Pixi: The evolution of cross-platform package manager

Presented By Don Setiawan

Table of contents

01

My Journey to Pixi

Reproducibility doesn't have to be hard

02

What is Pixi?

Hello, new best friend!

03

Demo

Here we go, let’s see what you can do

04

RL Pixi Examples

The true beauty

01

My Journey to Pixi

Reproducibility doesn't have to be hard

What’s my end goal?

Reproducibility

What’s my end goal?

Reproducibility

But really, I don’t want to bloat my laptop and allow others to have same development environment as me …

The timeline

Back in 2016, I was introduced to conda via Anaconda

Anaconda

Anaconda was so bloated, wanted a lighter management

Miniconda

OMG! The conda solver is so slow...

Mamba

Miniconda

Back to miniconda, libmamba solver is an option here now

Micromamba

I don't have to worry about having a 'base' environment that I shouldn't touch anyways??

Pixi

I can have my conda dependency and pip dependency in the same place?!

What really drew me to Pixi

Scott mentioned Pixi on Slack (2023)

Chatted Wolf in Scipy (Summer2024)

Full-stack project with backend, frontend, and cloud components (Fall 2024)

a Rust + Python software binding project (Winter 2024)

Concluded that it was flexible, blazing fast, and easy to work with... SOLD

02

What is Pixi?

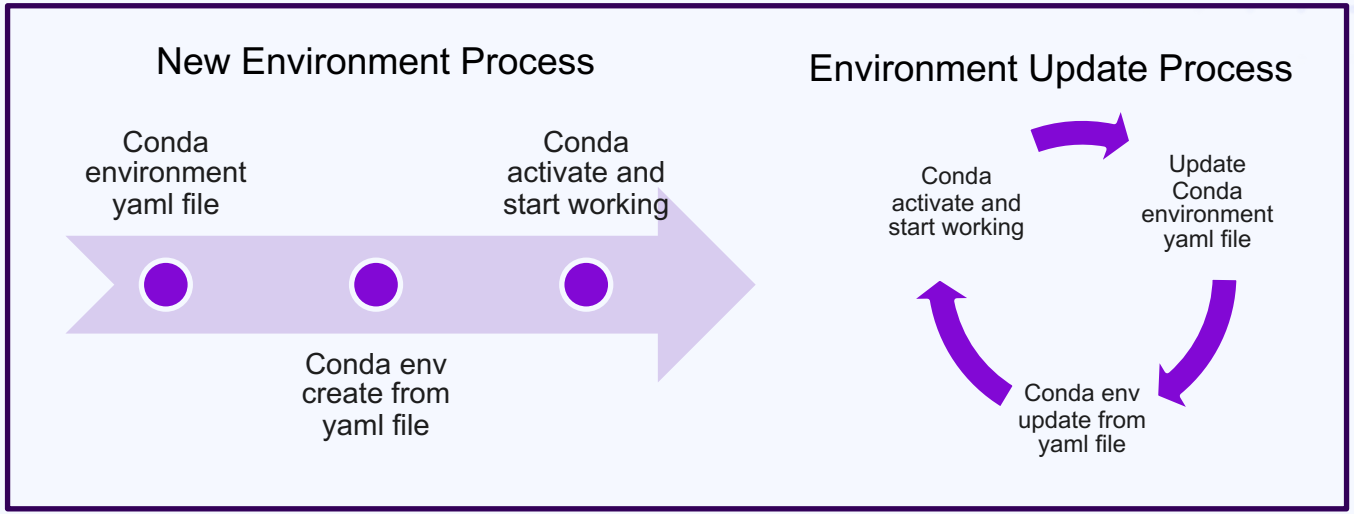
Hello, new best friend!

Before we talk about Pixi…

Let’s review our own current workflow with Conda.

This can be the classic conda or mamba.

Conda Workflow



Share environment file so others can also use it

Conda workflow pitfalls

- Environment yaml lack specific package version and build for the desired architectures, so may not work in others computer. (Can be done with separate tool called conda-lock)
- For python projects, we like having conda, but we need to setup the conda environment first, then install any Python dependency. (Now we have an environment.yml and requirements.txt or pyproject.toml)
- Conda doesn't integrate with pyproject.toml
- Conda doesn't have sub-dependencies, so what you define is what you get for the environment

Conda is still great

The idea and internals of conda is great and it will be still be used for many years to come.

Also, it has an amazing large community, especially from the conda-forge organization.

The new kid on the block...
Or is it?

A Prefix.dev project

pixi

conda package management simplified

- ★ Install packages globally or per-project
- ★ Simplified project management & auto-lockfile creation
- ★ Cross-platform, blazingly fast, written in Rust

How is it useful?

- Pixi installs conda AND PyPI packages, uses conda-first approach
- With Pixi everything is a project/environment
- Reproducibility, replicability, and portability are first class citizens in Pixi

See similarities to conda?

- Pixi installs conda AND PyPI packages, uses conda-first approach
- With Pixi everything is a project/environment
- Reproducibility, replicability, and portability are first class citizens in Pixi

So what’s actually different?

- Pixi installs conda AND PyPI packages, uses conda-first approach (written in Rust, uses resolvo for deps solver, and uv for PyPi package management)
- With Pixi everything is a project/environment (main emphasis on project-based approach)
- Reproducibility, replicability, and portability are first class citizens in Pixi (truly achieved those things)

Pixi Concepts

Pixi **Global**: Global environments, accessible from anywhere

Pixi **Environments**: The virtual environments where dependencies live

Pixi **Tasks**: The custom commands that we can chain

Pixi **Project**: The project defined in Pixi workspace manifest (pixi.toml or pyproject.toml)

03

Demo

Here we go, let’s see what you can do

04

RL Pixi Examples

The true beauty

Manifest Examples

1. **Resilience**: https://github.com/UW-THINKlab/resilience/blob/main/pixi.toml
2. **OpenIRE**: https://github.com/uw-ssec/open-ire/blob/main/pyproject.toml
3. **FutureDaws**: https://github.com/uw-ssec/futuredawgs/blob/main/pyproject.toml
4. **GNATSS Workshop**: https://github.com/seafloor-geodesy/gnatss-workshop/blob/main/pixi.toml
5. **HPyX**: https://github.com/uw-ssec/HPyX/blob/main/pixi.toml

Bonus Slide

New Conda Recipe Format is HERE!

See the Accepted Conda Enhancement Proposal:
https://github.com/conda/ceps/blob/main/cep-0013.md

You can now build conda recipe locally with ease with rattler-build:
https://prefix.dev/blog/the_love_of_building_conda_packages