# 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”

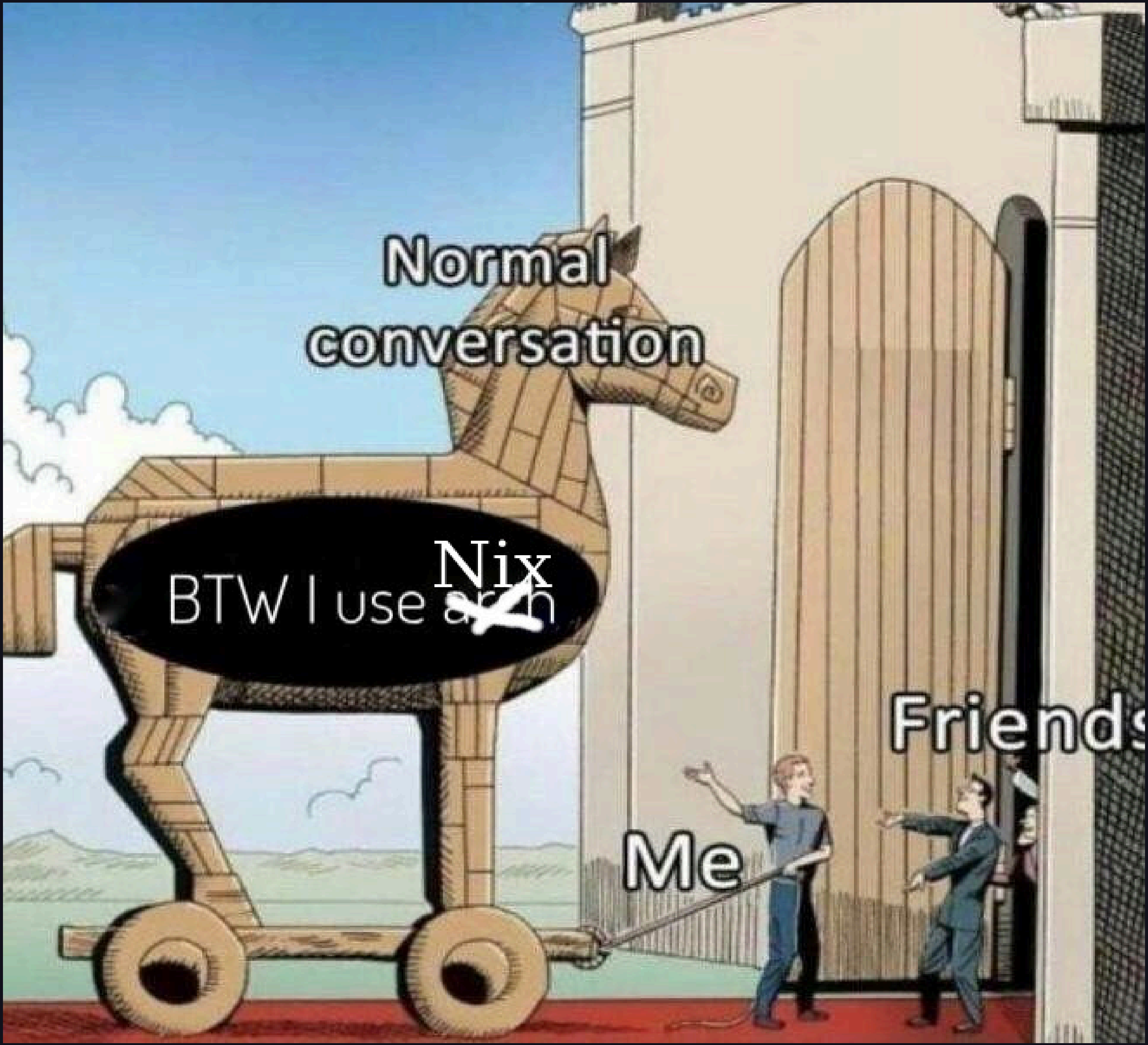
Credit

# 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;
}
```


Normal conversation
BTW I use ~~arch~~ Nix
Friends
Me

# Summary

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

```
>
```

# Golang

- Tools to aid development
  - lint
  - testing
  - docker

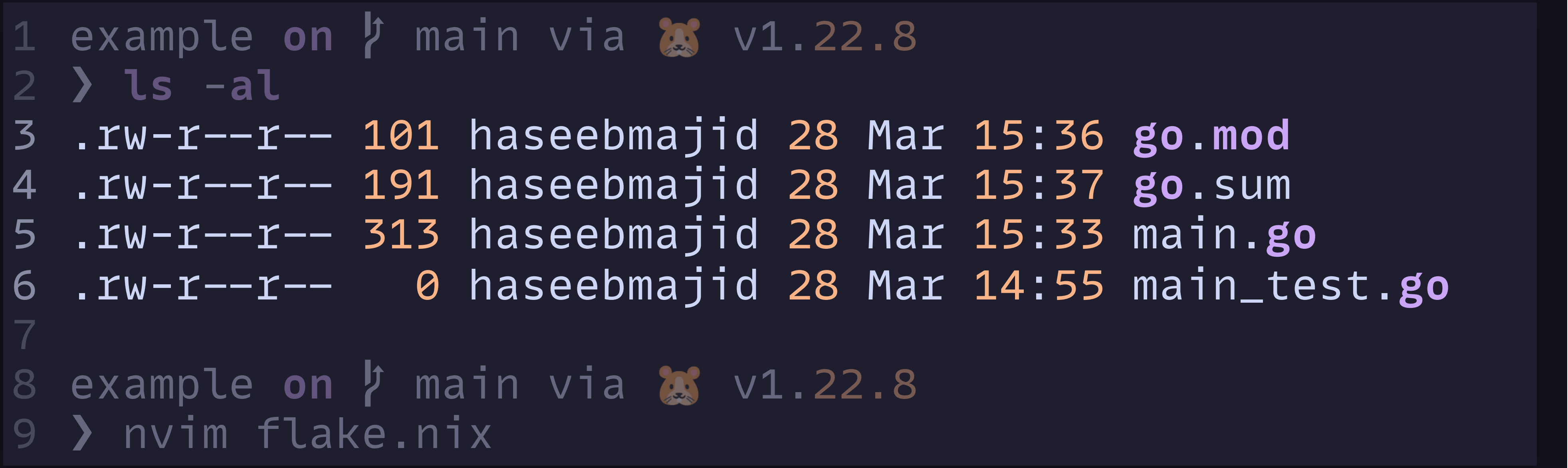
# tools.go

```go
// +build tools

package main

import (
_ "github.com/golangci/golangci-lint/cmd/golangci-lint"
_ "github.com/goreleaser/goreleaser"
)
```

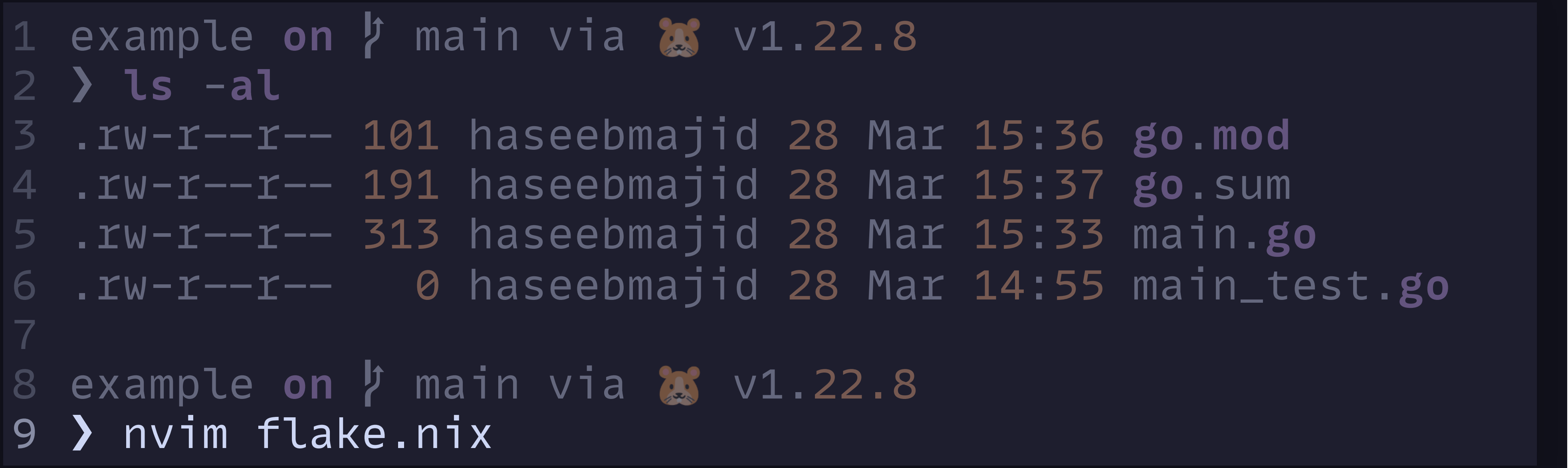
# tools.go

```go
// +build tools

package main

import (
_ "github.com/golangci/golangci-lint/cmd/golangci-lint"
_ "github.com/goreleaser/goreleaser"
)
```

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

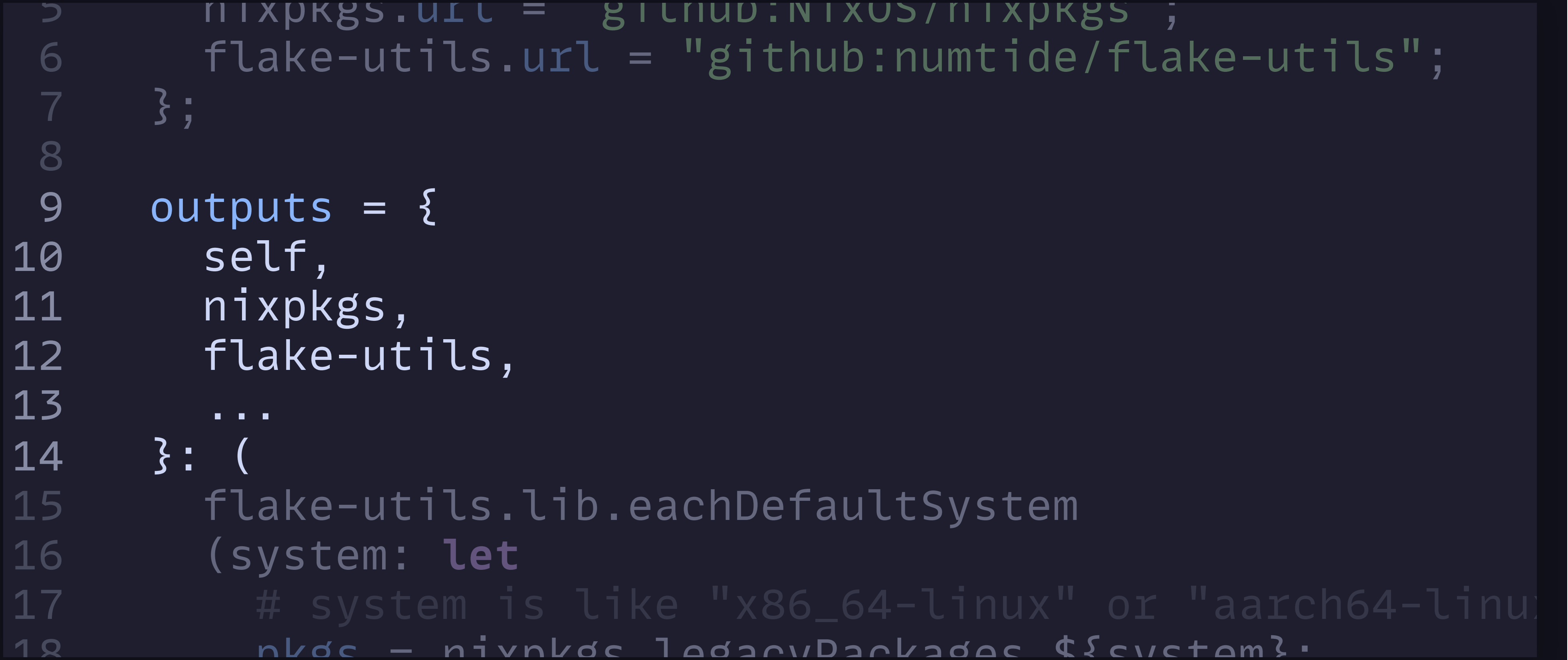


```
example on  main via 🐹 v1.22.8
❯ ls -al
.rw-r--r-- 101 haseebmajid 28 Mar 15:36 go.mod
.rw-r--r-- 191 haseebmajid 28 Mar 15:37 go.sum
.rw-r--r-- 313 haseebmajid 28 Mar 15:33 main.go
.rw-r--r--   0 haseebmajid 28 Mar 14:55 main_test.go

example on  main via 🐹 v1.22.8
❯ nvim flake.nix
```

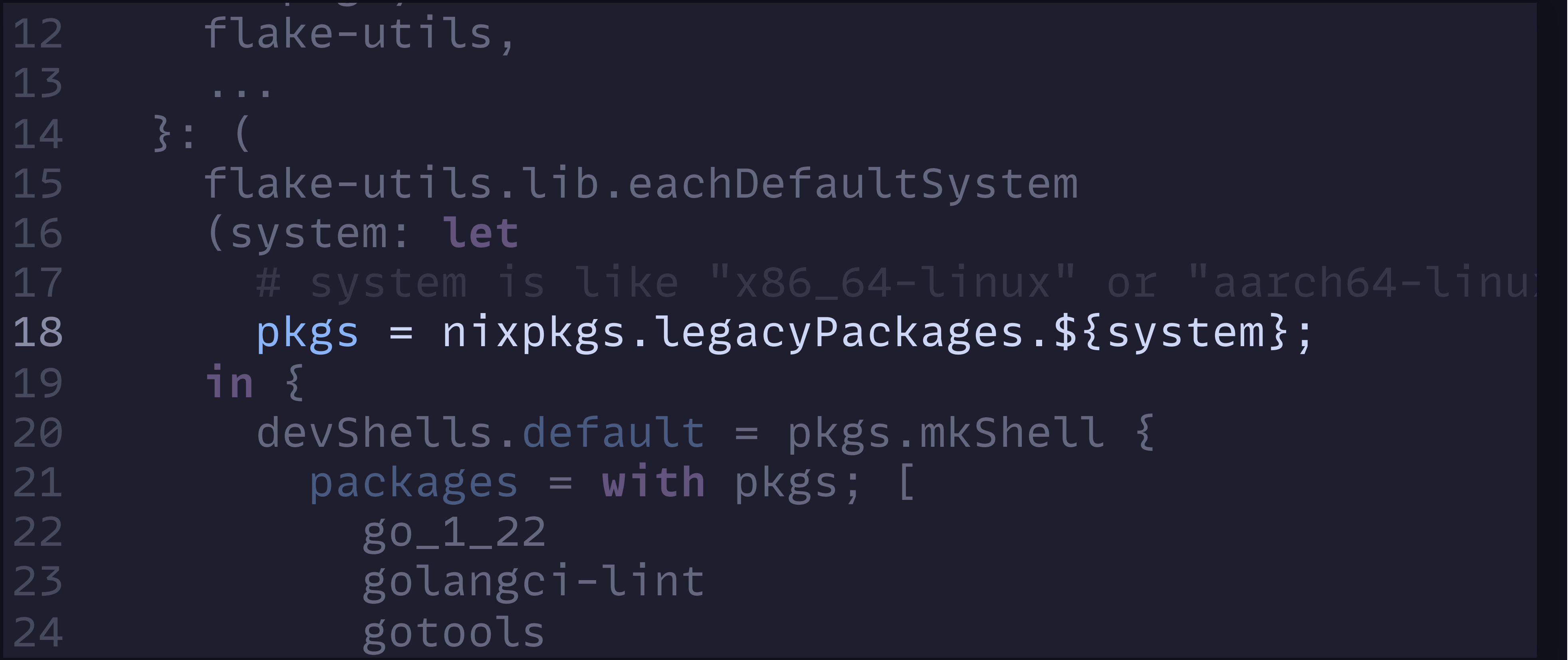


```
example on  main via 🐹 v1.22.8
❯ ls -al
.rw-r--r-- 101 haseebmajid 28 Mar 15:36 go.mod
.rw-r--r-- 191 haseebmajid 28 Mar 15:37 go.sum
.rw-r--r-- 313 haseebmajid 28 Mar 15:33 main.go
.rw-r--r--   0 haseebmajid 28 Mar 14:55 main_test.go

example on  main via 🐹 v1.22.8
❯ nvim flake.nix
```

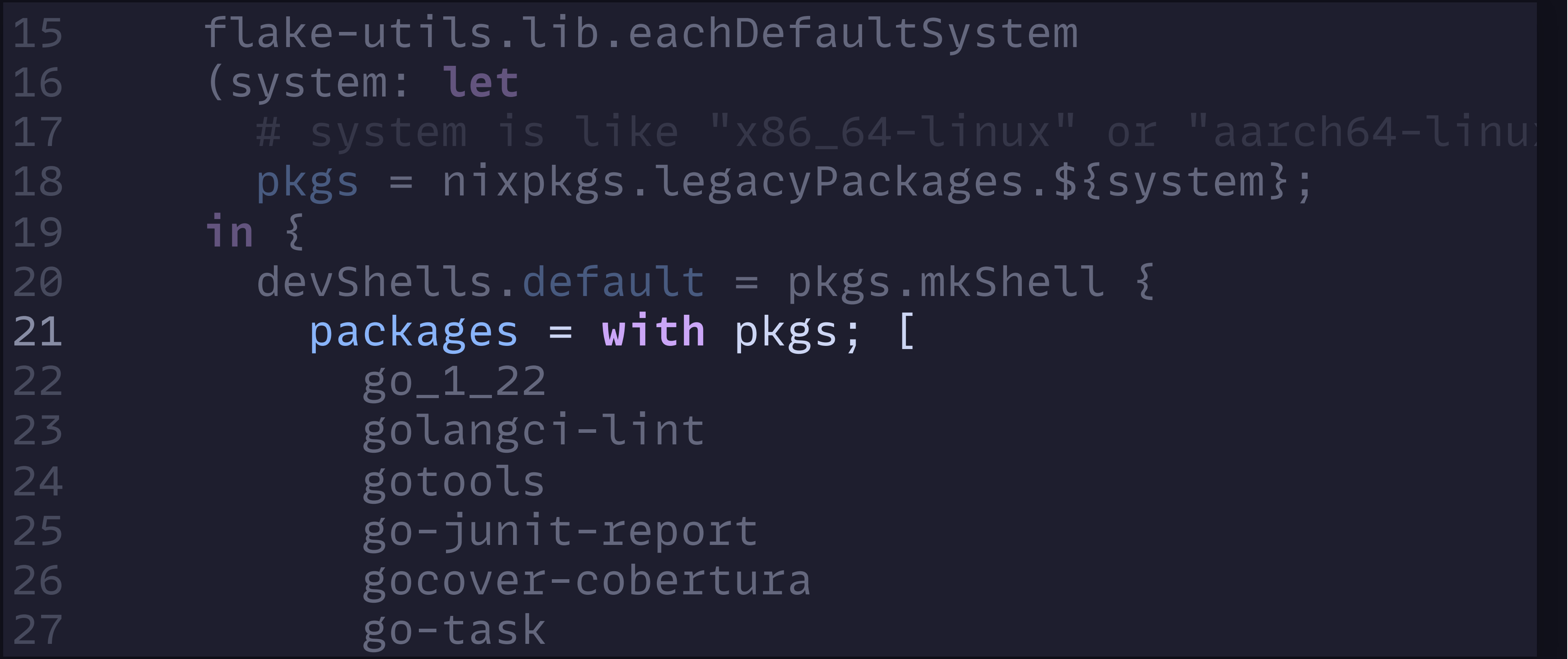
```nix
{
  description = "I use Nix btw!";

  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs";
    flake-utils.url = "github:numtide/flake-utils";
  };

  outputs = {
    self,
    nixpkgs,
    flake-utils,
    ...
```



```
    nixpkgs.url = "github:NixOS/nixpkgs";
    flake-utils.url = "github:numtide/flake-utils";
  };

  outputs = {
    self,
    nixpkgs,
    flake-utils,
    ...
  }: (
    flake-utils.lib.eachDefaultSystem
    (system: let
      # system is like "x86_64-linux" or "aarch64-linu
      pkgs = nixpkgs.legacyPackages.${system};
  outputs = {
    self,
    nixpkgs,
    flake-utils,
    ...
  }: (
    flake-utils.lib.eachDefaultSystem
    (system: let
      # system is like "x86_64-linux" or "aarch64-linu
      pkgs = nixpkgs.legacyPackages.${system};
    in {
      devShells.default = pkgs.mkShell {
        packages = with pkgs; [
```

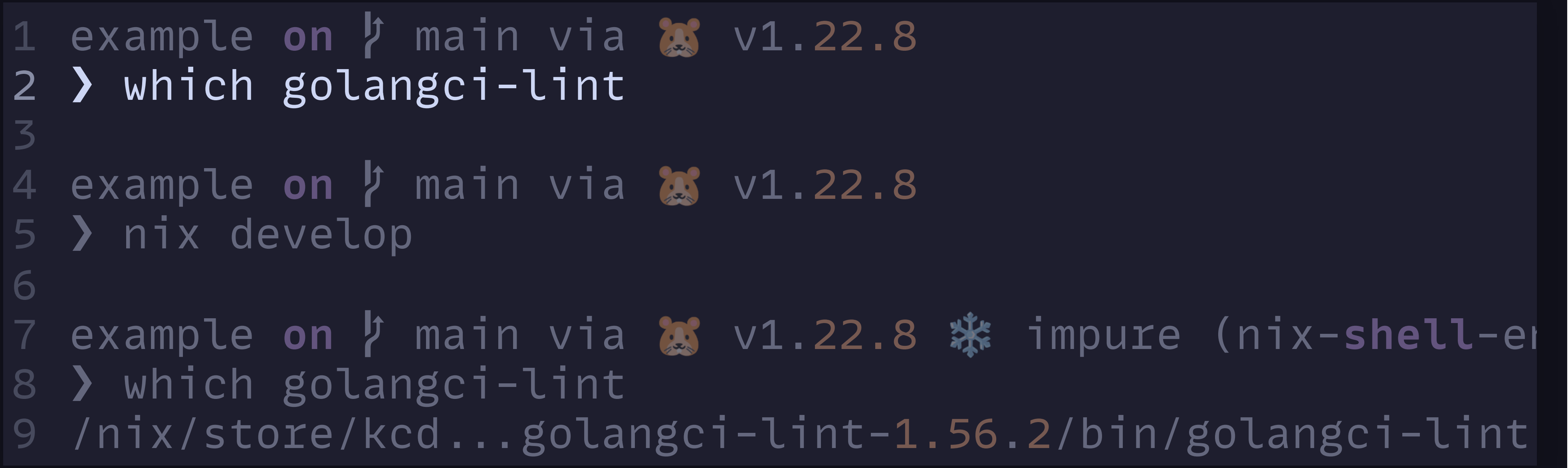


```
    flake-utils,
    ...
  }: (
    flake-utils.lib.eachDefaultSystem
    (system: let
      # system is like "x86_64-linux" or "aarch64-linu
      pkgs = nixpkgs.legacyPackages.${system};
    in {
      devShells.default = pkgs.mkShell {
        packages = with pkgs; [
          go_1_22
          golangci-lint
          gotools
  }: (
    flake-utils.lib.eachDefaultSystem
    (system: let
      # system is like "x86_64-linux" or "aarch64-linu
      pkgs = nixpkgs.legacyPackages.${system};
    in {
      devShells.default = pkgs.mkShell {
        packages = with pkgs; [
          go_1_22
          golangci-lint
          gotools
          go-junit-report
          gocover-cobertura
```

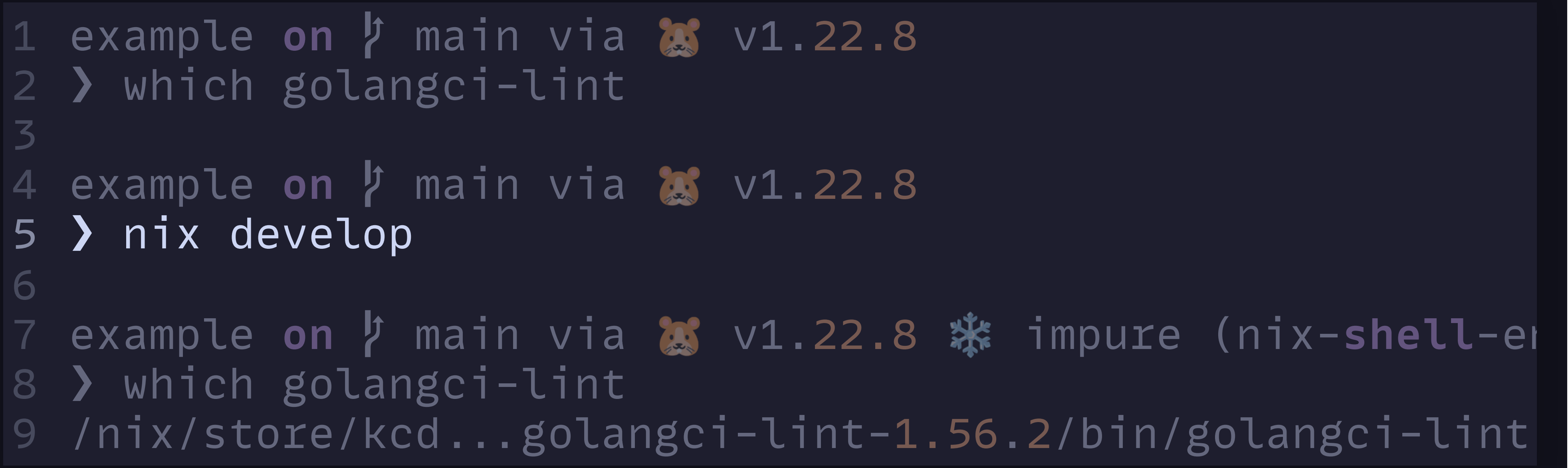


```
    flake-utils.lib.eachDefaultSystem
    (system: let
      # system is like "x86_64-linux" or "aarch64-linu
      pkgs = nixpkgs.legacyPackages.${system};
    in {
      devShells.default = pkgs.mkShell {
        packages = with pkgs; [
          go_1_22
          golangci-lint
          gotools
          go-junit-report
          gocover-cobertura
          go-task
```

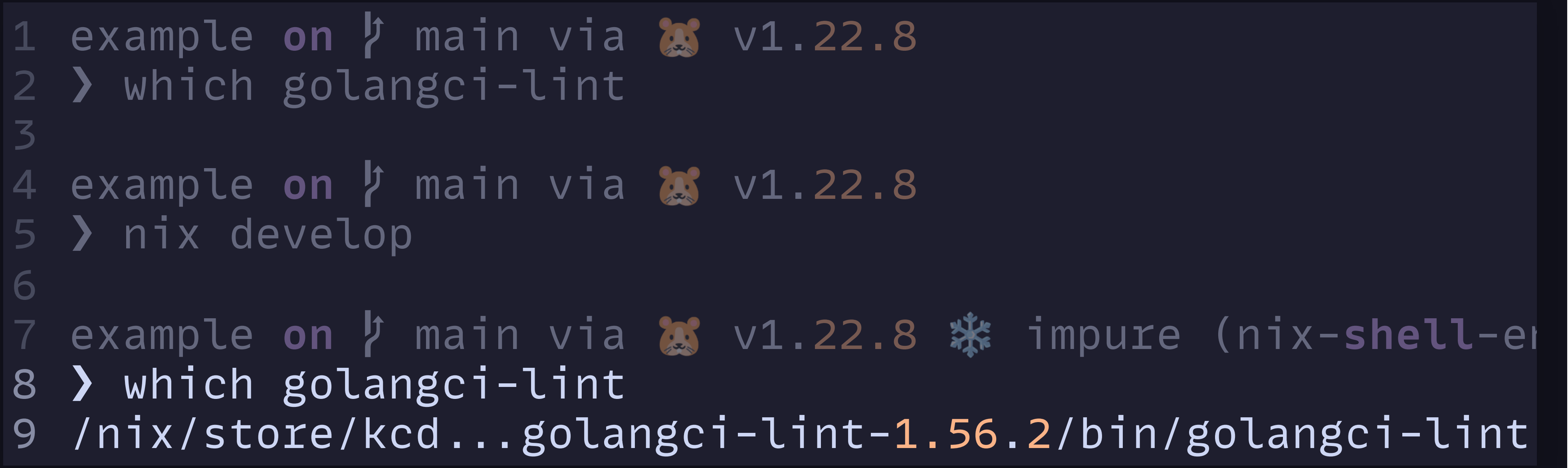
```nix
    in {
      devShells.default = pkgs.mkShell {
        packages = with pkgs; [
          go_1_22
          golangci-lint
          gotools
          go-junit-report
          gocover-cobertura
          go-task
          goreleaser
          sqlc
        ];
      };
    })
```



```
example on  main via 🐹 v1.22.8
❯ which golangci-lint

example on  main via 🐹 v1.22.8
❯ nix develop

example on  main via 🐹 v1.22.8 ❄️ impure (nix-shell-e
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```

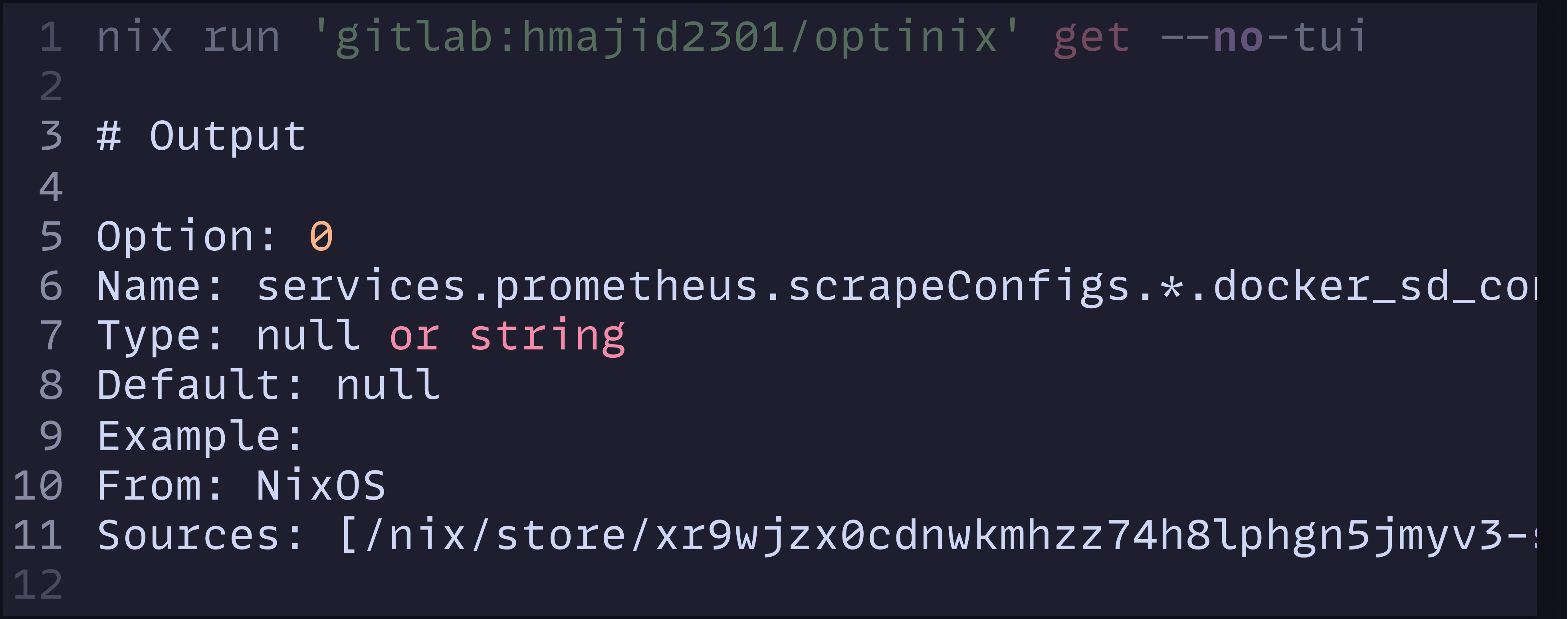


```
example on  main via 🐹 v1.22.8
❯ which golangci-lint

example on  main via 🐹 v1.22.8
❯ nix develop

example on  main via 🐹 v1.22.8 ❄️ impure (nix-shell-e
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```



```
example on  main via 🐹 v1.22.8
❯ which golangci-lint

example on  main via 🐹 v1.22.8
❯ nix develop

example on  main via 🐹 v1.22.8 ❄️ impure (nix-shell-e
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint
```

```
nix run 'gitlab:hmajid2301/optinix' get --no-tui

# Output

Option: 0
Name: services.prometheus.scrapeConfigs.*.docker_sd_co
Type: null or string
Default: null
Example:
From: NixOS
Sources: [/nix/store/xr9wjzx0cdnwkmhzz74h8lphgn5jmyv3-

```



```
nix run 'gitlab:hmajid2301/optinix' get --no-tui

# Output

Option: 0
Name: services.prometheus.scrapeConfigs.*.docker_sd_co
Type: null or string
Default: null
Example:
From: NixOS
Sources: [/nix/store/xr9wjzx0cdnwkmhzz74h8lphgn5jmyv3-
```

# 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

```
❯ which golangci-lint

❯ cd banterbus
direnv: error /home/haseeb/banterbus/.envrc
is blocked. Run `direnv allow` to approve
its content

banterbus on  main via 🐹 v1.22.7
❯ direnv allow
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...
```

# Usage

```
❯ which golangci-lint

❯ cd banterbus
direnv: error /home/haseeb/banterbus/.envrc
is blocked. Run `direnv allow` to approve
its content

banterbus on  main via 🐹 v1.22.7
❯ direnv allow
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...
```

# Usage

```
❯ which golangci-lint

❯ cd banterbus
direnv: error /home/haseeb/banterbus/.envrc
is blocked. Run `direnv allow` to approve
its content

banterbus on  main via 🐹 v1.22.7
❯ direnv allow
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...
```

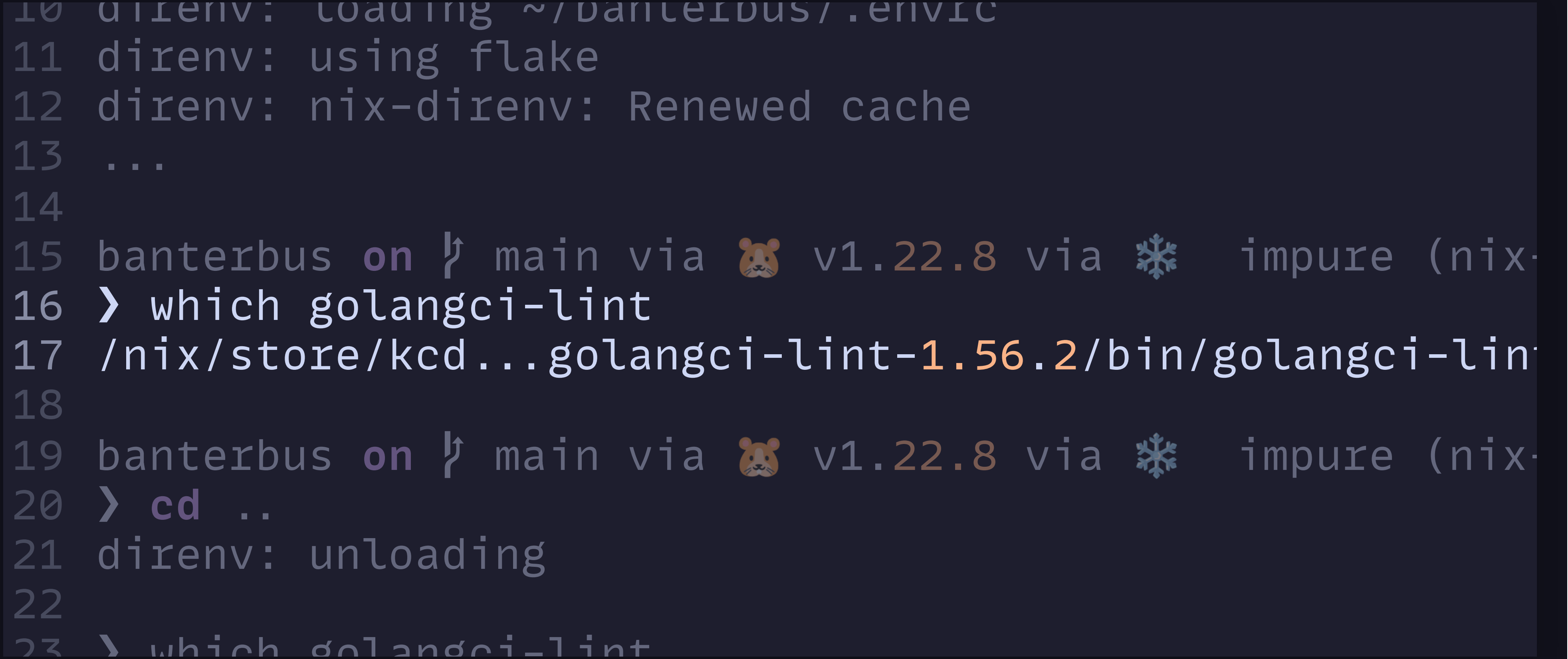
# Usage

```
❯ cd banterbus
direnv: error /home/haseeb/banterbus/.envrc
is blocked. Run `direnv allow` to approve
its content

banterbus on  main via 🐹 v1.22.7
❯ direnv allow
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
```

# Usage

```
is blocked. Run `direnv allow` to approve
its content

banterbus on  main via 🐹 v1.22.7
❯ direnv allow
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lin
```

# Usage



```
direnv: loading ~/banterbus/.envrc
direnv: using flake
direnv: nix-direnv: Renewed cache
...

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lint

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
❯ cd ..
direnv: unloading

❯ which golangci-lint
```

# Usage

```
direnv: using flake
direnv: nix-direnv: Renewed cache
...

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lin

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix-
❯ cd ..
direnv: unloading

❯ which golangci-lint
```

# Usage

```
direnv: using flake
direnv: nix-direnv: Renewed cache
...

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix
❯ which golangci-lint
/nix/store/kcd...golangci-lint-1.56.2/bin/golangci-lin

banterbus on  main via 🐹 v1.22.8 via ❄️  impure (nix
❯ cd ..
direnv: unloading

❯ which golangci-lint
```

# Remote Environments

```
# .envrc

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

# pre-commit

```
{
description = "Development environment for example pro
    
inputs = {
nixpkgs.url = "github:NixOS/nixpkgs";
flake-utils.url = "github:numtide/flake-utils";
pre-commit.url = "github:cachix/pre-commit-hooks.nix";
};

outputs = {
self,
nixpkgs,
flake-utils,
```

# pre-commit

```
}: (
flake-utils.lib.eachDefaultSystem
(system: let
  pkgs = nixpkgs.legacyPackages.${system};
  pre-commit-check = pre-commit.lib.${system}.run {
    src = ./.;
    hooks = {
      golangci-lint.enable = true;
      gotest.enable = true;
    };
  };
in {
  devShells.default = pkgs.mkShell {
    shellHook = pre-commit-check.shellHook;
```

# pre-commit

```nix
      golangci-lint.enable = true;
      gotest.enable = true;
    };
  };
in {
  devShells.default = pkgs.mkShell {
    shellHook = pre-commit-check.shellHook;
    packages = with pkgs; [
      go_1_22
      golangci-lint
      gotools
      go-junit-report
      gocover-cobertura
```

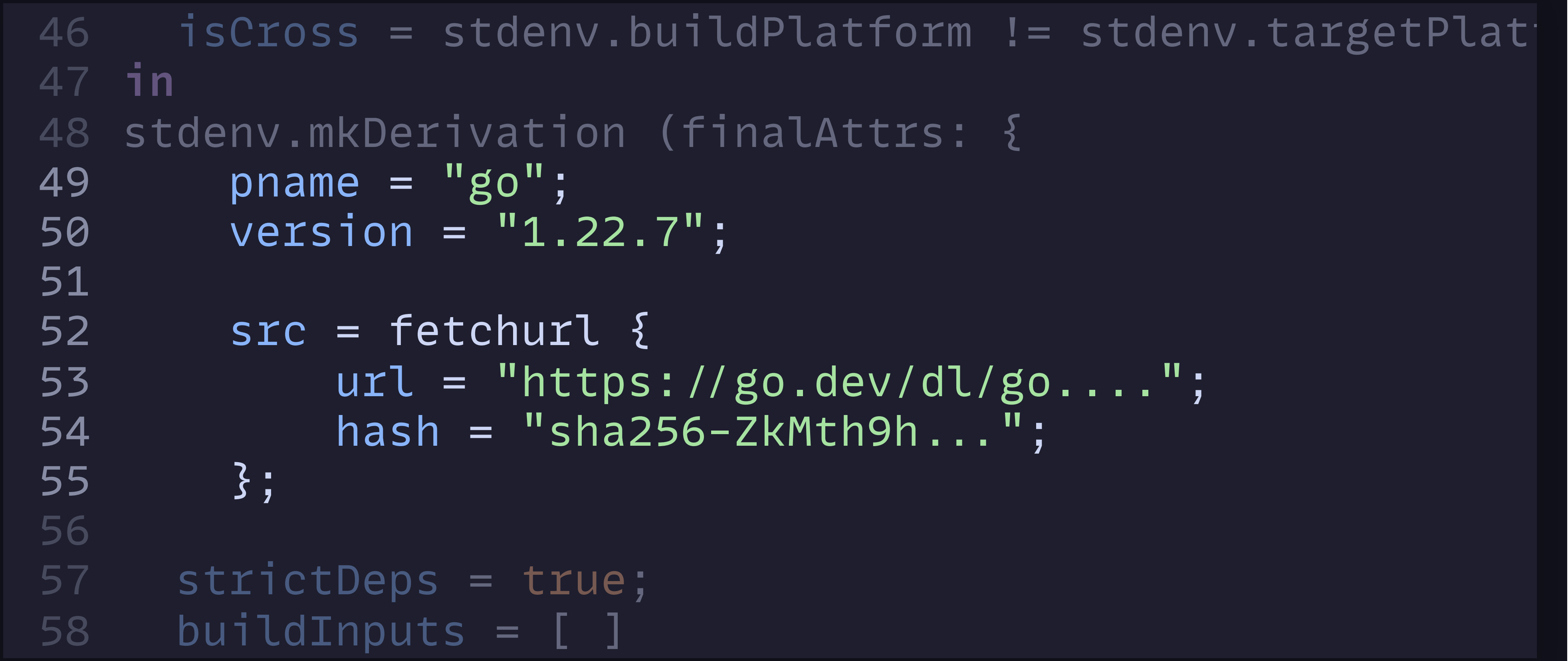
# Summary

- 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?

```
{ lib
, stdenv
, fetchurl
, tzdata
, substituteAll
, iana-etc
, Security
, Foundation
, xcbuild
, mailcap
, buildPackages
, pkgsBuildTarget
, threadsCross
```

# Go Nix Expression



```
  isCross = stdenv.buildPlatform != stdenv.targetPlat
in
stdenv.mkDerivation (finalAttrs: {
    pname = "go";
    version = "1.22.7";

    src = fetchurl {
        url = "https://go.dev/dl/go....";
        hash = "sha256-ZkMth9h...";
    };

  strictDeps = true;
  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

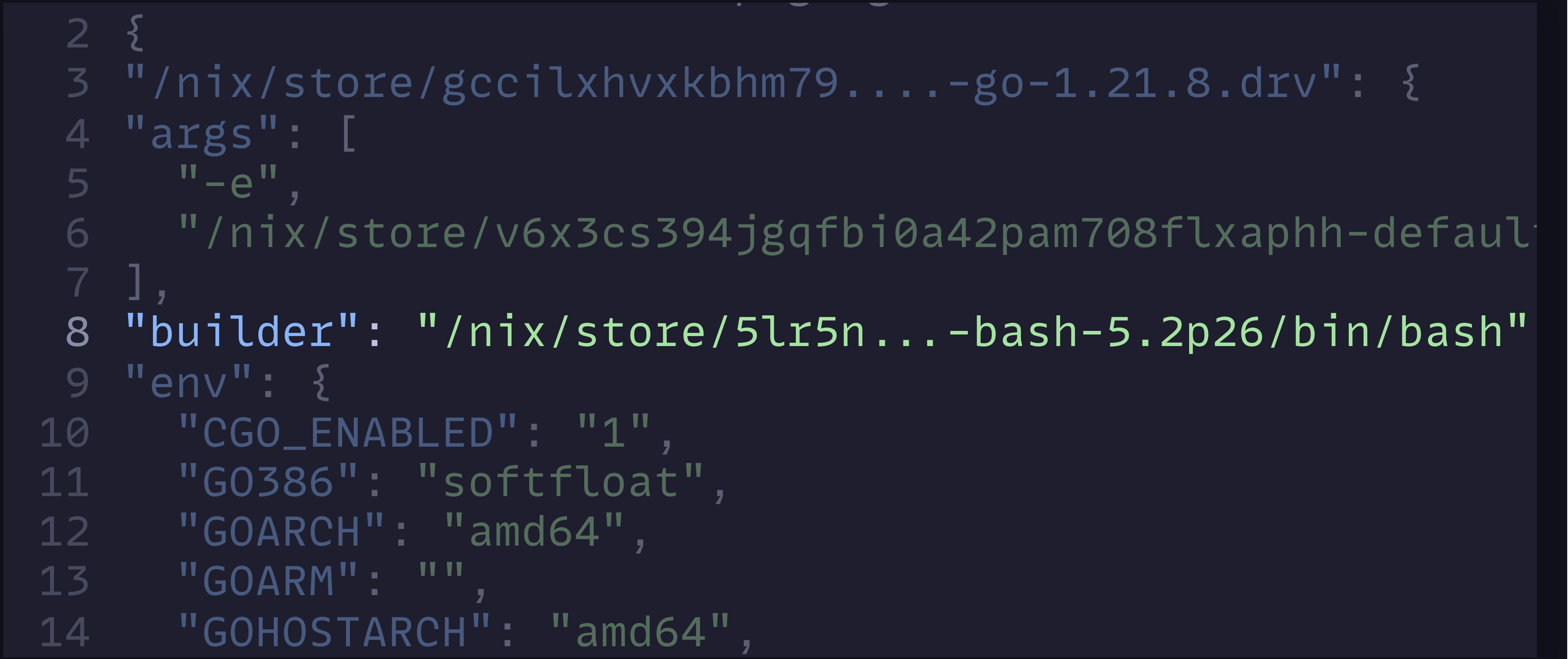
# Derivations

```
/nix/store/<hash>-<name>-<version>.drv
/nix/store/zg65r8ys8y5865lcwmmybrq5g...-go-1.21.8.drv
/nix/store/z45pk6pw3h4yx0cpi51fc5nwm...-go-1.22.1.drv
```

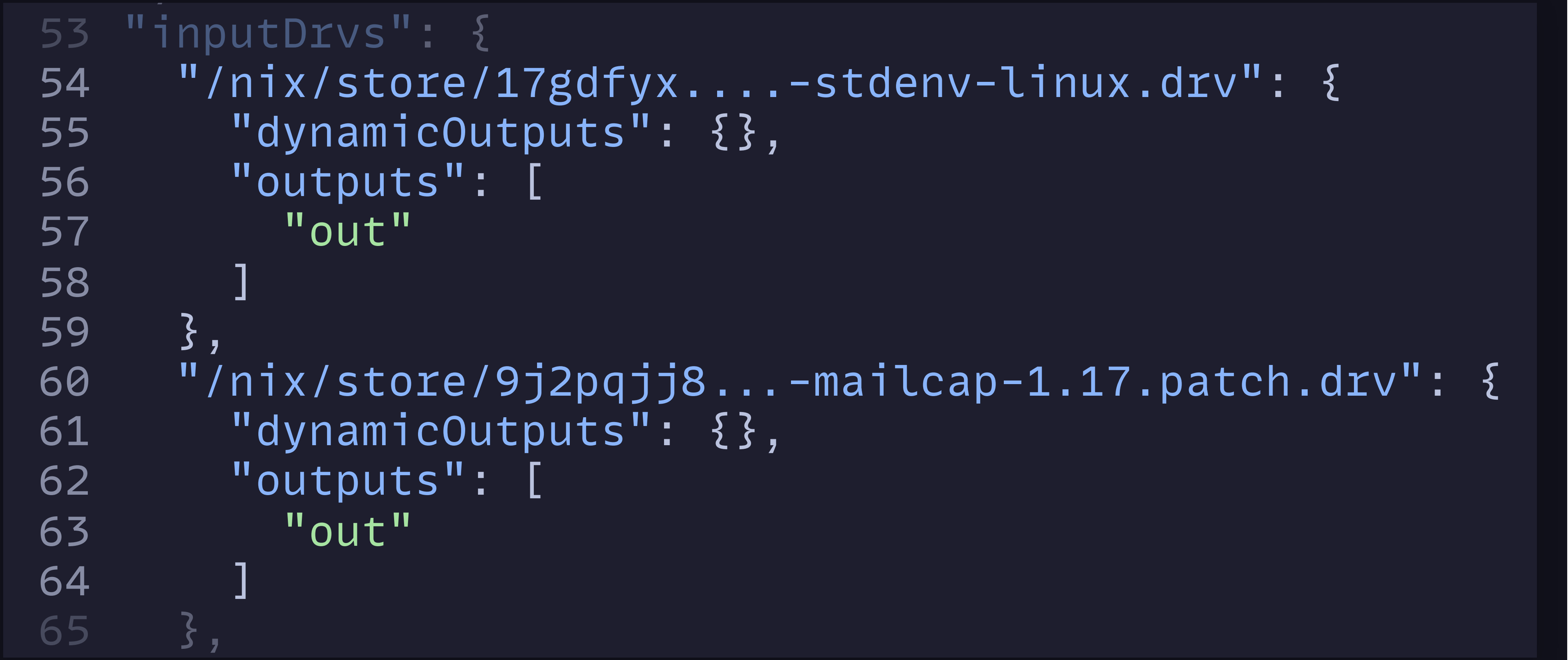
# Derivations

```
/nix/store/<hash>-<name>-<version>.drv
/nix/store/zg65r8ys8y5865lcwmmybrq5g...-go-1.21.8.drv
/nix/store/z45pk6pw3h4yx0cpi51fc5nwm...-go-1.22.1.drv
```

```
nix derivation show nixpkgs#go_1_21
{
"/nix/store/gccilxhvxkbhm79....-go-1.21.8.drv": {
"args": [
  "-e",
  "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-defaul
],
"builder": "/nix/store/5lr5n...-bash-5.2p26/bin/bash"
"env": {
  "CGO_ENABLED": "1",
  "GO386": "softfloat",
  "GOARCH": "amd64",
  "GOARM": "",
```



```
{
"/nix/store/gccilxhvxkbhm79....-go-1.21.8.drv": {
"args": [
  "-e",
  "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-defaul
],
"builder": "/nix/store/5lr5n...-bash-5.2p26/bin/bash"
"env": {
  "CGO_ENABLED": "1",
  "GO386": "softfloat",
  "GOARCH": "amd64",
  "GOARM": "",
  "GOHOSTARCH": "amd64",
"/nix/store/gccilxhvxkbhm79....-go-1.21.8.drv": {
"args": [
  "-e",
  "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-defaul
],
"builder": "/nix/store/5lr5n...-bash-5.2p26/bin/bash"
"env": {
  "CGO_ENABLED": "1",
  "GO386": "softfloat",
  "GOARCH": "amd64",
  "GOARM": "",
  "GOHOSTARCH": "amd64",
  "GOHOSTOS": "linux",
  "src": "/nix/store/p81s0316n7snx40fwkhda4p5jczf2pff
  "stdenv": "/nix/store/c8dj731bkcdzhgrpawhc8qvdgls4x
  "strictDeps": "1",
  "system": "x86_64-linux",
  "version": "1.21.8"
},
"inputDrvs": {
  "/nix/store/17gdfyx....-stdenv-linux.drv": {
    "dynamicOutputs": {},
    "outputs": [
      "out"
    ]
  },
```



```
"inputDrvs": {
  "/nix/store/17gdfyx....-stdenv-linux.drv": {
    "dynamicOutputs": {},
    "outputs": [
      "out"
    ]
  },
  "/nix/store/9j2pqjj8...-mailcap-1.17.patch.drv": {
    "dynamicOutputs": {},
    "outputs": [
      "out"
    ]
  },
  "/nix/store/m88mg4d43hwkkbip6dha7p858c0vm5c1-go_no_
  "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-defaul
  "/nix/store/x48d0s4gns4jrck6qkwrpqn7nh9ygpx6-remove
],
"name": "go-1.21.8",
"outputs": {
  "out": {
    "path": "/nix/store/afv3zwqxyw062...-go-1.21.8"
  }
},
"system": "x86_64-linux"
}
}
```

# 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

```
> nix-shell -p go_1_21

this path will be fetched
(39.16 MiB download, 204.47 MiB unpacked):
  /nix/store/k7chjapvryi....-go-1.21.8
copying path '/nix/store/k7chjapvryi....-go-1.21.8'
from 'https://cache.nixos.org'..
```

# Advantages

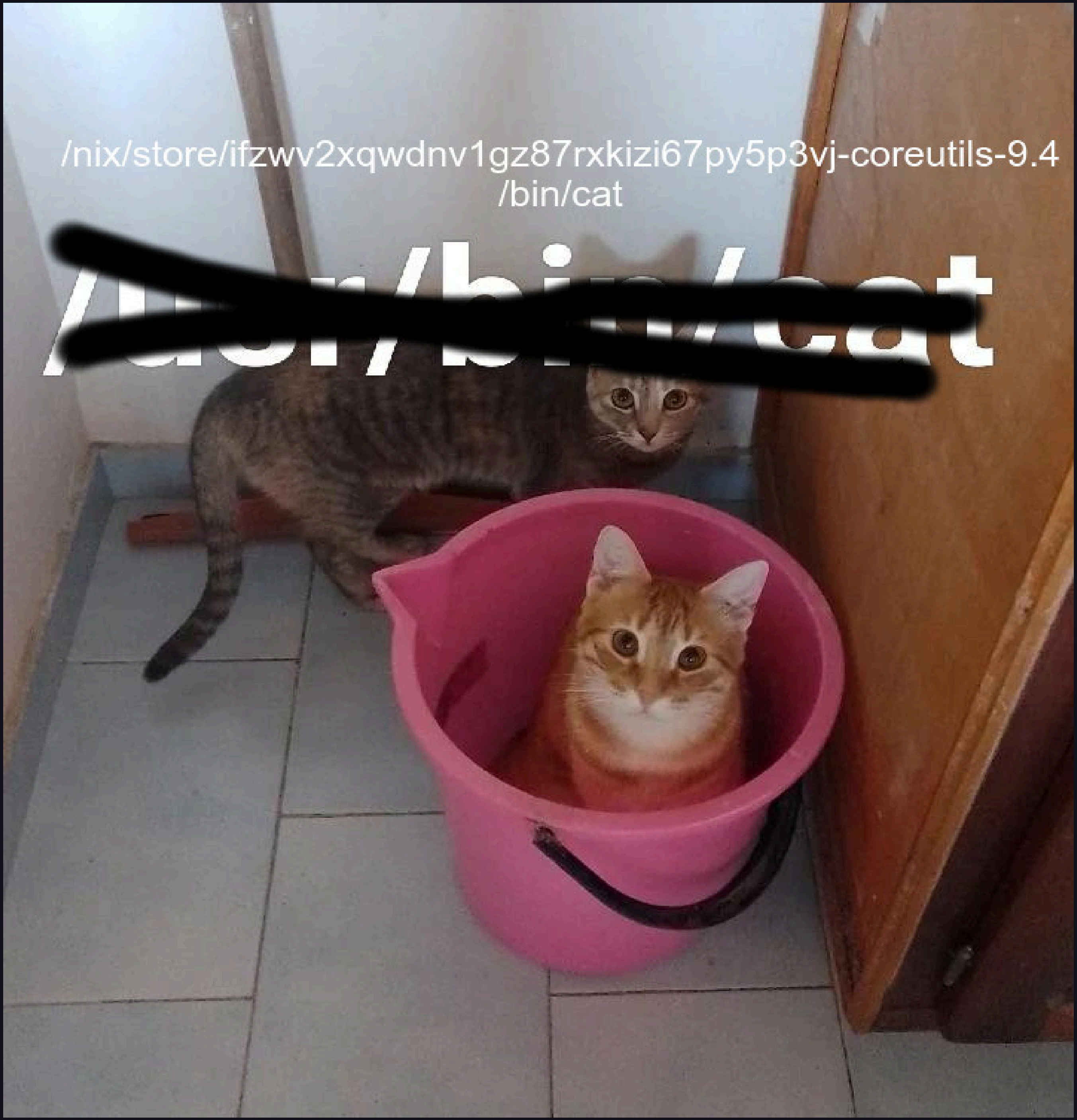
- Binary cache

```
> nix-shell -p go_1_21

this path will be fetched
(39.16 MiB download, 204.47 MiB unpacked):
  /nix/store/k7chjapvryi....-go-1.21.8
copying path '/nix/store/k7chjapvryi....-go-1.21.8'
from 'https://cache.nixos.org'..
```

# Advantages

- Atomic updates

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

lrwxrwxrwx - root  1 Jan  1970 , -> /nix/store/09irdfc2nq
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/09ir
lrwxrwxrwx - root  1 Jan  1970 applygnupgdefaults -> /nix
lrwxrwxrwx - root  1 Jan  1970 apropos -> /nix/store/09ir
lrwxrwxrwx - root  1 Jan  1970 atuin -> /nix/store/09irdf
```



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

~~/usr/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

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

# flake.lock

```
{
  "nodes": {
    "nixpkgs": {
      "locked": {
        "lastModified": 1668703332,
        // A SHA of the contents of the flake
        "narHash": "sha256-PW3vz3ODXaInogvp2I...=",
        // The GitHub org
        "owner": "NixOS",
        // The GitHub repo
        "repo": "nixpkgs",
        // The specific revision
        "rev": "de60d387a0e5737375ee61848872b1...",
```

# flake.lock

```
{
  "nodes": {
    "nixpkgs": {
      "locked": {
        "lastModified": 1668703332,
        // A SHA of the contents of the flake
        "narHash": "sha256-PW3vz3ODXaInogvp2I...=",
        // The GitHub org
        "owner": "NixOS",
        // The GitHub repo
        "repo": "nixpkgs",
        // The specific revision
        "rev": "de60d387a0e5737375ee61848872b1...",
```

# flake.lock

```
        "narHash": "sha256-PW3vz3ODXaInogvp2I...=",
        // The GitHub org
        "owner": "NixOS",
        // The GitHub repo
        "repo": "nixpkgs",
        // The specific revision
        "rev": "de60d387a0e5737375ee61848872b1...",
        // The type of input
        "type": "github"
      }
    },
    // Other inputs
  }
```

# 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”

# ci.nix

```nix
{
  pkgs,
  devPackages,
  ...
}:
pkgs.dockerTools.buildImage {
  name = "banterbus-dev";
  tag = "latest";
  copyToRoot = pkgs.buildEnv {
    name = "banterbus-dev";
    pathsToLink = ["/bin"];
    paths = with pkgs;
      [
```

# ci.nix

```nix
    pathsToLink = ["/bin"];
    paths = with pkgs;
      [
        coreutils
        gnugrep
        bash
        cacert.out
        curl
        git
      ]
      ++ devPackages;
  };
  config = {
```

# ci.nix

```
          gnugrep
          bash
          cacert.out
          curl
          git
        ]
        ++ devPackages;
    };
    config = {
      Env = [
        "NIX_PAGER=cat"
        # A user is required by nix
        # https://github.com/NixOS/nix/blob/9348f9291e5d
```

# flake.nix

```
# flake.nix
{
  description = "Development environment for BanterBus

  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs/nixos-unstable
    flake-utils.url = "github:numtide/flake-utils";
    pre-commit-hooks.url = "github:cachix/pre-commit-h
  };

  outputs = {
    self,
    nixpkgs,
```

# flake.nix

```
        go-task
        goreleaser
        sqlc
      ];
    in {
      packages.ci = pkgs.callPackage ./ci.nix {
        inherit pkgs;
        inherit devPackages;
      };
      devShells.default = pkgs.mkShell {
        packages = devPackages;
      };
    })
  );
```

# flake.nix

```nix
        sqlc
      ];
    in {
      packages.ci = pkgs.callPackage ./ci.nix {
        inherit pkgs;
        inherit devPackages;
      };
      devShells.default = pkgs.mkShell {
        packages = devPackages;
      };
    })
  );
}
```

# .gitlab-ci.yml

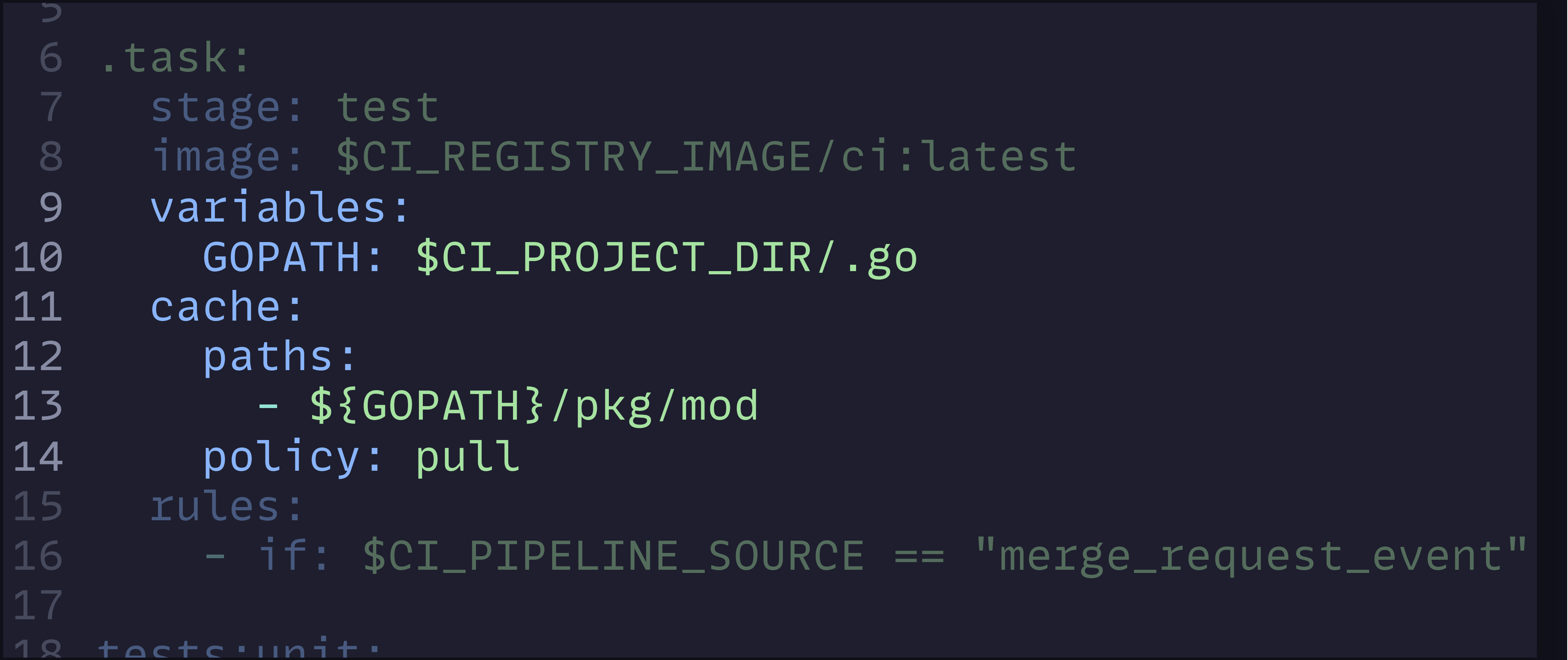
```yaml
publish:docker:ci:
  stage: pre
  variables:
    DOCKER_HOST: tcp://docker:2375
    DOCKER_DRIVER: overlay2
    DOCKER_TLS_CERTDIR: ""
    IMAGE: $CI_REGISTRY_IMAGE/ci
  rules:
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"
      changes:
        - "containers/ci.nix"
  services:
    - docker:25-dind
```

# .gitlab-ci.yml

```yaml
    changes:
      - "containers/ci.nix"
services:
  - docker:25-dind
script:
  - echo "experimental-features = nix-command flakes
  - nix-env -iA nixpkgs.docker
  - nix build .#ci
  - docker load < ./result
  - docker login -u $CI_REGISTRY_USER -p $CI_REGISTR
  - docker image tag banterbus-dev:latest $IMAGE:lat
  - docker push $CI_REGISTRY_IMAGE/ci:latest

```

```yaml

stages:
  - deps
  - test

.task:
  stage: test
  image: $CI_REGISTRY_IMAGE/ci:latest
  variables:
    GOPATH: $CI_PROJECT_DIR/.go
  cache:
    paths:
      - ${GOPATH}/pkg/mod
```

```
stages:
  - deps
  - test

.task:
  stage: test
  image: $CI_REGISTRY_IMAGE/ci:latest
  variables:
    GOPATH: $CI_PROJECT_DIR/.go
  cache:
    paths:
      - ${GOPATH}/pkg/mod
    policy: pull
```



```
.task:
  stage: test
  image: $CI_REGISTRY_IMAGE/ci:latest
  variables:
    GOPATH: $CI_PROJECT_DIR/.go
  cache:
    paths:
      - ${GOPATH}/pkg/mod
    policy: pull
  rules:
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"

tests:unit:
```

```yaml
    GOPATH: $CI_PROJECT_DIR/.go
  cache:
    paths:
      - ${GOPATH}/pkg/mod
    policy: pull
  rules:
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"

tests:unit:
  extends:
    - .task
  script:
    - go test -skip '^TestIntegration' ./internal/...
```

GET THAT THING OUT OF MY FACE!
NIX
chomp
NIX
chicken thoughts
imgflip.com

# 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/determin
    nix-installer/
- Use dev template
  - https://flakehub.com/f/the-nix-way/dev-templates
- Install direnv-nix
  - https://github.com/nix-community/nix-diren

```
curl --proto '=https' --tlsv1.2 -sSf -L \
 https://install.determinate.systems/nix | sh -s -- instal

nix flake init --template \
  "https://flakehub.com/f/the-nix-way/dev-templates/*#go"

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/

NIX All THE THINGS!

# 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=3hMIqxb
  - https://www.youtube.com/watch?app=desktop&v=TsZte_9GfPE&si=osBujLY3py

https://haseebmajid.dev/slides/london-gophers-reproducible-envs-with-nix/

# References & Thanks

- GIFs made with vhs
- Photos editted with pixlr
- All my friends who took time to give me feedback on this talk
- Some memes from https://github.com/gytis-ivaskevicius/high-quality-nix-content