Typhon
Nix-based continuous integration
You are a Nix enthusiast...

... and you are asked to set up CI at work.
You convince your boss to use Nix.
You install Hydra!
The sad reality:

- Hard to set up, stateful
- System-defined plugins
- Barely extensible
- Deployment is flaky and stateful

The dream:

- Simplicity, declarativity
- User-defined plugins
- Extensible with Nix
- Better deployment support
Typhon
Defining a jobset

github:alice/website#typhonJobs

```python
{
    "x86_64-linux".build = «derivation /nix/store/...»;
    "x86_64-linux".test = «derivation /nix/store/...»;
    ...
}
```
Defining a project

github:alice/website#typhonProject

mkGithubProject {
    owner = "alice";
    repo = "website";
    deploy.main.githubPages = true;
    secrets = ./secrets.age;
}

Actions!

- User-defined (by project)
- Built with Nix
- Sandboxed
- Triggered by Typhon

Action 1
- Setting statuses

Nix job

Action 2
- Setting statuses
- Deploying
{"job": "build",
"system": "x86_64-linux",
"secrets": ...
...}
A Nix library to write CI workflows

github:typhon-ci/typhon#lib

{  
  mkGithubProject = «lambda @ ...»;
  mkGitlabProject = «lambda @ ...»;
  ...
}

How it’s made
formated (x86_64-linux)
succeeded 35 mins ago in 1m 8s

1. unpackPhase
2. Running phase: unpackPhase
3. unpacking source archive /nix/store/1g7wb3yvb4xvz2ffv613k3gwz7qyml-8ah8v1pq5vsah6jihx19njdizjm25xr-source
4. source root is 8ah8v1pwx5vsah6jihx19njdizjm25xr-source
5. patchPhase
6. Running phase: patchPhase
7. updateAutotoolsGnuConfigScriptsPhase
8. Running phase: updateAutotoolsGnuConfigScriptsPhase
9. configurePhase
10. Running phase: configurePhase
11. no configure script, doing nothing
12. buildPhase
13. Running phase: buildPhase
Try it out!

Documentation:  https://typhon-ci.org

Repo:  https://github.com/typhon-ci/typhon

Disclaimer:  Typhon is a prototype! 🚧

Contribute:  Help us write cool actions!

Contact us:  Paul-Nicolas Madelaine <pnm@pnm.tf>
             Lucas Franceschino <lucas@franceschino.fr>