

# **Reproducible & Ephemeral Development Environments with Nix**

# 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**



imgflip.com

Credit



# What is Nix?

- Nix is a declarative package manager
- Nixlang the programming language that powers Nix
- NixOS: A Linux distribution that can be configured using Nixlang

# Declarative

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





Normal  
conversation

Nix  
BTW I use a ~~n~~

Friends

Me



# What is the problem?

- `/usr/local/bin/golangci-lint`
  - Dependencies
  - Configuration flags & Env Vars
  - Two versions of this package

There are various other packaging solutions that try to fix these issues:

- Snap/Flatpak
- asdf
- virtualenv

# Summary

- We want reproducible and ephemeral environments
- Nix is an ecosystem of tools
- Our current packaging system all have various different flaws



# Golang

- Tooling to aid development
- Use the same tool
- Same versions running on CI

# tools.go

```
// +build tools
```

```
package main
```

```
import (
```

```
    _ "github.com/golangci/golangci-lint/cmd/golangci-lint"  
    _ "github.com/goreleaser/goreleaser"  
    _ "github.com/spf13/cobra/cobra"
```

```
)
```

```
cat tools.go | grep _ | awk -F'"' '{print $2}' \  
| xargs -tI % go install %
```



# Creating our first dev environment

```
> 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
```

# flake.nix

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

# flake.nix

```
5     nixpkgs.url = "github:nixos/nixpkgs/nixos-unstable",
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         pkgs = nixpkgs.legacyPackages.${system};
```

# flake.nix

```
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; [
```

# flake.nix

```
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
```

# flake.nix

```
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
25          go-junit-report
26          gocover-cobertura
```

# flake.nix

```
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
25             go-junit-report
26             gocover-cobertura
27             go-task
```

# flake.nix

```
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             docker-compose
31         ];
32     };
```



```
> which golangci-lint
```

```
> nix develop
```

```
> which golangci-lint
```

```
/nix/store/kc58bqdmjdc6mfilih5pywprs7r7lrxw-golangci-lint-1.50
```

# Summary

- Leverage Flakes devshells for installing packages
  - Load into shell: `nix develop`
- Make sure each dev gets the same package
  - Update lock file: `nix flake update`

# direnv

```
# .envrc  
  
use flake
```

Direnv Code “use flake”

# Usage

```
1 > cd example
2 > which golangci-lint
3
4 > direnv allow
5 direnv: loading ~/projects/example/.envrc
6 direnv: using flake
7 direnv: nix-direnv: using cached dev shell
8 direnv: export +AR +AS +CC +CONFIG_SHELL +CXX +GOTOOLDIR +
9
10 > example on ↵ main via ❄ impure (nix-shell-env) took 5s
11 > which golangci-lint
12 /nix/store/kc58bqdmjdc6mfilih5pywprs7r7lxxw-golangci-lint-
13
```

# Usage

```
1 > cd example
2 > which golangci-lint
3
4 > direnv allow
5 direnv: loading ~/projects/example/.envrc
6 direnv: using flake
7 direnv: nix-direnv: using cached dev shell
8 direnv: export +AR +AS +CC +CONFIG_SHELL +CXX +GOTOOLDIR +
9
10 > example on ↵ main via ❄ impure (nix-shell-env) took 5s
11 > which golangci-lint
12 /nix/store/kc58bqdmjdc6mfilih5pywprs7r7lxxw-golangci-lint-
13
```

# Usage

```
4 > direnv allow
5 direnv: loading ~/projects/example/.envrc
6 direnv: using flake
7 direnv: nix-direnv: using cached dev shell
8 direnv: export +AR +AS +CC +CONFIG_SHELL +CXX +GOTOOLDIR +
9
10 > example on ↵ main via * impure (nix-shell-env) took 5s
11 > which golangci-lint
12 /nix/store/kc58bqdmjdc6mfilih5pywprs7r7lrxw-golangci-lint-
13
14 > cd ..
15 direnv: unloading
16 > which golangci-lint
```

# Usage

```
4 > direnv allow
5 direnv: loading ~/projects/example/.envrc
6 direnv: using flake
7 direnv: nix-direnv: using cached dev shell
8 direnv: export +AR +AS +CC +CONFIG_SHELL +CXX +GOTOOLDIR +
9
10 > example on ↵ main via * impure (nix-shell-env) took 5s
11 > which golangci-lint
12 /nix/store/kc58bqdmjdc6mfilih5pywprs7r7lrxw-golangci-lint-
13
14 > cd ..
15 direnv: unloading
16 > 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 proje
3
4   inputs = {
5     nixpkgs.url = "github:NixOS/nixpkgs/nixos-unstable";
6     flake-utils.url = "github:numtide/flake-utils";
7     pre-commit-hooks.url = "github:cachix/pre-commit-hooks
8   };
9
10  outputs = {
11    self,
12    nixpkgs,
13    flake-utils,
```

# pre-commit

```
10 } (
17 flake-utils.lib.eachDefaultSystem
18 (system: let
19     pkgs = nixpkgs.legacyPackages.${system};
20     pre-commit-check = pre-commit-hooks.lib.${system}.ru
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             gotools
34             go-junit-report
35             gocover-cobertura
```

# Summary

- We can use `direnv` to further reduce cognitive load
- Can even use flakes from remote git repository
  - Share between multiple projects
- We can manage pre-commit in Nix as well
  - However using an abstraction

# How does Nix work?

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

[mk.shell docs](#)

# How does Nix work?

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

[mk.shell docs](#)

# How does Nix work?

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

[mk.shell docs](#)

```
1 { buildGoModule, fetchFromGitHub, lib, installShellFiles }
2
3 buildGoModule rec {
4   pname = "golangci-lint";
5   version = "1.57.2";
6
7   src = fetchFromGitHub {
8     owner = "golangci";
9     repo = "golangci-lint";
10    rev = "v${version}";
11    hash = "sha256-d3U56fRIyntj/uKT0HuKFv0Zqh+6VtzYrbKDjcK";
12  };
13
```

## golangci-lint Nix expression



```
stdenv.mkDerivation
```

buildGoModule function

# 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/zg65r8ys8y58651cwmmmybrq5gn30n1az-go-1.21.8.drv
```

```
1 nix derivation show nixpkgs#go_1_21
2 {
3   "/nix/store/gccilxhvxbhm79hkmcczn0vxbb7d120-go-1.21.8.dr
4   "args": [
5     "-e",
6     "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-default-bu
7   ],
8   "builder": "/nix/store/5lr5n3qa4day8l1ivbwlcby2nknczqkq-b
9   "env": {
10    "CGO_ENABLED": "1",
11    "G0386": "softfloat",
12    "GOARCH": "amd64",
13    "GOARM": "",
```

```
2 {
3  "/nix/store/gccilxhvxbhm79hkmcczn0vxbb7d120-go-1.21.8.dr
4  "args": [
5    "-e",
6    "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-default-bu
7  ],
8  "builder": "/nix/store/5lr5n3qa4day8l1ivbwlcby2nknczqkq-b
9  "env": {
10   "CGO_ENABLED": "1",
11   "G0386": "softfloat",
12   "GOARCH": "amd64",
13   "GOARM": "",
14   "GOHOSTARCH": "amd64",
```



```
47   "src": "/nix/store/p81s0316n7snx40fwkhda4p5jczf2pff-go1
48   "stdenv": "/nix/store/c8dj731bkcdzhgrpawhc8qvdlg1s4xfjv-
49   "strictDeps": "1",
50   "system": "x86_64-linux",
51   "version": "1.21.8"
52 },
53 "inputDrvs": {
54   "/nix/store/17gdfyx2nzzcbhh8c2fm6zm8973nnrsd-stdenv-lin
55     "dynamicOutputs": {},
56     "outputs": [
57       "out"
58     ]
59   },
```

```
105     "/nix/store/m88mg4d43hwkkbip6dha7p858c0vm5c1-go_no_vend
106     "/nix/store/v6x3cs394jgqfbi0a42pam708flxaphh-default-bu
107     "/nix/store/x48d0s4gns4jrck6qkwrpqn7nh9ygp6-remove-too
108 ],
109 "name": "go-1.21.8",
110 "outputs": {
111     "out": {
112         "path": "/nix/store/afv3zwqxyw062vg2j220658jq0g1yadv-
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 (39.16 MiB download, 204.47 MiB u
4   /nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-go-1.21.8
5 copying path '/nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-g
6 https://cache.nixos.org' ..
```

# Advantages

- Binary cache

```
1 > nix-shell -p go_1_21
2
3 this path will be fetched (39.16 MiB download, 204.47 MiB u
4   /nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-go-1.21.8
5 copying path '/nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-g
6 https://cache.nixos.org' ..
```

# Advantages

- Forces us to make our dependency tree explicit

```
> nix-store -q --tree /nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-go-1.21.8
/nix/store/k7chjapvryivjixp01iil9z0z7yzg7z4-go-1.21.8
├── /nix/store/7vvggrs9367d3g9f123vjfyvsv10gb0r-mailcap-2.1.50
├── /nix/store/a1s263pmsci9zykm5xcdf7x9rv26w6d5-bash-5.2p26
│   ├── /nix/store/ddwyrxif62r8n6xclvskjyy6szdhvj60-glibc-2.39
│   │   ├── /nix/store/rxganm4ibf31qngal3j3psp20mak37yy-xgcc-10.3.0
│   │   ├── /nix/store/s32cldbh9pfzd9z82izi12mdlrw0yf8q-libidn2-2.3.4
│   │   │   ├── /nix/store/7n0mbqydcipkpbxm24fab066lxk68aqk-1.3.4
│   │   │   │   └── /nix/store/7n0mbqydcipkpbxm24fab066lxk68aqk-1.3.4
│   │   │   └── /nix/store/s32cldbh9pfzd9z82izi12mdlrw0yf8q-1.3.4
│   │   └── /nix/store/ddwyrxif62r8n6xclvskjyy6szdhvj60-glibc
│   └── /nix/store/a1s263pmsci9zykm5xcdf7x9rv26w6d5-bash-5.2p26
```

# Advantages

- Symlink
- Atomic updates

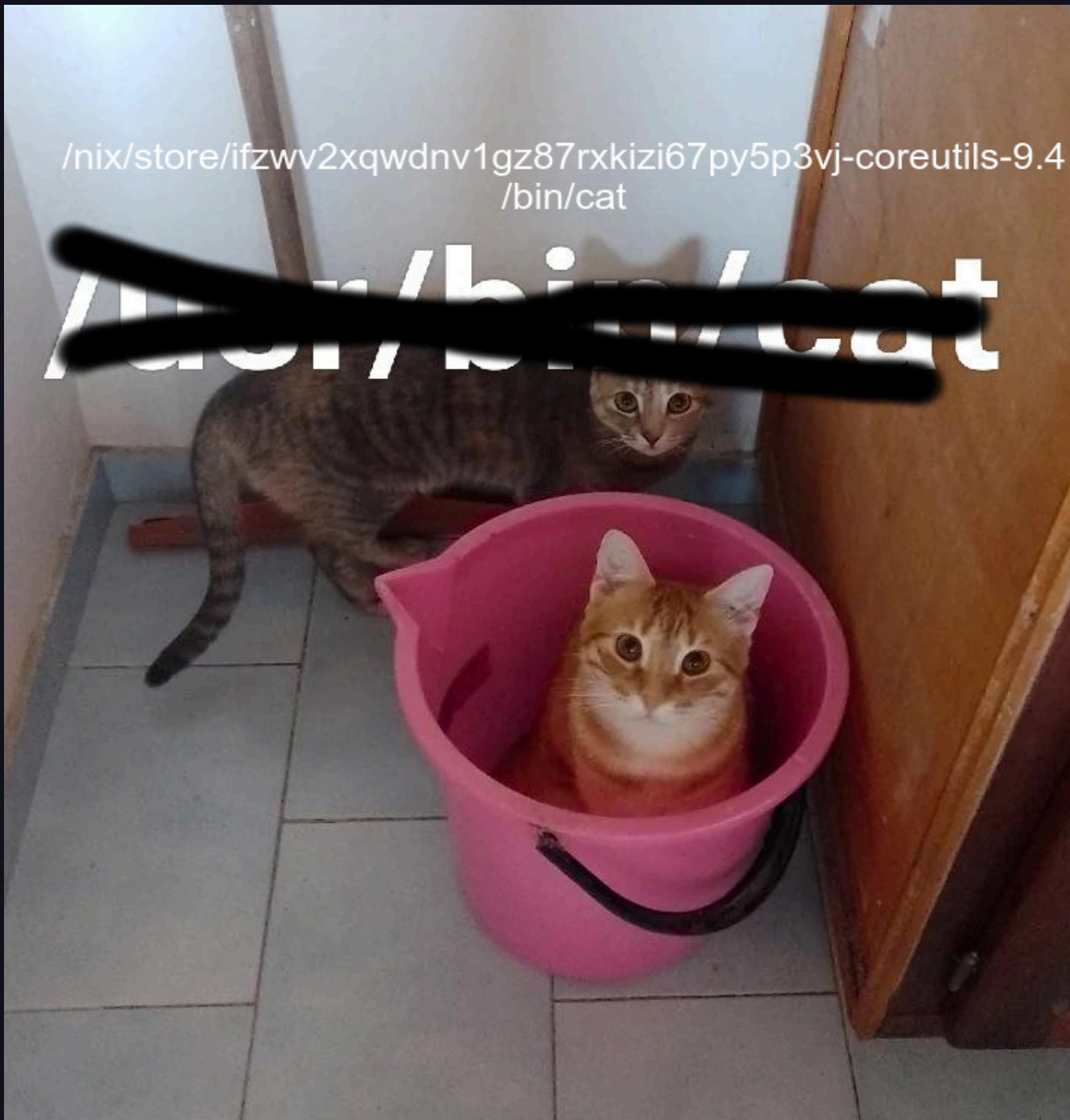
```
> ls ~/.nix-profile/bin
```

```
lrwxrwxrwx - root 1 Jan 1970 , -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 accessdb -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 addgnupghome -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 ag -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 animate -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 applygnupgdefaults -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 apropos -> /nix/store/09irdfc2nqr6p1l  
lrwxrwxrwx - root 1 Jan 1970 atuin -> /nix/store/09irdfc2nqr6p1l
```



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

~~/usr/bin/cat~~



# Summary

- Nix derivations allow us to have immutable packages
  - Require us to make dependencies explicit
- What if package is not in nixpkgs



# Nix Flakes

- Generates a lock file
- Improve reproducibility
- Use other git repo as inputs
- Define some structure

# 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-PW3vz30DXaInogvp2IQyDG9InwmGlf0
8         // The GitHub org
9         "owner": "NixOS",
10        // The GitHub repo
11        "repo": "nixpkgs",
12        // The specific revision
13        "rev": "de60d387a0e5737375ee61848872b1c8353f945e",
```

# flake.lock

```
7     "narHash": "sha256-PW3vz30DXaInogvp2IQyDG9InwmG1f0
8     // The GitHub org
9     "owner": "NixOS",
10    // The GitHub repo
11    "repo": "nixpkgs",
12    // The specific revision
13    "rev": "de60d387a0e5737375ee61848872b1c8353f945e",
14    // The type of input
15    "type": "github"
16  }
17 },
18 // Other inputs
19 }
```

# Summary

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

# CI

- Use same versions as local
- Leverage Nix “cachability”
  - Packages share dependencies

# Gitlab CI

```
1 .nix:  
2   image: registry.gitlab.com/cynerd/gitlab-ci-nix  
3   tags:  
4     - docker  
5   cache:  
6     key: "nix"  
7     paths:  
8       - ".nix-cache"  
9   before_script:  
10     - gitlab-ci-nix-cache-before  
11   after_script:  
12     - gitlab-ci-nix-cache-after
```

- Taken from <https://gitlab.com/Cynerd/gitlab-ci-nix>

# Gitlab CI

```
1 .nix:  
2   image: registry.gitlab.com/cynerd/gitlab-ci-nix  
3   tags:  
4     - docker  
5   cache:  
6     key: "nix"  
7     paths:  
8       - ".nix-cache"  
9   before_script:  
10     - gitlab-ci-nix-cache-before  
11   after_script:  
12     - gitlab-ci-nix-cache-after
```

- Taken from <https://gitlab.com/Cynerd/gitlab-ci-nix>




```
1 include:
2   - 'https://gitlab.com/Cynerd/gitlab-ci-nix/-/raw/master/'
3
4 stages:
5   - pre
6   - test
7   - release
8
9 install:deps:
10  stage: pre
11  extends: .nix
12  only:
13    - merge_request
```




```
6   - test
7   - release
8
9   install:deps:
10  stage: pre
11  extends: .nix
12  only:
13    - merge_request
14  script:
15    - nix develop
16
17  lint:
18    stage: test
```

```
14  script:
15    - nix develop
16
17  lint:
18    stage: test
19    extends: .nix
20    only:
21      - merge_request
22    script:
23      - nix develop -c golangci-lint run
24
25  tests:
26    stage: test
```

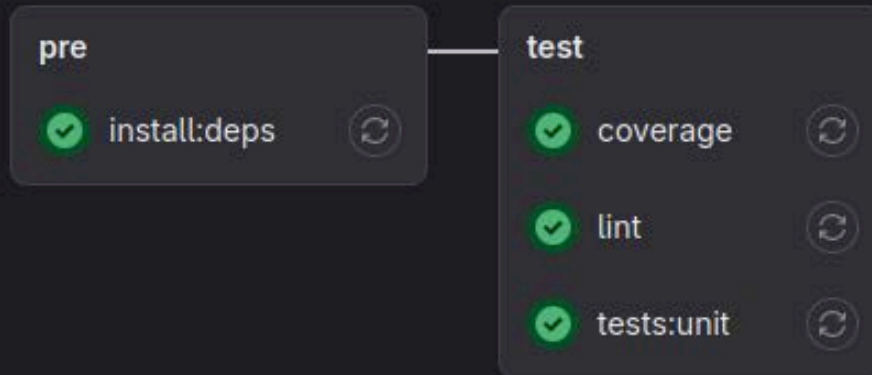
# chore: update

✔ Passed Haseeb Majid created pipeline for commit `5323524b`  2 weeks ago, finished 2 weeks ago

Related merge request [!22](#) to merge [full-text-search](#)

latest merge request  4 jobs  11.5  5 minutes 43 seconds, queued for 0 seconds

**Pipeline** Needs Jobs 4 Tests 18



# CI Logs

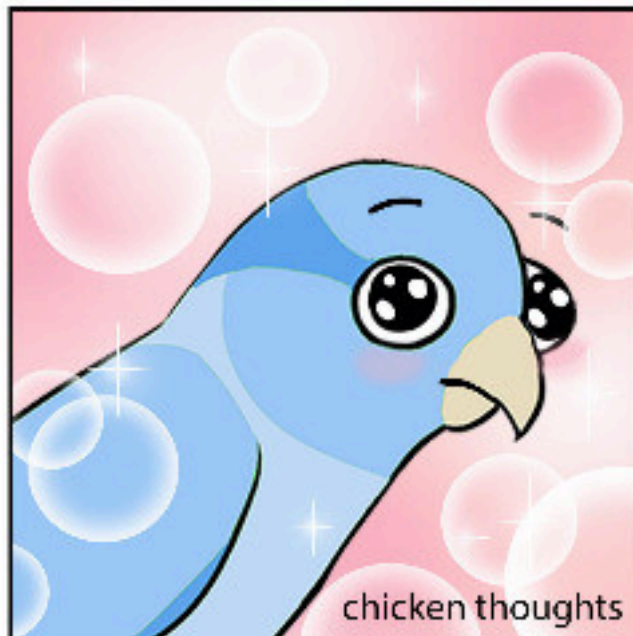
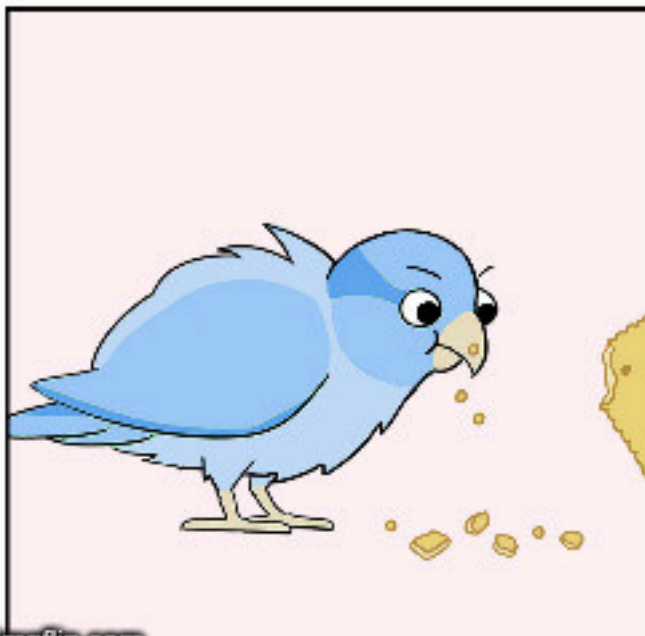
```
1 copying path '/nix/store/97j7mjdvbxb59jnkr97qmh1kk97mn1y8-s
2 from 'file:///builds/hmajid2301/optinix/.nix-cache' ...
3 copying path '/nix/store/yj1wxm9hh8610iyzqnz75kvs6x18j3my-s
4 copying path '/nix/store/gig8j85kj7ybjy3ksn6k3aich8j2k59y-s
5 copying path '/nix/store/lrq619bzwr9z810hac96cg7r5ga6fkq4-s
6 copying path '/nix/store/3ph17qxbvxvdg5b053q81mgqwf95gsiv-s
7 copying path '/nix/store/pgid9c9xfcrbqx2giry0an0bi0df7s5c-s
8 copying path '/nix/store/g16z4fs1mrbkxc4x6wm8xbrh13nc7aw4-s
```



GET THAT THING  
OUT OF MY FACE!



chomp



# Why not Docker?

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



# 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!**



# Slides

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

# My Links

- My Dotfiles Configured Using Nix
- Project using Nix Development Env
- Raspberry PI Cluster Deployed using Nix

# 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

- Better Nix Installer:

<https://determinate.systems/posts/determinate-nix-installer/>

```
> nixmei
```

- 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=I17oRkhg>

- 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://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

