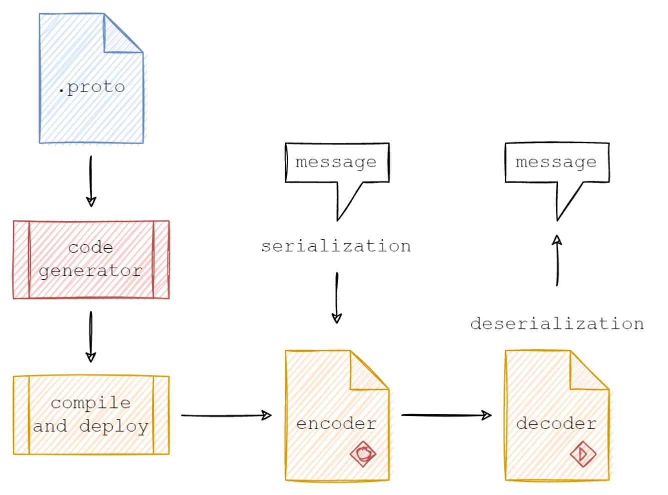
Nix for Reproducible Developer Environments

Oliver Pauffley

October 22, 2025

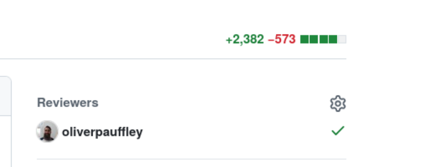
Protobuf


.proto
code generator
compile and deploy
message
serialization
encoder
message
deserialization
decoder

Version Number Wars

Version Number Wars

Everyone loves a PR with 2000+ changes.



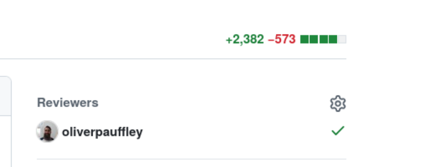
+2,382 −573 🟩🟩🟩🟩⬜

Reviewers

oliverpauffley ✓

Version Number Wars

Everyone loves a PR with 2000+ changes.

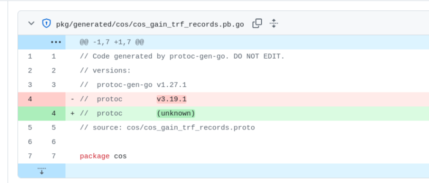


+2,382 −573 🟩🟩🟩🟩□

Reviewers

oliverpauffley ✓

But drilling down.



pkg/generated/cos/cos_gain_trf_records.pb.go

```
@@ -1,7 +1,7 @@
  // Code generated by protoc-gen-go. DO NOT EDIT.
  // versions:
  //  protoc-gen-go v1.27.1
- //  protoc        v3.19.1
+ //  protoc        (unknown)
  // source: cos/cos_gain_trf_records.proto

  package cos
```

The Problem

Every developer has a different version of the tools used to generate the protos!

Make - Install everything on my machine

Lets just include instructions on how to install things

```
PROTOC_VERSION := 3.14.0
.PHONY: install-protoc
install-protoc:
	curl -OL https://github.com/protocolbuffers/protobuf/releases/download/v$(PROTOC_VERSION)/$(PROTOC_FILE)
	sudo unzip -o $(PROTOC_FILE) -d /usr/local bin/protoc
	sudo unzip -o $(PROTOC_FILE) -d /usr/local 'include/*'
	rm -f $(PROTOC_FILE)
```

Ahh?

- Turns out this was already in the repo.

Ahh?

- Turns out this was already in the repo.
- I don’t really want to have curl installing things.

Ahh?

- Turns out this was already in the repo.
- I don’t really want to have curl installing things.
- You can just ignore it.

Install everything on someone else's machine

- If only there was a way to build a little mini machine with the binaries I need already on it.

Install everything on someone else's machine

- If only there was a way to build a little mini machine with the binaries I need already on it.
- Distribute the machine.

Install everything on someone else's machine

- If only there was a way to build a little mini machine with the binaries I need already on it.
- Distribute the machine.
- Docker!

Docker

```
FROM cimg/go:1.19-node as build-stage

ENV PROTOC_VERSION=3.18.1

# Install protoc
RUN curl -LO https://github.com/protocolbuffers/protobuf/releases/download/v${PROTOC_VERSION}/protoc-${PROTOC_VERSION}-linux-x86_64.zip
RUN unzip protoc-${PROTOC_VERSION}-linux-x86_64.zip -d $HOME/.local
RUN rm protoc-${PROTOC_VERSION}-linux-x86_64.zip

RUN mkdir -p /home/circleci/project/pkg/generated

# Install go/js tooling
ADD Makefile .
ADD go.mod .
ADD go.sum .
ADD package.json .
ADD yarn.lock .
RUN make install

ADD . .
# Generate code from proto files.
RUN make protos-generate

FROM scratch as export-stage

COPY --from=build-stage /home/circleci/project/pkg/generated /pkg/generated
```

Docker

Docker

Pros:

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

- We have actually only lifted the problem into docker

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

- We have actually only lifted the problem into docker
- `ADD package.json`

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

- We have actually only lifted the problem into docker
- `ADD package.json`
- This is only as deterministic as the package management we are using within docker.

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

- We have actually only lifted the problem into docker
- `ADD package.json`
- This is only as deterministic as the package management we are using within docker.
- Very slow

Docker

Pros:

- We can distribute this file and get the same(ish) system for everyone
- We can use it CI/CD

Cons:

- We have actually only lifted the problem into docker
- `ADD package.json`
- This is only as deterministic as the package management we are using within docker.
- Very slow
- Doesn't connect well with local tooling for devs

Nix - Install everything on my machine the same as everyone else’s

Demo

Is Nix for You?

Is Nix for You?

Almost definitely no. . .

But

Is Nix for You?

Almost definitely no...

- It's a little complicated to learn (But really not a big barrier to entry).

But

Is Nix for You?

Almost definitely no. . .

- It's a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.

But

Is Nix for You?

Almost definitely no...

- It's a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

Is Nix for You?

Almost definitely no. . .

- It's a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

- It's probably the future?

Is Nix for You?

Almost definitely no. . .

- It’s a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

- It’s probably the future?
- When it’s working and setup it’s incredibly powerful.

Is Nix for You?

Almost definitely no. . .

- It's a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

- It's probably the future?
- When it's working and setup it's incredibly powerful.
- You are probably rebuilding it's features yourself.

Is Nix for You?

Almost definitely no. . .

- It’s a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

- It’s probably the future?
- When it’s working and setup it’s incredibly powerful.
- You are probably rebuilding it’s features yourself.

Is Nix for You?

Almost definitely no. . .

- It’s a little complicated to learn (But really not a big barrier to entry).
- The payoff is complicated to explain.
- Error messages are difficult to understand.

But

- It’s probably the future?
- When it’s working and setup it’s incredibly powerful.
- You are probably rebuilding it’s features yourself.

I’ve been a happy Nix user for about 18 months now, and– well, not happy happy, but satisfied. . . no. . . not really satisfied either; perhaps it’s more of a resigned disgruntlement; a feeling that despite its many flaws, it’s still better than anything else out there, and I’ve invested so much time into it already that it would be a shame to give up now, so. . . am I describing Stockholm syndrome?
— Ian Henry

How to install Nix

`https://nixos.org/download/`

- Works on linux, mac and windows (through wsl)

How to install Nix

`https://nixos.org/download/`

- Works on linux, mac and windows (through wsl)
- You can use it as well as your other package managers

Haskell Book Club

I also run a monthly haskell book club. Meeting on the last Sunday of the month

Questions?