



KubeCon



CloudNativeCon

Europe 2021

*Virtual*



*Forward Together »*



KubeCon



CloudNativeCon

Europe 2021

# Beyond Federation

Automating Multi-cloud Workloads with K8s Native APIs

*Kevin Wang (Zefeng) @kevin-wangzefeng*

*Virtual*





KubeCon

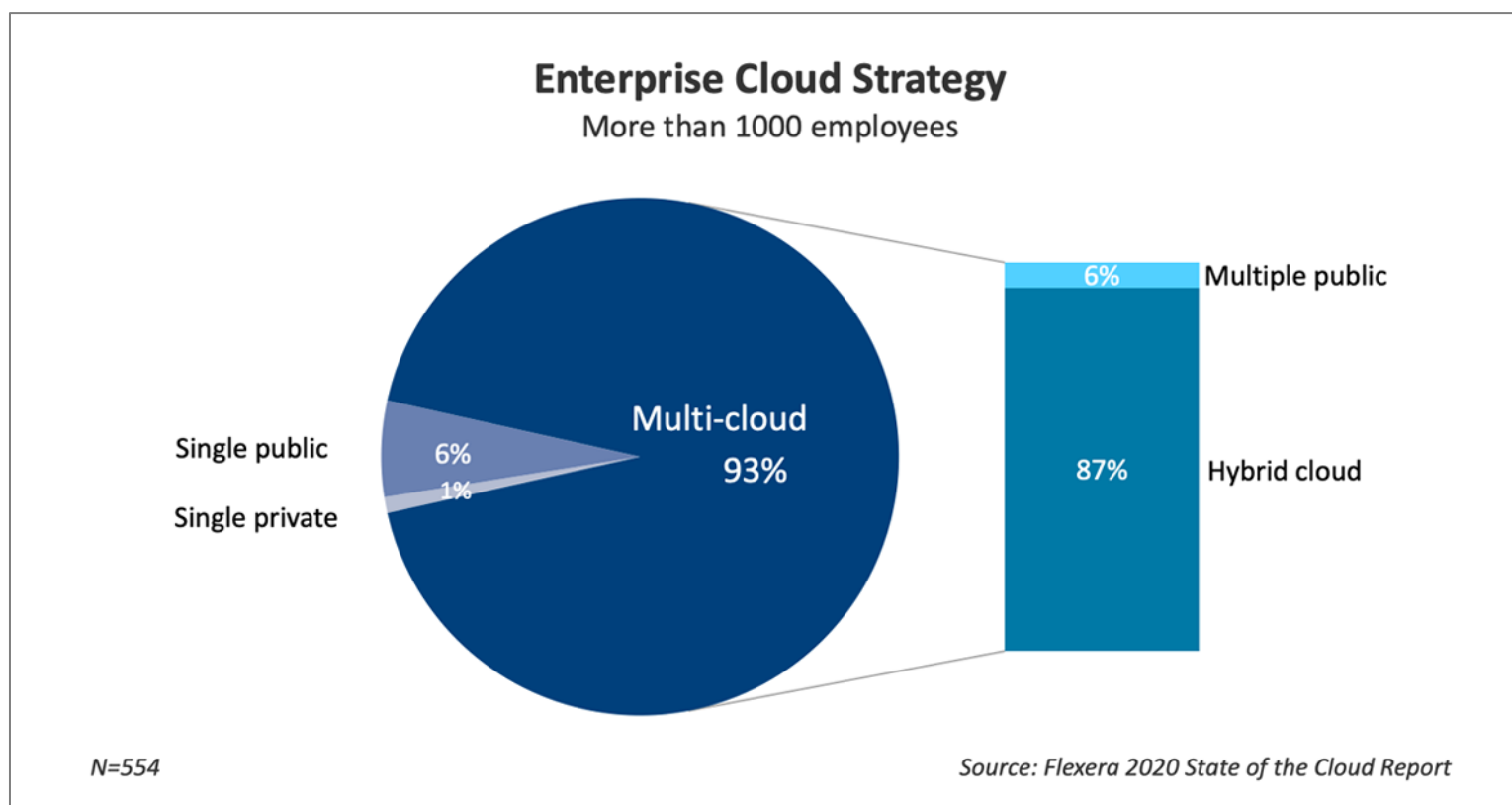


CloudNativeCon

Europe 2021

Virtual

# Multi-cloud has become the dominant enterprise strategy



# But Cloud-Native Multi-Cloud Is Challenging



KubeCon



CloudNativeCon

Europe 2021

*Virtual*

## Challenges of managing multiple container clusters

---

### Too Many Clusters

- Cumbersome and Repetitive setup
- Incompatible Cluster Lifecycle API
- Fragmented API endpoints

### Fragmentation of YAMLs

- Per-cluster customization for Apps
- Multi-cluster service discovery for Apps
- Sync Apps between clusters

### Boundary of Clusters

- Resource Scheduling
- Application Availability
- Horizontal Auto-scaling

### Vendor lock-in

- Deployment Gravity
- Lack of Migration Automation
- Lack of independent, neutral, open source multi-cluster management projects

# Lessons learnt from previous work



KubeCon



CloudNativeCon

Europe 2021

*Virtual*

- **Incompatible APIs don't help**
  - Coupled, Incompatible APIs, require extra learning and adoption efforts
- **Duplicated configs among similar applications are boring**
  - 1:1 mapping of federation API and workload, always too many fields to fill up
- **Building Blocks are Insufficient**
  - Too many customizations result in no standard

# Karmada: Open, Cloud-Native, Multi-Cloud Orchestration Engine



KubeCon



CloudNativeCon

Europe 2021

*Virtual*



Easily build infinitely scalable cluster pools with Karmada

Use multi-cloud clusters just like a single K8s cluster

## K8s Native API Compatible

Zero change upgrade: single-cluster → multi-cluster  
Seamless integration of existing K8s tool chain

## Out of the Box

Built-in policy sets for scenarios:  
Active-active, Remote DR, Geo Redundant

## Avoid Vendor Lock-in

Integration with mainstream cloud providers  
Automatic allocation, migration across clusters  
Not tied to proprietary vendor orchestration

## Fruitful Multi-Cluster Scheduling Policies

Cluster Affinity, Multi Cluster Splitting/Rebalancing,  
Multi-Dimension HA: Region/AZ/Cluster/Provider

## Centralized Management

Cluster location agnostic  
Support clusters in Public cloud, on-prem or edge

## Open and Neutral

Jointly initiated by Internet, finance,  
manufacturing, telecom, cloud providers, etc.  
Target for open governance with CNCF

# Karmada Architecture



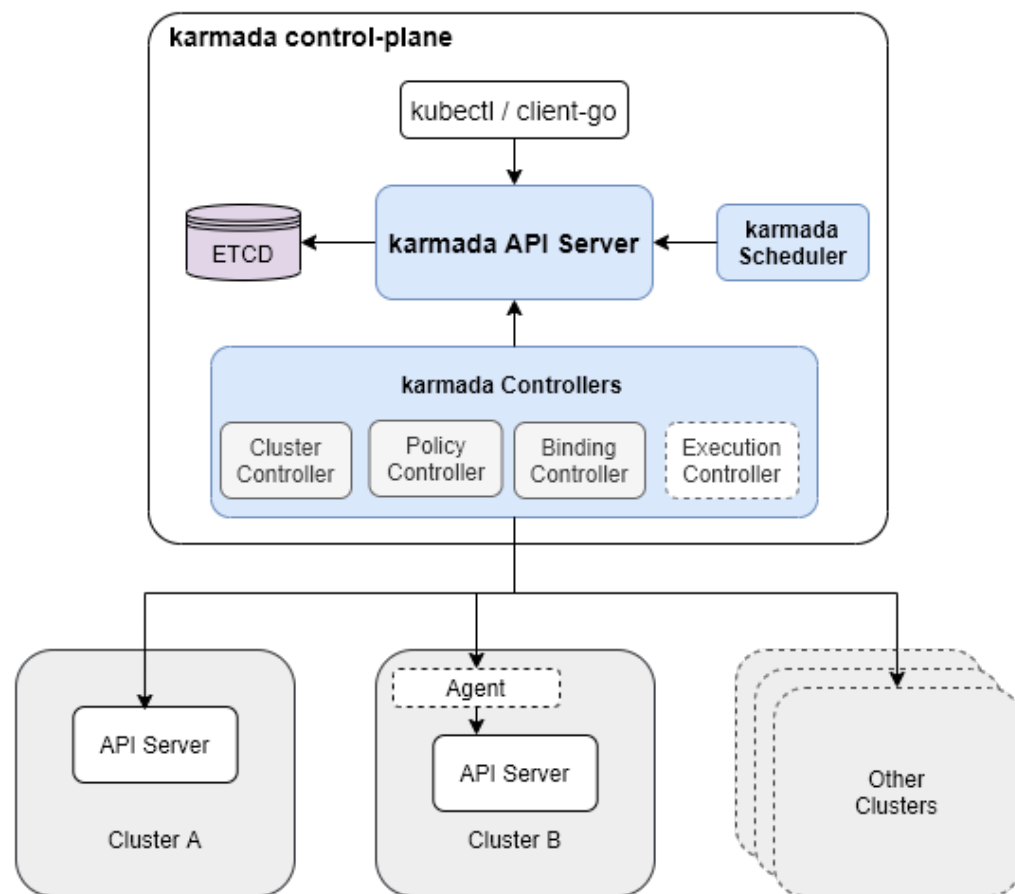
KubeCon



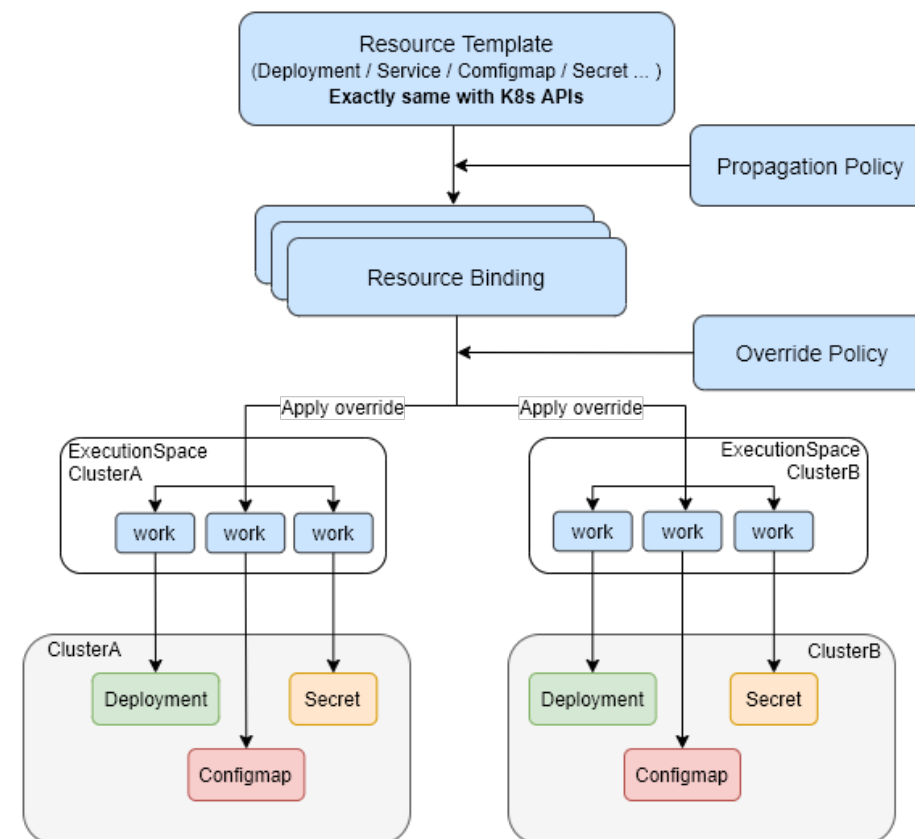
CloudNativeCon

Europe 2021

Virtual



## Karmada Concepts



# Propagation Policy



KubeCon



CloudNativeCon

Europe 2021

*Virtual*

```
apiVersion: policy.karmada.io/v1alpha1
kind: PropagationPolicy
metadata:
  name: multi-zone-replication
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      labelSelector:
        matchLabels:
          ha-mode: multi-zone-replication
  placement:
    spreadConstraints:
      - spreadByField: zone
        maxGroups: 3
        minGroups: 3
```

Example:

Reusable propagation policy for all apps that need multi AZ HA



# Override Policy



KubeCon



CloudNativeCon

Europe 2021

*Virtual*

```
apiVersion: policy.karmada.io/v1alpha1
kind: OverridePolicy
metadata:
  name: example-override
  namespace: default
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
  targetCluster:
    labelSelector:
      matchLabels:
        failedomain.kubernetes.io/region: dc1
  overrides:
    imageOverride:
      - component: prefix
        operator: replace
        value: "dc-1.registry.io"
```

Example:

Reusable override policy to rewrite all deployments' image prefix going into dc1

# Resource template



KubeCon



CloudNativeCon

Europe 2021

*Virtual*

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  app: nginx
  ha-mode: multi-zone-replication
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

```
export KUBECONFIG=~/.karmada/config
```

```
kubectl create -f nginx-deployment.yaml
```

Exactly the same yaml that you apply to Kubernetes

# Try it out!



KubeCon



CloudNativeCon

Europe 2021

*Virtual*



<https://github.com/karmada-io/karmada>



<https://karmada-io.slack.com>

Relevant Session at KubeCon:

[Sponsored Session: Huawei - Zero Change Transitioning to Multi-Cloud Architecture](#)

**Tuesday**, May 4 • 10:00 - **Friday**, May 7 • 15:10 CEST