Doubling down on the Go Replace directive to get to the fix we really want.

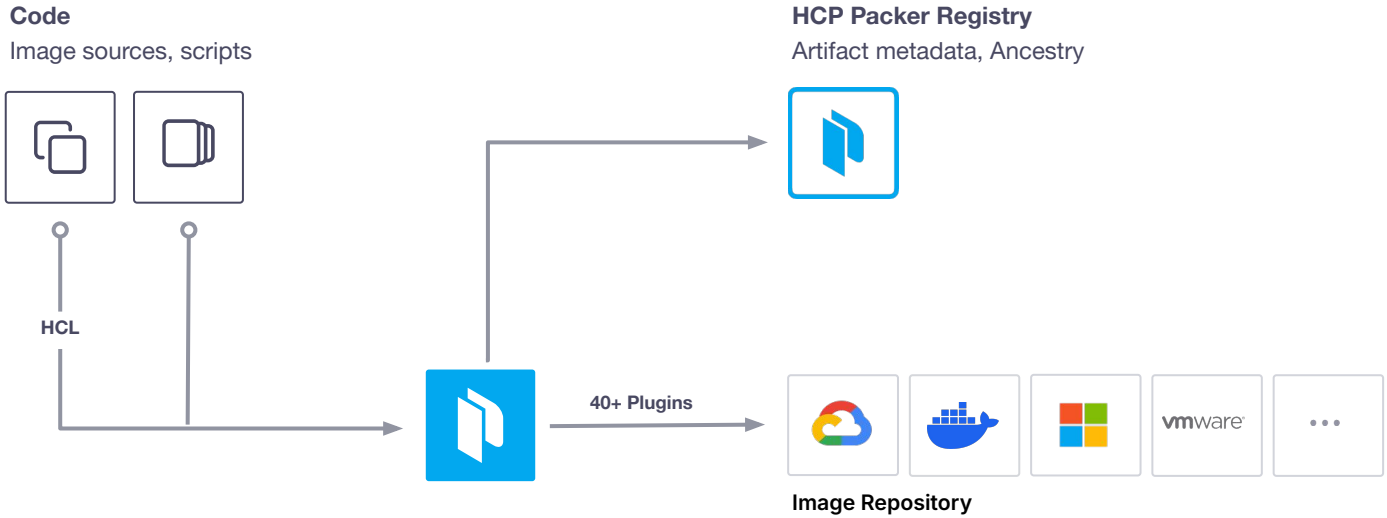
Embracing The Replace

Wilken Rivera

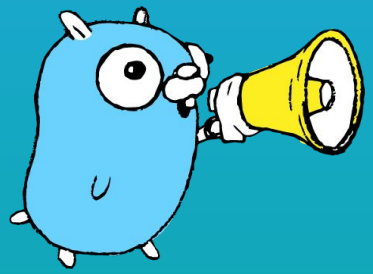
HashiCorp

HashiCorp Packer

+ Packer builds immutable artifacts for cloud providers and hypervisors.

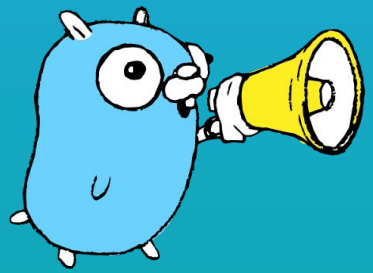


Broken Dependencies



Let me tell you what happend

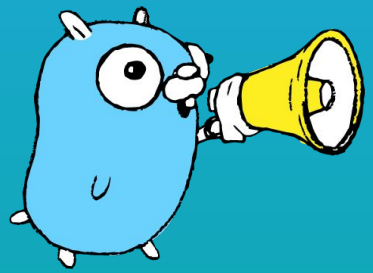
Dec 2019: Packer Adds HCL2 Support



Little Background

Dec 2019: Packer Adds HCL2 Support

Aug 2022: go-cty drops encoding/gob

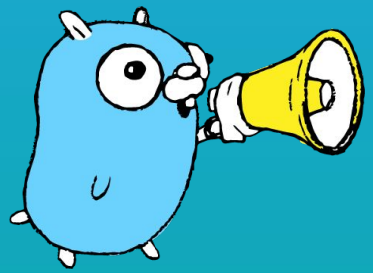


Let me tell you what happend

Dec 2019: Packer Adds HCL2 Support

Aug 2022: go-cty drops encoding/gob

Oct 2022: First plugin crash reported



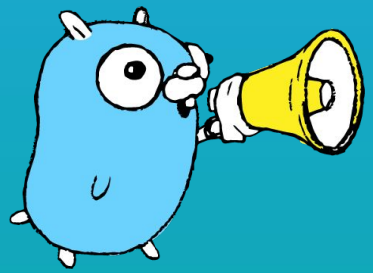
Little Background

Dec 2019: Packer Adds HCL2 Support

Aug 2022: go-cty drops encoding/gob

Oct 2022: First plugin crash reported

Apr 2023: We confirm the Issue



Little Background

Dec 2019: Packer Adds HCL2 Support

Aug 2022: go-cty drops encoding/gob

Oct 2022: First plugin crash reported

Apr 2023: We confirm the Issue

Apr 2023: Discover/Learn/Prototype

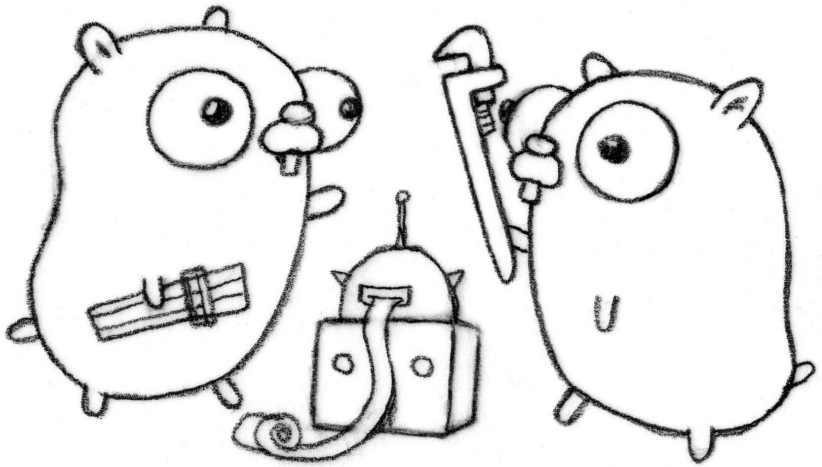
June 2023: Opened packer-plugin-sdk#188

> Just because you can, it doesn't mean you should.

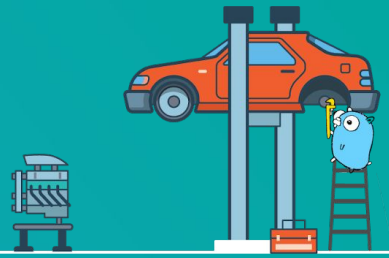
WILKEN RIVERA

Making the time to do things right

+ If you can't make the fix you want, do the next best thing to buy yourself time.



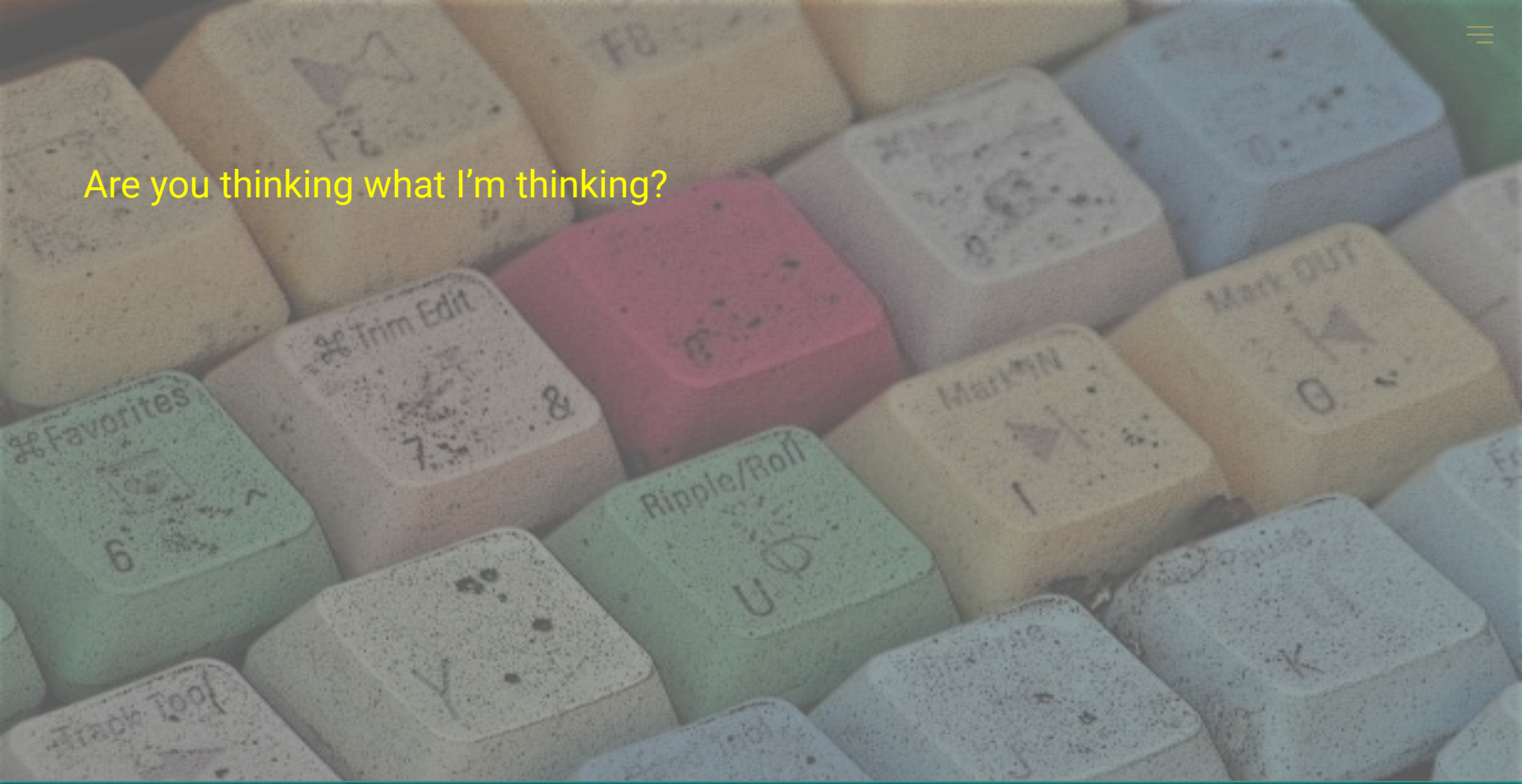
Making the fix without breaking the thing!



1. Created a fork of go-cty@1.11.0 for the SDK.
2. Reverted the gob removal; backported all new changes.
3. Updated the SDK to use go-cty fork with replace directive.
4. Tested the fix against our own plugins.
5. Rinse, repeat, and refine.
6. All is good ...

Quick Go replace directive tutorial

```
// Replace to local version of API for testing
replace github.com/example/api => ../api


// v0.3.11 panics for some reason on our tests
replace github.com/imdario/mergo => github.com/imdario/mergo v0.3.9
```


Are you thinking what I’m thinking?
Favorites
Trim Edit
Ripple/Roll
Track Tool


Manually adding a
go replace feels
messy.

For the sake of your users double down on the messy.

Automated module replacement using fix command

```
~> go get github.com/hashicorp/packer-plugin-sdk@v0.5.2
~> go mod edit -replace "github.com/zclconf/go-cty=github.com/nywilken/go-cty@v1.13.3"
~> go mod tidy
~> go test ./...
```

```
~> go get github.com/hashicorp/packer-plugin-sdk@v0.5.2
~> go install github.com/hashicorp/packer-plugin-sdk/cmd/packer-sdc@v0.5.2
~> packer-sdc fix .
```

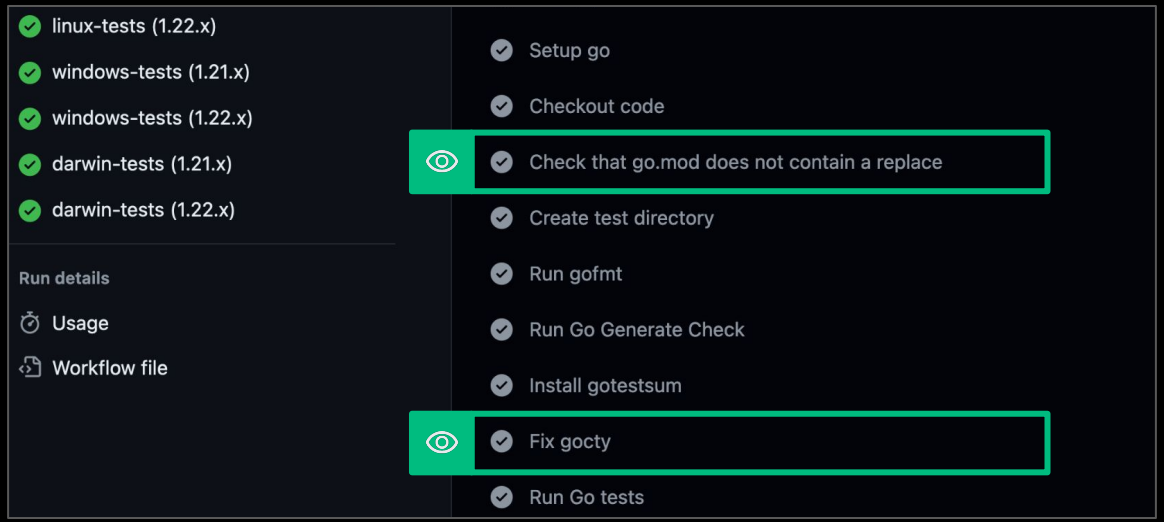
```
~> go mod tidy
~> go test ./...
```

Self documenting comments in the source

```
module go.mondoo.com/packer-plugin-cnspec

go 1.22.0

replace github.com/zclconf/go-cty => github.com/nywilken/go-cty v1.13.3 // added
by packer-sdc fix as noted in github.com/hashicorp/packer-plugin-sdk/issues/187

require (
    github.com/hashicorp/hcl/v2 v2.20.1
    github.com/hashicorp/packer-plugin-sdk v0.5.3
    github.com/mitchellh/mapstructure v1.5.0
    github.com/zclconf/go-cty v1.13.3
    golang.org/x/crypto v0.24.0
)
```

Used GitHub Tooling to self heal the ecosystem

+ GitHub actions to test our tooling on every push.



Used GitHub Tooling to self heal the ecosystem

+ GitHub actions to test our tooling on every push
+ Upgrade notes to the GitHub releases for users to follow.

v0.5.2

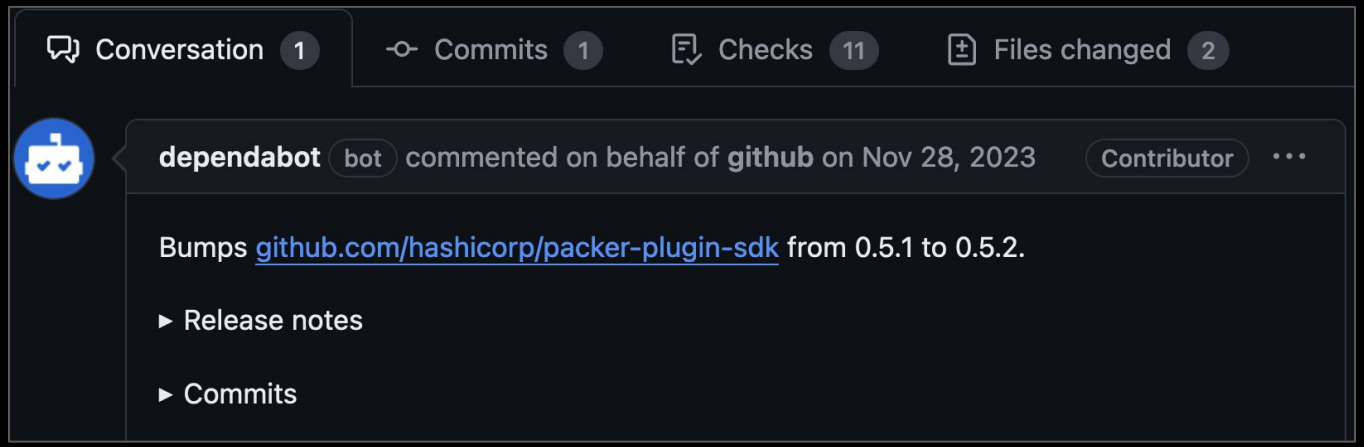
Upgrade Notes

Upgrading to this release may fail until you've applied one of the fixes documented in packer-plugin-sdk#187. Consumers of the Packer plugin SDK require a replace directive within their plugin's go module file to point to a compatible version of go-cty. The replace directive subject to change in future releases can be applied by running the `packer-sdc fix` sub-command to apply the replace directive to your plugin with a recommended version of the go-cty fork.

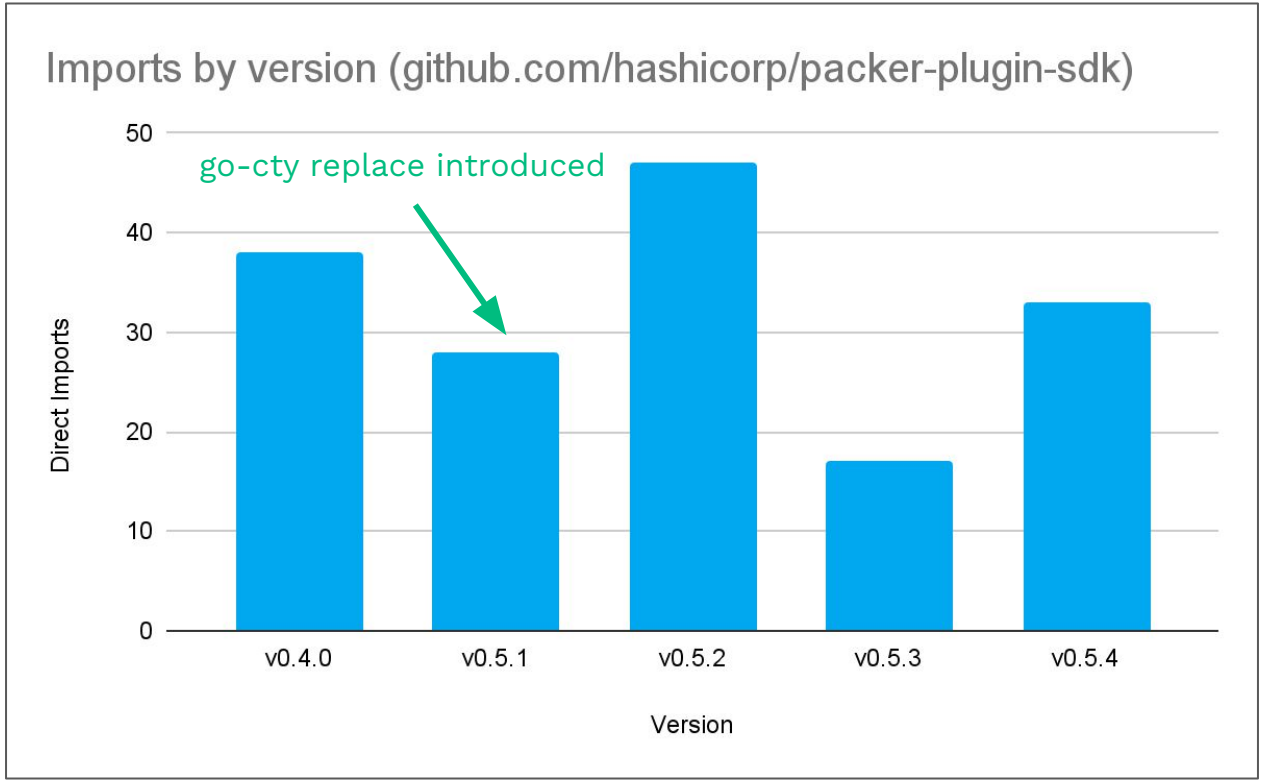
Plugins already working with Packer Plugin SDK v0.5.1 are advised to apply the updated SDK fixes by re-running `packer-sdc fix` against the plugin's root directory. The updated SDK fixes

Used GitHub Tooling to self heal the ecosystem

+ GitHub actions to test our tooling on every push
+ Upgrade notes to the GitHub releases for users to follow.
+ Dependabot to roll out SDK updates.



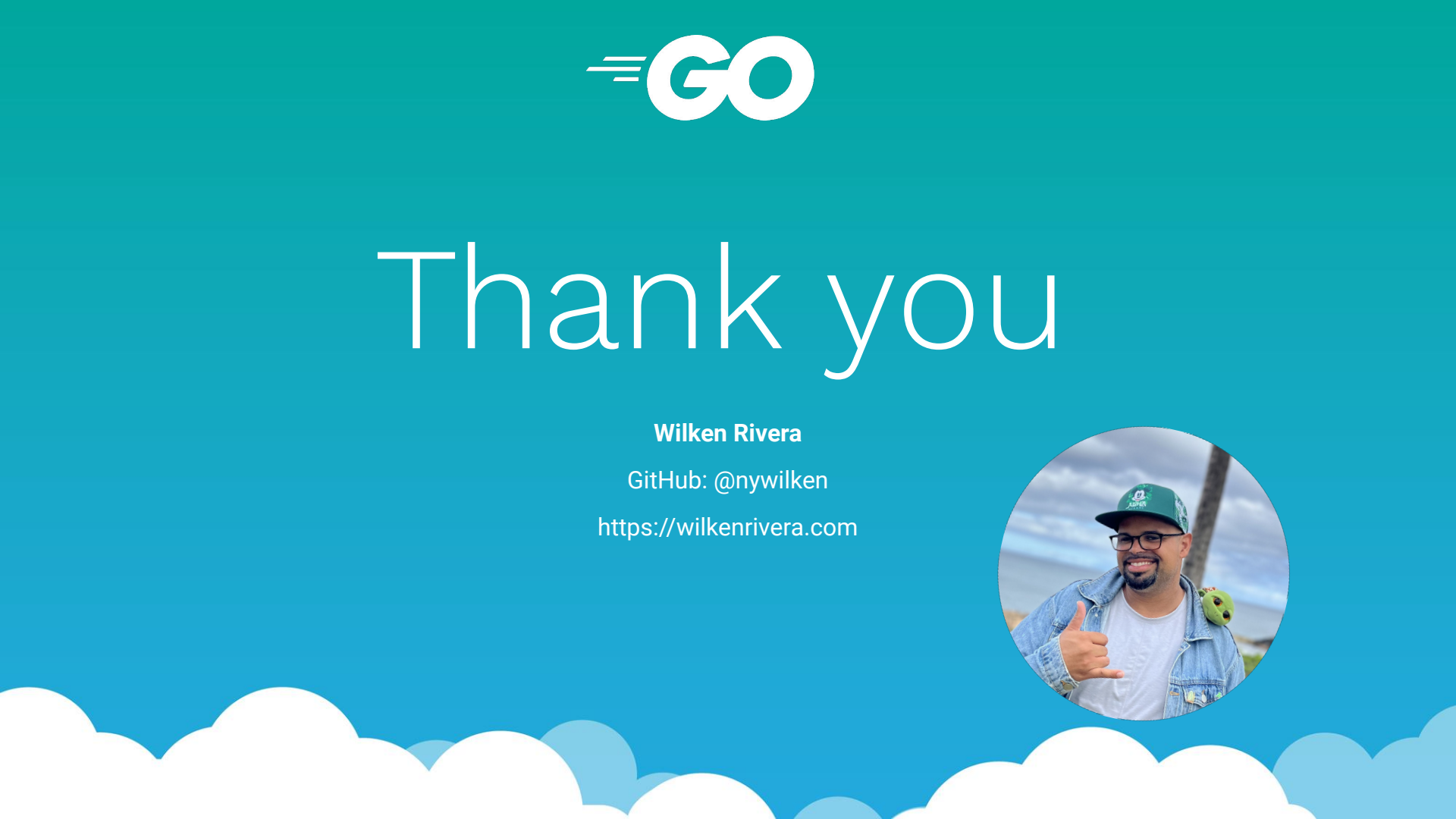
Current state of the fix



Takeaways

If you get anything out of this talk I want you to remember the following three things:

1. As maintainers, we can easily make breaking changes to fix our problem. But it comes at the cost of the user. That is not a good feeling.
2. The right fix feels like it needed to happen yesterday, but those fixes are not always easy. So the best thing to do is always look for ways to buy yourself time to get where you want to be.
3. Sometimes, the best solution is not the prettiest. When it does, double down on the ugly and make it as easy to adopt as possible.



Thank you

Wilken Rivera

GitHub: @nywilken

https://wilkenrivera.com