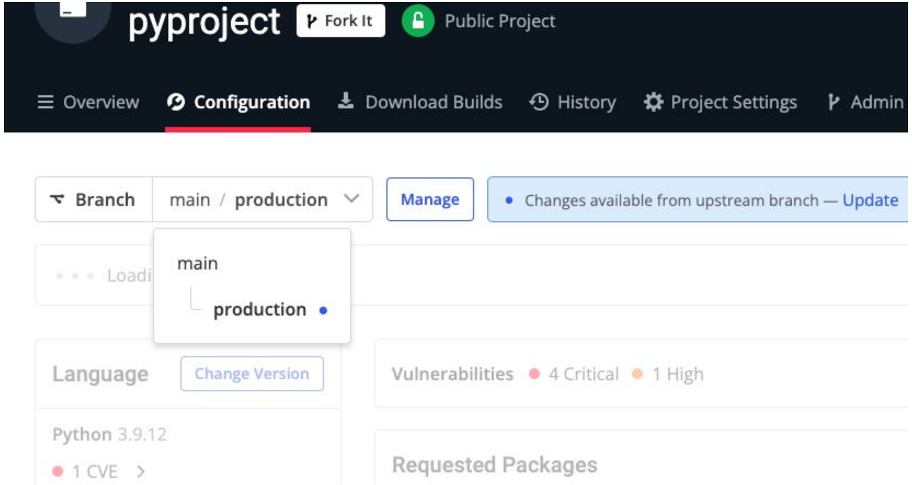
Linux Glibc 2.28
Build ready

Mac
Build ready

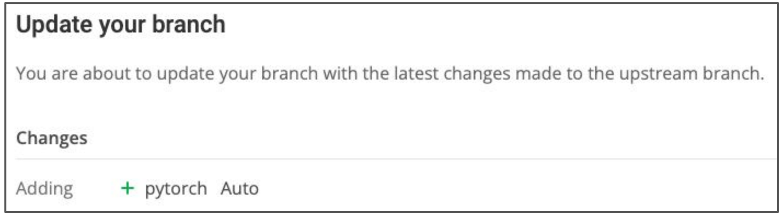
Windows 10
Build ready

Manage Runtime Configuration Drift

- Parent-child branches inherit changes between dev, test, prod, etc environments



Changes available from upstream branch — Update



Update your branch

You are about to update your branch with the latest changes made to the upstream branch.

Changes

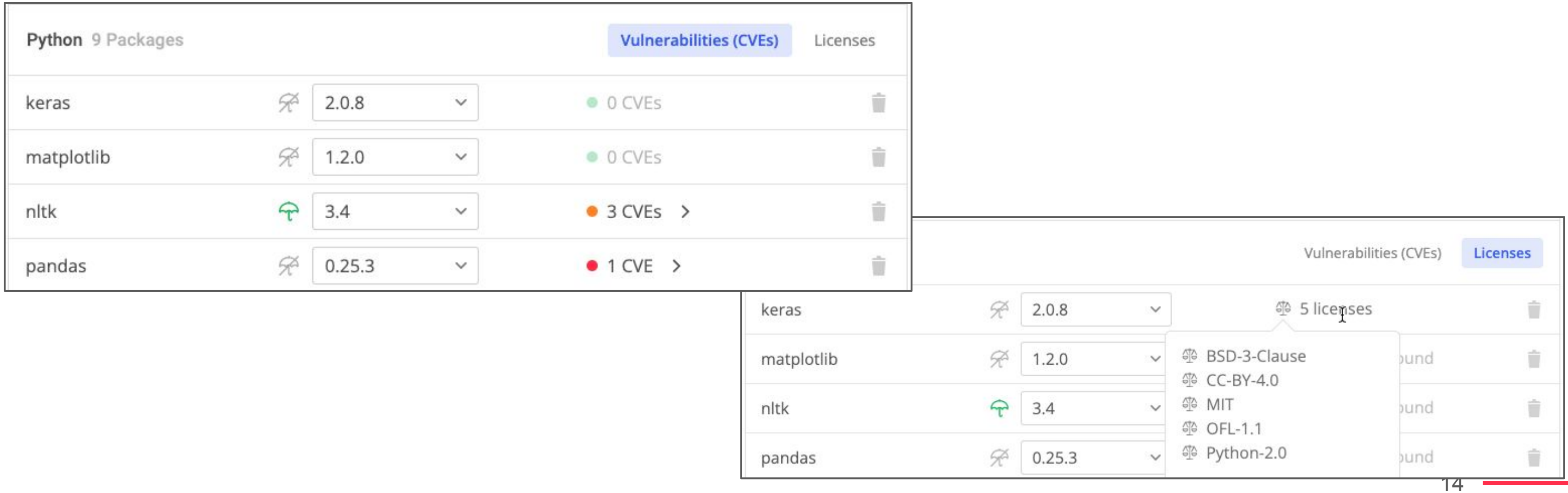
Adding + pytorch Auto

ActiveState

Help Developers Avoid Risk

Vetted Dependency Catalog

- Understand security and licensing risks at a glance



Python 9 Packages

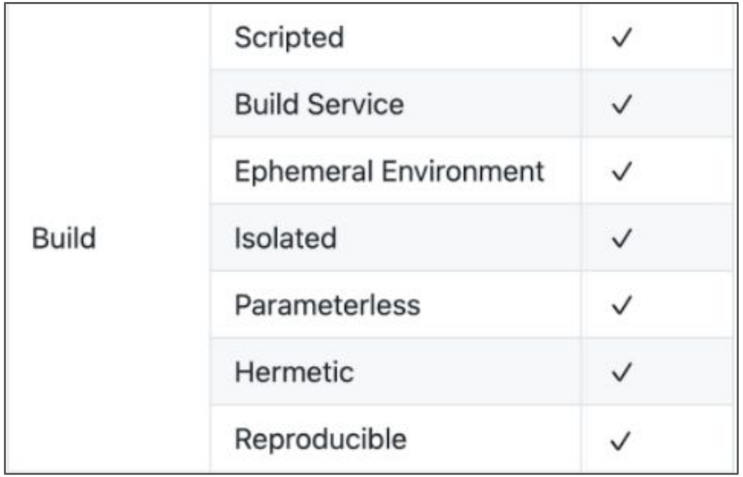
keras	2.0.8	● 0 CVEs
matplotlib	1.2.0	● 0 CVEs
nltk	3.4	● 3 CVEs ›
pandas	0.25.3	● 1 CVE ›

keras	2.0.8	5 licenses
matplotlib	1.2.0	ound
nltk	3.4	ound
pandas	0.25.3	ound

- BSD-3-Clause
- CC-BY-4.0
- MIT
- OFL-1.1
- Python-2.0

Secure Build Service

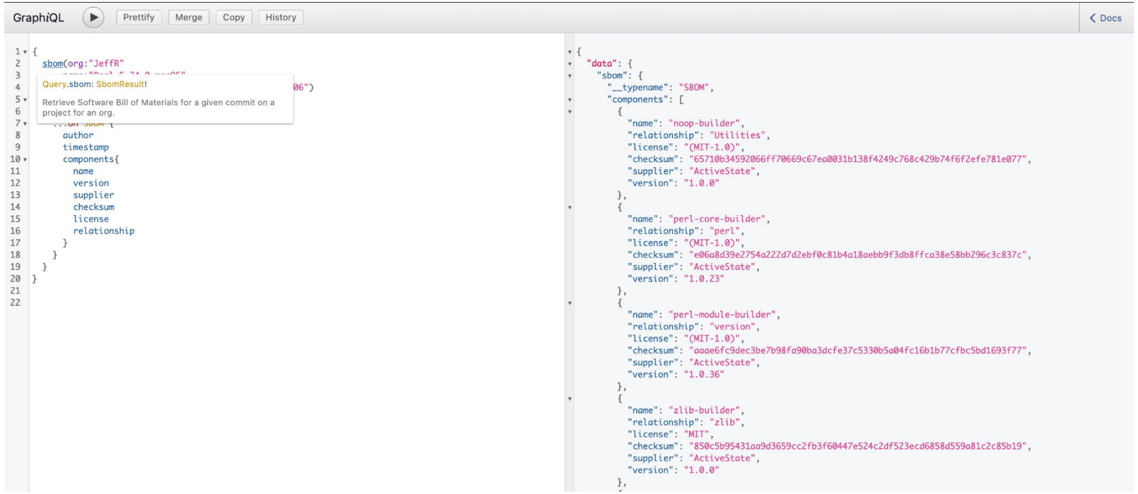
- Tamper-proof system creates reproducible builds of secure artifacts



Build	Scripted	✓
	Build Service	✓
	Ephemeral Environment	✓
	Isolated	✓
	Parameterless	✓
	Hermetic	✓
	Reproducible	✓

Automatically Generate SBOMs

- Software Bill Of Materials (SBOM) for each of your runtime environments



```
{
  sbom(org:"JeffR"
                                                 06")
     ...on SBOM {
      author
      timestamp
      components{
        name
        version
        supplier
        checksum
        license
        relationship
      }
    }
  }
}
```

Query.sbom: SbomResult!

Retrieve Software Bill of Materials for a given commit on a project for an org.

```
{
  "data": {
    "sbom": {
      "__typename": "SBOM",
      "components": [
        {
          "name": "noop-builder",
          "relationship": "Utilities",
          "license": "(MIT-1.0)",
          "checksum": "65710b34592066ff70669c67ea0031b138f4249c768c429b74f6f2efe781e077",
          "supplier": "ActiveState",
          "version": "1.0.0"
        },
        {
          "name": "perl-core-builder",
          "relationship": "perl",
          "license": "(MIT-1.0)",
          "checksum": "e06a8d39e2754a222d7d2ebf0c81b4a18aebb9f3db8ffca38e58bb296c3c837c",
          "supplier": "ActiveState",
          "version": "1.0.23"
        },
        {
          "name": "perl-module-builder",
          "relationship": "version",
          "license": "(MIT-1.0)",
          "checksum": "aaae6fc9dec3be7b98fa90ba3dcfe37c5330b5a04fc16b1b77cfbc5bd1693f77",
          "supplier": "ActiveState",
          "version": "1.0.36"
        },
        {
          "name": "zlib-builder",
          "relationship": "zlib",
          "license": "MIT",
          "checksum": "850c5b95431aa9d3659cc2fb3f60447e524c2df523ecd6858d559a81c2c85b19",
          "supplier": "ActiveState",
          "version": "1.0.0"
        },
```

Poll: To what extent do you automate your use of open source dependencies?

- Not at all
- Minimal automation (ie., automated import to repo)
- Some automation (ie., import + build system)
- Mostly automated (ie., import, build & vulnerability remediation)
- End to end automation (ie., import, build, vulnerabilities & environment setup/update, SBOM generation, etc)

Pros and Cons of Self-Vendoring

Pros:

- Avoid Dependency Conflicts
- Avoid Breaking the Build
- Consistent Deployments
- Faster Fixes

Cons:

- Need to Build Everything Yourself
- Outdated Dependencies
- Invisible Vulnerabilities
- Source Code/Repository Clutter

ActiveState's Managed Distributions

- Produce a catalog of dependencies
- Securely built
- Reproducible runtime environments across all operating systems
- Optionally distribute via artifact repositories
- Monitor dependencies for vulnerabilities and stale-datedness
- Optionally apply security fixes for EOL languages (e.g. Python 2)

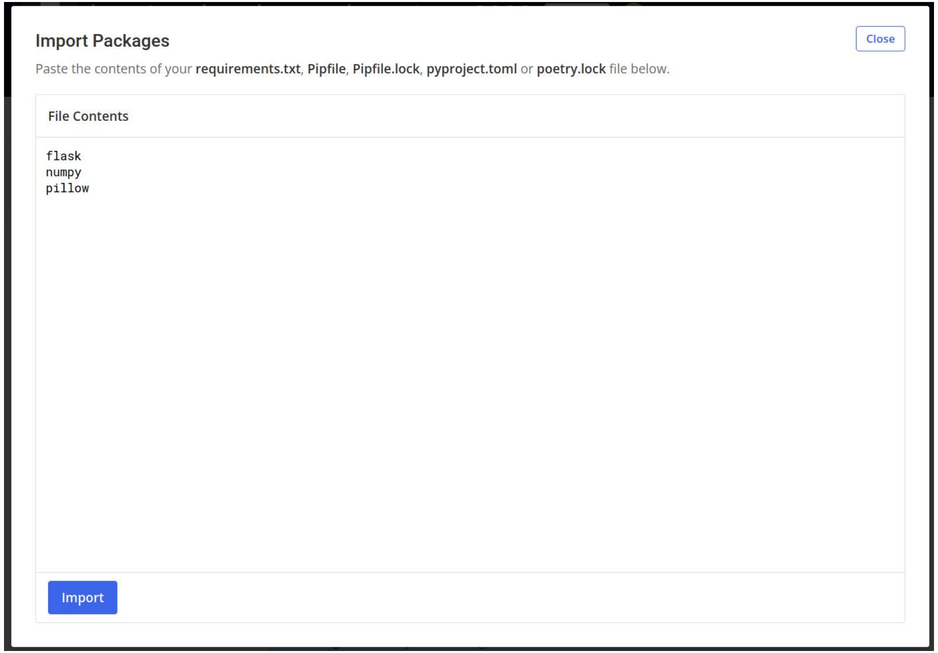
ActiveState

Platform Demo

Demo: How to Onboard New Developers Quicker

- A cross-platform project: https://platform.activestate.com/shanew/keeping-devs-happy-demo-june-2022
- Built on Linux, but let’s activate and run it elsewhere

Demo: Import Packages

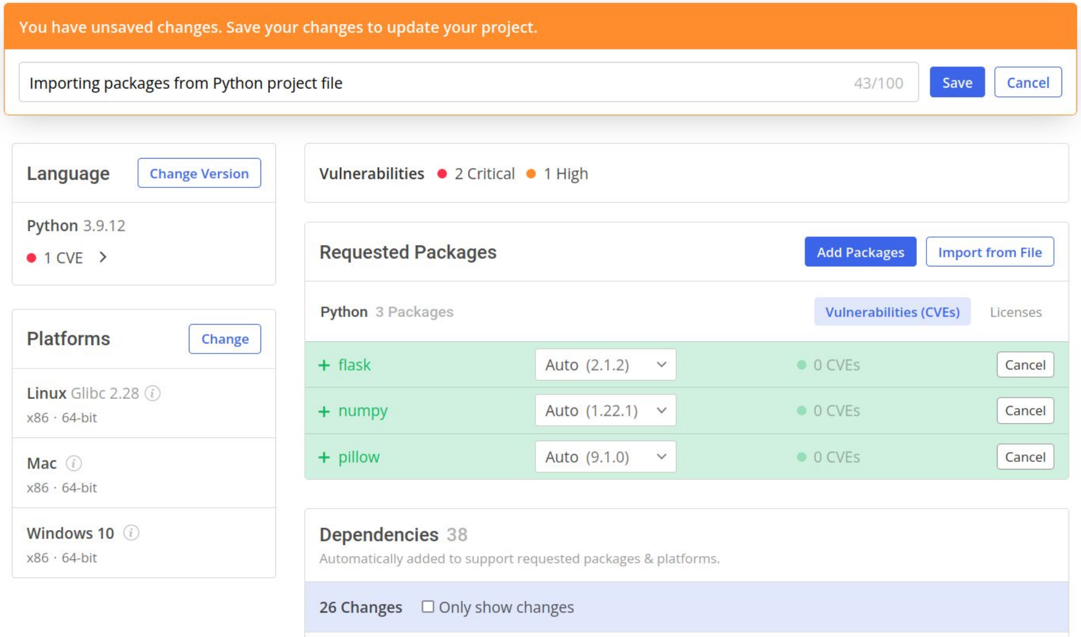


Paste the contents of your **requirements.txt**, **Pipfile**, **Pipfile.lock**, **pyproject.toml** or **poetry.lock** file below.

File Contents

```
flask
numpy
pillow
```

Demo: Dependency Hell Solved Automatically



You have unsaved changes. Save your changes to update your project.

Importing packages from Python project file 43/100

Language

Python 3.9.12

● 1 CVE

Platforms

Linux Glibc 2.28
x86 · 64-bit

Mac
x86 · 64-bit

Windows 10
x86 · 64-bit

Vulnerabilities ● 2 Critical ● 1 High

Requested Packages

Python 3 Packages

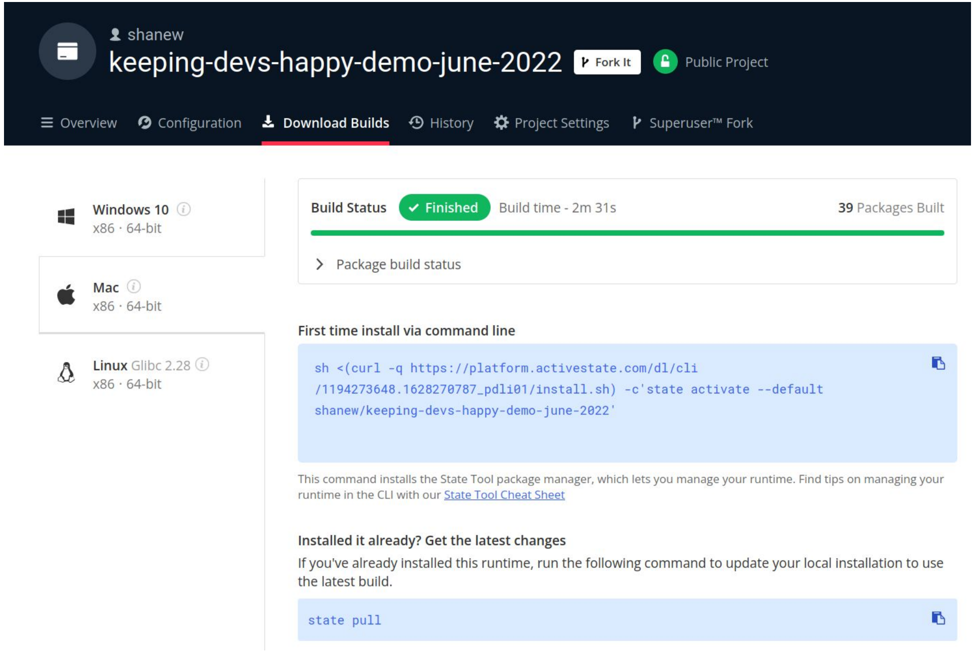
+ flask	Auto (2.1.2)	● 0 CVEs	Cancel
+ numpy	Auto (1.22.1)	● 0 CVEs	Cancel
+ pillow	Auto (9.1.0)	● 0 CVEs	Cancel

Dependencies 38

Automatically added to support requested packages & platforms.

26 Changes ☐ Only show changes

Demo: Built on All Platforms



Build Status ✓ Finished Build time - 2m 31s **39** Packages Built

Package build status

First time install via command line

```
sh <(curl -q https://platform.activestate.com/dl/cli
/1194273648.1628270787_pdli01/install.sh) -c'state activate --default
shanew/keeping-devs-happy-demo-june-2022'
```

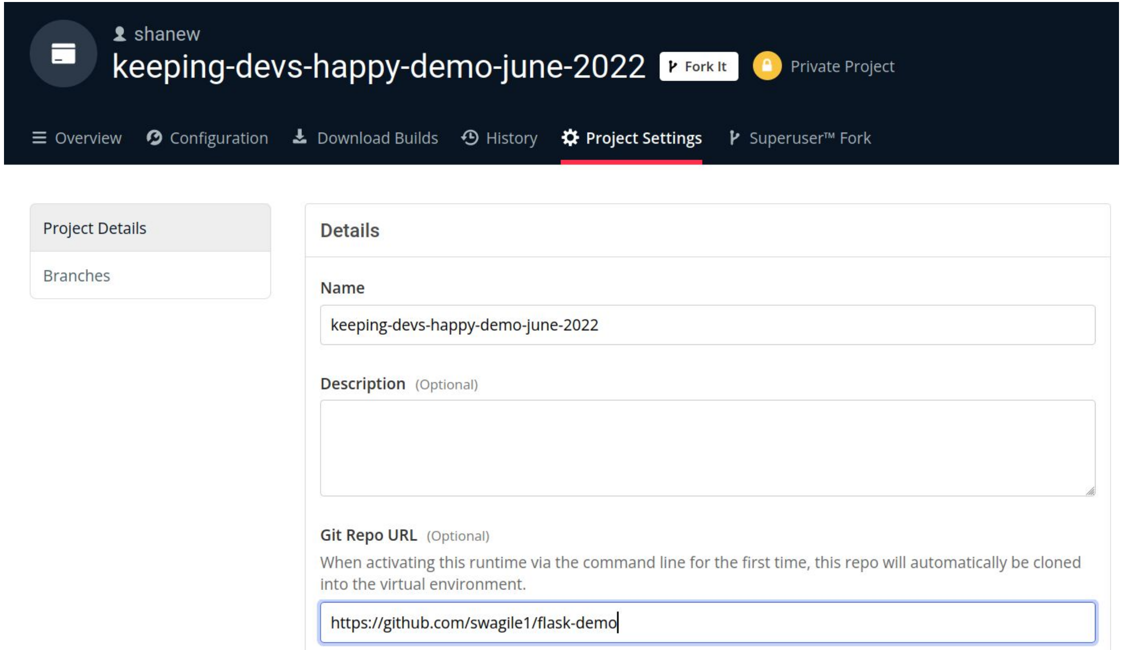
This command installs the State Tool package manager, which lets you manage your runtime. Find tips on managing your runtime in the CLI with our State Tool Cheat Sheet

Installed it already? Get the latest changes

If you've already installed this runtime, run the following command to update your local installation to use the latest build.

```
state pull
```

Demo: Associated With Your Repository



Details

Name

keeping-devs-happy-demo-june-2022

Description (Optional)

Git Repo URL (Optional)

When activating this runtime via the command line for the first time, this repo will automatically be cloned into the virtual environment.

https://github.com/swagile1/flask-demo

ActiveState

Q&A

ActiveState

Next Steps

Schedule a demo with our product experts:
https://www.activestate.com/get-demo/

Learn more about Managed Distributions:
https://www.activestate.com/blog/outsourced-dependency-vendoring/

Try the ActiveState Platform for free:
https://platform.activestate.com/

ActiveState

Webinar Feedback

Take our quick survey!

https://www.surveymonkey.com/r/keeping-devs-happy