

buildbot-nix

Richard Brežák

2024

There is no good open-source Nix CI/CD

1. some might say, we have no ...
2. we want something...

- easy to deploy

1. one temptation is to write everything from scratch. Multiple projects tried this and none of them succeeded. We needed something we can make work today, and iterate over.

- easy to deploy
- works out of the box

1. one temptation is to write everything from scratch. Multiple projects tried this and none of them succeeded. We needed something we can make work today, and iterate over.

- easy to deploy
- works out of the box
- natively integrates with GitHub and other forges

- easy to deploy
- works out of the box
- natively integrates with GitHub and other forges

1. one temptation is to write everything from scratch. Multiple projects tried this and none of them succeeded. We needed something we can make work today, and iterate over.

- easy to deploy
- works out of the box
- natively integrates with GitHub and other forges
- not a from-scratch project, something that works NOW

- easy to deploy
- works out of the box
- natively integrates with GitHub and other forges
- **not a from-scratch project, something that works NOW**

1. one temptation is to write everything from scratch. Multiple projects tried this and none of them succeeded. We needed something we can make work today, and iterate over.

2024-10-24

buildbot-nix

└ Hydra

Hydra

1. take Hydra, (then flip)

- heavy-weight, keeps track of every store path

└ Hydra

- heavy-weight, keeps track of every store path

1. take Hydra, (then flip)

- heavy-weight, keeps track of every store path
- `nixpkgs` scale CI for `nixpkgs` scale problems

└ Hydra

- heavy-weight, keeps track of every store path
- **nixpkgs** scale CI for **nixpkgs** scale problems

1. take Hydra, (then flip)

- heavy-weight, keeps track of every store path
- `nixpkgs` scale CI for `nixpkgs` scale problems
- doesn't integrate well with GitHub/Gitea

2024-10-24

buildbot-nix

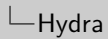
└ Hydra

Hydra

- heavy-weight, keeps track of every store path
- skipper scale CI for skipper scale problems
- doesn't integrate well with GitHub/Gitea

1. take Hydra, (then flip)

- heavy-weight, keeps track of every store path
- `nixpkgs` scale CI for `nixpkgs` scale problems
- doesn't integrate well with GitHub/Gitea, especially PRs



- heavy-weight, keeps track of every store path
- works on scale CI for workloads with scale problems
- doesn't integrate well with GitHub/Gitea, especially PRs

1. take Hydra, (then flip)

I attempted to run Hydra

I've succeeded and I'm still
running it



└─ GitHub/Gitea Actions

1. take Actions, (then flip)

- not Nix native

2024-10-24

buildbot-nix

└─ GitHub/Gitea Actions



1. take Actions, (then flip)

- not Nix native
- cannot use the build sandbox generally

└─ GitHub/Gitea Actions

GitHub/Gitea Actions

- not Nix native
- cannot use the build sandbox generally

1. take Actions, (then flip)

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful

└─ GitHub/Gitea Actions

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful

1. take Actions, (then flip)

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful
- build logs become a jumbled mess *OR* may repeat builds

└─ GitHub/Gitea Actions

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful
- build logs become a jumbled mess *OR* may repeat builds

1. take Actions, (then flip)

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful
- build logs become a jumbled mess *OR* may repeat builds
- if done naively, you will get an OOM

└─ GitHub/Gitea Actions

- not Nix native
- cannot use the build sandbox generally
- caching is hard to get right or wasteful
- build logs become a jumbled mess, CI may repeat builds
- **if done naively, you will get an OOM**

1. take Actions, (then flip)

PROGRAMS

HOW ~~STANDARDS~~ PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
2 THERE ARE
~~11~~ COMPETING
~~STANDARDS:~~
PROGRAMS

2 ~~11~~! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL ~~STANDARDS~~ PROGRAMS
THAT COVERS EVERYONE'S
USE CASES. YEAH!



SOON:
SITUATION:
3 THERE ARE
~~15~~ COMPETING
~~STANDARDS:~~
PROGRAMS



1. so we wrote one

buildbot-nix

2024-10-24

buildbot-nix

buildbot-nix

1. called buildbot-nix

I know of buildbot-nix

I've never heard of it

- CI/CD framework

2024-10-24

buildbot-nix

└─ buildbot?

buildbot?

- CI/CD framework

1. what's buildbot?

buildbot?

- CI/CD framework
- written in Python

2024-10-24

buildbot-nix

└─ buildbot?

buildbot?

- CI/CD framework
- written in Python

1. what's buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects



OpenZFS



2024-10-24

buildbot-nix

└─ buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects



OpenZFS



1. what's buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects

essentially...



Open**Z**FS



2024-10-24

buildbot-nix

└─ buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects

essentially...



The image shows a collection of logos for various technologies: OpenZFS (top left), Python (top right), a purple eye-like logo (middle left), and a red and orange logo (middle right). There is also a small logo of a bird or eagle at the bottom right.

1. what's buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects

essentially...

Jenkins but in Python



Open**Z**FS



2024-10-24

buildbot-nix

└─ buildbot?

buildbot?

- CI/CD framework
- written in Python
- used by many projects

essentially...

Jenkins but in Python



The slide features a dark header with the text 'buildbot?'. Below the header, there is a list of three bullet points: '• CI/CD framework', '• written in Python', and '• used by many projects'. Underneath the list, the text 'essentially...' is followed by 'Jenkins but in Python'. To the right of the text, there are three logos: OpenZFS (a blue square with a white grid), Python (the blue and yellow snake logo), and Jenkins (the red and white robot head logo).

1. what's buildbot?

Python
Anonymous ▾

Branch (all) ▾ Results (all) ▾

Builder Tags

aarch64 Android 3.13	3.13	stable	3m-3	(115)
aarch64 Android 3.x	3.x	stable	3m-3	(372)
aarch64 CentOS LTO + PGO 3.13	3.13	unstable	3x ppp nondebug test-2	(300)
aarch64 CentOS LTO + PGO 3.x	3.x	unstable	3x ppp nondebug test-2	(647)
aarch64 CentOS LTO 3.13	3.13	unstable	3x nondebug test-2	(299)
aarch64 CentOS LTO 3.x	3.x	unstable	3x nondebug test-2	(692)
aarch64 CentOS Refleaks 3.13	3.13	unstable	refleak test-2	(105)
aarch64 CentOS Refleaks 3.x	3.x	unstable	refleak test-2	(264)
aarch64 Fedora Rawhide 3.13	3.13	unstable	test-2	(126)
aarch64 Fedora Rawhide Clang 3.13	3.13	unstable	clang test-2	(127)
aarch64 Fedora Rawhide Clang	3.13	unstable	clang test-2	(127)

GitHub Pull Request #125709 (15 commits) 🔗

gh-125259: Fix error notes removal in enum initialization (GH-125647) 🔗

[3.13] gh-125710: [Enum] fix hashable->nonhashable comparisons for member values (GH-125735) (GH-125851) 🔗

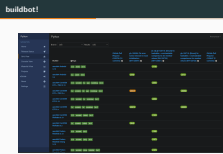
gh-125710: [Enum] fix hashable->nonhashable comparisons for member values (GH-125735) 🔗

GitHub Pull Request #125847 (2 commits) 🔗

2024-10-24

buildbot-nix

└─ buildbot!



1. used by Python for example

- CI/CD for Nix

- CI/CD for Nix
- plugin for buildbot

- CI/CD for Nix
- plugin for buildbot
- written in Python

- CI/CD for Nix
- plugin for buildbot
- written in Python
- can run on PRs safely – configuration is server-side

- CI/CD for Nix
- plugin for buildbot
- written in Python
- can run on PRs safely – configuration is server-side

Aiming for smaller projects

- CI/CD for nix
- plugin for buildbot
- written in Python



• can run on PRs safely – configuration is user-editable

Aiming for smaller projects



```
1  {
2  inputs = { ... };
3  outputs = { ... }:
4  {
5      checks.${system} = {
6          treefmt = treefmt...check;
7          packageTest = runCommand "package-test" { }
8              ↪ "exit 1";
9          };
10     formatter.${system} = treefmt...wrapper;
11 }
```





(not true Nix)

```
1 {
2   inputs = { ... };
3   outputs = { ... };
4   {
5     checks.${system} = {
6       treefmt = treefmt...check;
7       packageTest = runCommand "package-test" { }
8         -- "exit 1";
9     };
10    formatter.${system} = treefmt...wrapper;
11  };
12 }
```

1. what if you had a flake, like that, and wanted to run CI on it? well, run all the checks in CI!

Some checks were not successful ✕

1 failing and 3 successful checks

- ✕  **buildbot/nix-build** [github:nc-mirror-buildbot/nixcon2024-ex-01#checks.x86_64-linux.packageTest](#) - Build done. [Details](#)
- ✓  **buildbot/nix-build** - Build done. [Details](#)
- ✓  **buildbot/nix-build** [github:nc-mirror-buildbot/nixcon2024-ex-01#checks.x86_64-linux.treefmt](#) - Build done. [Details](#)
- ✓  **buildbot/nix-build** [github:nc-mirror-buildbot/nixcon2024-ex-01#checks.x86_64-linux.eval](#) - Build done. [Details](#)

(find the bug)

nc-mirror-buildbot/nixcon2024-ex-01/nix-eval/5 11 s failed finished (failure) **FAILURE**

0 worker_preparation 1 s worker buildbot-016 ready


1 git 2 s update

2 evaluate flake 4 s 'nix-eval-jobs --workers ...'

3 build flake0 builds,2 pending builds 4 s finished (failure)

None github:nc-mirror-buildbot/nixcon2024-ex-01#checks.x86_64-linux.packageTest/4 2 s failed 'nix build ...' (failure) **FAILURE**

None github:nc-mirror-buildbot/nixcon2024-ex-01#checks.x86_64-linux.treefmt/5 2 s build successful **SUCCESS**

scheduler [view all 13 lines](#)  Raw

4 Cleanup drv paths 0 s 'rm -rf ...'

How do I get this?

Easily!

```
1  services.buildbot-nix.master = {
2    enable = true;
3    jobReportLimit = null;
4    domain = "buildbot.example.org";
5    workersFile = "/secret/workers.json";
6    admins = ["your-nickname"];
7    authBackend = "github";
8    github = {
9      enable = true;
10     webhookSecretFile = "/secret/github_webhook_secret";
11     oauthId = "...";
12     oauthSecretFile = "/secret/github_oauth_secret";
13     topic = "build-with-buildbot";
14     authType.app = {
15       secretKeyFile =
16         ↪ "/secret/github_app_private_key.pem";
17       id = 881425;
18     };
19   };
20 };
21 };
```



```
1  services.buildbot-nix.master = {
2    enable = true;
3    jobReportLimit = null;
4    domain = "buildbot.example.org";
5    workersFile = "/secret/workers.json";
6    admins = ["your-nickname"];
7    authBackend = "github";
8    github = {
9      enable = true;
10     webhookSecretFile = "/secret/github_webhook_secret";
11     oauthId = "...";
12     oauthSecretFile = "/secret/github_oauth_secret";
13     topic = "build-with-buildbot";
14     authType.app = {
15       secretKeyFile =
16         ↪ "/secret/github_app_private_key.pem";
17       id = 881425;
18     };
19   };
20 };
21 };
```

```
1  services.buildbot-nix.master = {
2    enable = true;
3    jobReportLimit = null;
4    domain = "buildbot.example.org";
5    workersFile = "/secret/workers.json";
6    admins = ["your-nickname"];
7    authBackend = "github";
8    github = {
9      enable = true;
10     webhookSecretFile = "/secret/github_webhook_secret";
11     oauthId = "...";
12     oauthSecretFile = "/secret/github_oauth_secret";
13     topic = "build-with-buildbot";
14     authType.app = {
15       secretKeyFile =
16         ↪ "/secret/github_app_private_key.pem";
17       id = 881425;
18     };
19   };
20 };
21 };
```

```
1  services.buildbot-nix.master = {
2    enable = true;
3    jobReportLimit = null;
4    domain = "buildbot.example.org";
5    workersFile = "/secret/workers.json";
6    admins = ["your-nickname"];
7    authBackend = "github";
8    github = {
9      enable = true;
10     webhookSecretFile = "/secret/github webhook secret";
11     oauthId = "...";
12     oauthSecretFile = "/secret/github oauth secret";
13     topic = "build-with-buildbot";
14     authType.app = {
15       secretKeyFile =
16         ↪ "/secret/github app private key.pem";
17       id = 881425;
18     };
19   };
20 };
21 }
```

```
1  services.buildbot-nix.master = {
2    enable = true;
3    jobReportLimit = null;
4    domain = "buildbot.example.org";
5    workersFile = "/secret/workers.json";
6    admins = ["your-nickname"];
7    authBackend = "github";
8    github = {
9      enable = true;
10     webhookSecretFile = "/secret/github_webhook_secret";
11     oauthId = "...";
12     oauthSecretFile = "/secret/github_oauth_secret";
13     topic = "build-with-buildbot";
14     authType.app = {
15       secretKeyFile =
16         ↪ "/secret/github_app_private_key.pem";
17       id = 881425;
18     };
19   };
20 };
21 };
```

and you have buildbot running

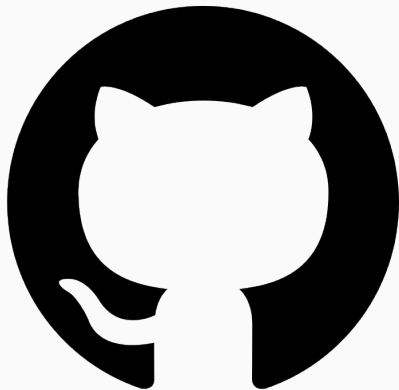
(after configuring `services.nginx.virtualHosts`)

What are the features?

Gitea/Forgejo



GitHub



Cachix

All [github:antifuchs/home#checks.nixos.gloria/346](https://github.com/antifuchs/home#checks.nixos.gloria/346) 10 s build successful **SUCCESS**

0 worker_preparation 1 s worker gloria-014 ready

1 Build flake attr 5 s 'nix build ...'

2 Upload cachix 4 s 'cachix push ...'

stdio [view all 39 lines](#) Raw

```
29 Pushing 4 paths (1985 are already present) using zstd to cache boinkornet-home 🕒
30
31 Pushing /nix/store/                r8-append-initrd-secrets (2.57 KiB)
32 Pushing /nix/store/                oq-etc-nix-registry.json (18.87 KiB)
33 Pushing /nix/store/                dd-etc (72.41 KiB)
34 Pushing /nix/store/                zk-nixos-system-gloria-24.11.2024
35
36 All done.
37 program finished with exit code 0
38 elapsedTime=4.080435
```

3 Register acroot 0 s finished

post build steps

```
# 0 worker_preparation 1 s worker buildbot-nix-0-001 ready
# 1 Build flake attr 1 s 'nix build ...'
# 2 sign-and-upload 8 s '/nix/store/f0xK59zxikmr9dwhxj0b2mlypnxdddpsi-postbuildstepper-alpha/bin/postbuildstepper'

stdio view all 73 lines [Download] [Raw]

57 [2024-10-21T15:19:14Z INFO postbuildstepper::business] owners ("steveej") are trusted! proceeding.
58 [2024-10-21T15:19:14Z DEBUG postbuildstepper::business] [aarch64-darwin.hello/include]: 'aarch64-[^\.]+\,hello' matched 'aarch64-darwin.hello': true
59 [2024-10-21T15:19:14Z DEBUG postbuildstepper::business] [aarch64-darwin.hello/include]: '.*tests.*' matched 'aarch64-darwin.hello': false
60 [2024-10-21T15:19:14Z DEBUG postbuildstepper::business] [aarch64-darwin.hello]: include: true, exclude: false, conclusion: true
61 added 1 signatures
62 [2024-10-21T15:19:14Z INFO postbuildstepper] successfully signed store path /nix/store/0bc591psx1x8s9n2nc5h96cd4yhhrh28-hello-2.12.1
63 querying info about '/nix/store/0bc591psx1x8s9n2nc5h96cd4yhhrh28-hello-2.12.1' on 's3://cache.holo.host'...
64 copying 1 paths...
65 copying path '/nix/store/0bc591psx1x8s9n2nc5h96cd4yhhrh28-hello-2.12.1' to 's3://cache.holo.host'...
66 copying path '/nix/store/0bc591psx1x8s9n2nc5h96cd4yhhrh28-hello-2.12.1' (121144 bytes, compressed 79.0% in 26 ms) to binary cache
67 uploaded 's3://cache.holo.host/0bc591psx1x8s9n2nc5h96cd4yhhrh28.ls' (155 bytes) in 359 ms
68 uploaded 's3://cache.holo.host/nar/041bbvhfdk351kn15952szc6t8nnp2yqhw5b1mws8bjwvsh721.nar.xz' (25432 bytes) in 343 ms
69 uploaded 's3://cache.holo.host/0bc591psx1x8s9n2nc5h96cd4yhhrh28.nar.info' (620 bytes) in 261 ms
70 [2024-10-21T15:19:22Z INFO postbuildstepper] successfully pushed store path /nix/store/0bc591psx1x8s9n2nc5h96cd4yhhrh28-hello-2.12.1

# 3 Register gcroot 0 s finished
# 4 Delete temporary gcroots 0 s 'rm -f ...'
# 5 Update build output 0 s finished (skipped)
```

per-repository settings (almost)

```
1 attribute = "checks"  
2 lock_file = "flake.lock"
```











per-repository effects with secrets (soon)

5 ▶ Evaluate effects

3 s 'buildbot-effects --rev ...'

6 ✔ Buildbot effect *5 builds, 0 pending builds*

22 s triggered antifuchs-home-run-effect, antifuchs-home-run-effect, antifuchs-home-run-effect, antifuchs-home-run-effect, antifuchs-home-run-effect, 5 successes

  None	deploy-fly-boinkornet-grapevine/104	21 s build successful	SUCCESS
  None	deploy-fly-boinkornet-heisenbridge/98	12 s build successful	SUCCESS
  None	deploy-fly-boinkornet-linkding/92	12 s build successful	SUCCESS
  None	deploy-fly-boinkornet-logs/43	16 s build successful	SUCCESS
  None	deploy-fly-boinkornet-monitor/92	14 s build successful	SUCCESS

buildbot-nix 1.0 is already here,
and is powering nix-community and nix-ngi-0

Get your very own today! (or at the hackday)

redalder.org
numtide.com
git.redalder.org/magic_rb/nixcon2024

Questions?



Questions?



1. thank users of buildbot-nix for the images and also for reporting bugs when they happen