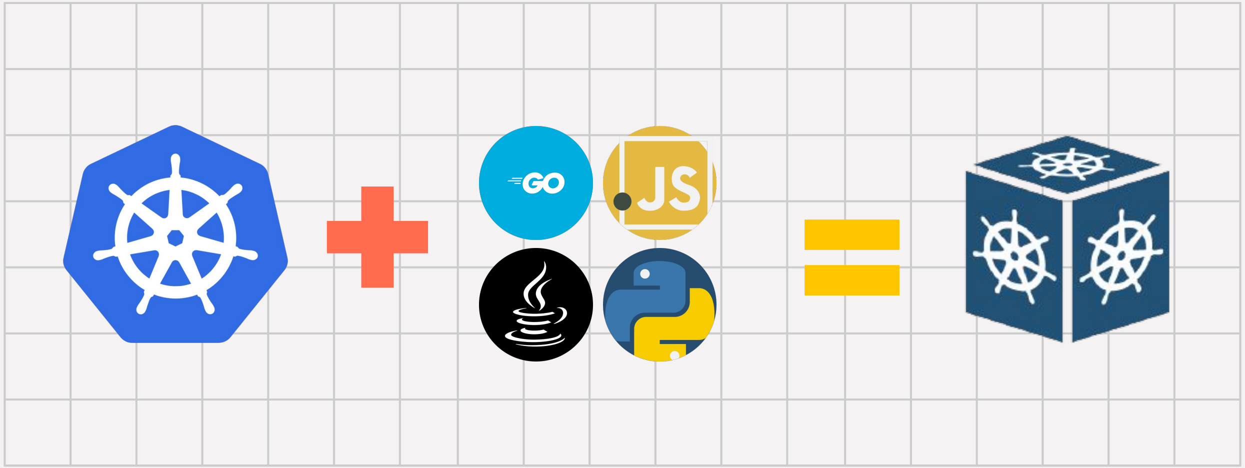
CDK8s: Unleash programming language power for correct and testable Kubernetes charts

Config Management Camp

Benjamin FÜHRMANN



About Me

❤ DevOps

DATADOG

Software engineer
Alerting product

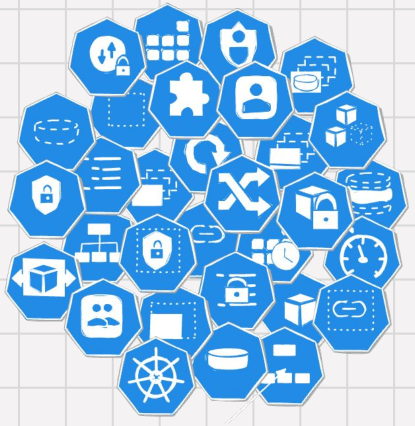
HELM

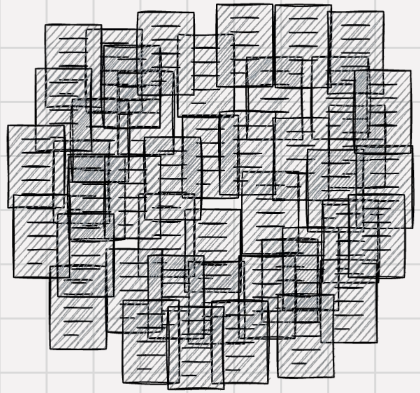
And infrastructure

Datadog ❤ Kubernetes

For Alerting

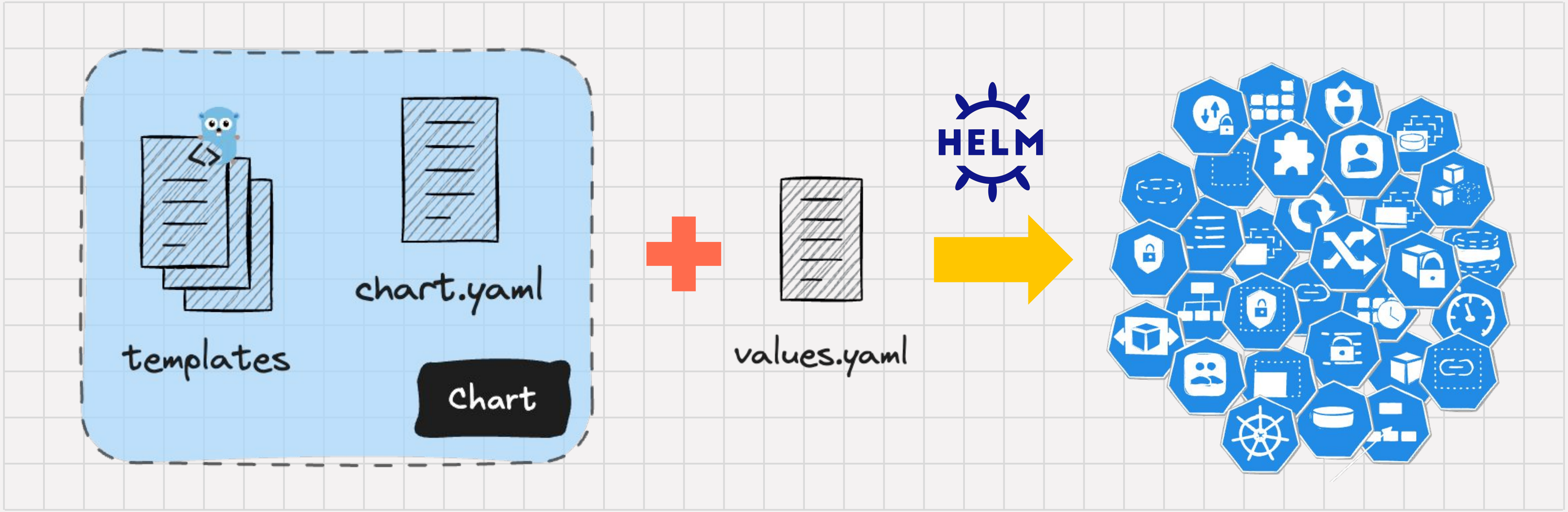
60+	Services
150+	Clusters
60+	Engineers

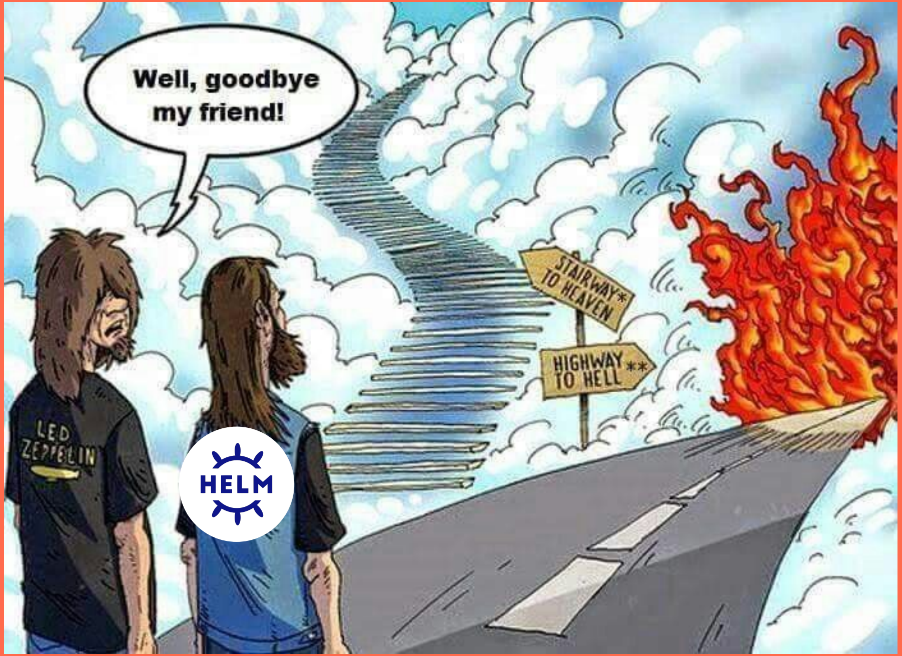

A lot of Kube resources


A lot of YAML documents...

Helm: the infra package manager

To package deployments, the “apt” for Kubernetes





When Helm becomes Hell...

Stripception

Where is my extra line?

```
metricSelector:
  matchLabels:
    {{- if default $.Values.extraLabel false -}}
    extraLabel: {{ $.Values.extraLabel }}
    {{- end -}}
```

```
metricSelector:
  matchLabels:extraLabel: my-extra-label
```

Think indent

My space is going outer space

```
{{- define "alerting.common-metadata" -}}
namespace: {{ .Release.Namespace }}
labels:
  domain: alerting
{{- end -}}

apiVersion: v1
kind: ServiceAccount
metadata:
  name: {{ $.Release.Name }}
  {{ include "alerting.common-metadata" . | nindent 2 | trim }}
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-release
  namespace: my-namespace
labels:
  domain: alerting
```

Readability

Forget about your IDE completion...

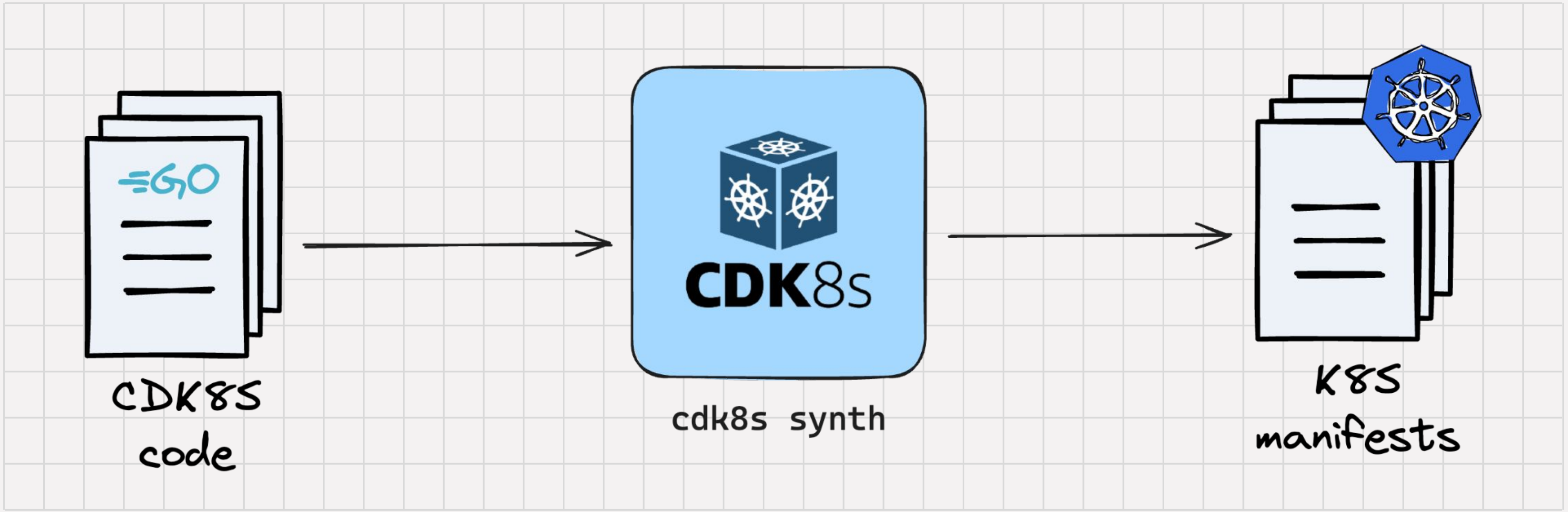
```
{{- if (($values.podAutoscaler).enabled) }}
{{- $deployName := $.Release.Name }}
{{- if and $.Values.shards $values.name }}
  {{- $deployName = printf "%s-%s" $.Release.Name $values.name }}
{{- end -}}

{{- /* Metric query filters */}}
{{- $releaseFilters := printf "service:%s,kube_deployment:%s" (include
"monitors-evaluation.metadata.service-name" $) (coalesce ($values.podAutoscaler.metricsFilter).deploymentName
$deployName) }}
{{- $dcFilter := printf ",datacenter:%s" $.Values.global.datacenter.datacenter }}
{{- $clusterFilter := ",kube_cluster_name:%%tag_kube_cluster_name%%" }}
{{- $shardFilter := "" }}
{{- $containerFilter := "" }}
{{- $containerName := "" }}
{{- if and $.Values.shards $values.name }}
  {{- $shardFilter = printf ",shard:%s" $values.name }}
{{- end }}
{{- if not (and $values.podAutoscaler.metricsFilter $values.podAutoscaler.metricsFilter.noContainerName) }}
  {{- $containerName = $.Chart.Name }}
{{- end }}
{{- if and $values.podAutoscaler.metricsFilter $values.podAutoscaler.metricsFilter.containerName  }}
  {{- $containerName = $values.podAutoscaler.metricsFilter.containerName }}
{{- end }}
```

The business watching me perform a complex engineering production fix
Me: Finding that extra space in the YAML file at line 127

How to get rid of templates?

CDK8s

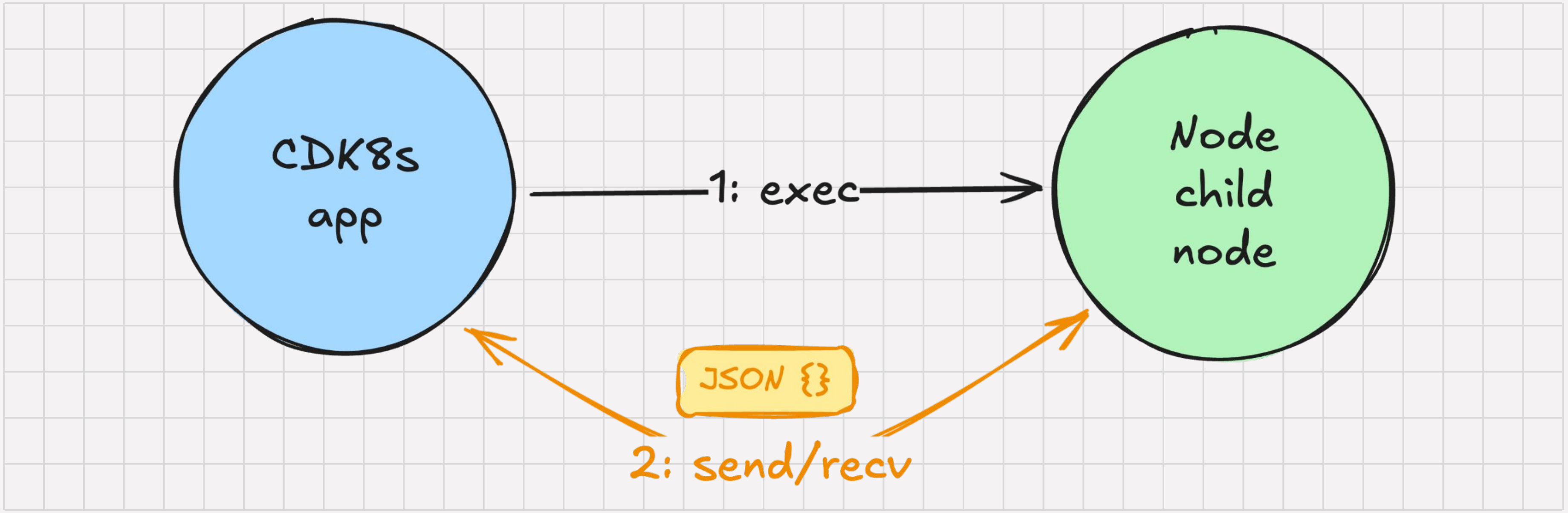
An open source framework to write your K8s resources as code!



Supported languages :

How does it work?

Based on AWS SDK Constructs



Setting up

Setup a new CDK8s project

```
> mkdir cdk8s-tutorial && cd cdk8s-tutorial
> cdk8s init go-app
Initializing a project from the go-app template
Importing k8s v1.25.0...
Importing resources, this may take a few moments...
==========================================================

 Your cdk8s Go project is ready!

   cat help      Prints this message
   cdk8s synth   Synthesize k8s manifests to dist/
   cdk8s import  Imports k8s API objects to "imports/k8s"

  Deploy:
   kubectl apply -f dist/

==========================================================
```

My first resource

Let's code a ServiceAccount in Go

```go
func main() {
    app := cdk8s.NewApp(nil)
    chart := cdk8s.NewChart(app, jsii.String("cdk8s-tutorial"), nil)
    k8s.NewKubeServiceAccount(
        chart, jsii.String("ServiceAccount"), &k8s.KubeServiceAccountProps{
        Metadata: &k8s.ObjectMeta{
            Name: jsii.String(ResourceName),
            Labels: &map[string]*string{
                "app.kubernetes.io/name":     jsii.String("hello-world"),
                "app.kubernetes.io/instance": jsii.String("cdk8s-tutorial"),
            },
        },
    })
    app.Synth()
}
```

Tada!

Synth and let the magic begin

```
> cdk8s synth
Synthesizing application
  - dist/cdk8s-tutorial.k8s.yaml
> cat dist/cdk8s-tutorial.k8s.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: cdk8s-tutorial-hello-world
  labels:
    app.kubernetes.io/name: hello-world
    app.kubernetes.io/instance: cdk8s-tutorial
```

Let’s have other resources

Same pattern, KISS

```
k8s.NewKubeConfigMap(...)

k8s.NewKubeCronJob(...)

k8s.NewKubeDeployment(...)

k8s.NewKubePodDisruptionBudget(...)

k8s.NewKubeService(...)

k8s.NewKubeStatefulSet(...)
```

Use direct K8s specs…

That’s tedious =/

```
k8s.NewKubeDeployment(chart, jsii.String("Deployment"), &k8s.KubeDeploymentProps{
    Metadata: &k8s.ObjectMeta{
        Name: jsii.String("my-deployment"),
    },
    Spec: &k8s.DeploymentSpec{
        Selector: &k8s.LabelSelector{
            MatchLabels: &map[string]*string{"app": jsii.String("app")},
        },
        Template: &k8s.PodTemplateSpec{
            Metadata: &k8s.ObjectMeta{
                Labels: &map[string]*string{"app": jsii.String("app")},
            },
            Spec: &k8s.PodSpec{
                Containers: &[]*k8s.Container{
                    {
                        Name: jsii.String("main"),
                        Image: jsii.String("nginx"),
```

…or use cdk8s+

High-level functions that abstract concept easily!

```go
// Selection label mapping is done automatically
deployment := cdk8splus.NewDeployment(
    chart, jsii.String("Deployment"), &cdk8splus.DeploymentProps{
    Containers: &[]*cdk8splus.ContainerProps{
        &cdk8splus.ContainerProps{Image: jsii.String("nginx")},
    },
})

// Expose the deployment easily through a Service
deployment.ExposeViaService(&cdk8splus.DeploymentExposeViaServiceOptions{
    ServiceType: cdk8splus.ServiceType_LOAD_BALANCER,
})
```

Testing

Like any unit test of your favorite language

Test the Props structures directly

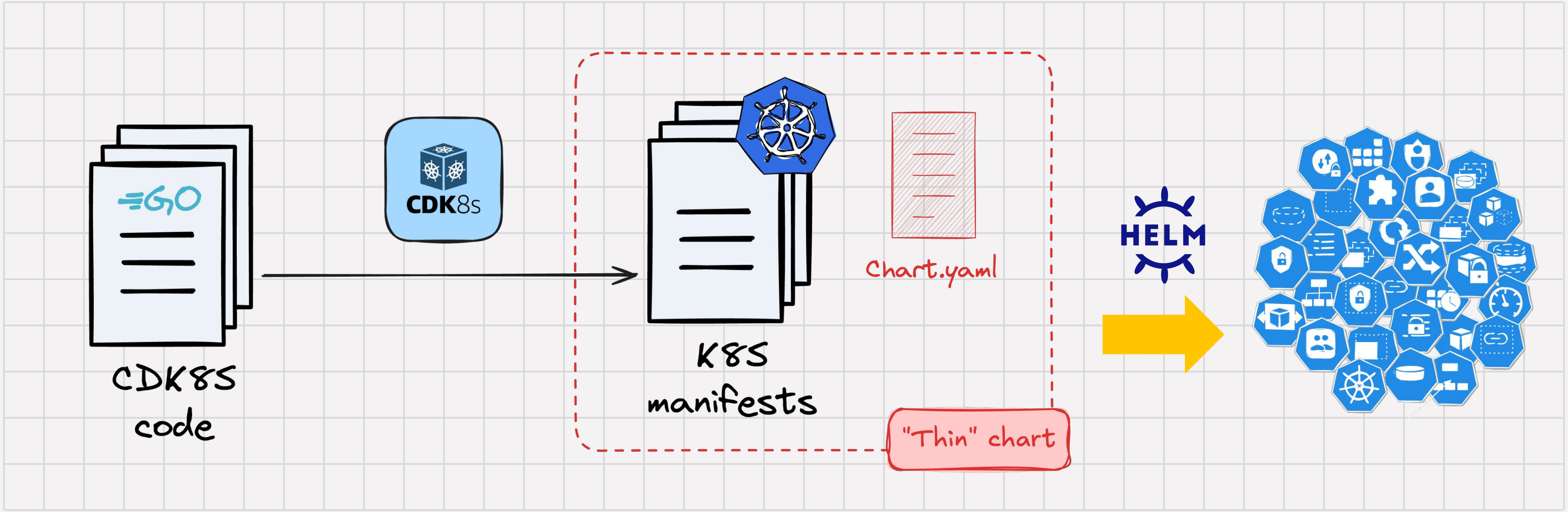
Or test the rendered manifests

```
// Mock CDK8s
app := cdk8s.Testing_App()
chart := cdk8s.Testing_Chart()
manifests := cdk8s.Testing_Synth(chart)
```

Integration with Datadog Alerting

Integration with Helm

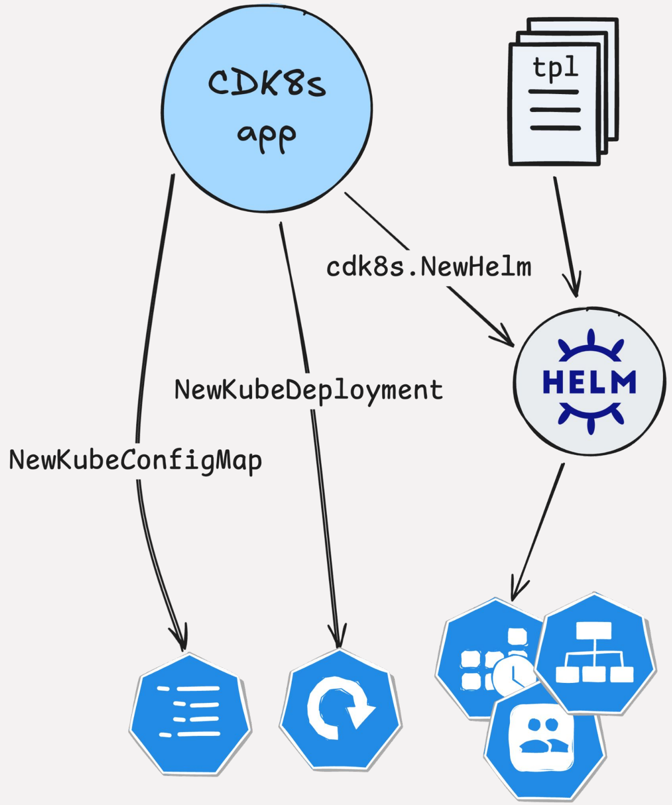
Keep the same workflow with CDK8s rendered manifests



Smooth transition

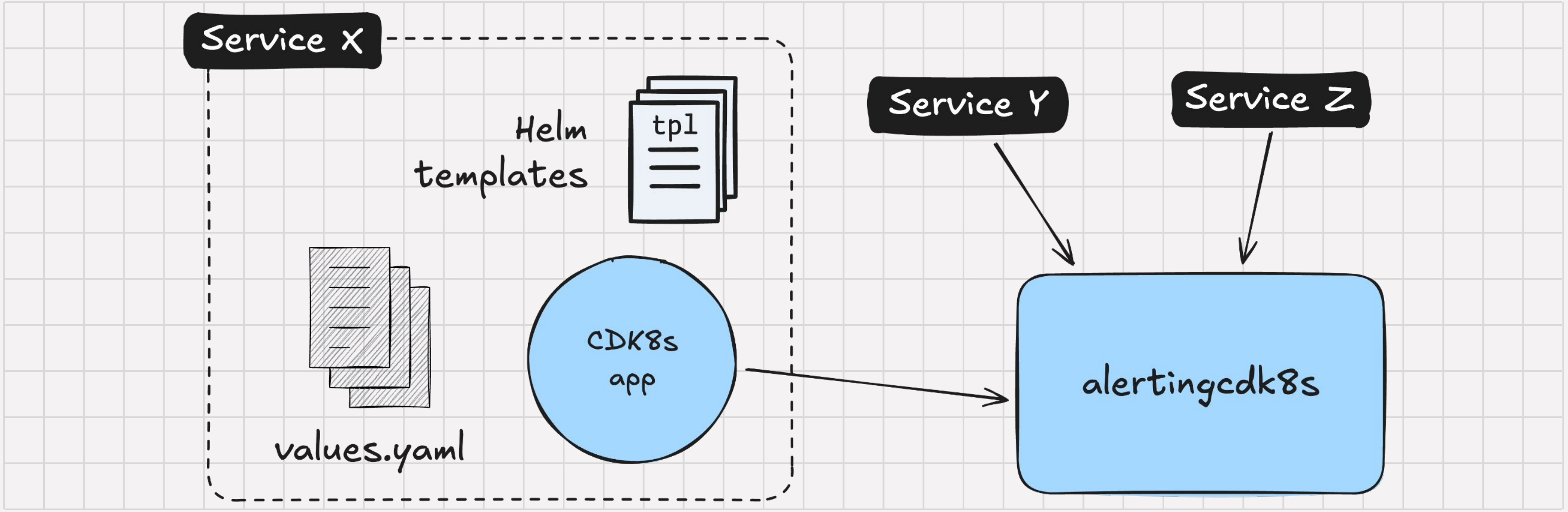
Having both templates and CDK8s code for the same service

```
app := cdk8s.NewApp(nil)
k8s.NewKubeDeployment(...)
k8s.NewKubeConfigMap(...)
cdk8s.NewHelm(
    app,
    jsii.String("imported-chart"),
    &cdk8s.HelmProps{
        Chart: jsii.String("imported-chart"),
        Values: &map[string]interface{}{
            "helm-value-a": "a",
        },
    },
)
app.Synth()
```



alertingcdk8s

One common library
to rule them all



Our XP at Datadog

Some pros and cons

- Easier to code
- Code sharing
- Standardization
- Testability
- Strong typing

- Long to rewrite
- Rendering not that fast
- Very verbose in Go

Thanks for your attention!

Questions?