

# Using Nix to Create Reproducible Go Development Environments

# Introduction

- Haseeb Majid
  - Backend Software Engineer at Curve
  - <https://haseebmajid.dev>
- Loves cats 
- Avid cricketer  #BazBall

# Who is this for?

- Interested in Nix
- Consistent development environments
  - Old project; still works
  - New developers
  - “It works on my machine”

# SAY IT AGAIN PLEASE



## "IT WORKS ON MY MACHINE"

imgflip.com

Credit



# What is Nix?

- Nix is a declarative package manager
- Nixlang the programming language that powers Nix
- Nixpkgs is the largest repository of Nix packages

# Declarative

```
{  
    wayland.windowManager.sway.enable = true;  
    xsession.windowManager.i3.enable = false;  
}
```





# Summary

- We want reproducible and ephemeral environments
- Nix is an ecosystem of tools



# Golang

- Tools to aid development
  - lint
  - testing
  - docker

# tools.go

```
1 // +build tools
2
3 package main
4
5 import (
6 _ "github.com/golangci/golangci-lint/cmd/golangci-lint"
7 _ "github.com/goreleaser/goreleaser"
8 )
```

# tools.go

```
1 // +build tools
2
3 package main
4
5 import (
6 _ "github.com/golangci/golangci-lint/cmd/golangci-lint"
7 _ "github.com/goreleaser/goreleaser"
8 )
```

```
go run github.com/....cmd/golangci-lint
```

```
1 example on ↵ main via 🐼 v1.22.8
2 > ls -al
3 .rw-r--r-- 101 haseebmajid 28 Mar 15:36 go.mod
4 .rw-r--r-- 191 haseebmajid 28 Mar 15:37 go.sum
5 .rw-r--r-- 313 haseebmajid 28 Mar 15:33 main.go
6 .rw-r--r--    0 haseebmajid 28 Mar 14:55 main_test.go
7
8 example on ↵ main via 🐼 v1.22.8
9 > nvim flake.nix
```

```
1 example on ↵ main via 🐾 v1.22.8
2 > ls -al
3 .rw-r--r-- 101 haseebmajid 28 Mar 15:36 go.mod
4 .rw-r--r-- 191 haseebmajid 28 Mar 15:37 go.sum
5 .rw-r--r-- 313 haseebmajid 28 Mar 15:33 main.go
6 .rw-r--r--    0 haseebmajid 28 Mar 14:55 main_test.go
7
8 example on ↵ main via 🐾 v1.22.8
9 > nvim flake.nix
```

```
1 {
2     description = "I use Nix btw!";
3
4     inputs = {
5         nixpkgs.url = "github:NixOS/nixpkgs";
6         flake-utils.url = "github:numtide/flake-utils";
7     };
8
9     outputs = {
10        self,
11        nixpkgs,
12        flake-utils,
13        ...
14 }
```

```
5      nixpkgs.url = "github:NIXOS/nixpkgs";
6      flake-utils.url = "github:numtide/flake-utils";
7  };
8
9  outputs = {
10    self,
11    nixpkgs,
12    flake-utils,
13    ...
14 }: (
15   flake-utils.lib.eachDefaultSystem
16   (system: let
17     # system is like "x86_64-linux" or "aarch64-linux"
18     nixpkgs = nixpkgs_letOverPackages $system;
```

```
9   outputs = {
10    self,
11    nixpkgs,
12    flake-utils,
13    ...
14  }: (
15    flake-utils.lib.eachDefaultSystem
16    (system: let
17      # system is like "x86_64-linux" or "aarch64-linux"
18      pkgs = nixpkgs.legacyPackages.${system};
19    in {
20      devShells.default = pkgs.mkShell {
21        packages = with pkgs; [
```

```
12     flake-utils,
13     ...
14 }: ((
15     flake-utils.lib.eachDefaultSystem
16     (system: let
17         # system is like "x86_64-linux" or "aarch64-linux"
18         pkgs = nixpkgs.legacyPackages.${system};
19     in {
20         devShells.default = pkgs.mkShell {
21             packages = with pkgs; [
22                 go_1_22
23                 golangci-lint
24                 gotools
```

```
14  }: (
15    flake-utils.lib.eachDefaultSystem
16    (system: let
17      # system is like "x86_64-linux" or "aarch64-linux"
18      pkgs = nixpkgs.legacyPackages.${system};
19    in {
20      devShells.default = pkgs.mkShell {
21        packages = with pkgs; [
22          go_1_22
23          golangci-lint
24          gotoools
25          go-junit-report
26          gocover-cobertura
```

```
15 flake-utils.lib.eachDefaultSystem
16 (system: let
17     # system is like "x86_64-linux" or "aarch64-linu
18     pkgs = nixpkgs.legacyPackages.${system};
19   in {
20     devShells.default = pkgs.mkShell {
21       packages = with pkgs; [
22         go_1_22
23         golangci-lint
24         gotools
25         go-junit-report
26         gocover-cobertura
27         go-task
```

```
19     in {
20         devShells.default = pkgs.mkShell {
21             packages = with pkgs; [
22                 go_1_22
23                 golangci-lint
24                 gotools
25                 go-junit-report
26                 gocover-cobertura
27                 go-task
28                 goreleaser
29                 sqlc
30             ];
31         };
32     }
```

```
1 example on ↗ main via 🐼 v1.22.8
2 > which golangci-lint
3
4 example on ↗ main via 🐼 v1.22.8
5 > nix develop
6
7 example on ↗ main via 🐼 v1.22.8 * impure (nix-shell-environment)
8 > which golangci-lint
9 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```

```
1 example on ↗ main via 🐼 v1.22.8
2 > which golangci-lint
3
4 example on ↗ main via 🐼 v1.22.8
5 > nix develop
6
7 example on ↗ main via 🐼 v1.22.8 * impure (nix-shell-environment)
8 > which golangci-lint
9 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```

```
1 example on ↗ main via 🐼 v1.22.8
2 > which golangci-lint
3
4 example on ↗ main via 🐼 v1.22.8
5 > nix develop
6
7 example on ↗ main via 🐼 v1.22.8 * impure (nix-shell-environment)
8 > which golangci-lint
9 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```

```
1 nix run 'gitlab:hmajid2301/optinix' get --no-tui
2
3 # Output
4
5 Option: 0
6 Name: services.prometheus.scrapeConfigs.*.docker_sd_co
7 Type: null or string
8 Default: null
9 Example:
10 From: NixOS
11 Sources: [/nix/store/xr9wjzx0cdnwkmhzz74h8lphgn5jmyv3-
12
```

```
1 nix run 'gitlab:hmajid2301/optinix' get --no-tui
2
3 # Output
4
5 Option: 0
6 Name: services.prometheus.scrapeConfigs.*.docker_sd_co
7 Type: null or string
8 Default: null
9 Example:
10 From: NixOS
11 Sources: [/nix/store/xr9wjzx0cdnwkmhzz74h8lphgn5jmyv3-
12
```

# Summary

- Leverage dev shells for installing packages
  - `nix develop`
- Make sure each dev gets the same package
  - `nix flake update`

# direnv

```
# .envrc  
  
use flake
```

Direnv Code “use flake”

# Usage

```
1 > which golangci-lint
2
3 > cd banterbus
4 direnv: error /home/haseeb/banterbus/.envrc
5 is blocked. Run `direnv allow` to approve
6 its content
7
8 banterbus on ↗ main via 🐾 v1.22.7
9 > direnv allow
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
```

# Usage

```
1 > which golangci-lint
2
3 > cd banterbus
4 direnv: error /home/haseeb/banterbus/.envrc
5 is blocked. Run `direnv allow` to approve
6 its content
7
8 banterbus on ↗ main via 🐾 v1.22.7
9 > direnv allow
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
```

# Usage

```
1 > which golangci-lint
2
3 > cd banterbus
4 direnv: error /home/haseeb/banterbus/.envrc
5 is blocked. Run `direnv allow` to approve
6 its content
7
8 banterbus on ↗ main via 🐾 v1.22.7
9 > direnv allow
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
```

# Usage

```
3 > cd banterbus
4 direnv: error /home/haseeb/banterbus/.envrc
5 is blocked. Run `direnv allow` to approve
6 its content
7
8 banterbus on ↗ main via 🐾 v1.22.7
9 > direnv allow
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
14
15 banterbus on ↗ main via 🐾 v1.22.8 via * impure (nix-)
```

# Usage

```
5 is blocked. Run direnv allow to approve
6 its content
7
8 banterbus on ↗ main via 🐕 v1.22.7
9 > direnv allow
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
14
15 banterbus on ↗ main via 🐕 v1.22.8 via * impure (nix-)
16 > which golangci-lint
17 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
18
```

# Usage

```
10 direnv: loading ~/banterbus/.envrc
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
14
15 banterbus on ↗ main via 🐀 v1.22.8 via * impure (nix-)
16 > which golangci-lint
17 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lin
18
19 banterbus on ↗ main via 🐀 v1.22.8 via * impure (nix-)
20 > cd ..
21 direnv: unloading
22
23 > which golangci-lint
```

# Usage

```
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
14
15 banterbus on ↗ main via 🐾 v1.22.8 via * impure (nix-)
16 > which golangci-lint
17 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
18
19 banterbus on ↗ main via 🐾 v1.22.8 via * impure (nix-)
20 > cd ..
21 direnv: unloading
22
23 > which golangci-lint
```

# Usage

```
11 direnv: using flake
12 direnv: nix-direnv: Renewed cache
13 ...
14
15 banterbus on ↗ main via 🐾 v1.22.8 via * impure (nix-)
16 > which golangci-lint
17 /nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
18
19 banterbus on ↗ main via 🐾 v1.22.8 via * impure (nix-)
20 > cd ..
21 direnv: unloading
22
23 > which golangci-lint
```

# Remote Environments

```
# .envrc  
  
use flake "github:the-nix-way/dev-templates?dir=go"
```

# pre-commit

```
1 {
2   description = "Development environment for example pro
3
4   inputs = {
5     nixpkgs.url = "github:NixOS/nixpkgs";
6     flake-utils.url = "github:numtide/flake-utils";
7     pre-commit.url = "github:cachix/pre-commit-hooks.nix";
8   };
9
10  outputs = {
11    self,
12    nixpkgs,
13    flake-utils,
```

# pre-commit

```
16 to : {
17   flake-utils.lib.eachDefaultSystem
18   (system: let
19     pkgs = nixpkgs.legacyPackages.${system};
20     pre-commit-check = pre-commit.lib.${system}.run {
21       src = ./.;
22       hooks = {
23         golangci-lint.enable = true;
24         gotest.enable = true;
25       };
26     };
27   in {
28     devShells.default = pkgs.mkShell {
29       shellHook = pre-commit-check shellHook;
```

# pre-commit

```
23     golangci-lint.enable = true;
24     gotest.enable = true;
25   };
26 };
27 in {
28   devShells.default = pkgs.mkShell {
29     shellHook = pre-commit-check.shellHook;
30     packages = with pkgs; [
31       go_1_22
32       golangci-lint
33       gotoools
34       go-junit-report
35       gocover-cobertura
```

# Summary

- We can use direnv to reduce cognitive load
- Use dev shell from remote git repository
- We can manage pre-commit in Nix as well

# How does Nix work?

```
1 # shell.nix
2
3 {pkgs, ...}:
4 pkgs.mkShell {
5   packages = with pkgs; [
6     go_1_22
7     golangci-lint
8   ];
9 }
```

[mkShell docs](#)

# How does Nix work?

```
1 # shell.nix
2
3 {pkgs, ...}:
4 pkgs.mkShell {
5   packages = with pkgs; [
6     go_1_22
7     golangci-lint
8   ];
9 }
```

[mkShell docs](#)

# How does Nix work?

```
1 # shell.nix
2
3 {pkgs, ...}:
4 pkgs.mkShell {
5   packages = with pkgs; [
6     go_1_22
7     golangci-lint
8   ];
9 }
```

[mkShell docs](#)

```
# flake.nix

{
  devShells.default = pkgs.callPackage ./shell.nix {
    inherit pkgs;
  };
}
```



Back to nixos.org

Packages

NixOS options

Flakes

Experimental

# Q Search more than 100 000 packages

go

Search

Channel: 24.05 unstable

## Package sets

No package set	1132
haskellPackages	417
python312Packages	355
python311Packages	355
rPackages	258
emacsPackages	119
texlivePackages	58
sbc1Packages	44
gnomeExtensions	28
vimPlugins	26
perl536Packages	24
perl538Packages	23
home-assistant-component-tests	18
ocamlPackages	12
vscode-extensions	9
libsForQt5	9
kdePackages	9
jetbrains	9
flutterPackages-source	6
nodePackages	5

## Showing results 1-50 of 2989 packages.

Sort: Best match ▾

Data from nixpkgs [bc947f54](#).**go**

Go Programming language

Name: [go](#) Version: 1.22.7 [Homepage](#) [Source](#) License: BSD 3-clause "New" or "Revised" License**gox**

Dead simple, no frills Go cross compile tool

Name: [gox](#) Version: 1.0.1 [Homepage](#) [Source](#) License: Mozilla Public License 2.0**got**

Version control system which prioritizes ease of use and simplicity over flexibility

Name: [got](#) Version: 0.103 [Homepage](#) [Source](#) License: ISC License**gom**

GObject to SQLite object mapper

Name: [gom](#) Version: 0.4 Outputs: [out](#) [py](#) [Homepage](#) [Source](#) License: GNU Lesser General Public License v2.1 or later**gol**

Command-line utility for creating and managing Geographic Object Libraries

Name: [gol](#) Version: 0.2.0 [Homepage](#) [Source](#) License: GNU Affero General Public License v3.0 only**goa**

Design-based APIs and microservices in Go

Name: [goa](#) Version: 3.19.1 [Homepage](#) [Source](#) License: MIT License

## Licenses

MIT License

719

[search.nixos.org](https://search.nixos.org)

```
1 { lib
2 , stdenv
3 , fetchurl
4 , tzdata
5 , substituteAll
6 , iana-etc
7 , Security
8 , Foundation
9 , xcbuild
10 , mailcap
11 , buildPackages
12 , pkgsBuildTarget
13 , threadsCross
```

## Go Nix Expression

```
46  isCross = stdenv.buildPlatform != stdenv.targetPlatform
47  in
48  stdenv.mkDerivation (finalAttrs: {
49      pname = "go";
50      version = "1.22.7";
51
52      src = fetchurl {
53          url = "https://go.dev/dl/go....";
54          hash = "sha256-ZkMth9h...";
55      };
56
57      strictDeps = true;
58      buildInputs = [ ]
```

## Go Nix Expression

# Evaluation

- Evaluation Time: Nix Expression (.nix) is parsed and returns a derivation set .drv
- Build Time: The derivation is built into a package

# Derivations

```
1 /nix/store/<hash>-<name>-<version>.drv
2 /nix/store/zg65r8ys8y5865lcwmmybrq5g...-go-1.21.8.drv
3 /nix/store/z45pk6pw3h4yx0cpipi51fc5nwm...-go-1.22.1.drv
```

# Derivations

```
1 /nix/store/<hash>-<name>-<version>.drv
2 /nix/store/zg65r8ys8y5865lcwmmybrq5g...-go-1.21.8.drv
3 /nix/store/z45pk6pw3h4yx0cpipi51fc5nwm...-go-1.22.1.drv
```

```
1 nix derivation show nixpkgs#go_1_21
2 {
3   "/nix/store/gccilxhvxbhm79....-go-1.21.8.drv": {
4     "args": [
5       "-e",
6       "/nix/store/v6x3cs394jgqfbioa42pam708flxaphh-default",
7     ],
8     "builder": "/nix/store/5lr5n...-bash-5.2p26/bin/bash"
9     "env": {
10       "CGO_ENABLED": "1",
11       "GO386": "softfloat",
12       "GOARCH": "amd64",
13       "GOARM": "",
```

```
2 {
3 "/nix/store/gccilxhvxbhm79....-go-1.21.8drv": {
4 "args": [
5   "-e",
6   "/nix/store/v6x3cs394jgqfbioa42pam708flxaphh-default",
7 ],
8 "builder": "/nix/store/5lr5n...-bash-5.2p26/bin/bash"
9 "env": {
10   "CGO_ENABLED": "1",
11   "GO386": "softfloat",
12   "GOARCH": "amd64",
13   "GOARM": "",
14   "GOHOSTARCH": "amd64",
```



```
47  "src": "/nix/store/p81s0316n7snx40fwhda4p5jczf2pff
48  "stdenv": "/nix/store/c8dj731bkcdzhgrpaawhc8qvdlgls4x
49  "strictDeps": "1",
50  "system": "x86_64-linux",
51  "version": "1.21.8"
52 },
53 "inputDrv": {
54   "/nix/store/17gdfyx....-stdenv-linux.drv": {
55     "dynamicOutputs": {},
56     "outputs": [
57       "out"
58     ]
59   },

```

```
53 "inputDrv": {
54   "/nix/store/17gdfyx....-stdenv-linuxdrv": {
55     "dynamicOutputs": {},
56     "outputs": [
57       "out"
58     ]
59   },
60   "/nix/store/9j2pqjj8...-mailcap-1.17.patchdrv": {
61     "dynamicOutputs": {},
62     "outputs": [
63       "out"
64     ]
65   },

```

```
105    "/nix/store/m88mg4d43hwkkbip6dha7p858c0vm5c1-go_no_"
106    "/nix/store/v6x3cs394jgqfbio42pam708flxaphh-default"
107    "/nix/store/x48d0s4gns4jrck6qkwrpqn7nh9ygpx6-remove"
108 ],
109 "name": "go-1.21.8",
110 "outputs": {
111     "out": {
112         "path": "/nix/store/afv3zwqxyw062...-go-1.21.8"
113     }
114 },
115 "system": "x86_64-linux"
116 }
117 }
```

# Advantages

- A derivation is immutable

```
/nix/store/afv3zwqxyw062vg2j220658jq0g1yadv-go-1.21.8
└── bin
    └── go -> ../../share/go/bin/go
    └── gofmt -> ../../share/go/bin/gofmt
└── share
    └── go
        ├── api
        ├── bin
        ├── doc
        ├── go.env
        ├── lib
        ├── misc
        └── pkg
```

# Advantages

- Binary cache

```
1 > nix-shell -p go_1_21
2
3 this path will be fetched
4 (39.16 MiB download, 204.47 MiB unpacked):
5   /nix/store/k7chjapvryi....-go-1.21.8
6 copying path '/nix/store/k7chjapvryi....-go-1.21.8'
7 from 'https://cache.nixos.org'..
```

# Advantages

- Binary cache

```
1 > nix-shell -p go_1_21
2
3 this path will be fetched
4 (39.16 MiB download, 204.47 MiB unpacked):
5   /nix/store/k7chjapvryi....-go-1.21.8
6 copying path '/nix/store/k7chjapvryi....-go-1.21.8'
7 from 'https://cache.nixos.org'..
```

# Advantages

- Forces us to make our dependency tree explicit

```
> nix-store -q --tree /nix/store/k7chj...-go-1.21.8
```

```
/nix/store/k7chj...-go-1.21.8
└── /nix/store/7vvgggrs9367d3...-mailcap-2.1.53
    ├── /nix/store/a1s263pmisci9z...-bash-5.2p26
    │   └── /nix/store/ddwyrxif6...-glibc-2.39-5
    │       ├── /nix/store/rxgan...-xgcc-13.2.0-libgcc
    │       ├── /nix/store/s32cl...-libidn2-2.3.7
    │       │   └── /nix/store/7n...-libunistring-1>
    │           └── /nix/store/7n0mbqydcipkpbxm24fab066lxl
    │               └── /nix/store/s32cldbh9pfzd9z82izi12mdlrw0yf8
    └── /nix/store/ddwyrxif62r8n6xclvskjyy6szdhvj60-g
    └── /nix/store/a1s263pmisci9zykm5xcdf7x9rv26w6d5-bash-
```

# Advantages

- Atomic updates

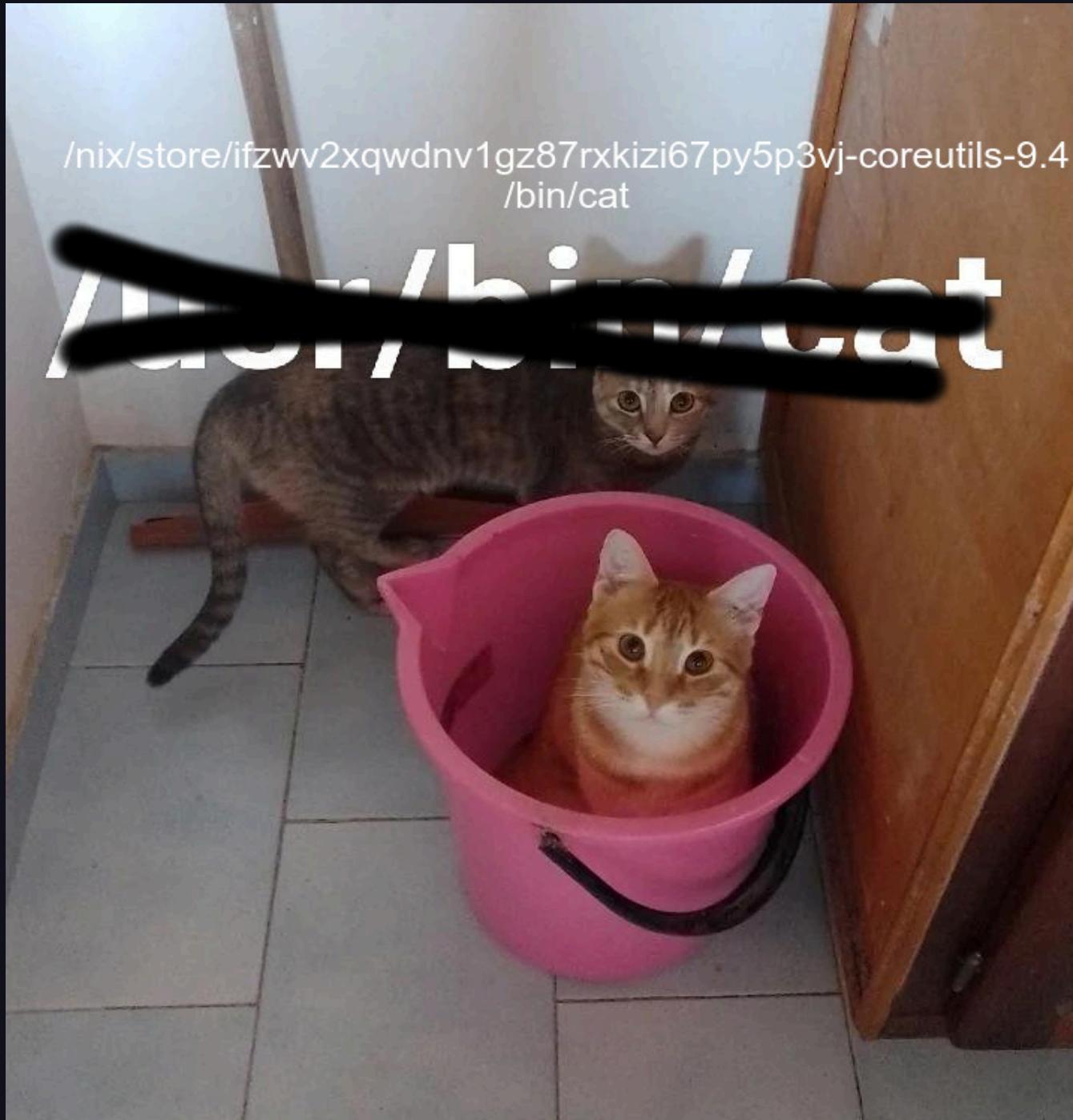
```
> ls -al ~/.nix-profile/bin
```

```
lrwxrwxrwx - root 1 Jan 1970 , -> /nix/store/09irdfc2nqj
lrwxrwxrwx - root 1 Jan 1970 accessdb -> /nix/store/09i
lrwxrwxrwx - root 1 Jan 1970 addgnupghome -> /nix/store/
lrwxrwxrwx - root 1 Jan 1970 ag -> /nix/store/09irdfc2n
lrwxrwxrwx - root 1 Jan 1970 animate -> /nix/store/09i
lrwxrwxrwx - root 1 Jan 1970 applygnupgdefaults -> /nix/
lrwxrwxrwx - root 1 Jan 1970 apropos -> /nix/store/09i
lrwxrwxrwx - root 1 Jan 1970 atuin -> /nix/store/09irdfc
```



/nix/store/ifzwv2xqwdnv1gz87rxkizi67py5p3vj-coreutils-9.4  
/bin/cat

/bin/cat



# Summary

- Derivations are a key building block of Nix
- Nix derivations -> immutable packages

# Nix Flakes

- Per project dependencies
  - `flake.lock`
- Improves reproducibility

# flake.nix

```
{  
  inputs = {  
    # Aliased to "nixpkgs";  
    nixpkgs.url = "github:NixOS/nixpkgs";  
  };  
  outputs = {};  
}
```

# flake.lock

```
1 {
2   "nodes": {
3     "nixpkgs": {
4       "locked": {
5         "lastModified": 1668703332,
6         // A SHA of the contents of the flake
7         "narHash": "sha256-PW3vz30DXaInogvp2I...",
8         // The GitHub org
9         "owner": "NixOS",
10        // The GitHub repo
11        "repo": "nixpkgs",
12        // The specific revision
13        "rev": "de60d387a0e5737375ee61848872b1...",
```

# flake.lock

```
1 {
2   "nodes": {
3     "nixpkgs": {
4       "locked": {
5         "lastModified": 1668703332,
6         // A SHA of the contents of the flake
7         "narHash": "sha256-PW3vz30DXaInogvp2I...",
8         // The GitHub org
9         "owner": "NixOS",
10        // The GitHub repo
11        "repo": "nixpkgs",
12        // The specific revision
13        "rev": "de60d387a0e5737375ee61848872b1...",
```

# flake.lock

```
7      "narHash": "sha256-PW3vz30DXaInogvp2I...",
8      // The GitHub org
9      "owner": "NixOS",
10     // The GitHub repo
11     "repo": "nixpkgs",
12     // The specific revision
13     "rev": "de60d387a0e5737375ee61848872b1...",
14     // The type of input
15     "type": "github"
16   },
17 },
18 // Other inputs
19 }
```

# Summary

- Nix Flakes improve reproducibility
  - Lock dependencies
- Provide a more standard way to configure Nix
- Are an EXPERIMENTAL feature still

# CI

- Use same versions as local
- Leverage Nix “cachability”

# GitLab CI

```
1 image: nixos/nix
2
3 tests:unit:
4   only:
5     - merge_request
6   before_script:
7     - echo "experimental-features = nix-command flakes"
8   script:
9     - nix develop -c task tests:unit
```

# GitLab CI

```
1 image: nixos/nix
2
3 tests:unit:
4   only:
5     - merge_request
6   before_script:
7     - echo "experimental-features = nix-command flakes"
8   script:
9     - nix develop -c task tests:unit
```

# GitLab CI

```
1 image: nixos/nix
2
3 tests:unit:
4   only:
5     - merge_request
6   before_script:
7     - echo "experimental-features = nix-command flakes"
8   script:
9     - nix develop -c task tests:unit
```

```
1 tasks:
2   tests:unit:
3     desc: Runs all the unit tests.
4     cmds:
5       - go test -skip '^TestIntegration' ./internal/...
```

```
1 copying path '/nix/store/49mrmsvafx8lscgi...-source'
2 from 'https://cache.nixos.org'...
3 copying path '/nix/store/v6gqc89sr4gvh3gl75ncg0ajc4rbal'
4 copying path '/nix/store/ba7r274fm1v4r9zfgjr4qfsby1hxij'
5 copying path '/nix/store/8rvn0r46zg5zd5chc9wqdpz0cva2p'
6 copying path '/nix/store/dj5kdz9m149apk5hsvancfm5fksx7'
7 copying path '/nix/store/vr6ig5i2y7g8dn50qdj3ym5gkfs7s'
8 copying path '/nix/store/yyhjvp15mhffbfxsnfy3f0br47yfk'
9 copying path '/nix/store/hhxx6ns9cn6ljkphbf1jchcrsa43zi'
10 copying path '/nix/store/wak6dggawz5c00cy1iplzkiiwscy4'
11 copying path '/nix/store/g4jq20cqxnmlgz5sidrwiahmx67r'
12 copying path '/nix/store/fn1y6zydm7mgxrm7b08h1w1c9qkrz'
13 copying path '/nix/store/pd8xxiyn2xi21fgg9qm7r0qghsk87'
```

```
29 copying path '/nix/store/ymltac07j3pq0xdk44ckcd78gjxssw'
30 copying path '/nix/store/kvi43jy0kzsbkq4kmfgmrk6yw596b'
31 copying path '/nix/store/560z0zfybsjb8m76n67x6c1k7gpm0'
32 copying path '/nix/store/7k4cvc2cm6am5zq1kf5bzx7hfwm52'
33 copying path '/nix/store/awnmm98ja9nrlw5qw6ikq1gp88fph'
34 # ...
35 task: [tests:unit] go test ./...
36 go: downloading modernc.org/sqlite v1.31.1
37 go: downloading github.com/remyhanenay/slog-otel v1.1.0
38 go: downloading github.com/pressly/goose/v3 v3.21.1
39 go: downloading github.com/sethvargo/go-envconfig v1.1.0
40 go: downloading github.com/muesli/go-app-paths v0.2.2
41 go: downloading github.com/flexstack/uuid v1.0.0
42 go: downloading github.com/obhwas/wc v1.1.0
```

```
1  {
2    pkgs,
3    myPackages,
4    ...
5  }:
6  pkgs.dockerTools.buildImage {
7    name = "banterbus-dev";
8    tag = "latest";
9    copyToRoot = pkgs.buildEnv {
10      name = "banterbus-dev";
11      pathsToLink = ["/bin"];
12      paths = with pkgs;
13      [
```

```
11     pathsToLink = ["/bin"];  
12     paths = with pkgs;  
13     [  
14         coreutils  
15         gnugrep  
16         bash  
17         cacert.out  
18         curl  
19         git  
20     ]  
21     ++ myPackages;  
22 };  
23 config = {
```

```
15      gnugrep
16      bash
17      cacert.out
18      curl
19      git
20    ]
21    ++ myPackages;
22  };
23  config = {
24    Env = [
25      "NIX_PAGER=cat"
26      # A user is required by nix
27      # https://github.com/NixOS/nix/blob/9348f9291e5d
```



```
29      go-task
30      goreleaser
31      sqlc
32    ];
33  in {
34    packages.ci = pkgs.callPackage ./ci.nix {
35      inherit pkgs;
36      inherit myPackages;
37    };
38    devShells.default = pkgs.mkShell ./shell.nix {
39      inherit pkgs;
40      inherit myPackages;
41    }
42  }.
```

```
33 in {
34     packages.ci = pkgs.callPackage ./ci.nix {
35         inherit pkgs;
36         inherit myPackages;
37     };
38     devShells.default = pkgs.mkShell ./shell.nix {
39         inherit pkgs;
40         inherit myPackages;
41     }
42     };
43   );
44 );
45 }
```



```
10    changes:
11        - "containers/ci.nix"
12  services:
13      - docker:25-dind
14  script:
15      - echo "experimental-features = nix-command flakes"
16      - nix-env -iA nixpkgs.docker
17      - nix build .#ci
18      - docker load < ./result
19      - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD
20      - docker image tag banterbus-dev:latest $IMAGE:latest
21      - docker push $CI_REGISTRY_IMAGE/ci:latest
22
```

# CI Improved

```
1
2 stages:
3   - deps
4   - test
5
6 .task:
7   stage: test
8   image: $CI_REGISTRY_IMAGE/ci:latest
9   variables:
10     GOPATH: $CI_PROJECT_DIR/.go
11   cache:
12     paths:
13       - ${GOPATH}/pkg/mod
```

# CI Improved

```
2 stages:
3   - deps
4   - test
5
6 .task:
7   stage: test
8   image: $CI_REGISTRY_IMAGE/ci:latest
9   variables:
10     GOPATH: $CI_PROJECT_DIR/.go
11   cache:
12     paths:
13       - ${GOPATH}/pkg/mod
14   policy: pull
```

# CI Improved

```
6 .task:
7   stage: test
8   image: $CI_REGISTRY_IMAGE/ci:latest
9   variables:
10     GOPATH: $CI_PROJECT_DIR/.go
11   cache:
12     paths:
13       - ${GOPATH}/pkg/mod
14   policy: pull
15   rules:
16     - if: $CI_PIPELINE_SOURCE == "merge_request_event"
17
18 download-dependency:
```

# CI Improved

```
17
18 download:dependency:
19   extends: .task
20   stage: deps
21   rules:
22     - if: $CI_PIPELINE_SOURCE == "merge_request_event"
23       changes:
24         - go.mod
25         - go.sum
26   script:
27     - go mod download
28   cache:
29     policy: pull-push
30
```

# CI Improved

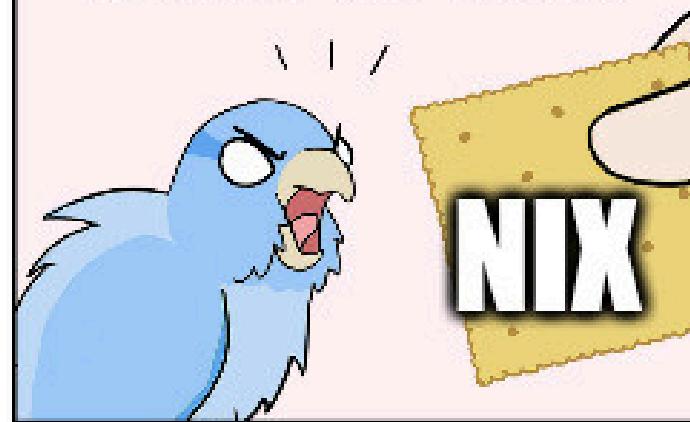
```
27      - go mod download
28 cache:
29     policy: pull-push
30
31 tests:unit:
32   extends:
33     - .task
34   script:
35     - task tests:unit
36
37 format:
38   extends:
39     - .task
```

# Time Improvement

- unit tests job
  - 2 minutes 28 seconds
  - 54 seconds



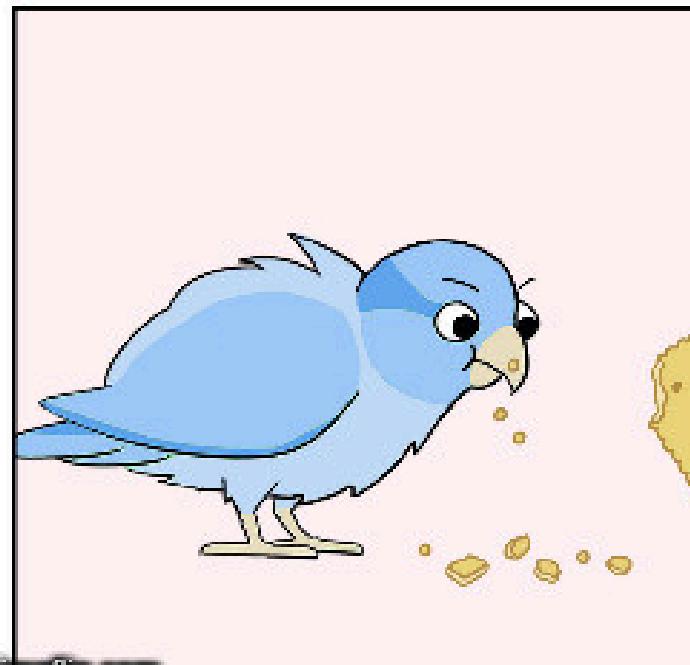
GET THAT THING  
OUT OF MY FACE!



chomp



NIX



chicken thoughts

# Why not Docker?

- Docker is imperative
  - Repeatable but not reproducible
- Hard to personalise
  - bash vs fish vs zsh

# Try Nix!

- Install Nix
  - <https://determinate.systems/posts/deterministic-nix-installer/>
- Use dev template
  - <https://flakehub.com/f/the-nix-way/dev-templates>
- Install direnv-nix
  - <https://github.com/nix-community/nix-direnv>

```
curl --proto '=https' --tlsv1.2 -sSf -L https://install.de  
nix flake init --template "https://flakehub.com/f/the-nix-  
nix profile install nixpkgs#nix-direnv
```

# Further

- gomod2nix: <https://www.tweag.io/blog/2021-03-04-gomod2nix/>
- Build Docker image:  
<https://jameswillia.ms/posts/go-nix-containers.html>
- Arion: Manage docker-compose with nix
- devenv: <https://devenv.sh/>



# Slides

- Slides: <https://haseebmajid.dev/slides/go-lab-reproducible-envs-with-nix/>

# My Links

- My Dotfiles Configured Using Nix
- How I setup dev shell for a Go project
  - Project using dev shell

# Appendix

- Useful Articles:
  - <https://blog.ysndr.de/posts/guides/2021-12-01-nix-shells/>
  - <https://serokell.io/blog/what-is-nix>
  - <https://shopify.engineering/what-is-nix>
  - nix-shell vs nix shell vs nix develop:  
<https://blog.ysndr.de/posts/guides/2021-12-01-nix-shells/>

# Useful Tools

```
> namei
```

- Get started with Nix

- <https://zero-to-nix.com/>
- <https://nixos.org/guides/nix-pills/why-you-should-give-it-a-try>
- <https://nixos-and-flakes.thiscute.world/introduction/>

More about flakes:

- <https://nixos.wiki/wiki/Flakes>
- <https://zero-to-nix.com/concepts/flakes>

- Useful Channels/Videos

- <https://www.youtube.com/@vimjoyer>
- Docker and Nix (Dockercon 2023):  
<https://www.youtube.com/watch?v=l17oRkhg>
- How to build a new package in Nix:  
<https://www.youtube.com/watch?v=3hMlqxb>
- [https://www.youtube.com/watch?app=desktop&v=TsZte\\_9GfPE&si=osBujLY3py](https://www.youtube.com/watch?app=desktop&v=TsZte_9GfPE&si=osBujLY3py)

# References & Thanks

- GIFs made with [vhs](#)
- Photos edited with [pixlr](#)
- All my friends who took time to give me feedback on this talk