



# 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?

## Features

## Kubernetes

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

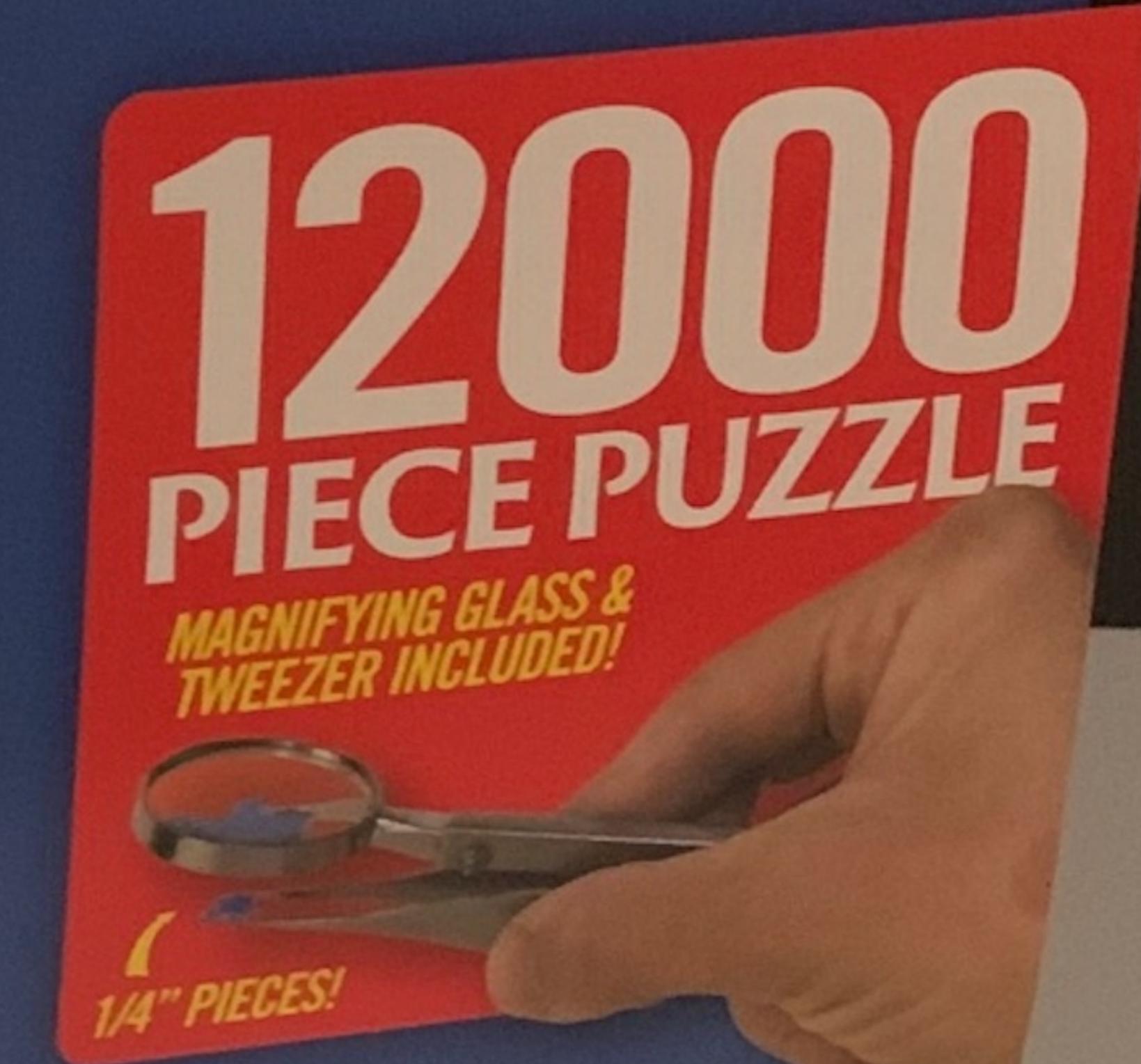
## On Kubernetes:



Your Startup's

**MVP**

On Kubernetes:



# Kubernetes Best Features

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

## Features

## Kubernetes

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

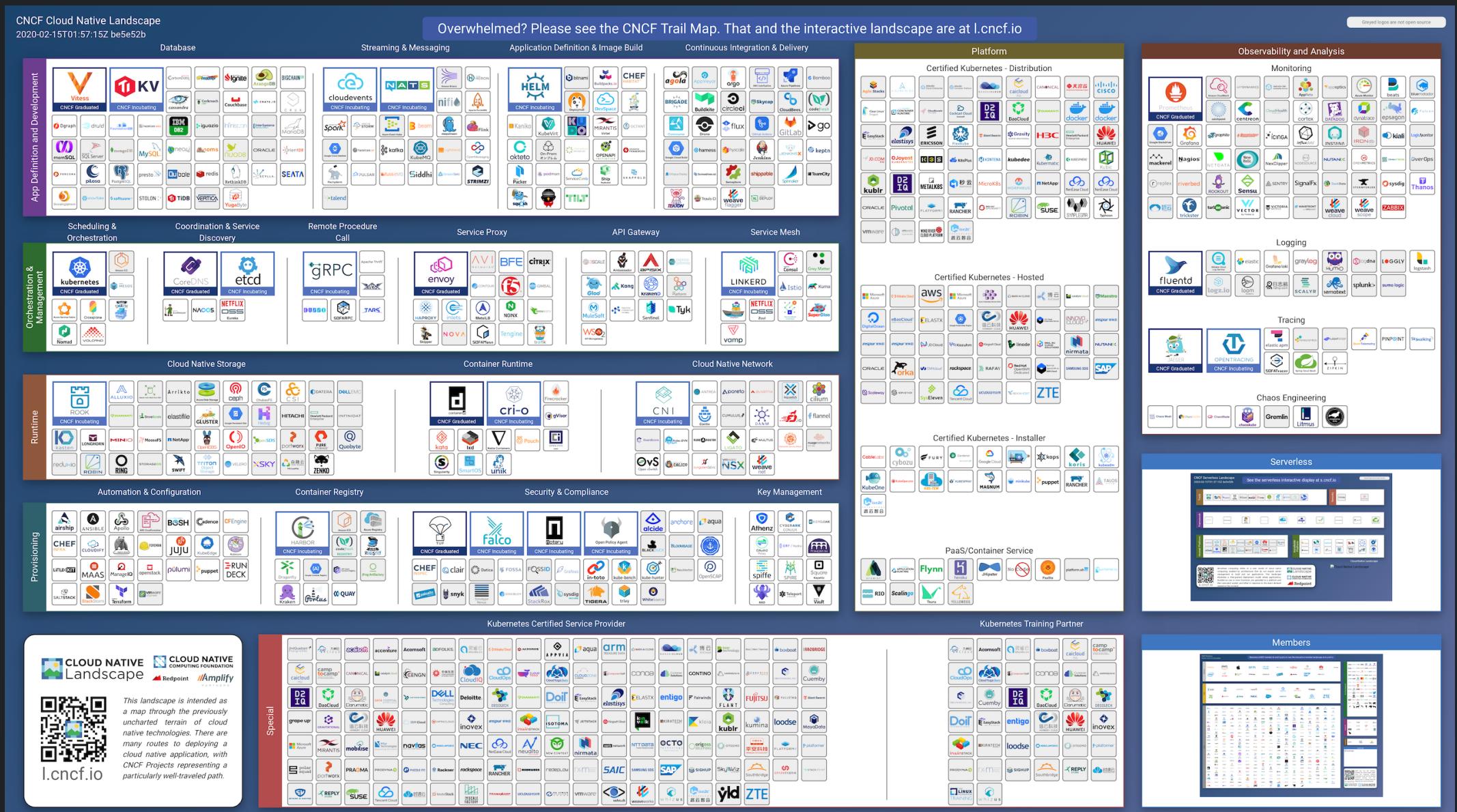
## Learned Complexity

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

# What can you deploy today?

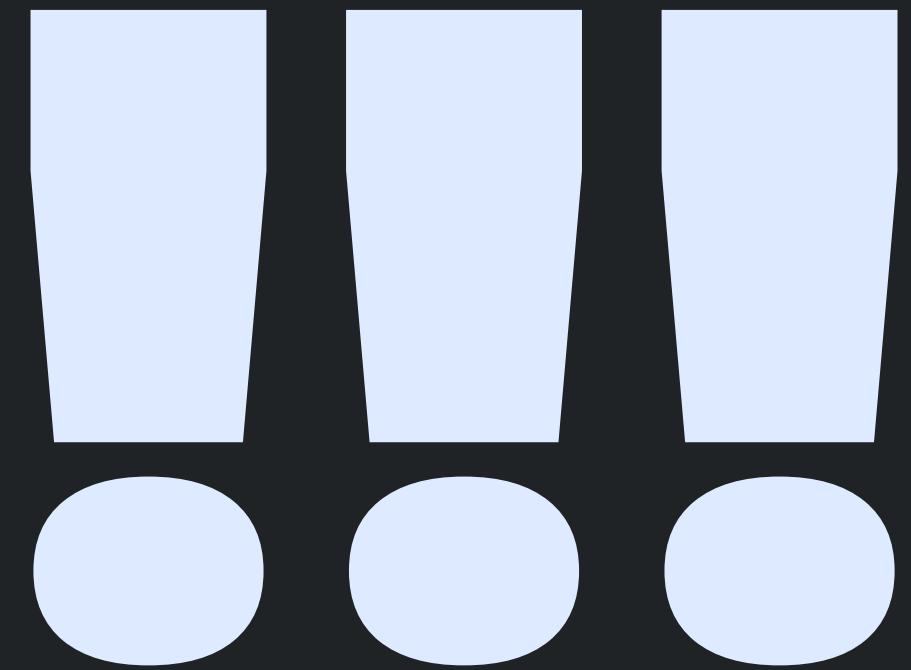
- Workloads
- Batch jobs
- Load balancers
- DNS records
- Machine learning models and pipelines (kubeflow)
- DynamoDB Tables, S3 buckets, BigQuery Tables

# The CNCF and Kubernetes Community

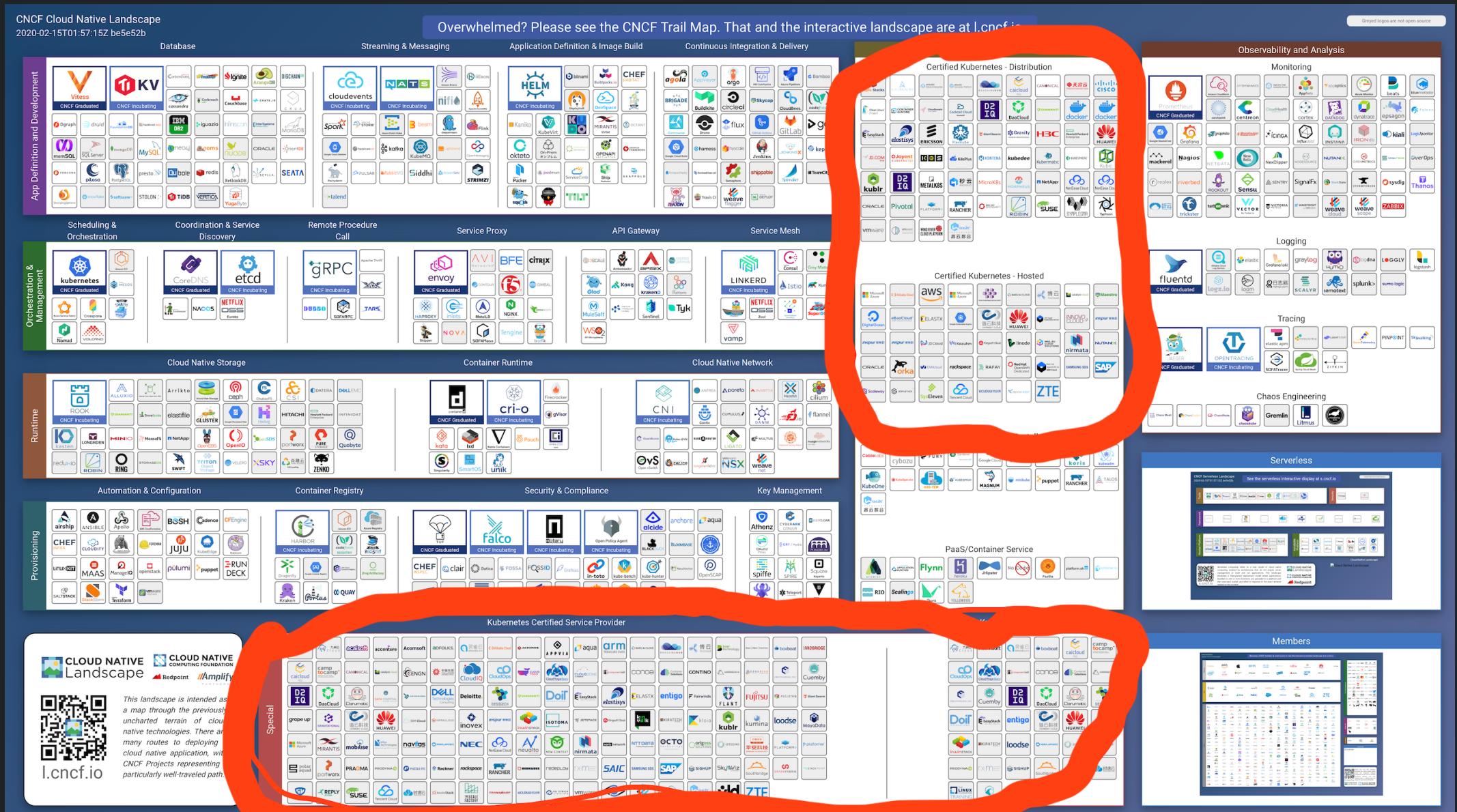


Source: CNCF Cloud Native Interactive Landscape

HOOHYY



# 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

# The Risk of Kubernetes



**Don't Expose Kubernetes to Developers**

**Use  
Continuous  
Delivery**

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



The background of the image is a dark, moody underwater environment. A large, multi-limbed sea creature, possibly a giant squid or章鱼, is the central figure. It has numerous tentacles and suction cups, some of which are illuminated with a faint blue light. Small fish are scattered throughout the water, and the overall atmosphere is mysterious and deep-sea-like.

**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
```

Excerpt 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?**

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	✓

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	-	✓
Service Discovery / DNS	Service, Endpoint, EndpointSlices, external-dns	-	✓
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	-	✓

**WARNING!**

*gamma ray*

**ALERT**

# Deployments

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  replicas: 3
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
```

# Pod Resources & QoS: BestEffort

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  replicas: 3
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
          resources: {}
```

# Pod Resources & QoS: Burstable

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  replicas: 3
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
          resources:
            limits:
              cpu: 500m
              memory: 200Mi
            requests:
              cpu: 250m
              memory: 200Mi
```

# Pod Resources & QoS: Guaranteed

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  replicas: 3
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
          resources:
            limits:
              cpu: 500m
              memory: 200Mi
```

"LimitRange can be used to apply resource requests and limits to all pods in a Namespace"

```
better_together master % k get pods
```

NAME	READY	STATUS	RESTARTS	AGE
better-together-8486b786dc-9tv6v	1/1	Running	0	19m
better-together-8486b786dc-hf9hc	1/1	Running	0	19m
better-together-8486b786dc-kqbv8	1/1	Running	0	19m

```
better_together master % k get pods/better-together-8486b786dc-9tv6v -oyaml | tail -n 10
```

```
state:
```

```
  running:
```

```
    startedAt: "2020-03-01T00:41:12Z"
```

```
hostIP: 10.128.0.6
```

```
phase: Running
```

```
podIP: 10.60.1.9
```

```
podIPs:
```

```
- ip: 10.60.1.9
```

```
qosClass: Guaranteed
```

```
startTime: "2020-03-01T00:41:08Z"
```

```
better_together master %
```

# QoS Summary

Requests	Limits	Class	CPU Affinity
none	none	BestEffort	✗
present	none	Burstable	✗
none or =limits	present	Guaranteed	✗
present, integer, and =limits	present	Guaranteed	✓

# Deployment Strategy

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  replicas: 4
  strategy:
    type: RollingUpdate # or Recreate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 50%
```

zsh

⌘1

zsh

⌘2

+

better\_together master % kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
better-together-6645c745b9-8rkqp	1/1	Running	0	89s
better-together-6645c745b9-f8v7x	1/1	Running	0	88s
better-together-6645c745b9-kzk7h	1/1	Running	0	89s
better-together-6645c745b9-ln6nx	1/1	Running	0	89s

better\_together master % █

better\_together master % kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
better-together-6645c745b9-8rkqp	1/1	Running	0	2m27s
better-together-6645c745b9-f8v7x	1/1	Running	0	2m26s
better-together-6645c745b9-kzk7h	1/1	Running	0	2m27s
better-together-6645c745b9-ln6nx	1/1	Running	0	2m27s

better\_together master % make deploy -s; kubectl get pods

service/better-together-headless unchanged

service/better-together-web unchanged

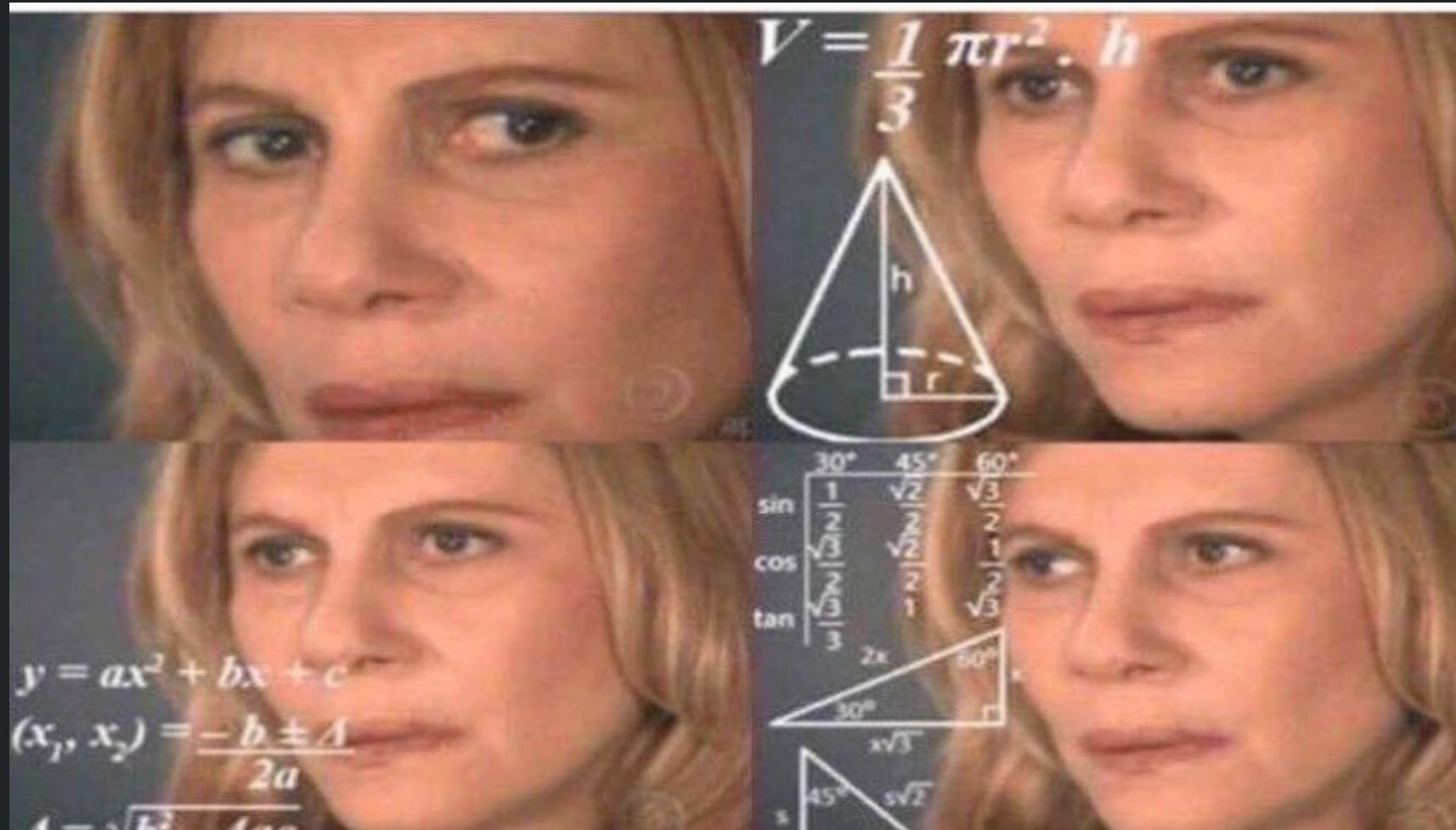
deployment.apps/better-together configured

NAME	READY	STATUS	RESTARTS	AGE
better-together-6645c745b9-8rkqp	1/1	Running	0	2m41s
better-together-6645c745b9-f8v7x	1/1	Running	0	2m40s
better-together-6645c745b9-kzk7h	1/1	Running	0	2m41s
better-together-6645c745b9-ln6nx	<u>1/1</u>	Running	0	2m41s
better-together-c4c6445d6-6ssm5	0/1	ContainerCreating	0	0s
better-together-c4c6445d6-snraqd	0/1	ContainerCreating	0	0s

better\_together master %

# Health Checks

A pod without health checks is the Math Lady of Kubernetes  
— Nick Young



# Health Checks

```
readinessProbe:  
  initialDelaySeconds: 5  
  periodSeconds: 60  
  httpGet:  
    path: /health  
    port: 4000  
livenessProbe:  
  initialDelaySeconds: 5  
  periodSeconds: 60  
  exec:  
    command:  
      - mix myapp.how_you_doing_fam
```

# Affinity

- Affinity or AntiAffinity
- Pod or Node
- Preferred or Required

# podAntiAffinity

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
      affinity:
        podAntiAffinity:
          preferredDuringSchedulingIgnoredDuringExecution:
            # requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - better-together
              topologyKey: "kubernetes.io/hostname"
            # topoloyKey: failure-domain.beta.kubernetes.io/zone
            # topologyKey: failure-domain.beta.kubernetes.io/region
```

# Horizontal and vertical Autoscalers

A large, multi-decked cruise ship is shown sailing on the ocean. The ship has several levels of decks and a prominent funnel. The water is slightly choppy, and the sky above is filled with scattered clouds.

# HPA: Pod CPU

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: better-together
  namespace: prod
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: better-together
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
    target:
      type: Utilization
      averageUtilization: 50
```

# HPA: Pod Custom Metrics

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: better-together
  namespace: prod
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: better-together
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Pods
    pods:
      metric:
        name: packets-per-second
    target:
      type: AverageValue
      averageValue: 1k
```

# HPA: External Metrics

```
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: my-app
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: my-app
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - external:
      metricName: my.external.kafka|consumer_group|lag
      metricSelector:
        matchLabels:
          resource.labels.topic: good-topic
      targetAverageValue: "2000"
    type: External
```

# Vertical Pod Autoscalers

```
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: better-together
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind: Deployment
    name: better-together
  updatePolicy:
  updateMode: "Off"
```

# VPA: recommendations

recommendation:

  containerRecommendations:

- containerName: my-container

  lowerBound:

    cpu: 536m

    memory: 262144k

  target:

    cpu: 587m

    memory: 262144k

  upperBound:

    cpu: 27854m

    memory: "545693548"

# Service Discovery



---

```
apiVersion: v1
kind: Service
metadata:
  name: better-together-epmd
  namespace: prod
spec:
  selector:
    app: better-together
  clusterIP: None
```

SVC\_NAME.NAMESPACE.svc.cluster.local  
better-together-epmd.prod.svc.cluster.local

# Service Discovery + libcluster

```
config :libcluster,
topologies: [
  k8s: [
    strategy: Cluster.Strategy.Kubernetes.DNS,
    config: [
      service: "better-together-epmd",
      application_name: "better_together",
      polling_interval: 10_000
    ]
  ]
]
```

```
iex(better_together@10.60.0.7)1> Node.list()
[:"better_together@10.60.0.6", :"better_together@10.60.1.10",
 :"better_together@10.60.2.4"]
iex(better_together@10.60.0.7)2> Node.list()
[:"better_together@10.60.0.6", :"better_together@10.60.1.10",
 :"better_together@10.60.2.4", :"better_together@10.60.1.11" ] ← new deploy
 :"better_together@10.60.2.5"] ← previous deploy
iex(better_together@10.60.0.7)3> *** ERROR: Shell process terminated! (^G to start new job) ***
command terminated with exit code 137
make: *** [shell] Error 137
better_together master %
```

# PriorityClass



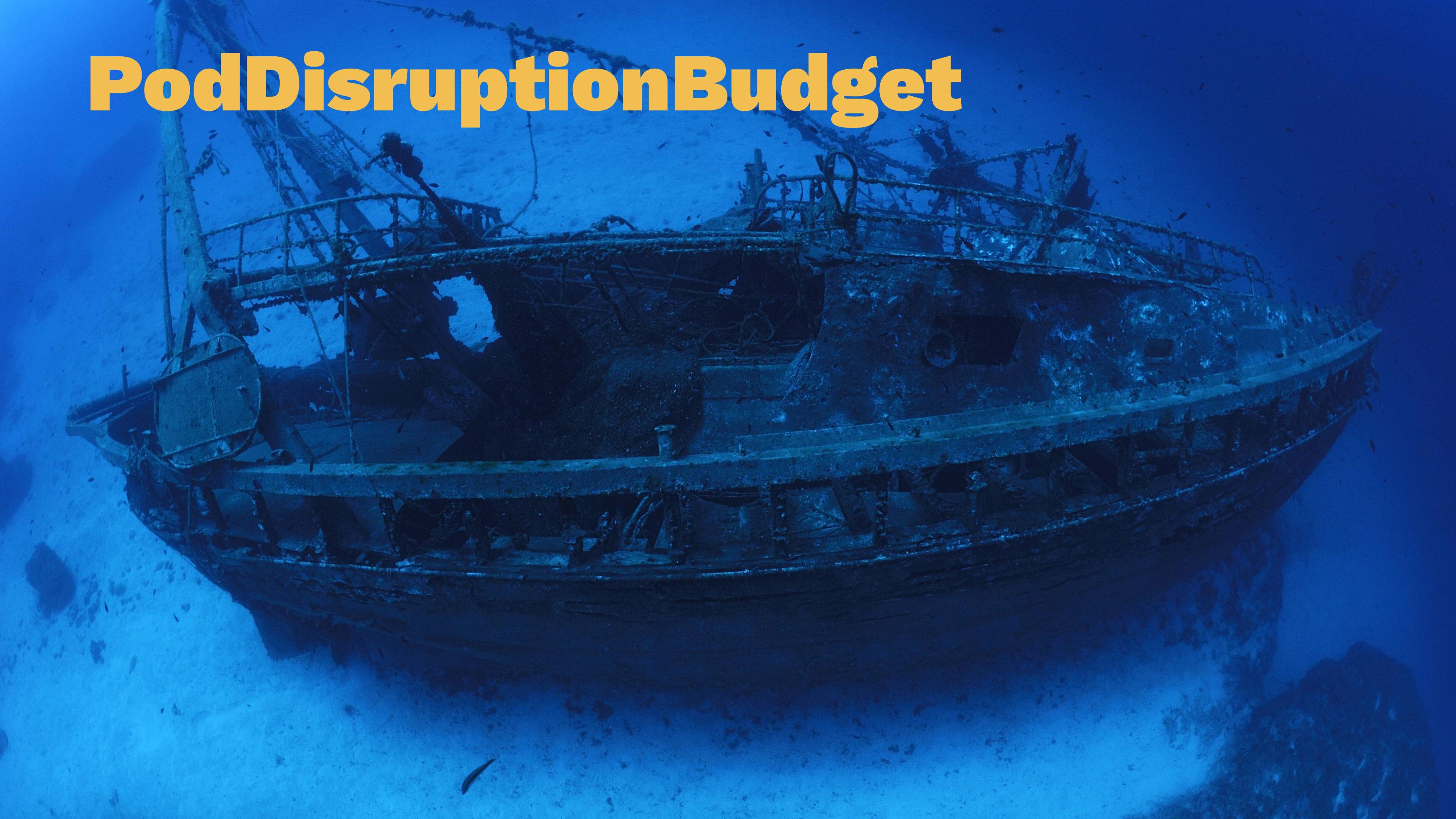
# PriorityClass

```
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: checkout-service
value: 1000000
globalDefault: false
description: "This priority class should be used by checkout service pods only."
```

# PriorityClass

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ecommerce-app
spec:
  template:
    spec:
      priorityClassName: checkout-service
    containers:
      - name: ecommerce-app
        image: quay.io/coryodaniel/ecommerce-app:latest
```

# PodDisruptionBudget

A photograph of a shipwreck underwater. The ship is a dark-colored wooden vessel, heavily damaged with large holes in its hull. A diver is visible near the bow. The water is a deep blue, and small fish can be seen swimming around the wreck.

# PodDisruptionBudget

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: front-end
spec:
  minAvailable: 2
  selector:
    matchLabels:
      app: front-end
```

# Bonus Round:

# Security

## Bonus Round: **Security**

- PodSecurityPolicy
- pod.spec.securityContext
- pod.spec.containers.securityContext

# securityContext

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: better-together
spec:
  template:
    spec:
      containers:
        - name: better-together
          image: quay.io/coryodaniel/better_together:latest
      securityContext:
        allowPrivilegeEscalation: false
        readOnlyRootFilesystem: true
        runAsNonRoot: true
        runAsUser: 65534
```

# Security Tools

- Distroless
- kubeaudit
- kube-bench
- kube-hunter

# Takeaways

1. Boats are friggin dangerous.
2. Kubernetes is complex, but learned complexity is a feature.
3. Simple, extendable API.
4. Community built, powerful features to make you go webscale.

**Bon voyage**

