# Pants 2.8 adds Golang, Autoflake & Pyupgrade, and file-level metadata

We're pleased to announce Pants 2.8.0, the latest release of Pants, the scalable and ergonomic build system.

To update, set `pants_version = "2.8.0"` in your `pants.toml`. See upgrade tips, including a new `update-build-files` goal to automate some of the upgrade.

## Experimental Golang support

Pants now augments Go's (already excellent!) tooling with these unique benefits:

1. A consistent interface for all languages and tools in your repository.
2. Integration with Git + advanced project introspection.
3. Remote caching and execution.
4. Unlike legacy build systems, these benefits all come with minimal boilerplate.

Check out our announcement post for more, and an example repository to try it out.

## Autoflake and Pyupgrade

Pants now supports Autoflake and Pyupgrade, in addition to already supporting Black, isort, Flake8, Bandit, Docformatter, MyPy, Pylint, Yapf, Shfmt, and Shellcheck!

Autoflake removes unused imports automatically, and Pyupgrade modernizes your Python syntax to use new features like f-strings.

Before:

```
# app.py
import os

name = "pantsbuild"
print("Hello %s!" %  x)
```

After running `./pants fmt app.py`:

```
name = "pantsbuild"
print(f"Hello {name}!")
```

# Publishing Docker images

Pants 2.7 added experimental Docker support, which gives you a single command to streamline deploying your Python applications with Docker.

Pants 2.8 expands its Docker support to include publishing images. Now, `./pants publish` will first build the Docker image—including copying in any relevant PEX binaries, Python wheels, and files. Pants will then publish the image to the registries you specify.

```
❯ ./pants publish src/docker::
14:41:51.98 [INFO] Built docker image: docker.registry.net:8181/test-example:1.2.5
The push refers to repository [docker.registry.net:8181/test-example]
7c521f914a2d: Layer already exists
...
62a747bf1719: Layer already exists
1.2.5: digest: sha256:961beb6126320d277ad4236f9f12694ad4e077a16f73b896b3292023007a002c
size: 2428

✓ docker.registry.net:8181/test-example:1.2.5 published.
```

Pants 2.8 now also can run your Docker images with `./pants run`, making sure that the image is first built with all your binaries and files copied into it.

# Google Cloud Functions

Pants can now create a .zip file understood by Google Cloud Function, allowing you to develop your functions alongside the rest of your project.

This functionality uses the same technology that allows Pants to create AWS Lambda functions.

# PEP-517 packaging support

As explained in a previous post, Pants can now act as a PEP 517 build frontend.

Pants has long had the ability to generate setup.py files for you, to streamline creating and managing distributions. Pants 2.8 extends this with support for PEP 517, the Python standard for specifying how distributions should be built.

Under this standard, you can specify a “build backend” in a pyproject.toml file, and Pants will install and invoke it appropriately. This means that Pants can now manage run any distribution build, including building native extensions with setup.py and building Poetry projects.

## File-level metadata with `overrides`

Since Pants 2.0, Pants has operated at the file level. Unlike legacy build tools, Pants runs tests on individual files, for example, rather than "targets" (usually directories). Likewise, Pants understands metadata like your dependencies at the file level, which gives you fine-grained caching.

All this time, Pants has been dynamically generating one "target" per file, but we hid that from you. Now, we added new targets like `python_test` and `python_source` which correspond to individual files. You can manually create those, or let Pants generate the targets for you:

```
python_tests(
    name="tests",
    # Generates two `python_test` targets.
    sources=["osutil_test.py", "dirutil_test.py"],
)
```

You can now also easily override metadata at the file-level through a new `overrides` field:

```
python_tests(
    name="tests",
    overrides={
        "dirutil_test.py": {"timeout": 10},
        ("osutil_test.py", "contextutil_test.py"): {"dependencies": [":test_data"]},
    },
)
```

That is, you get precise, file-level metadata. But without demanding the boilerplate of legacy tools like Bazel.

### Impact to existing users

Improving this modeling required making some other changes:

- Some targets were renamed, like `python_library` to `python_sources`. Run `./pants update-build-files` to automatically rename your BUILD files. (The old names will still work in Pants 2.8.)
- `conftest.py` no longer belongs to the `sources` of `python_tests` by default. Instead, it belongs to `python_test_utils`. Run `./pants tailor` to automatically add these new targets.
- `./pants filter --target-type` and `./pants peek` both now differentiate between `python_tests` and `python_test` targets, for example. Usually you will want to update your scripts to use `python_test` rather than `python_tests` and so on.
- The command line arguments `dir:` and `dir::` now include generated targets for project introspection goals like `list` and `peek`.

## Other changes

- TOML dictionaries work for `dict` options in `pants.toml`, e.g.

```
[tailor]
alias_mapping = { python_sources = "custom_python_sources", python_tests = "custom_python_tests" }
```

or

```
[tailor.alias_mapping]
python_sources = "custom_python_sources"
python_tests = "custom_python_tests"
```

- You can set up CLI aliases, like `./pants green ::` being an alias for `./pants fmt lint check ::`

```
[cli.alias]
green = "fmt lint check"
```

- `[coverage-py].fail_under` option added to fail if Python test coverage is below a certain number.

- `pex_binary` has a `script` field to allow setting console scripts, which allows running third-party dependencies.
- `pex_binary` has a `restarable` field, which you can set so that Pants will restart the binary when input files have changed. This is particularly useful for auto-reloading Flask and Django servers, for example.
- Added `--tailor-check` option as a dry run for automatically adding new targets to your BUILD files.
- Added `[tailor].ignore_paths` and `[tailor].ignore_adding_targets` options to avoid adding targets that are false positives.
- Added `./pants help tools` to list all tools installed by Pants, including their versions.
- `typecheck` is deprecated in favor of `check`, which will run MyPy and compilation for languages like Go and Java.

See the full changelog for more changes.

# Upcoming in Pants 2.9

We are continuing to work on language support: adding more Go features and getting Java/Scala into alpha/beta state.

We also plan to resume the redesign of Python lockfiles started with Pants 2.7's tool lockfiles.

--

Try out our example Python repository, and let us know what you think in Slack!

## Thanks

Special shout-outs to these contributors, along with everyone who shared feedback on changes in 2.8 and who tested 2.8 release candidates!

- Chris Williams for Google Cloud Functions
- Andreas Stenius for Docker publishing and CLI aliases.
- Tom Dyas and Eric Arellano for Go support.
- Benjy Weinberger for PEP 517 support.
- Asher Foa for Pyupgrade.
- Patrick Lawson for Autoflake.
- Stu Hood for auto-reloading when a PEX binary changes.