

# BEAM and Kubernetes: Better Together?

# Cory O'Daniel

- Maintainer: Bonny, k8s.ex, Arbor
- Twitter: @coryodaniel
- GitHub: @coryodaniel

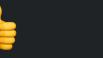
Pronounces kubectl: **kube cuddle** or **kube cartel** depending on my mood.



garamond

Are they  
better  
together?

<b>Features</b>	<b>Kubernetes</b>
Automated Rollouts / Rollbacks	Deployment, ReplicaSet, StatefulSet
Automated Scheduling (instance placement)	pod/node affinity/anti-affinity, resources
Bin Packing	Policy, LimitRange, resources
Batch Jobs / Execution	Job, CronJob
Service Discovery / DNS	Service, Endpoint, EndpointSlices, external-dns
Load Balancing	internal & external load balancers
Storage Orchestration	PersistentVolume, StorageClass, and CSI
Secret "Management"	Secret
Config Management	ConfigMap
Health Checks	Shell, TCP, and HTTP Health Checks
Horizontal Scaling	HorizontalPodAutoscaler
Vertical Scaling	VerticalPodAutoscaler, resources
QoS	PodDisruptionBudget, resources
Security Templates	PodSecurityPolicy, securityContext
Metrics	metric-server, custom-metrics-server

<b>Feature</b>	<b>Kubernetes</b>	<b>Beam or Erlang/OTP</b>
Automated Rollouts / Rollbacks	Deployment, ReplicaSet, StatefulSet	Hot Code Loading
Automated Scheduling (instance placement)	pod/node affinity/anti-affinity, resources	-
Bin Packing	Policy, LimitRange, resources	-
Batch Jobs / Execution	Job, CronJob	Processes
Service Discovery / DNS	Service, Endpoint, EndpointSlices, external-dns	-
Load Balancing	internal & external load balancers	-
Storage Orchestration	PersistentVolume, StorageClass, and CSI	-
Secret "Management"	Secret	-
Config Management	ConfigMap	vm.args, sys.config
Health Checks	Shell, TCP, HTTP Health Checks	Supervisors, -heart
Horizontal Scaling	HorizontalPodAutoscaler	Add nodes 
Vertical Scaling	VerticalPodAutoscaler, resources	Add CPUs / RAM 
QoS	PodDisruptionBudget, resources	-
Security Templates	PodSecurityPolicy, securityContext	-
Metrics	metric-server, custom-metrics-server	erlang:system_info/1 erlang:statistics/1

Are they  
better  
together?

# **YOUR\_ORG** and Kubernetes: Better Together?



# Your Startup's

# MVP

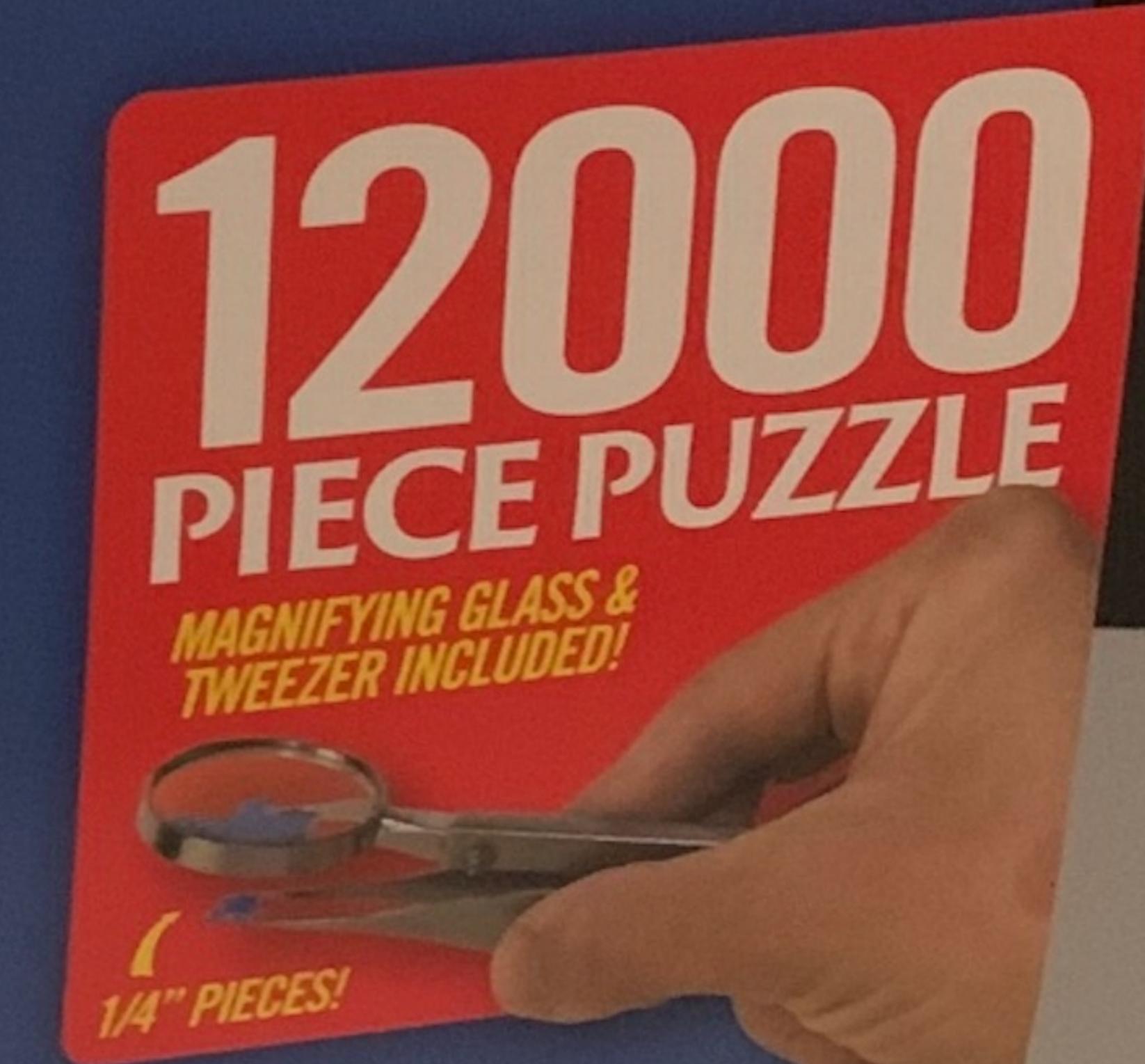
## On Kubernetes:



# Your Startup's

# MVP

## On Kubernetes:



# Kubernetes Best Features

- A simple, extendable API and client
- Learned Complexity
- Community

<b>Features</b>	<b>Kubernetes</b>
Automated Rollouts / Rollbacks	Deployment, ReplicaSet, StatefulSet
Automated Scheduling (instance placement)	pod/node affinity/anti-affinity, resources
Bin Packing	Policy, LimitRange, resources
Batch Jobs / Execution	Job, CronJob
Service Discovery / DNS	Service, Endpoint, EndpointSlices, external-dns
Load Balancing	internal & external load balancers
Storage Orchestration	PersistentVolume, StorageClass, and CSI
Secret "Management"	Secret
Config Management	ConfigMap
Health Checks	Shell, TCP, and HTTP Health Checks
Horizontal Scaling	HorizontalPodAutoscaler
Vertical Scaling	VerticalPodAutoscaler, resources
QoS	PodDisruptionBudget, resources
Security Templates	PodSecurityPolicy, securityContext
Metrics	metric-server, custom-metrics-server

# Kubernetes Best Features

- A simple, extendable API and client
- Learned Complexity
- Community

**Apps start out  
so cute...**



- Packer for VM Images
- Terraform for VM Instances
- Salt or Chef for Configuration
- KMS for Secrets
- Deployin' with bash
- LOL for rollbacks
- Terraform or aws-cli for Autoscaling Groups
- Load Balancers, DNS Records, Health Checks, Access Control
- Ports, Firewalls, IAM
- Do I need a sErvlce MeSH?!1!



# WHEW!

That's a lot of tooling.



**Aarggh!**  
Our VMs are  
only using  
**8.3%**  
of their CPU.



**What business  
value are you  
**creating** by  
reinventing the  
**wheel**  
**helm?****



Kubernetes

is

complicated

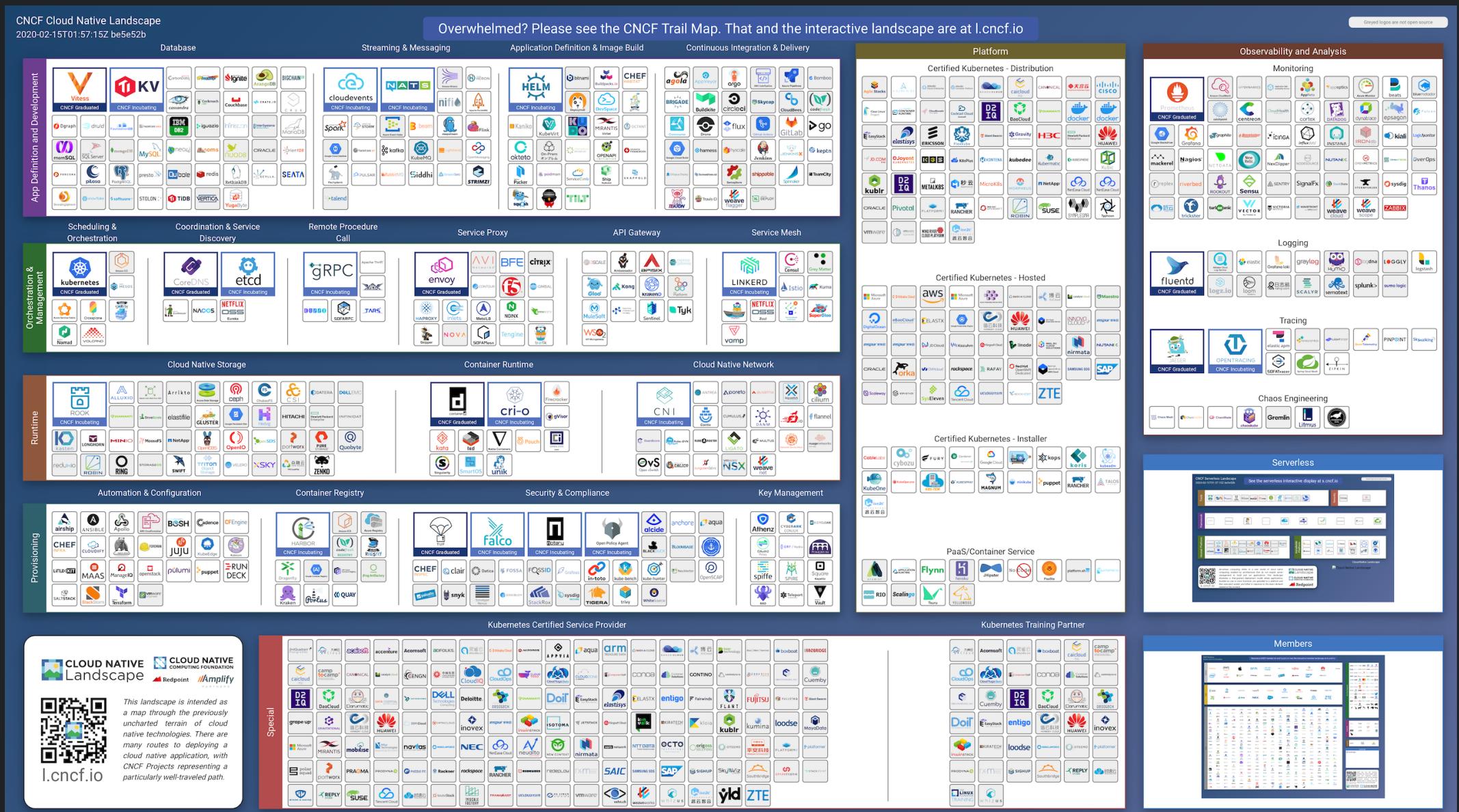
Kubernetes  
is  
complicated  
**Learned Complexity**

In the next few years, the  
Kubernetes API will become the  
common API for interfacing with  
cloud resources.

# What can you deploy today?

- Workloads
- Batch jobs
- Load balancers
- DNS records
- Machine learning models and pipelines (kubeflow)
- DynamoDB Tables, S3 buckets, BigQuery Tables

# The CNCF and Kubernetes Community

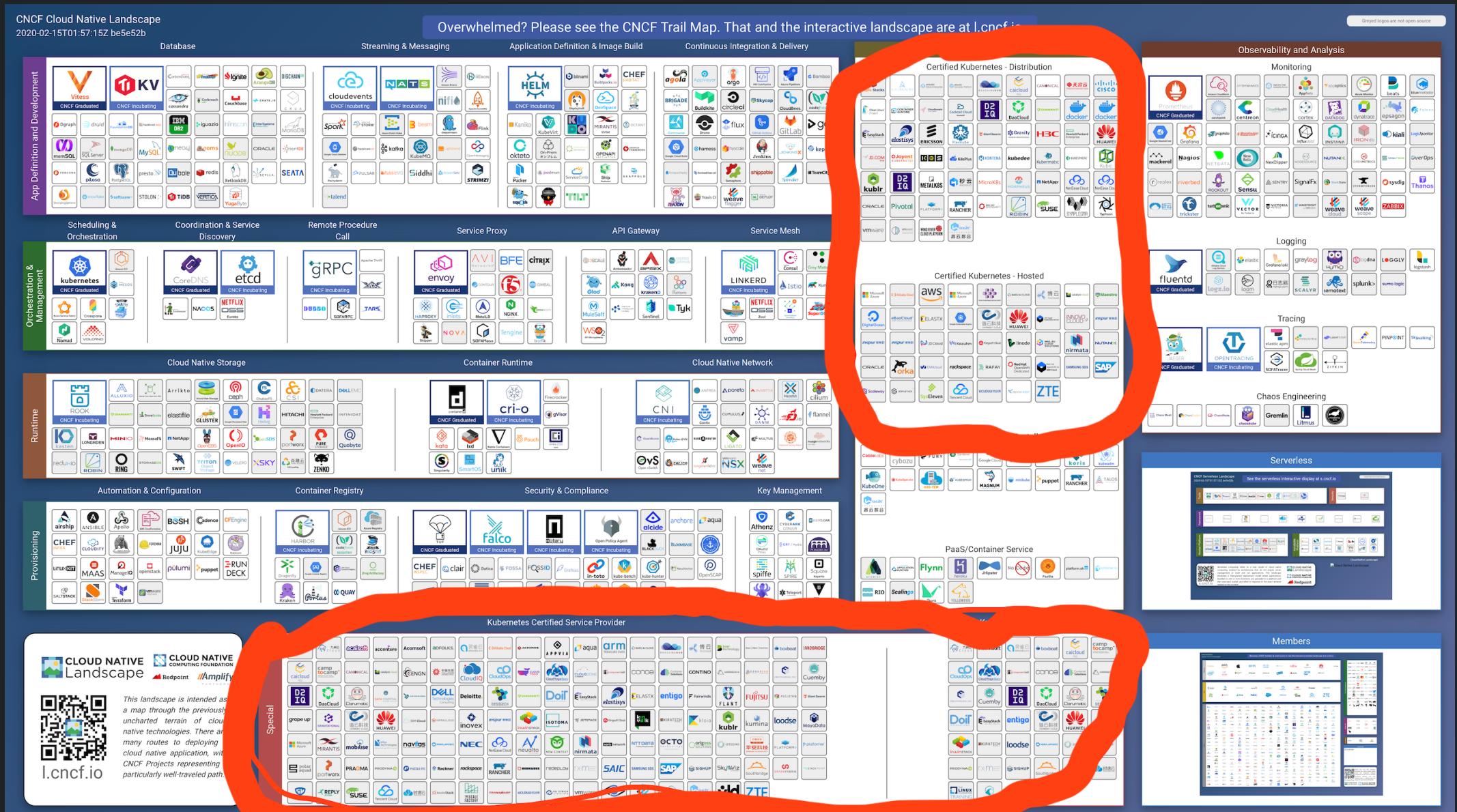


Source: CNCF Cloud Native Interactive Landscape

HOLY



# The CNCF and Kubernetes Community



Source: CNCF Cloud Native Interactive Landscape

## Recap:

### What to pitch people on when considering Kubernetes

- A simple, extendable API and client
- Learned Complexity
- Community

A large, heavily rusted shipwreck or hull is shown partially submerged in water, likely at a shipbreaking yard. The metal structures are dark brown and orange from corrosion. In the background, there are green trees and some industrial buildings under a clear sky.

# The risk of Kubernetes

**Don't Expose Kubernetes to Developers**

**Use  
Continuous  
Deployment**

Don't be a gatekeeper, though, curiosity drives innovation.

# Leaky Abstraction?!

## Kubernetes Has A Simple, Declarative Interface!!!

**Spoiler Alert:**

**No**  
**It**  
**doesn't**

# Spoiler Alert:

No

It

doesn't

**Spoiler Alert:**

**No**

**It**

**Doesn't**

# A "declarative" interface

One person's declarative is another person's imperative

— me

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    k8s-app: eviction-operator
  name: eviction-operator
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: eviction-operator
  template:
    metadata:
      labels:
        k8s-app: eviction-operator
    spec:
      containers:
        - image: quay.io/coryodaniel/eviction-operator:0.1.1
          name: eviction-operator
          resources:
            limits:
              cpu: 200m
              memory: 200Mi
            requests:
              cpu: 200m
              memory: 200Mi
          securityContext:
            allowPrivilegeEscalation: false
            readOnlyRootFilesystem: true
            runAsNonRoot: true
            runAsUser: 65534
      serviceAccountName: eviction-operator
```

B4cerpt from the eviction-operator manifest generated with Bonny. Originally ~180 lines.

F\*ck that noise.

— Johann Bach

**Ops wants  
Kubernetes,  
Devs want  
Heroku**

**ops wants  
Kubernetes,  
Devs want  
Heroku**

git push

so...  
.

# **BEAM and Kubernetes:** **Better Together?**

## @ WORK IN PROGRESS...

- Deployments, rollout, and resources (QoS)
  - busywait, schedulers
- Service discover, DNS, process handoff
- hpa/vpa
- affinity, anti-affinity (terminate a node)
- poddisruptionbudget
- [bonus] securityContext, PodSecurityPolicy, Distroless

Feature	Kubernetes	Beam or Erlang/OTP	Can k8s Help?
Automated Rollouts / Rollbacks	Deployment, ReplicaSet, StatefulSet	Hot Code Loading	✗
Automated Scheduling (instance placement)	pod/node affinity/anti-affinity, resources	-	✓
Bin Packing	Policy, LimitRange, resources	-	✓
Batch Jobs / Execution	Job, CronJob	Processes	✓
Service Discovery / DNS	Service, Endpoint, EndpointSlices, external-dns	-	✓
Load Balancing	internal & external load balancers	-	✓
Storage Orchestration	PersistentVolume, StorageClass, and CSI	-	✓
Secret "Management"	Secret	-	✓
Config Management	ConfigMap	vm.args, sys.config	✓
Health Checks	Shell, TCP, HTTP Health Checks	Supervisors, -heart	✓
Horizontal Scaling	HorizontalPodAutoscaler	Add nodes 👍	✓
Vertical Scaling	VerticalPodAutoscaler, resources	Add CPUs / RAM 👍	✓
QoS	PodDisruptionBudget, resources	-	✓
Security Templates	PodSecurityPolicy, securityContext	-	✓
Metrics	metric-server, custom-metrics-server	erlang:system_info/1 erlang:statistics/1	✓

# Thanks