

Nix for PHP Developers (and everyone else)

Oliver Davies (opdavies)

<https://www.oliverdavies.uk>

**You do not need to be
a PHP or Drupal Developer**

About Me

- HTML, CSS, PHP since 2007
- Drupal since 2008
- JavaScript, TypeScript, Go
- Full-time Linux since ~2015
- Nix/NixOS since 2022
- Homelabber since 2024







Presentations

Since September 2012, I have given 104 public talks and workshops at various conferences and meetups, in-person and remotely, on topics including PHP, Drupal, automated testing, Git, CSS, and systems administration.

Building Static Websites with Sculpin

How to use Sculpin to generate static HTML websites.

TDD - Test-Driven Drupal

How to write automated tests for Drupal, and how to create a new Drupal module using test driven development.

Taking Flight with Tailwind CSS

An introduction to the utility-first approach to writing CSS with a focus on the Tailwind CSS framework.

[Explore](#)[Help](#)[\[← Sign in](#)[Repositories](#)[Users](#)[Organizations](#)

Filter ▾

Sort ▾

**opdavies / nix-config**

Updated 7 hours ago

Nix ☆ 0 🍷 0

**opdavies / reading-college-demos**

Updated 7 hours ago

Nix ☆ 0 🍷 0

**opdavies / oliverdavies.uk**

Updated 19 hours ago

Twig ☆ 0 🍷 0

**opdavies / presentations**

Updated 2 days ago

Nix ☆ 0 🍷 0

**opdavies / drupal-nix-flake-example**

Updated 2 weeks ago

Nix ☆ 0 🍷 0

**opdavies / oliverdavies.uk-tome**

Updated 3 weeks ago

Twig ☆ 0 🍷 0

**opdavies / phpsw-sculpin-demo**

Updated 2 months ago

Twig ☆ 0 🍷 0

**opdavies / bootstrap-with-tailwind**`php` `sculpin` `tailwind-css` `twig`

Updated 2 months ago

Twig ☆ 0 🍷 0

**opdavies / sculpin-demo**

Updated 2 months ago

Twig ☆ 0 🍷 0

**opdavies / ssg-test** `Archived`

Updated 3 months ago

HTML ☆ 0 🍷 0

**opdavies / sculpin-from-scratch-app**

Updated 4 months ago

JavaScript ☆ 0 🍷 0

**opdavies / learn-go-with-tests**

Updated 4 months ago

Go ☆ 0 🍷 0

1-100 of 161 ⏪ ⏩ ✂ 📺 A-Z 📄



2012
2009



Airplane!
1980



The Amazing Spider-...
2012



Angels & Demons
2009



Anger Management
2003



Apollo 13
1995



Avatar
2009



The Avengers
2012



Avengers: Endgame
2019



Back to the Future
1985



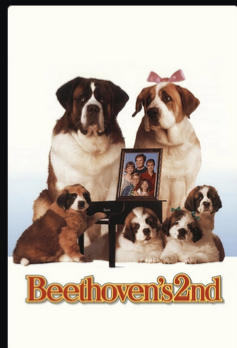
Back to the Future Pa...
1989



Back to the Future Pa...
1990



Beethoven
1992



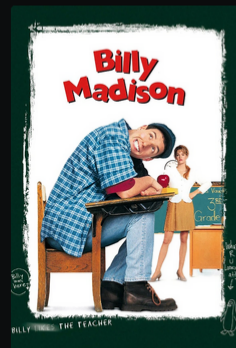
Beethoven's 2nd
1993



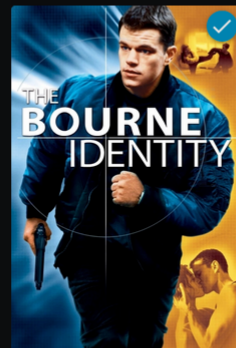
Beetlejuice
1988



Bill & Ted's Excellent ...
1989



Billy Madison
1995



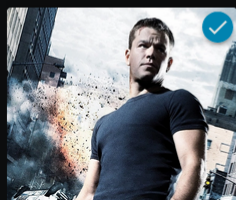
The Bourne Identity
2002



THE BOURNE LEGACY



THE BOURNE
SUPREMACY



Bruce
ALMIGHTY



THE
BUCKET LIST



CAPTAIN AMERICA
THE FIRST AVENGER



Photos

Explore

Map

Sharing

LIBRARY

Favorites

Albums

Recent

Camera

DrupalCamp E...

Utilities

Archive

Trash

Storage space

1.6 TiB of 1.8 TiB used

Buy Immich



Server Online

v1.132.3

Mon 7 Apr



2025

2024

2023

2022

2021

2020

2019

2018

2017

2016

2015



Nix for PHP Developers

Oliver Davies

**Who has PHP installed
on their computer?**

**Who has multiple versions
of PHP installed?**

Installing PHP

- WAMP (Windows), MAMP (macOS), XAMPP
- Homebrew (`brew install php`)
- `apt` , `dnf` , `yum` , `pacman` `yay`
- Laravel Herd/Sail
- Symfony CLI?
- Containers (Docker, Podman, LXC, etc)
- Virtual machines (VMWare, Virtualbox, etc)



What is Nix?

- Package manager with more than **120,000** packages (`nixpkgs`)
- Operating system with more than **20,000** packages (NixOS)
- Tool for declaratively building software reproducibly
- A functional domain-specific configuration language
- Home Manager, `nix-darwin` , `devenv`
- Started as a research project by Eelco Dolstra around 2003

What is Nix?

- Package manager
- Build tool
- Operating system
- Language

[Back to nixos.org](#)[Packages](#)[NixOS options](#)[Flakes](#)[Experimental](#)[NixOS Wiki](#)

🔍 Search more than **120 000 packages**

Channel:

Please help us improve the search by [reporting issues](#).

❤️ Elasticsearch instance graciously provided by [Bonsai](#). Thank you! ❤️

Package sets

php84Extensions 111

php83Extensions 111

php82Extensions 111

php81Extensions 111

No package set 49

emacsPackages 29

php84Packages 19

php83Packages 19

php82Packages 19

php81Packages 19

vimPlugins 7

haskellPackages 6

vscode-extensions 5

rPackages 5

python313Packages 4

python312Packages 4

Showing results 1-50 of 643 packages.

Sort: Best match ▼

Data from nixpkgs [a58390ab](#).

php

HTML-embedded scripting language

Name: [php-with-extensions](#) Version: 8.4.11 [Homepage](#) [Source](#) License: [PHP License v3.01](#)

▼▼▼ Show more package details ▼▼▼

php83

HTML-embedded scripting language

Name: [php-with-extensions](#) Version: 8.3.24 [Homepage](#) [Source](#) License: [PHP License v3.01](#)

▼▼▼ Show more package details ▼▼▼

php82

HTML-embedded scripting language

Name: [php-with-extensions](#) Version: 8.2.29 [Homepage](#) [Source](#) License: [PHP License v3.01](#)

▼▼▼ Show more package details ▼▼▼

php81

HTML-embedded scripting language

Name: [php-with-extensions](#) Version: 8.1.33 [Homepage](#) [Source](#) License: [PHP License v3.01](#)

kdePackages	3
sbclPackages	2
libsForQt5	2
apacheHttpdPackages	1

Licenses

PHP License v3.01	302
MIT License	127
BSD 3-clause "New" or "Revised" License	53
Apache License 2.0	48
Unfree	22
BSD 2-clause "Simplified" License	14
GNU General Public License v2.0 or later	11
GNU Library General Public License v2 or later	8
GNU General Public License v3.0 or later	6

phpunit

PHP Unit Testing framework

Name: `phpunit` Version: 12.1.5 [Homepage](#) [Source](#) License: BSD 3-clause "New" or "Revised" License

▼▼▼ Show more package details ▼▼▼

phpactor

Mainly a PHP Language Server

Name: `phpactor` Version: 2025.04.17.0 [Homepage](#) [Source](#) License: MIT License

▼▼▼ Show more package details ▼▼▼

vimPlugins.nvim-treesitter-parsers.php

Name: `vimplugin-treesitter-grammar-php` [Homepage](#) [Source](#)

▼▼▼ Show more package details ▼▼▼

apacheHttpdPackages.php

HTML-embedded scripting language

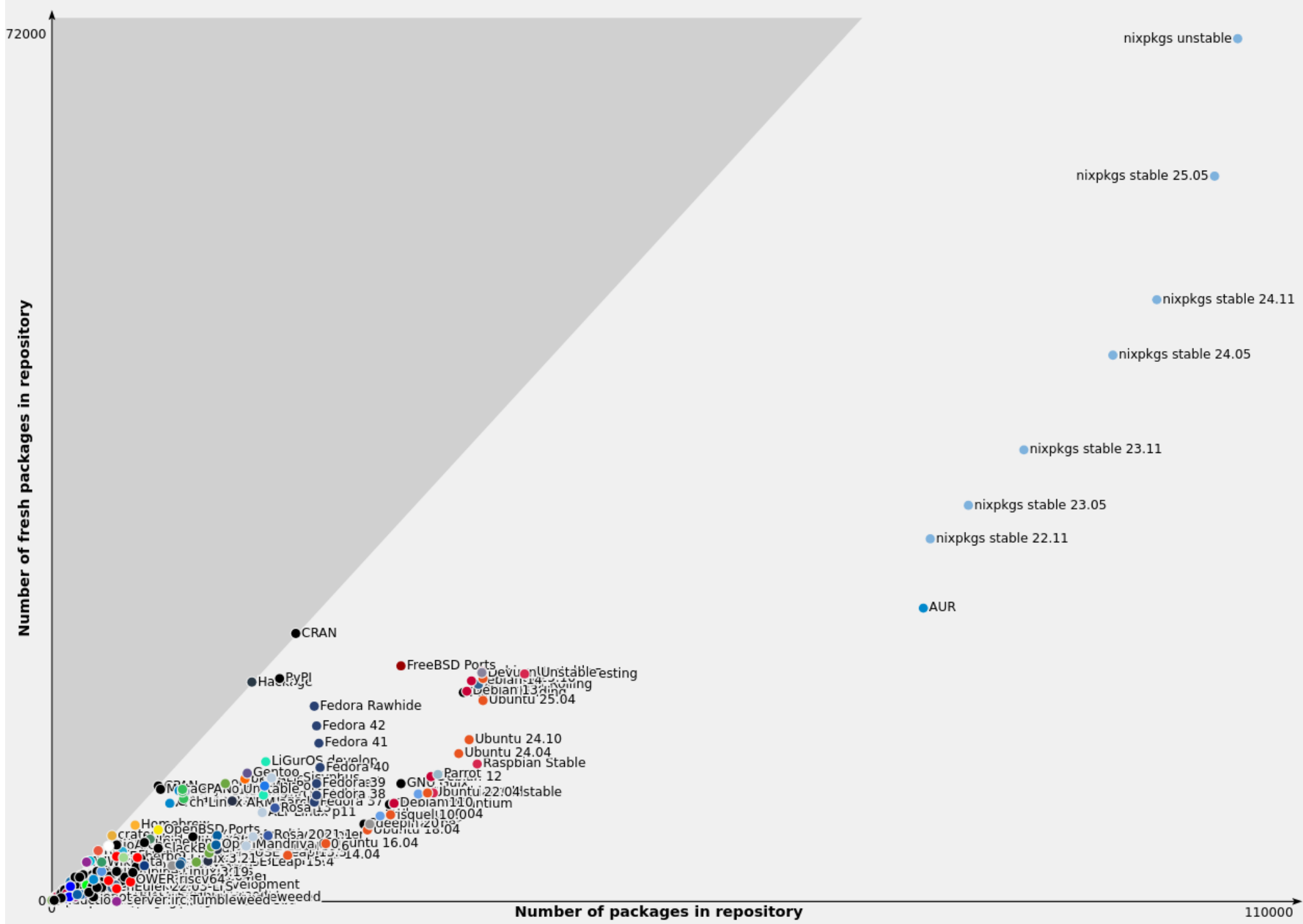
Name: `php-with-extensions` Version: 8.4.11 [Homepage](#) [Source](#) License: PHP License v3.01

▼▼▼ Show more package details ▼▼▼

phpdocumentor

PHP documentation generator

Name: `phpdocumentor` Version: 3.7.1 [Homepage](#) [Source](#) License: MIT License



Installing Nix

<https://nixos.org/download>

```
$ sh <(curl --proto '=https' --tlsv1.2 -L https://nixos.org/nix/install) \
--daemon
```

Different instructions for Linux, macOS and Windows (WSL2).

Different instructions for NixOS.

```
$ nix --version

nix (Nix) 2.28.4
```

This installation tool will set up your computer with the Nix package manager. This will happen in a few stages:

1. Make sure your computer doesn't already have Nix. If it does, I will show you instructions on how to clean up your old install.
2. Show you what I am going to install and where. Then I will ask if you are ready to continue.
3. Create the system users (uids [30001..30032]) and groups (gid 30000) that the Nix daemon uses to run builds. To create system users in a different range, exit and run this tool again with `NIX_FIRST_BUILD_UID` set.
4. Perform the basic installation of the Nix files daemon.
5. Configure your shell to import special Nix Profile files, so you can use Nix.
6. Start the Nix daemon.

Installing PHP with Ubuntu

```
$ apt update -y
```

```
$ apt-get install php
```

```
$ which php
```

```
/usr/bin/php
```

```
$ php -v
```

```
PHP 8.3.6
```

Running PHP with Nix

```
$ nix run nixpkgs#php -- -v
```

```
PHP 8.4.11
```

```
$ nix shell nixpkgs#php
```

```
$ which php  
/nix/store/s4kv9bzqawhqcyjg9xm2hji3jap6d6kd-php-with-extensions-8.4.11/bin/php
```

```
$ php -v
```

```
PHP 8.4.11
```

Running PHP with Nix

```
$ nix run nixpkgs#php83 -- -v
```

```
PHP 8.3.25
```

```
$ nix run nixpkgs#php82 -- -v
```

```
PHP 8.2.29
```

Using a `shell.nix` file

```
{ pkgs ? import <nixpkgs> {} }:  
  
pkgs.mkShell {  
  packages = with pkgs; [  
    php  
    phpPackages.composer  
    phpactor  
  ];  
}
```

```
$ nix-shell shell.nix
```

```
[nix-shell:~/my-project]$ composer -V
```

```
Composer version 2.8.5 2025-01-21 15:23:40
```

Using a `flake.nix` file

```
{
  inputs.nixpkgs.url = "github:nixos/nixpkgs/nixos-unstable";

  outputs = inputs:
    let
      system = "x86_64-linux";
      pkgs = import inputs.nixpkgs { inherit system; };
    in {
      devShells.${system}.default = (import ./shell.nix {
        inherit pkgs;
      });
    };
}
```

Using a `flake.nix` file

```
{
  inputs.nixpkgs.url = "github:nixos/nixpkgs/nixos-unstable";

  outputs = inputs:
    let
      system = "x86_64-linux";
      pkgs = import inputs.nixpkgs { inherit system; };
    in {
      devShells.${system}.default = pkgs.mkShell {
        packages = with pkgs; [
          php
          phpPackages.composer
          phpactor
        ];
      };
    };
}
```

Structure of a Flake-based project

```
.  
├── composer.json  
├── composer.lock  
├── flake.lock  
├── flake.nix  
├── output_dev  
├── source  
└── vendor
```

As well as **flake.nix**, we also get **flake.lock**.

Pinning packages

```
# flake.nix
```

```
inputs = {  
  nixpkgs.url = "github:nixos/nixpkgs/nixos-unstable";  
  nixpkgs-stable.url = "github:nixos/nixpkgs/nixos-24.11";  
  nixpkgs-pinned.url = "github:nixos/nixpkgs/45570c299dc2";  
}
```

```
{  
  let  
    pkgs-stable = import inputs.nixpkgs-stable { inherit system; };  
  in {  
    packages = [ pkgs-stable.php ];  
  };  
}
```

Managing services

With NixOS:

```
services.mysql.enable = true;  
  
services.mysql.initialDatabases = [  
  { name = "my_project"; }  
  { name = "another_project"; }  
];
```

Without NixOS:

<https://github.com/juspay/services-flake>

Managing services

```
{
  inputs = {
    flake-parts.url = "github:hercules-ci/flake-parts";

    nixpkgs.url = "github:nixos/nixpkgs/nixpkgs-unstable";

    process-compose.url = "github:Platonic-Systems/process-compose-flake";

    services.url = "github:juspay/services-flake";
  };

  # ...
}
```

Managing services

```
imports = [  
  inputs.process-compose.flakeModule  
];  
  
perSystem = { config, lib, pkgs, ... }: {  
  process-compose."default" = {  
    imports = [  
      inputs.services.processComposeModules.default  
    ];  
  
    services = { # ... };  
  
    settings.processes = { # ... };  
  }
```

Managing services

```
services = {  
  mysql."mysql1" = {  
    enable = true;  
  
    initialDatabases = [  
      { name = "drupal_nix_flake_example"; }  
    ];  
  };  
};
```

Managing services

```
settings.processes = {  
  php = {  
    command = pkgs.writeShellApplication {  
      name = "php-local-server";  
  
      text = "${getExe php} -S 127.0.0.1:${toString webPort} -t web";  
    };  
  
    depends_on."mysql1".condition = "process_healthy";  
  };  
};
```

Managing services

```
devShells.default = pkgs.mkShell {  
  inputsFrom = [  
    config.process-compose."default".services.outputs.devShell  
  ];  
  
  nativeBuildInputs = with pkgs; [  
    php  
    phpPackages.composer  
  ];  
};
```

Run `nix run .` and go to <http://localhost:8000>.

Version: v1.46.0
Hostname: t480
Processes: 2/3
RAM | CPU: 104.3 MiB | 1.0%

Process Compose 🔥

PID(P)	NAME(N)	NAMESPACE(C)	STATUS(S)	AGE(A)	HEALTH(H)	MEM(M)	CPU(U)	RESTARTS(R)	EXIT CODE(E)
649743	mysql1	mysql.mysql1	Running	18s	Ready	101.0 MiB	0.9%	-	-
649759	mysql1-configure	mysql.mysql1	Completed	-	-	-	-	-	0
649760	php	default	Running	15s	-	3.2 MiB	0.1%	-	-

mysql1

```
2025-08-20 8:47:56 0 [Note] Starting MariaDB 10.11.11-MariaDB source revision e69f8cae1a15e15b9e4f5e0f8497e1f17bdc81a4 server_uid SBUT5+V4gzRHX5cSl7/yqJApnrs= as process 649743
2025-08-20 8:47:56 0 [Note] InnoDB: Compressed tables use zlib 1.3.1
2025-08-20 8:47:56 0 [Note] InnoDB: Number of transaction pools: 1
2025-08-20 8:47:56 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-08-20 8:47:56 0 [Note] InnoDB: Using liburing
2025-08-20 8:47:56 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
2025-08-20 8:47:56 0 [Note] InnoDB: Completed initialization of buffer pool
2025-08-20 8:47:56 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2025-08-20 8:47:56 0 [Note] InnoDB: End of log at LSN=46980
2025-08-20 8:47:56 0 [Note] InnoDB: 128 rollback segments are active.
2025-08-20 8:47:56 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
2025-08-20 8:47:56 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
2025-08-20 8:47:56 0 [Note] InnoDB: log sequence number 46980; transaction id 14
2025-08-20 8:47:56 0 [Note] Plugin 'FEEDBACK' is disabled.
2025-08-20 8:47:56 0 [Note] InnoDB: Loading buffer pool(s) from /home/opdavies/tmp/drupal-nix-flake-example/data/mysql1/ib_buffer_pool
2025-08-20 8:47:56 0 [Note] InnoDB: Buffer pool(s) load completed at 250820 8:47:56
2025-08-20 8:47:56 0 [Note] Server socket created on IP: '0.0.0.0'.
2025-08-20 8:47:56 0 [Note] Server socket created on IP: '::'.
2025-08-20 8:47:56 0 [Note] mysqld: ready for connections.
Version: '10.11.11-MariaDB' socket: '/home/opdavies/tmp/drupal-nix-flake-example/data/mysql1/mysql.sock' port: 3306 MariaDB Server
```

Drupal 11.1.6

1 Choose language

2 Choose profile

3 Verify requirements

4 Set up database

5 Install site

6 Configure site

Choose language

English, British



Translations will be downloaded from the [Drupal Translation web-site](#). If you do not want this, select [English](#).

Save and continue

Multiple PHPs

- Laravel Herd/Sail
- Containers
- `phpenv`, `brew-php-switcher`, `phpbrew`
- Vagrant
- VMWare, VirtualBox
- Different physical machines?

**Nix is like `nvm`
for everything**

My ideal environment

```
$ php -v
```

```
The program 'php' is currently not installed.
```

```
$ cd ~/my-project
```

```
$ php -v
```

```
PHP 8.4.10
```

```
$ cd ~/another-project
```

```
$ php -v
```

```
PHP 8.3.24
```

Using direnv

<https://direnv.net>

It augments existing shells with a new feature that can load and unload environment variables depending on the current directory.

In NixOS:

```
{  
  programs.direnv.enable = true;  
  programs.direnv.nix-direnv.enable = true;  
}
```

The .envrc file

If you have a `flake.nix`:

```
use flake
```

Using a remote Flake:

```
use flake "git+https://git.oliverdavies.uk/opdavies/dev-shells#php84"
```

```
use flake "github:opdavies/dev-shells#php84"
```

Packaging for Nix

```
stdenv.mkDerivation (finalAttrs: {  
  pname = "hello";  
  version = "2.12.2";  
  
  src = fetchurl {  
    url = "mirror://gnu/hello/hello-${finalAttrs.version}.tar.gz";  
    hash = "sha256-WpqZbcKSzCTc9BH06H6S9qr luNE72caBm0x6nc4IGKs=";  
  };  
  
  env = lib.optionalAttrs stdenv.hostPlatform.isDarwin {  
    NIX_LDFLAGS = "-liconv";  
  };  
  
  doCheck = true;  
  doInstallCheck = true;  
  
  # ...
```

Packaging for Nix

```
pkgs.writeShellApplication {  
  name = "preview";  
  
  runtimeInputs = with pkgs.nodePackages; [  
    browser-sync  
  ];  
  
  text = ''  
    browser-sync start  
      --ignore '**/*.*' \  
      --no-notify \  
      --no-ui \  
      -sW  
  '';  
}
```

Packaging PHP for Nix

```
php.buildComposerProject2 (finalAttrs: {  
  pname = "phpactor";  
  version = "2025.07.25.0";  
  
  src = fetchFromGitHub {  
    owner = "phpactor";  
    repo = "phpactor";  
    tag = finalAttrs.version;  
    hash = "sha256-9XWlWwq+xvqPgKIc7IGoMVTxajjYsrPo/ra/0JIE168=";  
  };  
  
  vendorHash = "sha256-3xkt0QjytW4B0CgZdevat7zkSuZTPPvwz3yptiq5zoo=";  
  
  # ...
```

Packaging PHP for Nix

```
php.buildComposerProject2 (finalAttrs: {  
  pname = "pest";  
  version = "3.7.4";  
  
  src = fetchFromGitHub {  
    owner = "pestphp";  
    repo = "pest";  
    tag = "v${finalAttrs.version}";  
    hash = "sha256-ddsdVx/Vsg7GG11fGASouBU3HAJLSjs1AQGHx52TWzA=";  
  };  
  
  composerLock = ./composer.lock;  
  vendorHash = "sha256-r0J6PFp4Xfe89usoH455EAT30d2Tu3zd3+C/6K/kGBw=";  
  
  # ...
```

Packaging Sculpin

Packaging Sculpin was essentially the same:

<https://git.oliverdavies.uk/opdavies/lab/src/branch/main/nix/sculpin>

```
nix build  
  
./result/bin/sculpin generate
```

```
.  
├── flake.lock  
├── flake.nix  
├── output_dev  
│   └── index.html  
└── source  
    └── index.md
```

Packaging npm

```
pkgs.buildNpmPackage rec {  
  pname = "openapi-generate-html";  
  version = "0.5.3";  
  
  src = pkgs.fetchFromGitHub {  
    owner = "qazsato";  
    repo = pname;  
    rev = "v${version}";  
    hash = "sha256-+RmwoRhvfkaJ/d3EwID7E6noVV+M3h6pe7IEVYyuUwk=";  
  };  
  
  dontNpmBuild = true;  
  
  npmDepsHash = "sha256-7yYM43fAR2HLAC0QNw7N/t8Lk+17qNfeDKzfb1wx/0U=";  
}
```

Packaging Go

```
pkgs.buildGoModule rec {  
    pname = "openapi-mock";  
    version = "0.3.9";  
  
    src = pkgs.fetchFromGitHub {  
        owner = "muonsoft";  
        repo = pname;  
        rev = "v${version}";  
        hash = "sha256-7u//uwcVV1/EI6Rr3ju7K0wMYt/dXivyvBWIpTaoWZk=";  
    };  
  
    vendorHash = "sha256-KPCRunuCIbBX+YpHgshixmrXM3Ey0LIdeC0Z4CtpQoI=";  
}
```

Building Nix packages

```
$ nix repl  
Nix 2.28.4  
Type :? for help.
```

```
nix-repl> :l .  
Added 24244 variables.
```

```
nix-repl> :b phpactor
```

This derivation produced the following outputs:

```
out -> /nix/store/nhq4js2lsj8lb1cz6dj4vrf6xb2r2zbd-phpactor-2025.03.28.0
```

```
/nix/store/nhq4js2lsj8lb1cz6dj4vrf6xb2r2zbd-phpactor-2025.03.28.0/bin/phpactor
```

```
Phpactor 2025.03.28.0
```

NixOS in production?

```
services.phpfpm.enable = true;  
services.mysql.enable = true;  
services.nginx.enable = true;
```

NixOS in my homelab

```
services.jellyfin.enable = true;  
services.immich.enable = true;  
services.paperless.enable = true;
```

Not sure?

Want to try it, but you're not sure?

```
$ docker run --rm -it nixos/nix
```

```
$ podman run --rm -it nixos/nix
```

Thanks!

- <https://nixos.org>
- <https://github.com/nixos/nixpkgs>
- <https://wiki.nixos.org/wiki/PHP>
- <https://nixos.org/manual/nixpkgs/stable/#ssec-building-php-projects>
- <https://git.oliverdavies.uk/opdavies/lab>
- <https://git.oliverdavies.uk/opdavies/nix-config> (laptop, Neovim, home server)
- <https://books.oliverdavies.uk/nix-for-php-developers> (book, in progress)