We're pleased to announce Pants 2.18.0, the latest release of [Pantsbuild, the scalable and ergonomic build system](https://www.pantsbuild.org).
To update, set `pants_version = "2.18.0"` in your `pants.toml`. If you're not using Pants to build, format, test (and more) your codebase, [get started now](https://www.pantsbuild.org/docs/getting-started).

Highlights in 2.18.0 include:

- Even more improvements to AWS Lambda and Google Cloud Function support, including simpler configuration and support for AWS Lambda Layers.
- Many more options can now be controlled when building Docker images.
- Significant feature work and bug fixes to "deploy" backends like Helm and Terraform.

There's [much more](https://github.com/pantsbuild/pants/blob/2.18.x/src/python/pants/notes/2.18.x.md), too: performance optimizations, new backends, bug fixes and features!

Starting from the previous 2.17.0 release, you can now [support Pants financially](https://www.pantsbuild.org/docs/sponsorship).

Pants is an open-source project that is not owned nor controlled by any one company or organization, and does incur some expenses. These expenses are managed by Pants Build, a non-profit that was established for this purpose. This non-profit's only source of revenue is sponsorship by individuals and companies that support Pants.

We offer [formal sponsorship tiers for companies](https://www.pantsbuild.org/docs/sponsorship), as well as individual sponsorships via [GitHub](https://github.com/sponsors/pantsbuild).

Experimental backends!

Did you know that Pants has a lot of "experimental" backends? These include support for languages, tools and ecosystems that are still in preview. The [documentation of backends](https://www.pantsbuild.org/v2.18/docs/enabling-backends) now lists all these backends. Please go check it out, try backends relevant to you and let us know how they go for your use cases, [report bugs](https://www.pantsbuild.org/v2.18/docs/getting-help) and/or [submit fixes](https://www.pantsbuild.org/v2.18/docs/contributor-overview)! People like you trying them is how we can perfect the features and squash the bugs to stabilize them.

Functions-as-a-service improvements for Python

Pants includes support for building function-as-a-service artifacts for Python, via the [`pants.backend.awslambda.python`](https://www.pantsbuild.org/v2.18/docs/awslambda-python) and [`pants.backend.google_cloud_function.python`](https://www.pantsbuild.org/v2.18/docs/google-c

loud-function-python) backends. These backends make it easy to generate zip artifacts appropriate for uploading to cloud services.

These backends had significant improvements to their start-up performance and size in 2.17, and the improvements have continued in 2.18:

- The [new `python_aws_lambda_layer` target](https://www.pantsbuild.org/v2.18/docs/awslambda-python#building-a-lambda-layer) can be used to build Layers for AWS Lambda. For instance, this allows putting third-party requirements from PyPI into a layer, and only first party-sources from your repository into the function code, making for faster builds and uploads. (For consistency, the `python_awslambda` target has been renamed to `python_aws_lambda_function`.)

-The `runtime` field is no longer necessary when the interpreter constraints unambiguously refer to an entire single major version. For instance, with `[python].interpreter_constraints = ["==3.11.*"]` in `pants.toml`, one doesn't need to specify `runtime="python3.11"` to `python_aws_lambda_function` or `runtime="python311"` to `python_google_cloud_function`, as it will be inferred.

- For AWS Lambda targets, the `complete_platforms` argument now has default values provided by Pants, which gives more reliable builds, especially when building on a different platform than the AWS Lambda itself. This change also means that only one of the `complete_platforms` or `platforms` fields should be provided.

Simplification of Python tool lockfiles

You now install custom versions of Python-based tools (such as Pytest, MyPy and so on) by pointing the tool's `install_from_resolve` option at a regular Python [resolve](https://www.pantsbuild.org/docs/python-lockfiles). You create the lockfiles for these resolves just as you would for your own code's resolves. You can have one resolve per tool, or one resolve for multiple tools, and you can even install tools from the same resolve you use for your code. This is useful when you have tools that also provide runtime libraries, such as Pytest. See [the documentation](https://www.pantsbuild.org/docs/python-lockfiles#lockfiles-for-tools) for more details.

Note that this means that you can no longer `export` the built-in, default lockfile for a tool, as that is not represented by a named resolve. If you need to export Python tool virtualenvs for consumption outside of Pants, e.g., by an IDE, you must first create a resolve for them.

Infrastructure-as-code iteration

Pants has nascent support for describing and deploying infrastructure via the [`pants.backend.experimental.helm`](https://www.pantsbuild.org/v2.18/docs/helm-overview) and `pants.backend.experimental.terraform` backends. These have seen several improvements in 2.18.0:

- Improved testing with [`helm_unittest_tests` target](https://www.pantsbuild.org/v2.18/docs/helm-overview#helm-unit-tests) as it can now be automatically created by the `tailor` goal, and supports [snapshot testing](https://www.pantsbuild.org/v2.18/docs/helm-overview#snapshot-testing) via the new `generate-snapshots` goal

- Support for running Kubeconform on Helm targets via [the new `pants.backend.experimental.helm.check.kubeconform` backend](https://www.pantsbuild.org/v2.18/docs/helm-kubeconform)

- The [new `terraform_deployment` target](https://www.pantsbuild.org/v2.18/docs/reference-terraform_deployment) allows configuring a `terraform_module` to be deployed with [the `experimental-deploy` goal](https://www.pantsbuild.org/v2.18/docs/reference-experimental-deploy).

- A new `pants.backend.experimental.terraform.lint.tfsec` backend has been added to run the `tfsec` linter as part of the `lint` goal.

In addition to those major improvements, many smaller bug fixes and features are included in 2.18.0.

Docker finesse

The `pants.backend.docker` backend now supports many more options to control how images are built:

- on [the `[docker]` subsystem](https://www.pantsbuild.org/v2.18/docs/reference-docker): `build_hosts`, `build_no_cache`

- on [the `docker_image` target](https://www.pantsbuild.org/v2.18/docs/reference-docker_image): `build_network`, `build_platform`, `extra_build_hosts`

These new options allow, for instance, more easily building a x86-64 image while running on an arm64 host.

More visible visibility

[The `pants.backend.experimental.visibility` backend](https://www.pantsbuild.org/v2.18/docs/validating-dependencies) now also runs as a linter, such as when running `pants lint ::`, to report all visibility violations at one time.

This backend allows applying rules and restrictions to which targets can depend on other targets, for example, a repository with two applications and some shared library code can use the visibility backend to enforce that code in one application cannot depend on code in the other application.

Finding paths between addresses that expand to multiple targets

Pants previously allowed finding all paths in the dependency graph between two individual targets, like a `python_source` and a `python_test`. You can now find all paths that exist between multiple targets, for instance, by passing a directory address.This will help you to answer various questions such as “How is this Python module connected to a Python package, if at all?” or “How tightly are two Java packages connected?”.

This also enables sophisticated dependency graph analytics. By listing all the paths in the repo, for example, between sources and tests, one can find the source modules that have most paths leading to tests, identify very long paths, and many more.

For example, this command will count all paths between source and test targets in a repository:

```

$ pants paths --from=tests:: --to=src:: | jq length

42

```

Much more!

Pants 2.18 involved several months of work from dozens of contributors, thank you everyone!

See the [release notes](https://github.com/pantsbuild/pants/blob/2.18.x/src/python/pants/notes/2.18.x.md) for all of the new features and fixes.

To update, set `pants_version = "2.18.0"` in your `pants.toml`. If you’re not using Pants to build, format, test (and more) your codebase, [get started now](https://www.pantsbuild.org/docs/getting-started).