Over engineering the core of Kubernetes kops

Kris Nova

Kris Nova

About Me

Kris Nova

“I work in the cloud…”

Kris Nova

In my free time I help run a Kubernetes SIG..

Kris Nova

..that brings an

open source

Google

project..

Kris Nova

..to

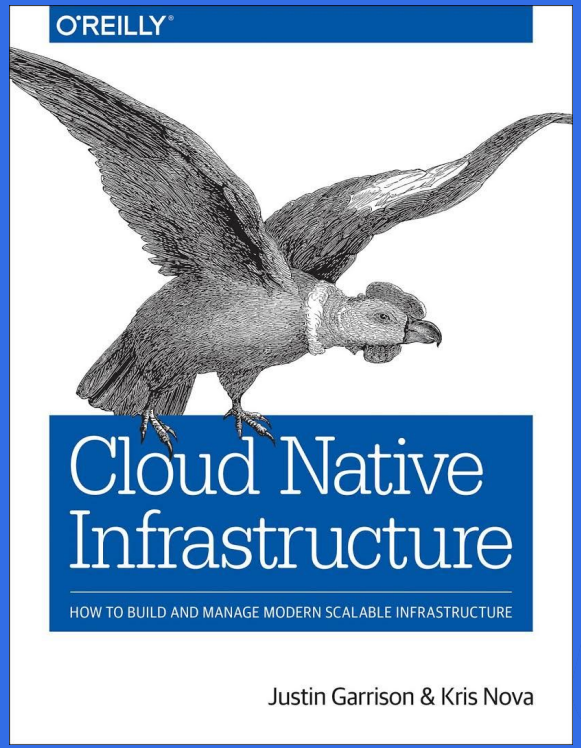
amazon web services

Kris Nova

..while working at...

Microsoft ACS

Kris Nova


O'REILLY
Cloud Native Infrastructure
HOW TO BUILD AND MANAGE MODERN SCALABLE INFRASTRUCTURE
Justin Garrison & Kris Nova

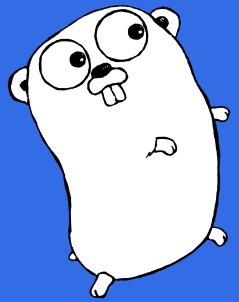
Kris Nova

go/AUTHORS

```
570    Kris Nova <kris@nivenly.com>
```

Kris Nova

r/golang

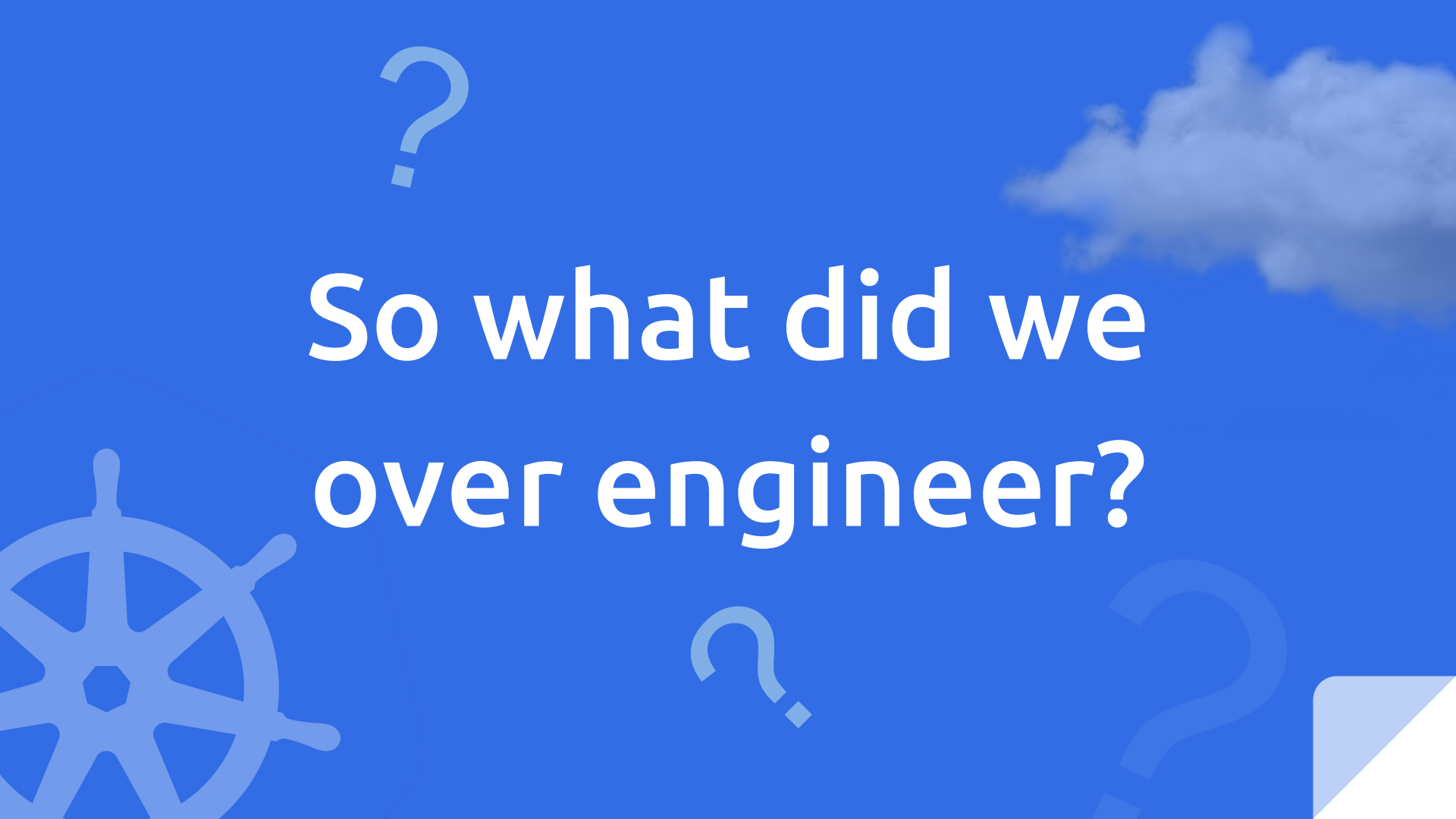


Thanks Renee French!

Kris Nova

"I speak for the software.."

Over Engineering Go in Kubernetes


So what did we over engineer?

Kubernetes Kops

Kubernetes Operations

Kubernetes Kops

Kubernetes Operations

"Kops models a cluster then realizes it in a cloud"

- Kris Nova

Kubernetes Kops

Kubernetes Operations

Statically linked CLI tool written in Go

Kubernetes Kops

Kubernetes Operations

import “k8s.io/kubernetes”

Kubernetes Kops

Kubernetes Operations

Cluster Model → Apply → Kubernetes

Kops Model

Kubernetes Operations

apiVersion: “kops/v1alpha2”
kind: “Cluster”
kubernetesVersion: “1.7.0”
networkCIDR: “172.20.0.0/16”
masterPublicName: “api.nivenly.com”

So how does it work ?

Kops 1.4

Kops 1.4

We have to track our work

```
tasks := []*Task
```

Where do tasks come from

Kops 1.4

```
# SSH is open to AdminCIDR set
{{ if IsTopologyPublic }}
{{ range $index, $cidr := AdminCIDR }}
securityGroupRule/ssh-external-to-master-{{ $index }}:
  securityGroup: securityGroup/masters.{{ ClusterName }}
  cidr: {{ $cidr }}
  protocol: tcp
  fromPort: 22
  toPort: 22
{{ end }}
{{ end }}
```

How do we parse them

Kops 1.4

embed in .go file

Kops 1.4

embed in .go file

import “text/template”

Kops 1.4

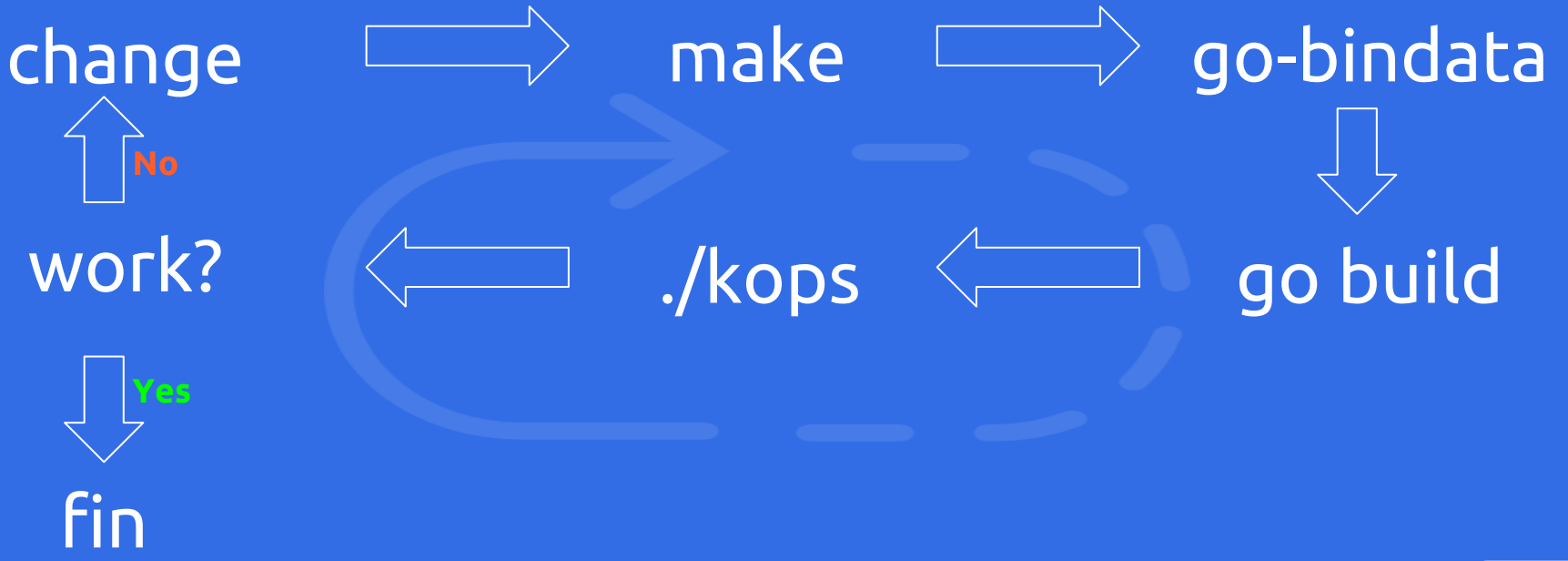
embed in .go file

import “text/template”

parse at runtime

How do we develop

Kops 1.4



Kops 1.4

We were **really** good at dealing with YAML

Kops 1.4

And we had a lot of it..

Kops 1.4

```
$: git checkout tags/v1.4.4
$: cd upup/models/cloudup
$: find . | xargs wc -l | grep
total
    total 1076
```

The next release

Kops 1.5

wanted new features

Kops 1.5

wanted new features

brittle dev process

Kops 1.5

wanted new features

brittle dev process

falling behind

Kops 1.5

..also

Kops 1.5

..also

700 open GitHub issues

Kops 1.5

..also

We couldn’t test our “text/template” code

Kops 1.5

..also

We would still get panics at runtime..

Kops 1.5

List of things we needed to fix:

1. Our shit

Kops 1.5

The Sandlot (maintainers)

Kops 1.5

The Beast (templates)

Kops 1.5

Smalls (me)

What did we do?


YAML
text/template
fed6831

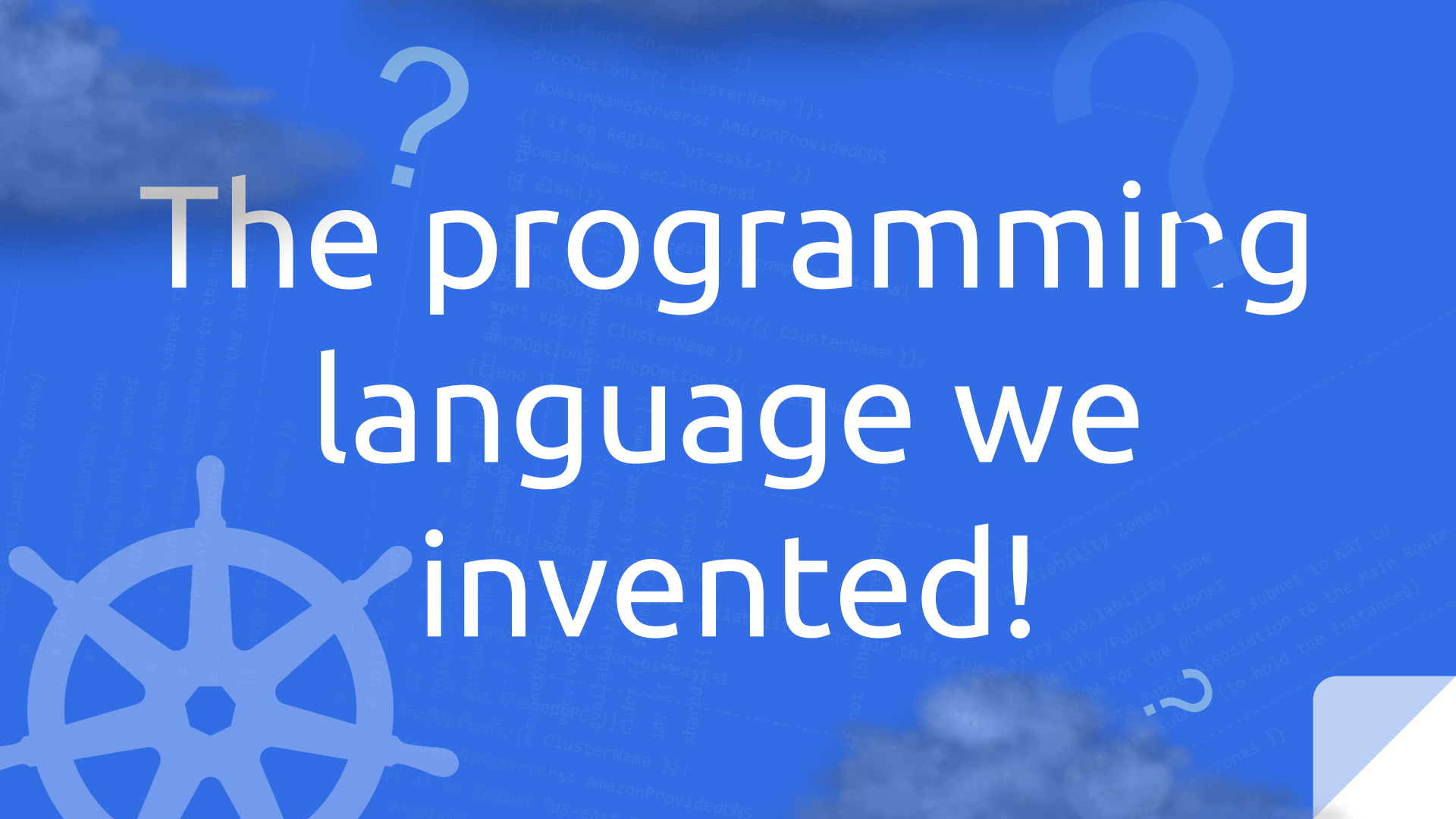
Kops 1.5

P.F. Flyers (rm -rf)

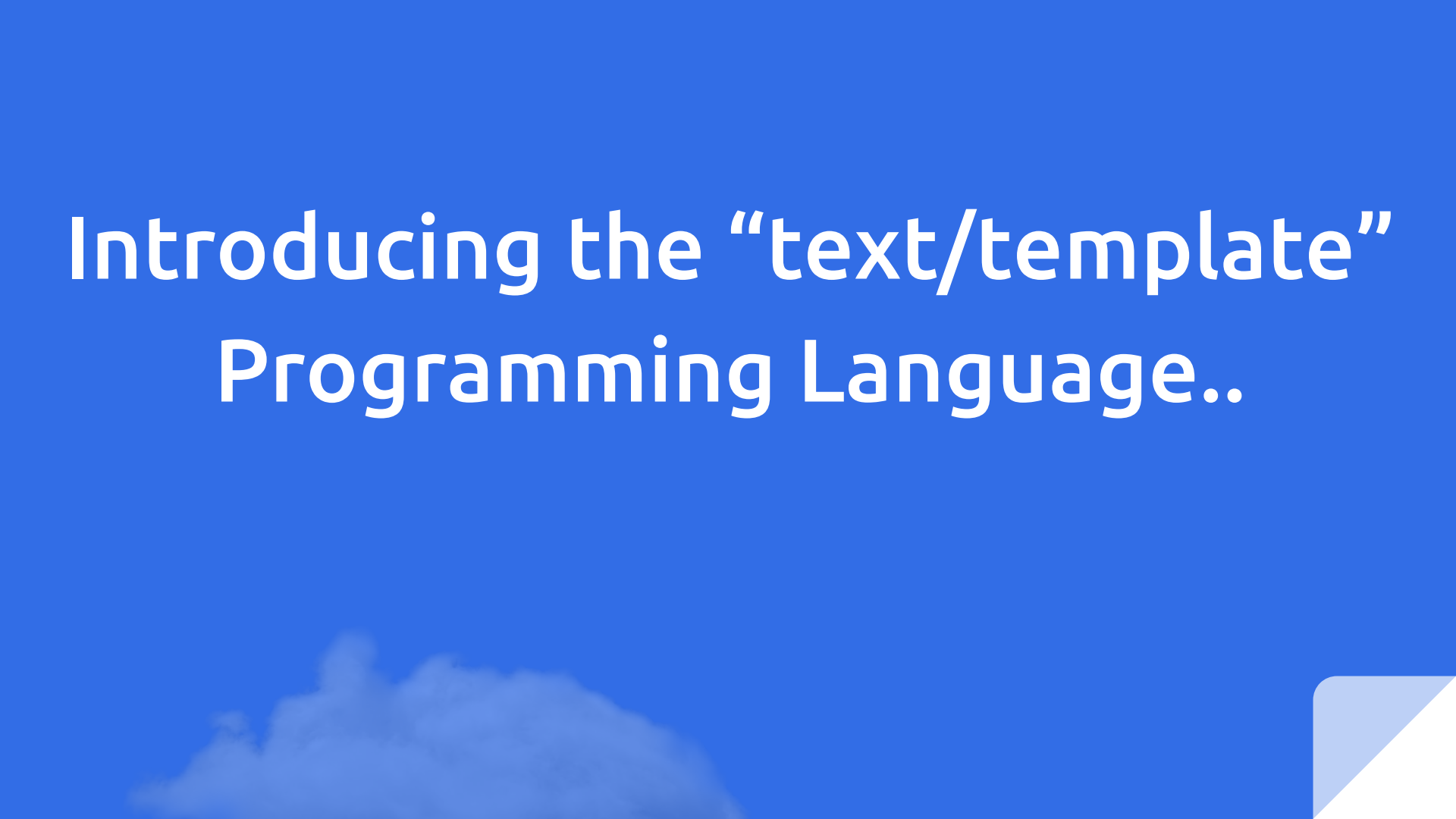
Kops 1.5

The Baseball (make test)

What did we remove?


The programming language we invented!

Introducing the “text/template” Programming Language..



The “text/template” Programming Language

The “text/template” Programming Language

type **FuncMap**

FuncMap is the type of the map defining the mapping from names to functions. Each function must have either a single return value, or two return values of which the second has type error. In that case, if the second (error) return value evaluates to non-nil during execution, execution terminates and Execute returns that error.

When template execution invokes a function with an argument list, that list must be assignable to the function's parameter types. Functions meant to apply to arguments of arbitrary type can use parameters of type interface{} or of type reflect.Value. Similarly, functions meant to return a result of arbitrary type can return interface{} or reflect.Value.

```
type FuncMap map[string]interface{}
```

The “text/template” Programming Language

Calling arbitrary functions...

The “text/template” Programming Language

Calling arbitrary functions...

=

Turing complete language

{{ nbd }}

Go vs. text/template

Round 1 Errors

Go	text/template
1	0

Go:
enforced error handling

text/template:
panic

Round 2 Compiling

Go	text/template
2	0

Go:
Won’t compile on invalid syntax

text/template:
Will compile on invalid syntax

Round 3 Debugging

Go	text/template
3	0

Go:
line numbers and meaningful output

text/template:
it broke

Round 4 Developing

Go	text/template
4	0

Go:
wonderful IDE support

nope

Round 5 Testing

Go	text/template
5	0

go test

uh. . .

Go	text/template
5	0

Winner: Go

Thanks Ashley!

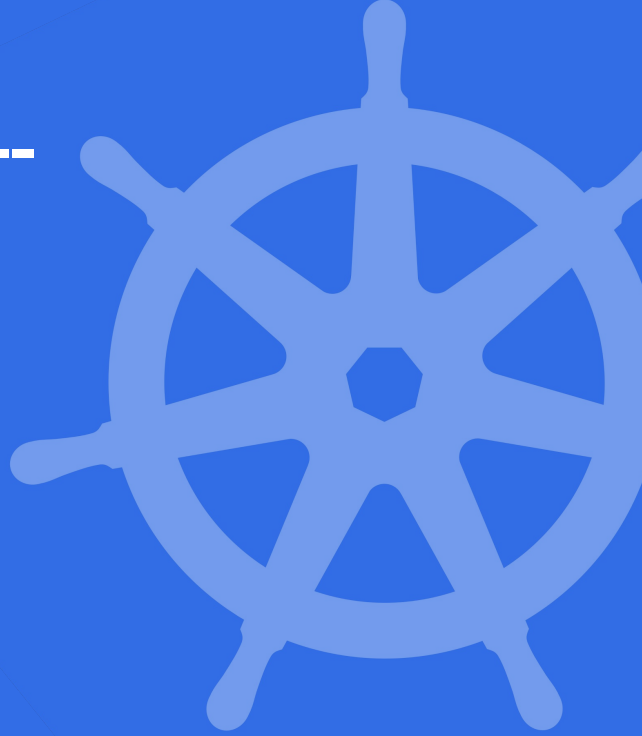
So we moved our tasks into Go..

{{ if isSomething }} ⟹ if isSomething() {}

.yaml .go

We wrote small Go systems instead of YAML systems

```
# ---------------- BEFORE ----------------
# Logic to include small system
{{ if IsSomething }}
BigSystem:
    SmallSystem:
        Name: {{  Name }}
        Data: {{ Data }}
{{ end }}
```

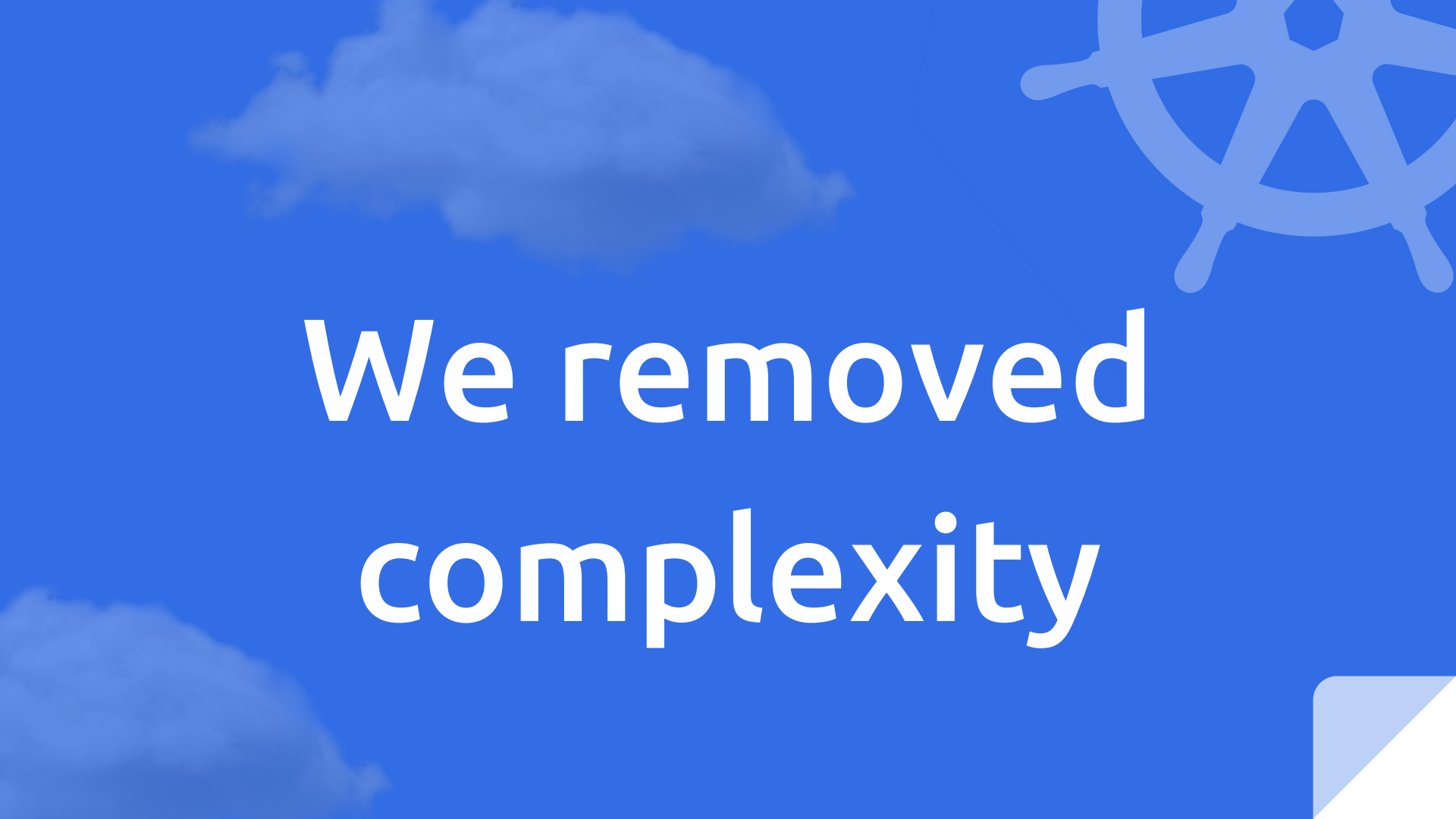


```
// ---------------- AFTER ----------------
func (b *bigSystem) smallSystem(api *api) error {
    if isSomething(api.Value){
        return b.ensureTask(&task{
            Name: “myTask”,
            Data: “data”,
        })
    }
    return nil
}
```

We turned 1,000 lines of yaml into 10 structs


We wrote real tests..

..that exercised real code


We removed
complexity

We deleted entire packages


We made it
exciting to develop

We fixed our shit

Simple Go wins.

Kris Nova

@kris__nova