



KubeCon



CloudNativeCon

Europe 2020

Virtual

Standardizing Applications for the Cloud at a Global Scale

Jared Watts, Upbound
Lei Zhang, Alibaba

How The Story Begin?



How Does Alibaba Ensure the Performance of System Components in a 10,000-node Kubernetes Cluster?

Alibaba Developer October 24, 2019 22,961 0

This article looks the problems and challenges that Alibaba Cloud overcame for Kubernetes to function at an ultra-large scale and the specific solutions proposed.

By Zeng Fansong, senior technical expert for the Alibaba Cloud Container Platform, and Chen Jun, systems technology expert at Ant Financial.

This article will take a look at some of the problems and challenges that Alibaba and its ecosystem partner [Ant Financial](#) had to overcome for [Kubernetes](#) to function properly at mass scale, and will cover the solutions proposed to the various problems the Alibaba engineers encountered. Some of these solutions include improvements to the underlying architecture of the Kubernetes deployment, such as enhancements to the performance and stability of etcd, the kube-apiserver, and kube-controller. These were all crucial for Alibaba to ensure the support needed for the 2019 [Tmall 618 Shopping Festival](#) to take full advantage of the 10,000-node Kubernetes cluster deployment. They are also important lessons for any enterprise interested in following Alibaba's footsteps.



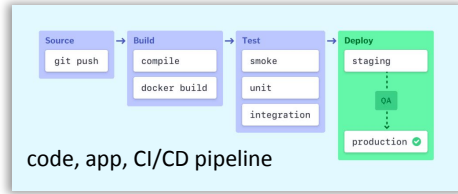
Me & team (*platform builders*)



My users (*developers, operators*)

What's Happened?

API & Primitives



users' expectation

Levels of Abstraction

scaling

- auto scale +100 instances when latency > 10%

rollout

- promote the canary instance with step of 10% when it passes baseline analysis

User Interfaces



GUI



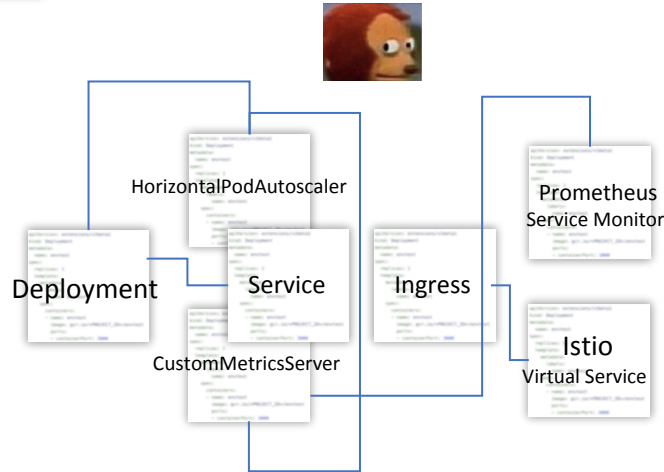
CLI



IaC

what we provide

Deployment	Pod
HPA	Controller
Sidcar	Node
NetworkPolicy	CR/CRD



```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-depl
spec:
  revisionHistoryLimit: 10
  minReadySeconds: 30
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.19.0
          ports:
            - containerPort: 80
  app: nginx
  deployment: distelli

apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  revisionHistoryLimit: 5
  minReadySeconds: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.19.0
          ports:
            - containerPort: 80
  app: nginx
  deployment: distelli
```



KubeCon



CloudNativeCon

Europe 2020

Virtual



Let's Build K8s App Platforms for Users!



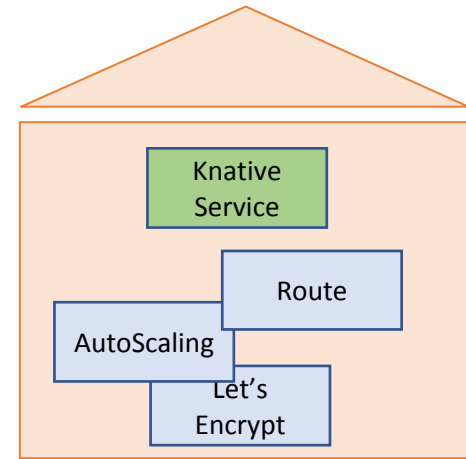
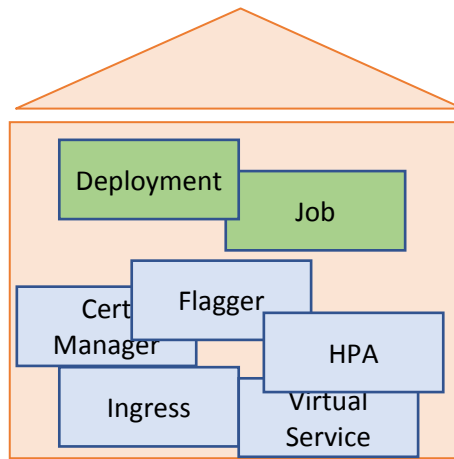
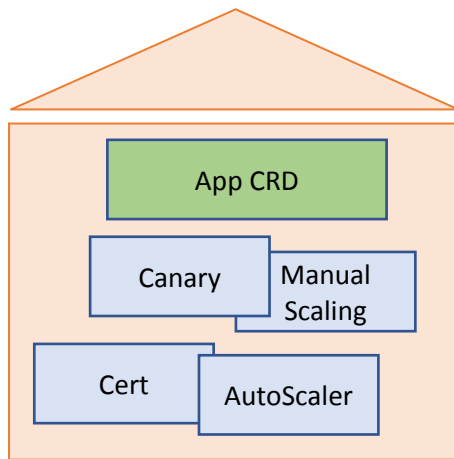
Early 2019: Silos Created ...



I run stateful workloads!

I run stateless apps!

I run stateless serverless containers!



Kubernetes

- Fragmentation:** ~11 PaaS/Serverless
- Silos:** no interoperability, reusability, or portability
- Close:** many in-house wheels due to in-house app crd



Can we build application platforms based on k8s, which are **user friendly, highly extensionable, in a standard approach?**

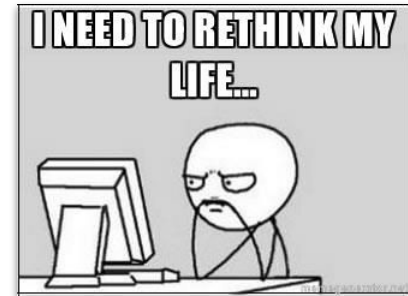
Build abstractions!

Knative, OpenFaaS, or
DIY your own
abstraction!

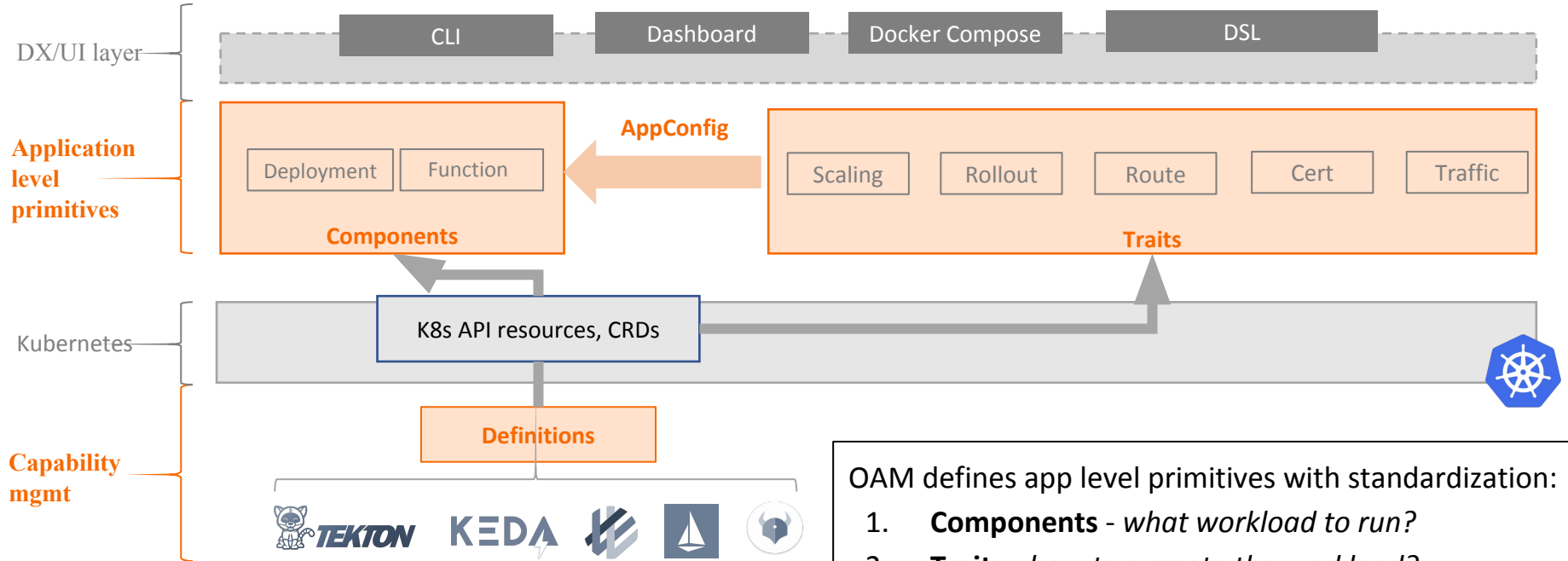
Leverage k8s
extensionability!

Container, ksvc, VM,
auto scaling, manual
scaling, canary
blue-green, just name it!

How ???



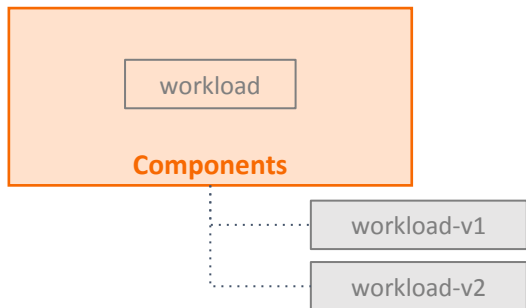
Open Application Model (OAM)



OAM defines app level primitives with standardization:

1. **Components** - what workload to run?
2. **Traits** - how to operate the workload?
3. **AppConfig** - bind trait with component
4. **Definitions** - register CRD as workload/trait

Components



Component is versionized template for your workload

```
$ kubectl get components
```

NAME	WORKLOAD TYPE
frontend	deployment.apps.k8s.io

```
$ kubectl get deployment
```

NAME	REVISION	AGE
frontend-c8bb659c5	1	2d15h
frontend-a8eb65xfe	2	10m

```
apiVersion: core.oam.dev/v1alpha2
kind: Component
metadata:
  name: frontend
  annotations:
    description: Container workload
spec:
  workload:
    apiVersion: apps/v1
    kind: Deployment
    spec:
      template:
        spec:
          containers:
            - name: web
              image: 'php:latest'
              env:
                - name: OAM_TEXTURE
                  value: texture.jpg
          ports:
            - containerPort: 8001
              name: http
              protocol: TCP
```

Persona: App Developer

Workloads

Developers are free to define **workloads at any abstraction level**, including cloud resources.

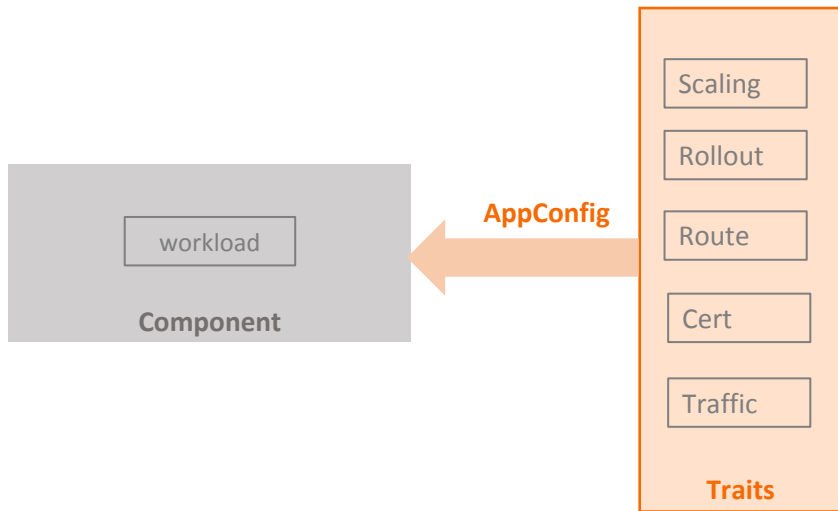
```
apiVersion: core.oam.dev/v1alpha2
kind: Component
metadata:
  name: frontend
  annotations:
    description: Container
workload
spec:
  workload:
    apiVersion: apps/v1
    kind: Deployment
    spec:
      replicas: 3
      selector:
        matchLabels: app: nginx
      template:
        metadata:
          labels:
            app: nginx
        spec:
          containers:
            - name: nginx
              image: nginx:1.14.2
              ports:
                - containerPort: 80
```

Abstraction level: **low**

```
apiVersion: core.oam.dev/v1alpha2
kind: Component
metadata:
  name: frontend
  annotations:
    description: Container workload
spec:
  workload:
    apiVersion: apps.alibaba-inc/v1
    kind: Containerized
    spec:
      image: nginx:1.14.2
      deploy:
        replicas: 3
```

Abstraction level: **high**

Traits and AppConfig



- Traits
 - Declarative abstractions for operational capabilities
- AppConfig (Application Configuration)
 - Bind given trait to component

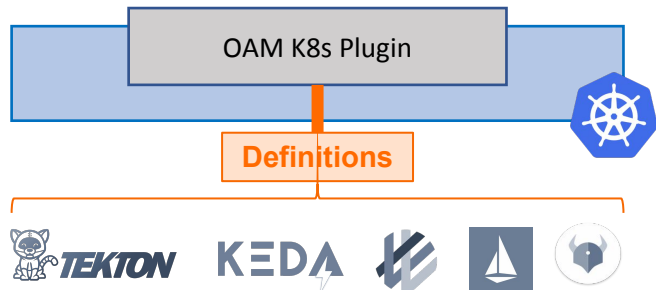
```
apiVersion: core.oam.dev/v1alpha2
kind: ApplicationConfiguration
metadata:
  name: helloworld
spec:
  components:
    # 1st component
    - componentName: frontend
      traits:
        - trait:
            apiVersion: autoscaling/v2beta2
            kind: HorizontalPodAutoscaler
            spec:
              minReplicas: 1
              maxReplicas: 10
        - trait:
            apiVersion: networking.alibaba-inc.com/v1
            kind: APIGateway
            spec:
              hostname: app.alibaba.com
              path: /
              service_port: 8001
    # 2nd component
    - componentName: redis
```

Persona: App Operator

Definitions



Register and discover k8s capabilities
(API resources) as workloads or traits



```
apiVersion: core.oam.dev/v1alpha2
kind: TraitDefinition
metadata:
  name: virtualservices.networking.istio.io
  annotations:
    alias: traffic
spec:
  appliesTo:
    - *.apps.k8s.io
  conflictsWith:
    - traffic-split.alimesh.io
  definition: virtualservices.networking.istio.io
```

Persona: Platform Builder/Infra Operator

e.g.: Register Istio VirtualService as Traffic trait

\$ kubectl get traits

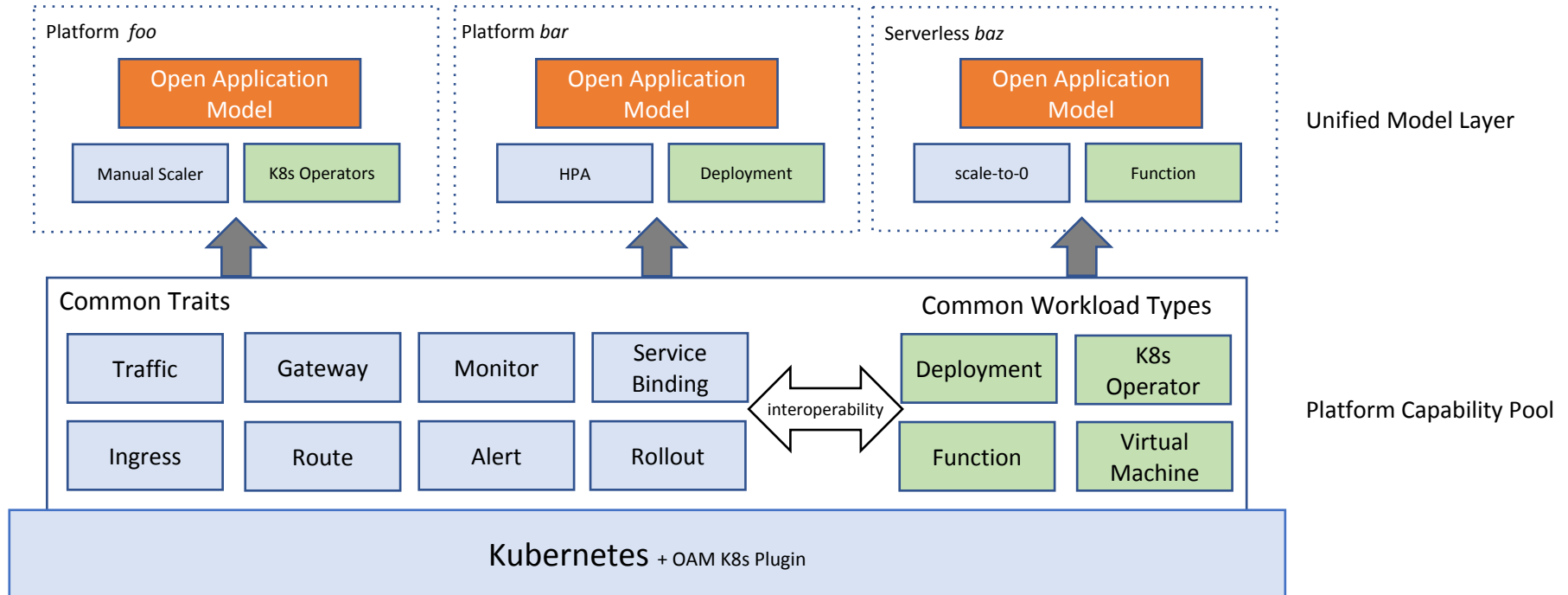
NAME	DEFINITION	APPLIES TO	CONFLICTS WITH
traffic	virtualservices.networking.istio.io	*.apps.k8s.io	traffic-split.alimesh.io
route	route.core.oam.dev	*.apps.k8s.io	
cert	cert.core.oam.dev	*.apps.k8s.io	

In 2020: Build Standard Platforms



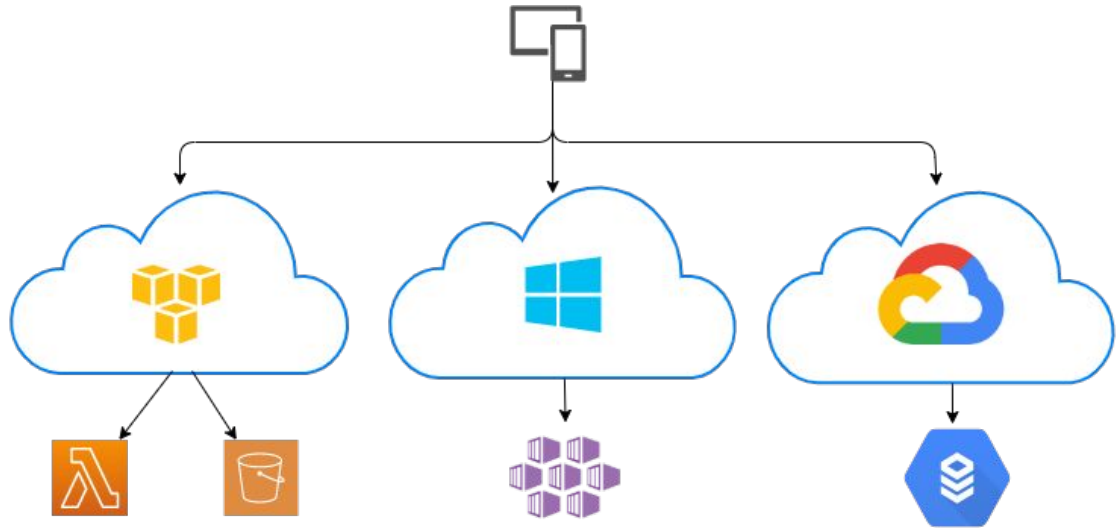
Open Application Model (OAM) is:

1. a building block to create standard app platforms
 - a. with developer centric primitives and your own level of abstraction
2. a standard and runtime agnostic app definition
 - a. enable global scale app distribution



Globally Distributed Apps

- A lot of real life applications don't live in just 1 place
- App components and infrastructure can be spread across:
 - Cloud providers
 - Regions and zones
 - Clusters
- Justifications
 - Availability
 - Resiliency
 - Cost
 - Unique services



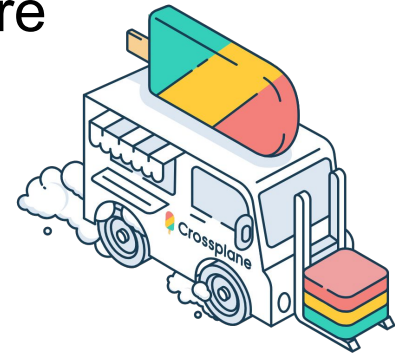
Global Application Challenges



- Know who and what you're building for - act with intention
- Understand the infrastructure needs of your app
- Tool explosion
 - how many dashboards, GUIs, consoles do you want to live in?
 - how many skills sets do you need to learn or hire for?
- Monitoring, management, policy, operations...

- A set of components that make management and orchestration decisions for the **entire** solution
- **Centralizes** decision making into an authoritative place
 - Single entry point API
- Kubernetes has a control plane
 - schedules pods/resources across nodes
- Global control plane is similar - but bigger scope
 - Provision infrastructure needed by app
 - Deploy application components to clouds, regions, etc.

- CNCF sandbox project
- Open source **control plane** for applications and their infrastructure
 - Based on Kubernetes control plane
- Provision infrastructure declaratively using the K8s API
- Publish your own declarative infrastructure API without code
 - self-service, on-demand, policy, config, best practices
- Run and deploy applications alongside infrastructure
 - **The** OAM implementation for Kubernetes



Consuming Infrastructure



KubeCon



CloudNativeCon

Europe 2020

Virtual

- Standardized apps need infrastructure - databases, caches, buckets, networking, etc.
- **Control plane API** - centralized place for provisioning & consuming infrastructure
- Define the right API for your organization
 - Abstractions to
 - hide complexity & environment knowledge
 - codify policy & best practices
 - enable self-service by apps

Consuming Infrastructure



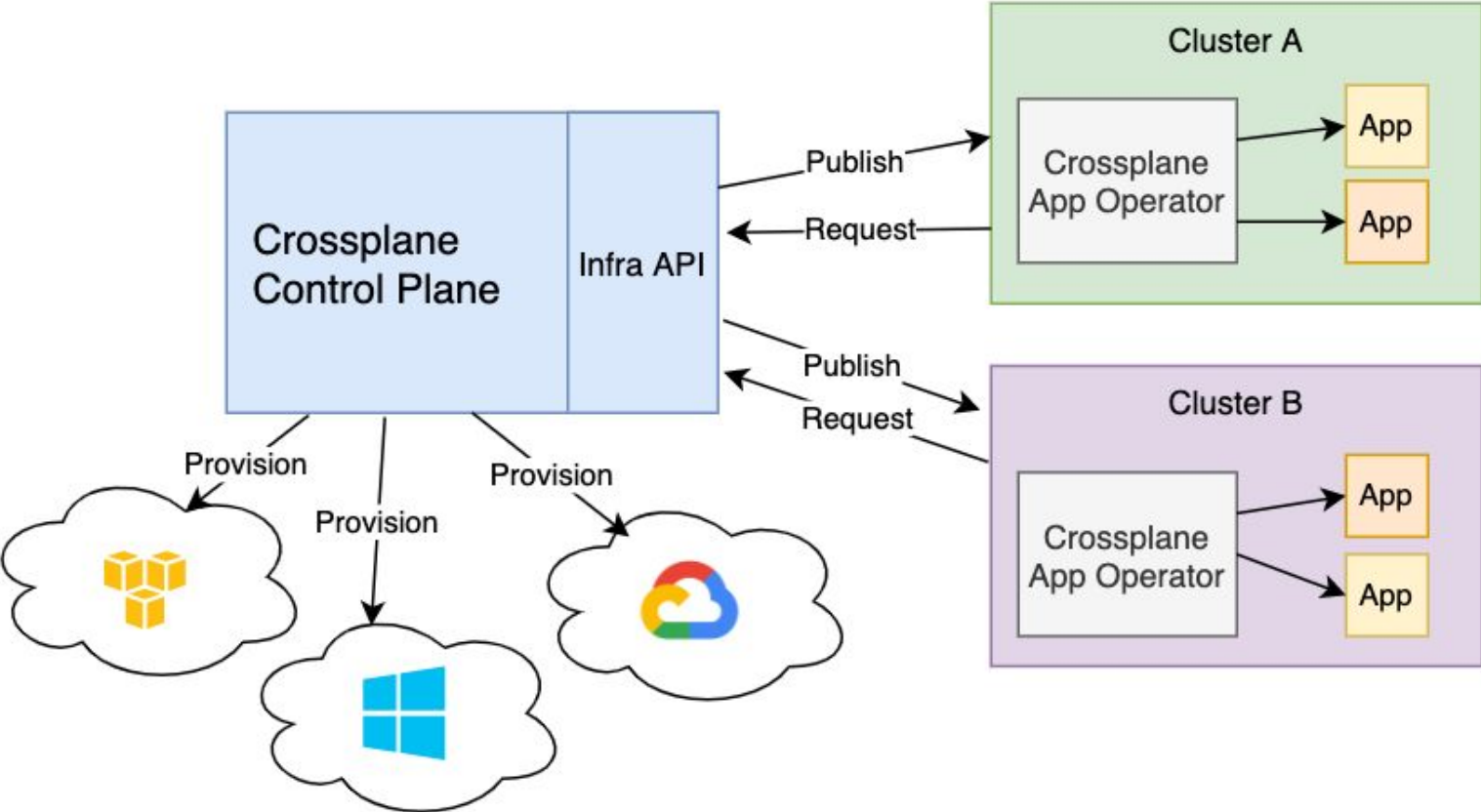
KubeCon



CloudNativeCon

Europe 2020

Virtual





KubeCon



CloudNativeCon

Europe 2020

Virtual

DEMO

Building a global control plane API for standardized apps and infrastructure

Get Involved!



KubeCon



CloudNativeCon

Europe 2020

Virtual

- **OAM**

- <https://oam.dev/>
- <https://github.com/oam-dev/spec>
- Gitter - <https://gitter.im/oam-dev/>
- Twitter - https://twitter.com/oam_dev
- [Community meetings](#)

- **Crossplane**

- <https://crossplane.io/>
 - Try it out with the quick start docs!
- <https://github.com/crossplane/crossplane>
- Slack - <https://slack.crossplane.io/>
- Twitter - https://twitter.com/crossplane_io
- [Community meetings](#) and [live streams](#)



KubeCon



CloudNativeCon

Europe 2020

Virtual

Questions?

<https://crossplane.io/>

<https://oam.dev/>



KubeCon



CloudNativeCon

Europe 2020

Virtual

Thank you!

<https://crossplane.io/>

<https://oam.dev/>



KubeCon



CloudNativeCon

Europe 2020



Virtual



KEEP CLOUD NATIVE

CONNECTED

