

Streamline Your Development Workflow: Harnessing the power of NixOS and Nix Package Manager

By **Salar Rahamanian** (www.softinio.com)



About Salar Rahamanian

- ▶ Also known as **softinio**
- ▶ My Blog: www.softinio.com
- ▶ NixOS enthusiast and contributor since around 2018



Lifecycle of delivering code

Developers Laptop

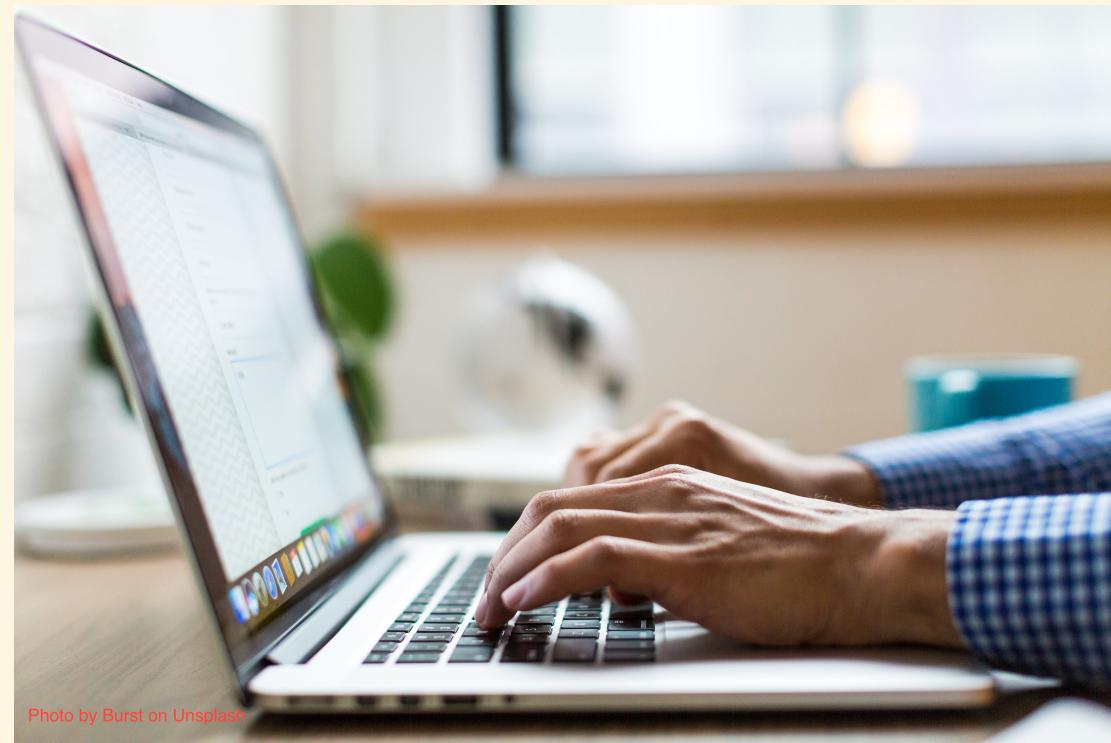
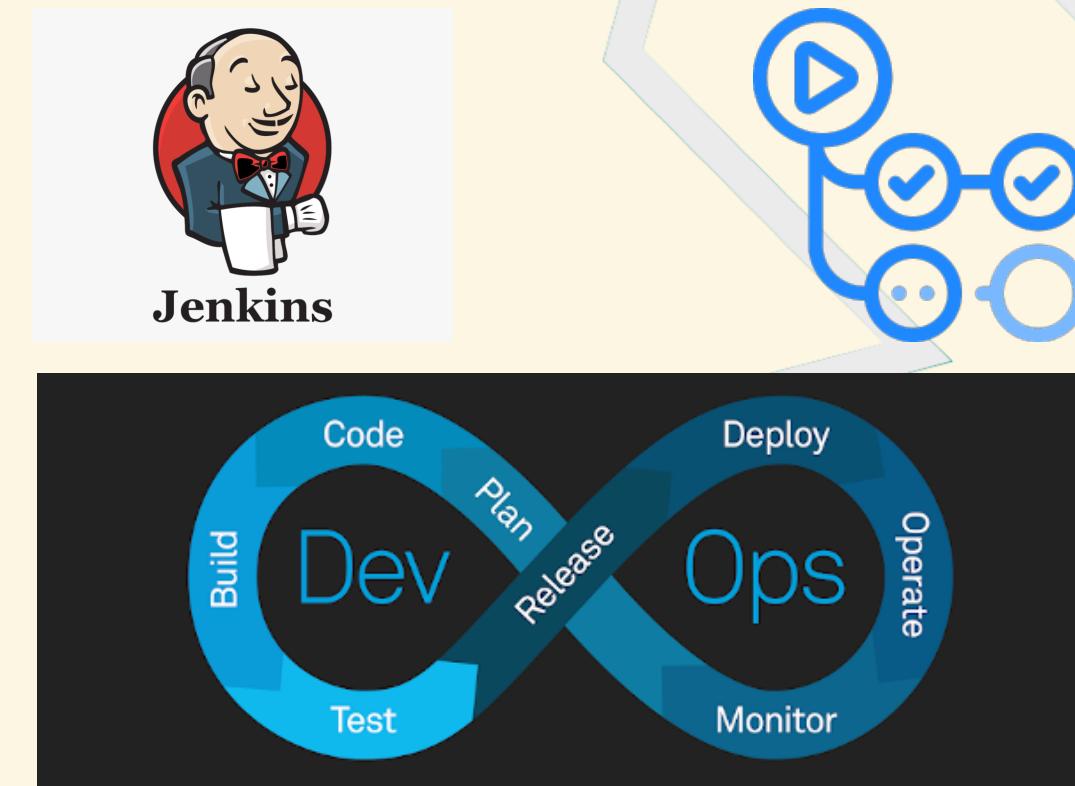
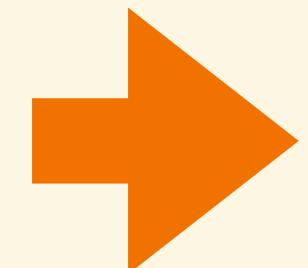


Photo by Burst on Unsplash



Cloud / Server

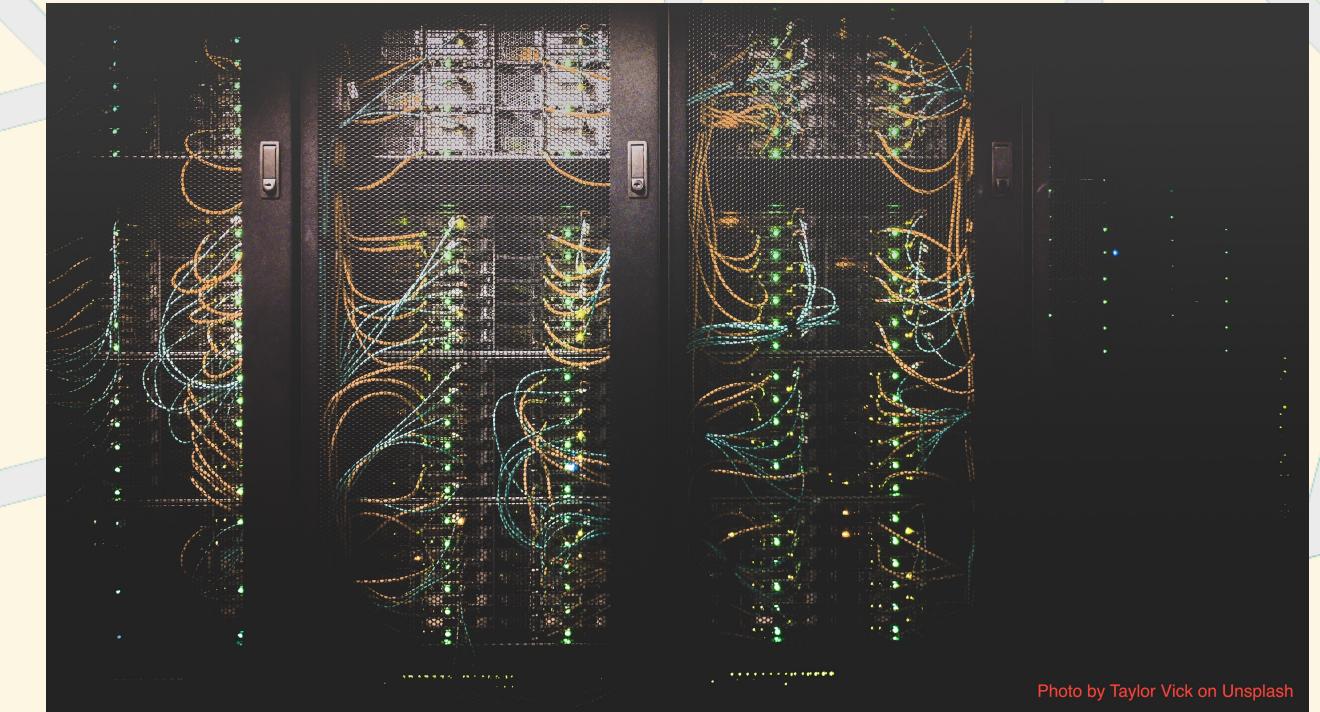


Photo by Taylor Vick on Unsplash

- ▶ Operating system: MacOS or a Linux variant
- ▶ Install OS developer tools
- ▶ Install packages using homebrew or download from various websites

- ▶ Install your programming language, its packages, tools and libraries (eg python/pip, Scala/sbt, Java/maven/gradle etc)
- ▶ Install your services like postgresql, Kafka , spark

Problem Statement

Variations between our various environments and tools have consequences

- ▶ Non-deterministic
- ▶ Not Reproducible
- ▶ Unreliable



Photo by Yogendra Singh on Unsplash

Lost Developer Time

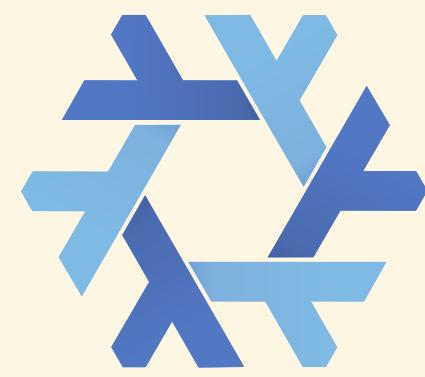
There is a cost: MONEY 💰 and reduced VELOCITY 🚀 to deliver

How is our time spent:

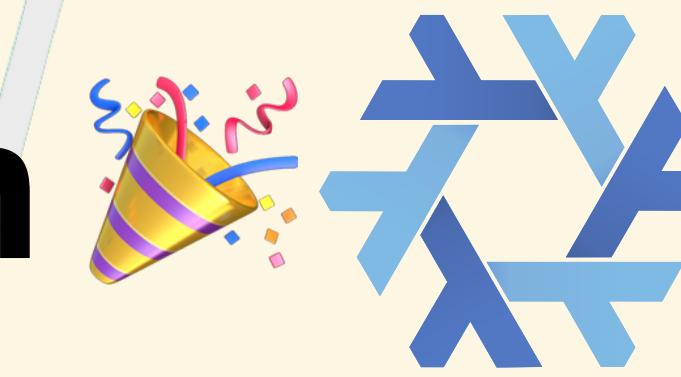
- ▶ Fixing issues
- ▶ Configuration and setup of our projects in the different environments
- ▶ Setting up services our project needs
- ▶ Testing our work in the different environments



Photo by Yogendra Singh on Unsplash



Nix is the solution





Nix is the solution



- ▶ Nix Package Manager
- ▶ Nix Expression Language
- ▶ NixOS

What is Nix?

Nix Package Manager



- ▶ It's a package manager
- ▶ It's Programming Language agnostic
- ▶ One tool, one way of building and testing,
everywhere 
-but many programming languages!

What is Nix?

Nix Expression Language

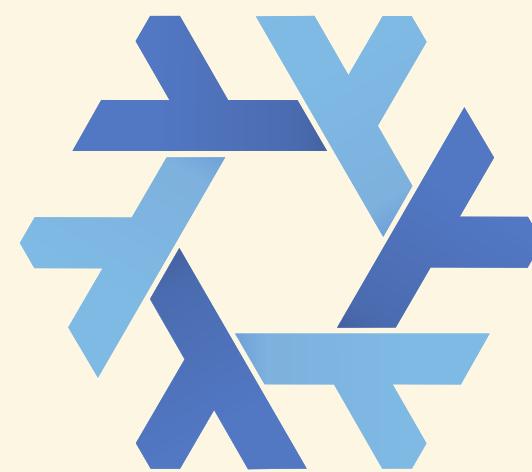


- ▶ It's a Functional Programming Language
- ▶ It's used to add packages to Nix Package manager
- ▶ You use it to tell Nix how to build, test & deploy
- ▶ You use it to configure and setup your environment and services

What is Nix?

NixOS

- It is its own Linux OS distribution
- It uses Nix package manager
- It uses Nix Expression Language to configure and set it up



NixOS

Nix Shell



I can't believe I don't have
python or **wget** installed!

talks/SBTB2023 on ⚡ master [!?]

λ

⋮



Nix Shell



Let's create a nix shell that has
python or **wget** installed!

talks/SBTB2023 on ⚡ master [!?]

λ



Nix Shell



- ➡ How you define a shell?
- ➡ Create a **shell.nix** and/or **default.nix**
- ➡ You can have as many shells as you want
- ➡ Name file: **<what ever I want>.nix**

```
talks/SBTB2023/1_nix_shell on ⚡ master [!?]
λ cat shell.nix
```

File: shell.nix

```
1 { pkgs ? import (fetchTarball "https://github.com/NixOS/nixpkgs/archive/06278c77b5d162e62df170fec307e83f1812d94b.tar.gz") {} }:
2
3 pkgs.mkShell {
4   buildInputs = [
5     ];
6
7   shellHook = ''
8   '';
9 }
10
11 }
```

Nix Shell



- How you define a shell?
- Purpose of **shell.nix** and **default.nix**
- We want some dependencies

```
talks/SBTB2023/1_nix_shell on ✘ master [!?]
λ cat shell_1.nix
```

File: shell_1.nix

```
1 { pkgs ? import (fetchTarball "https://github.com/NixOS/nixpkgs
2   /archive/06278c77b5d162e62df170fec307e83f1812d94b.tar.gz") {} }
3 :
4
5 pkgs.mkShell {
6   buildInputs = [
7     pkgs.wget
8     pkgs.ruff
9     (pkgs.python3.withPackages (ps: [
10       ps.pytest
11     ]))
12   ];
13   shellHook = ''
14     echo "hello"
15   '';
16 }
17 }
```

Nix Shell



We need postgresql for our project

```
talks/SBTB2023/1_nix_shell on ✘ master [!?]
λ cat shell_2.nix

File: shell_2.nix

1 { pkgs ? import (fetchTarball "https://github.com/NixOS/nixpkgs/archive/06278c77b5d162e62df170fec307e83f1812d94b.tar.gz") {} };

2 pkgs.mkShell {
3   buildInputs = [
4     pkgs.wget
5     pkgs.ruff
6     (pkgs.python3.withPackages (ps: [
7       ps.pytest
8     ]))
9     pkgs.postgresql
10   ];
11
12
13   shellHook = ''
14   export PGDATA=$PWD/.env/data/postgres_data
15   export PGHOST=$PWD/.env/data/postgres
16   export LOG_PATH=$PWD/.env/data/postgres/LOG
17   export PGDATABASE=postgres
18   export DATABASE_URL="postgresql:///postgres?host=$PGHOST"
19   if [ ! -d $PGHOST ]; then
20     mkdir -p $PGHOST
21   fi
22   if [ ! -d $PGDATA ]; then
23     echo 'Initializing postgresql database ... '
24     initdb $PGDATA --auth=trust >/dev/null
25   fi
26   pg_ctl start -l $LOG_PATH -o "-c listen_addresses= -c unix_socket_directories=$PGHOST"
27   trap "'pg_ctl' stop --silent -W" EXIT
28   '';
29
30 }
31 }
```

```
talks/SBTB2023/1_nix_shell on ✘ master [!?]
```

```
λ cat shell_2.nix
```

```
File: shell_2.nix

1 { pkgs ? import (fetchTarball "https://github.com/NixOS/nixpkgs/archive/06278c77b5d162e62df170fec307e83f1812d94b.tar.gz") {} }:

2 pkgs.mkShell {
3     buildInputs = [
4         pkgs.wget
5         pkgs.ruff
6         (pkgs.python3.withPackages (ps: [
7             ps.pytest
8         ]))
9         pkgs.postgresql
10    ];
11
12    shellHook = ''
13        export PGDATA=$PWD/.env/data/postgres_data
14        export PGHOST=$PWD/.env/data/postgres
15        export LOG_PATH=$PWD/.env/data/postgres/LOG
16        export PGDATABASE=postgres
17        export DATABASE_URL="postgresql://postgres?host=$PGHOST"
18        if [ ! -d $PGHOST ]; then
19            mkdir -p $PGHOST
20        fi
21        if [ ! -d $PGDATA ]; then
22            echo 'Initializing postgresql database ... '
23            initdb $PGDATA --auth=trust >/dev/null
24        fi
25        pg_ctl start -l $LOG_PATH -o "-c listen_addresses= -c unix_socket_directories=$PGHOST"
26        trap "'pg_ctl' stop --silent -W" EXIT
27        '';
28    '';
29
30 }
31
```

```
talks/SBTB2023/1_nix_shell on ✘ master [!?]
```

```
λ └
```

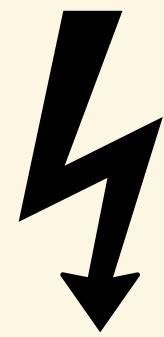
```
talks/SBTB2023/1_nix_shell on ✘ master [!?]
```

```
λ └
```

What is DEVENV?

- Define your development environment declaratively
- Define by configuration so less verbose and complex
- Let's you define common tools you need
- Manage your scripts
- Reproducible
- You can include it all in your git repository 

DEVENV



Initialize your project

```
$ devenv init
```

talks/SBTB2023 on ♫ master [!?]

λ



Initialize your project

```
.../SBTB2023/SBTB2023_project_1 on ✘ master [!?] via * impure (devenv-shell)
λ l
Permissions Size User Date Modified Git Name
drwxr-xr-x    - salar 11 Nov 09:35 -I .devenv
drwxr-xr-x    - salar 11 Nov 09:35 -I .direnv
.rw-r--r-- 3.4k salar 11 Nov 09:35 -I * .devenv.flake.nix
.rw-r--r-- 176 salar 11 Nov 09:35 -I .envrc
.rw-r--r-- 93 salar 11 Nov 09:35 -N git .gitignore
.rw-r--r-- 4.1k salar 11 Nov 09:35 NM 🔒 devenv.lock
.rw-r--r-- 567 salar 11 Nov 09:35 -N * devenv.nix
.rw-r--r-- 66 salar 11 Nov 09:35 -N ? devenv.yaml
```

Import nix package repositories you need

```
.../SBTB2023/SBTB2023_project_1 on ✘ master [!?] via * impure (devenv-shell)
λ cat devenv.yaml
```

	File: devenv.yaml
1	inputs:
2	nixpkgs:
3	url: github:NixOS/nixpkgs/nixpkgs-unstable

```
.../SBTB2023/SBTB2023_project_1 on ✘ master [!?] via * impure (devenv-shell)
λ cat devenv.nix
```

	File: devenv.nix
1	{ pkgs, ... }:
2	{
3	# https://devenv.sh/basics/ env.GREET = "devenv";
4	# https://devenv.sh/packages/ packages = [pkgs.git];
5	# https://devenv.sh/scripts/ scripts.hello.exec = "echo hello from \$GREET";
6	enterShell = '' hello git --version '';
7	# https://devenv.sh/languages/ # languages.nix.enable = true;
8	# https://devenv.sh/pre-commit-hooks/ # pre-commit.hooks.shellcheck.enable = true;
9	# https://devenv.sh/processes/ # processes.ping.exec = "ping example.com";
10	# See full reference at https://devenv.sh/reference/options/ }

DEVENV

Import nix package repositories you need

```
.../SBTB2023/SBTB2023_project_1 on ✘ master [!?] via * impure (devenv-shell)
λ cat devenv.yaml
```

File: devenv.yaml

```
1 inputs:
2   nixpkgs:
3     url: github:NixOS/nixpkgs/nixpkgs-unstable
```

Import nix python package repositories

```
λ cat devenv.yaml
```

File: devenv.yaml

```
1 inputs:
2   nixpkgs:
3     url: github:NixOS/nixpkgs/nixpkgs-unstable
4   nixpkgs-python:
5     url: github:cachix/nixpkgs-python
6
```

DEVENV

```
.../SBTB2023/SBTB2023_project_1 on ✘ master [!?] via * impure (devenv-shell)
λ cat devenv.nix

File: devenv.nix

1 { pkgs, ... }:
2
3 {
4     # https://devenv.sh/basics/
5     env.GREET = "devenv";
6
7     # https://devenv.sh/packages/
8     packages = [ pkgs.git ];
9
10    # https://devenv.sh/scripts/
11    scripts.hello.exec = "echo hello from $GREET";
12
13    enterShell = ''
14        hello
15        git --version
16    '';
17
18    # https://devenv.sh/languages/
19    # languages.nix.enable = true;
20
21    # https://devenv.sh/pre-commit-hooks/
22    # pre-commit.hooks.shellcheck.enable = true;
23
24    # https://devenv.sh/processes/
25    # processes.ping.exec = "ping example.com";
26
27    # See full reference at https://devenv.sh/reference/options/
28 }
```

```
λ cat devenv.nix

File: devenv.nix

1 { pkgs, ... }:
2
3 {
4     # https://devenv.sh/basics/
5     env.GREET = "Scale By the Bay";
6
7     # https://devenv.sh/packages/
8     packages = [ pkgs.git ];
9
10    # https://devenv.sh/scripts/
11    scripts.hello.exec = "echo hello from $GREET";
12
13    enterShell = ''
14        hello
15        git --version
16    '';
17
18    languages.python = {
19        enable = true;
20        version = "3.11.3";
21    };
22
23    services.postgres = {
24        enable = true;
25        initialDatabases = [{ name = "sbtbdb"; }];
26    };
27
28    pre-commit.hooks.black.enable = true;
29
30    # https://devenv.sh/languages/
31    # languages.nix.enable = true;
32
33    # https://devenv.sh/pre-commit-hooks/
34    # pre-commit.hooks.shellcheck.enable = true;
35
36    # https://devenv.sh/processes/
37    # processes.ping.exec = "ping example.com";
38
39    # See full reference at https://devenv.sh/reference/options/
40 }
```



DEVENV

```
λ cat devenv.nix
```

```
File: devenv.nix

1 { pkgs, ... }:
2
3 {
4     # https://devenv.sh/basics/
5     env.GREET = "Scale By the Bay";
6
7     # https://devenv.sh/packages/
8     packages = [ pkgs.git ];
9
10    # https://devenv.sh/scripts/
11    scripts.hello.exec = "echo hello from $GREET";
12
13    enterShell = ''
14        hello
15        git --version
16    '';
17
18    languages.python = {
19        enable = true;
20        version = "3.11.3";
21    };
22
23    services.postgres = {
24        enable = true;
25        initialDatabases = [{ name = "sbtbdb"; }];
26    };
27
28    pre-commit.hooks.black.enable = true;
29
30    # https://devenv.sh/languages/
31    # languages.nix.enable = true;
32
33    # https://devenv.sh/pre-commit-hooks/
34    # pre-commit.hooks.shellcheck.enable = true;
35
36    # https://devenv.sh/processes/
37    # processes.ping.exec = "ping example.com";
38
39    # See full reference at https://devenv.sh/reference/options/
40 }
```

```
talks/SBTB2023 on λ master [!?]
```

```
λ
```

What is Nix Flakes

- Package manager for your project
- Conventions on how to build, run and deploy your project
- Cross platform
- Has a distinct schema
- It's not yaml 😂😂😂

Nix Flakes

Flakes Schema Overview

1. Input: Tell flakes what repositories to include
2. Output: List of systems to produce output for
 - a. Linux - x86
 - b. Linux - Arm
 - c. MacOS - x86
 - d. MacOS - Arm/M1
3. Build Package
 - a. Specify python version
 - b. Package name
 - c. Python dependencies
 - d. Source root
4. Build Application
5. Build Development Shell
 - a. Developer shell environment like nix-shell / devenv
 - b. List dependencies you need during development
 - c. shellHook - bash commands run on shell startup
 - d. Can use in conjunction with devenv

```
1  description = "Python example flake";
2
3  inputs = {
4      nixpkgs.url = "github:NixOS/nixpkgs?ref=nixos-unstable";
5  };
6
7
8  outputs = { self, nixpkgs }:
9      let
10         allSystems = [ "x86_64-linux" "aarch64-linux" "x86_64-darwin" "aarch64-darwin" ];
11
12         forAllSystems = nixpkgs.lib.genAttrs allSystems;
13
14         nixpkgsFor = forAllSystems (system: import nixpkgs { inherit system; });
15
16
17         in
18     {
19         # Create your python package
20         packages = forAllSystems (system:
21             let
22                 pkgs = nixpkgsFor.${system};
23                 in
24                 {
25                     default =
26                         let
27                             python = pkgs.python39;
28                             in
29                             python.pkgs.buildPythonApplication {
30                                 name = "sbtb";
31
32                                 buildInputs = with python.pkgs; [ pip ];
33
34                                 src = ./.;
35                             };
36
37                     # Use the package you created to create your python app
38                     apps = forAllSystems (system: {
39                         default =
40                             {
41                                 program = "${self.packages.${system}.default}/bin/sbtb";
42                                 type = "app";
43                             };
44                     });
45
46                     # Your development environment
47                     devShells = forAllSystems (system:
48                         let
49                             pkgs = nixpkgsFor.${system};
50                             in
51                             {
52                                 default = pkgs.mkShell {
53                                     buildInputs = with pkgs; [ ruff wget ];
54                                     shellHook =
55                                         ''
56                                         echo "You have entered the wonderful world of nix"
57                                         '';
58                                 };
59                             };
60                         );
61                     );
62                 };
63             );
64         );
65     );
66 }
```

3_flake_only on ✘ main [x!+] via 🐍 v3.9.6

λ

Nix Templates



Nix Templates

[NixOS / templates](#) Public

Code Issues 18 Pull requests 11 Actions Projects Security Insights

master 2 branches 0 tags Go to file Code

Author	Commit Message	Date
domenkozar	Merge pull request #63 from peterbecich/haskell-template-with-ha...	98bc26d last month 87 commits
	.github Initial work on README.md	2 years ago
	bash-hello attribute self.lastModifiedDate missing in newer versions of flakes	2 years ago
	c-hello Use pname+version, don't overwrite unpackPhase (#35)	9 months ago
	compat Add a template that provides default.nix and shell.nix	2 years ago
	full full: Mark nixosModule as deprecated (#70)	3 months ago
	go-hello Add example for devShells in go-hello (#60)	9 months ago
	haskell-flake remove test.sh , unnecessary	2 months ago
	haskell-hello Add support for x86_64-darwin	last year
	haskell.nix update haskell.nix Flake template	9 months ago
	hercules-ci templates/hercules-ci: init	2 years ago
	pandoc-xelatex better formatting for abstract	last year
	python python: refactor and cleanup	last year
	rust-web-server self.lastModifiedDate fix for all templates	last year
	rust Fix wrong defaultApp in Rust template	last year
	simple-container Doh	3 years ago
	trivial Update trivial template with nix 2.7 default package syntax	last year
	gitignore update haskell.nix flake template (#57)	9 months ago

About

Flake templates

- Readme
- MIT license
- Code of conduct
- Security policy
- Activity
- 289 stars
- 11 watching
- 67 forks

Report repository

Releases

No releases published

Sponsor this project

 opencollective.com/nixos

Packages

No packages published

Nix Templates

To use a template from nix own github: `nix flake init --template templates#python`

```
talks/SBTB2023 on ⚡ master [!?]
λ █
```



Nix Templates

[NixOS / templates](#) Public

Code Issues 18 Pull requests 11 Actions Projects Security Insights

master 2 branches 0 tags Go to file Code

Author	Commit Message	Date
domenkozar	Merge pull request #63 from peterbecich/haskell-template-with-ha...	98bc26d last month 87 commits
	.github Initial work on README.md	2 years ago
	bash-hello attribute self.lastModifiedDate missing in newer versions of flakes	2 years ago
	c-hello Use pname+version, don't overwrite unpackPhase (#35)	9 months ago
	compat Add a template that provides default.nix and shell.nix	2 years ago
	full full: Mark nixosModule as deprecated (#70)	3 months ago
	go-hello Add example for devShells in go-hello (#60)	9 months ago
	haskell-flake remove test.sh , unnecessary	2 months ago
	haskell-hello Add support for x86_64-darwin	last year
	haskell.nix update haskell.nix Flake template	9 months ago
	hercules-ci templates/hercules-ci: init	2 years ago
	pandoc-xelatex better formatting for abstract	last year
	python python: refactor and cleanup	last year
	rust-web-server self.lastModifiedDate fix for all templates	last year
	rust Fix wrong defaultApp in Rust template	last year
	simple-container Doh	3 years ago
	trivial Update trivial template with nix 2.7 default package syntax	last year
	gitignore update haskell.nix flake template (#57)	9 months ago

About

Flake templates

- Readme
- MIT license
- Code of conduct
- Security policy
- Activity
- 289 stars
- 11 watching
- 67 forks

Report repository

Releases

No releases published

Sponsor this project

 opencollective.com/nixos

Packages

No packages published

Nix Templates

To use a template from a github repo: `nix flake init --template github:the-nix-way/dev-templates#go`

The screenshot shows the GitHub repository page for 'the-nix-way/dev-templates'. The repository is public and has 38 forks and 384 stars. The 'Code' tab is selected, showing the main branch commit history. The commits are as follows:

Author	Commit Message	Date	Time	
lucperkins	Minor change to workflow YAML	on Aug 22	163	
	.github	Minor change to workflow YAML	3 months ago	
	clojure	Revamp all templates to remove flake-u...	4 months ago	
	csharp	Update README with new versions	4 months ago	
	cue	Update README with new versions	4 months ago	
	dhall	Update README with new versions	4 months ago	
	elixir	Revamp all templates to remove flake-u...	4 months ago	
	elm	Revamp all templates to remove flake-u...	4 months ago	
	gleam	Revamp all templates to remove flake-u...	4 months ago	
	go	Revamp all templates to remove flake-u...	4 months ago	
	hsobi	Revamp all templates to remove flake-u...	4 months ago	

About

Dev environments for numerous languages based on Nix flakes [maintainer=@lucperkins]

Tags

- nix
- reproducibility
- flakes

Repository Details

- Readme
- Activity
- 384 stars
- 7 watching
- 38 forks
- Report repository

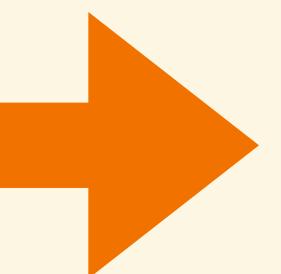
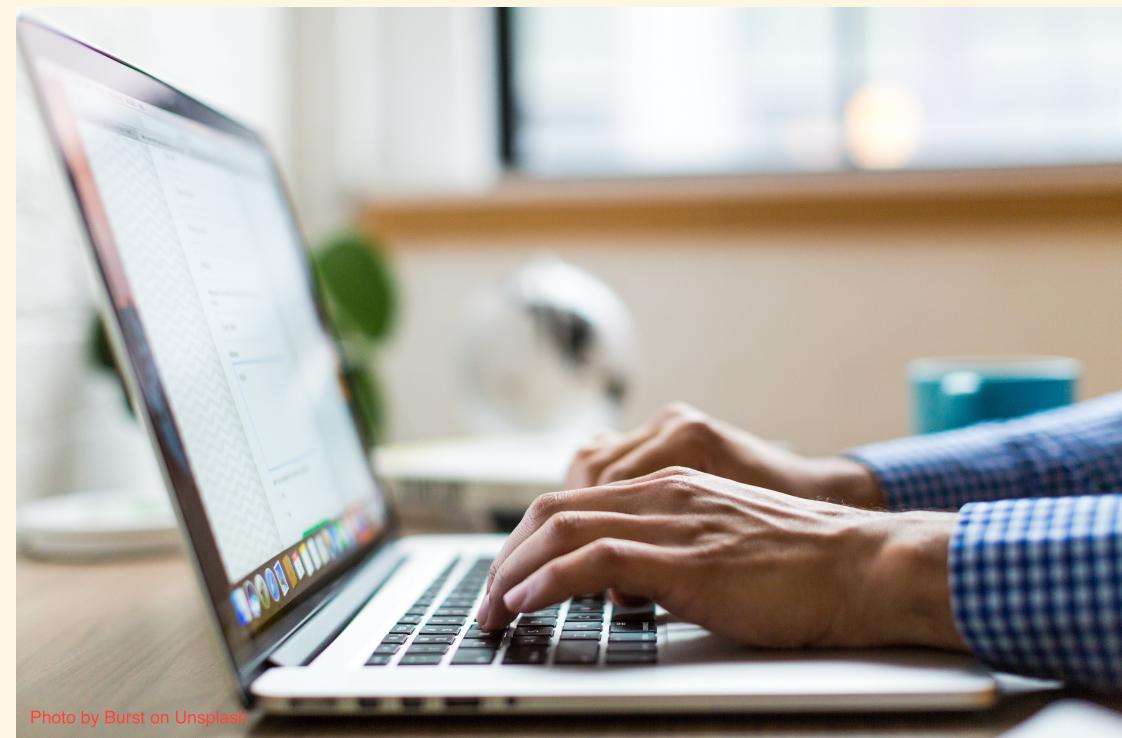
Releases

- 1 tags

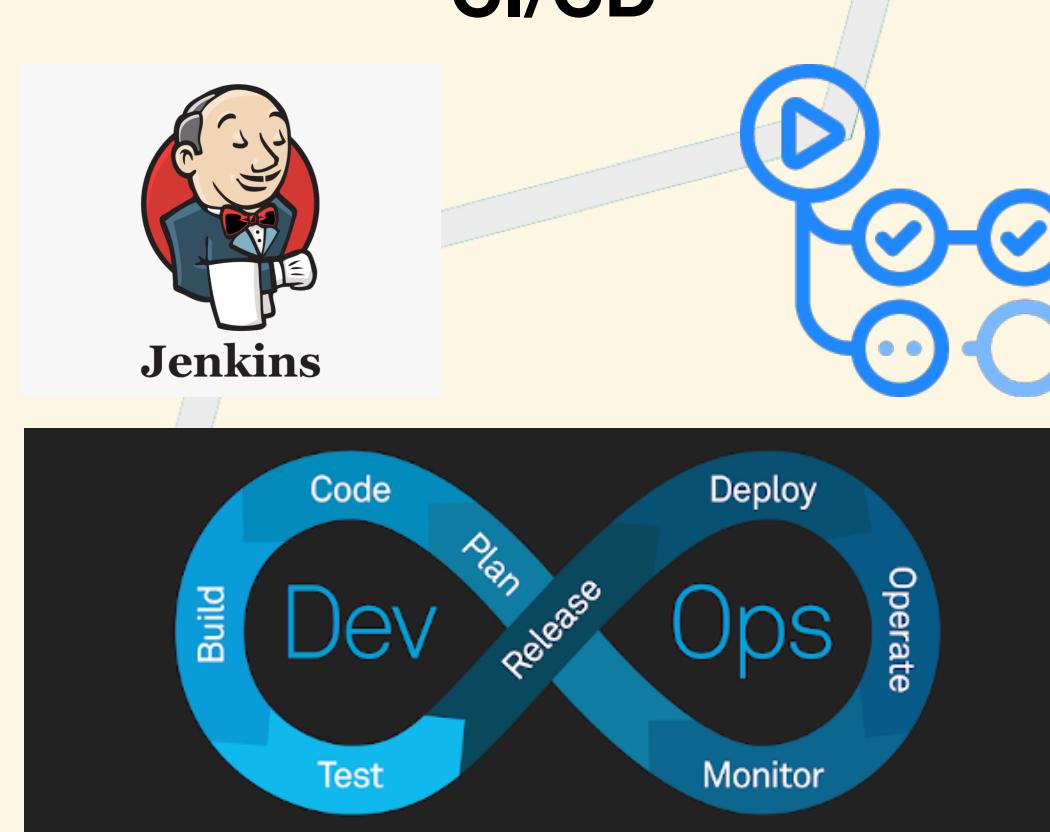
🔥 Solved by Nix Flakes 🚀

Lifecycle of delivering code

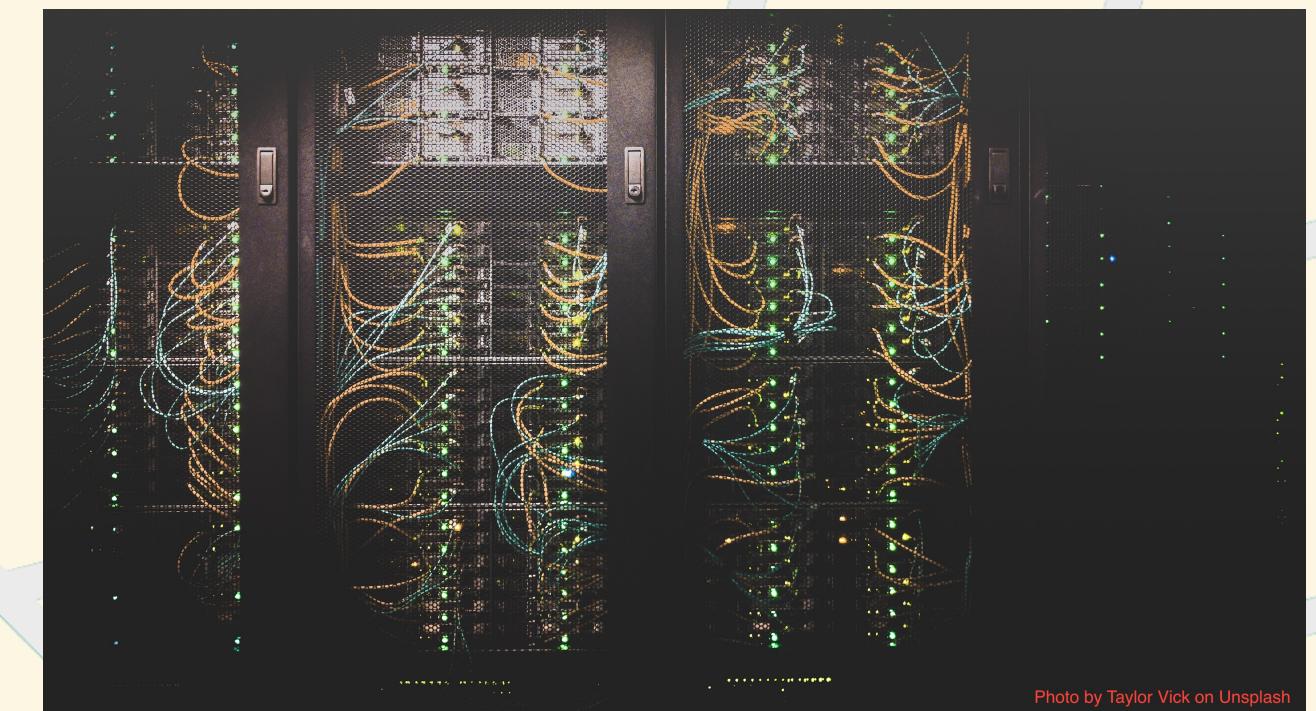
Developers Laptop



CI/CD



Cloud / Server



Resources

- ▶ <https://nixos.org/>
- ▶ <https://zero-to-nix.com/>
- ▶ <https://devenv.sh/>
- ▶ <https://www.youtube.com/c/NixCon>
- ▶ <https://scrive.github.io/nix-workshop/index.html>
- ▶ <https://www.tweag.io/blog/2020-05-25-flakes/>
- ▶ <https://watch.softinio.com>
- ▶ <https://www.softinio.com>

🙏 **Thank You for attending my talk** 🙏

Follow my work and keep in touch via ➡ www.softinio.com ➡