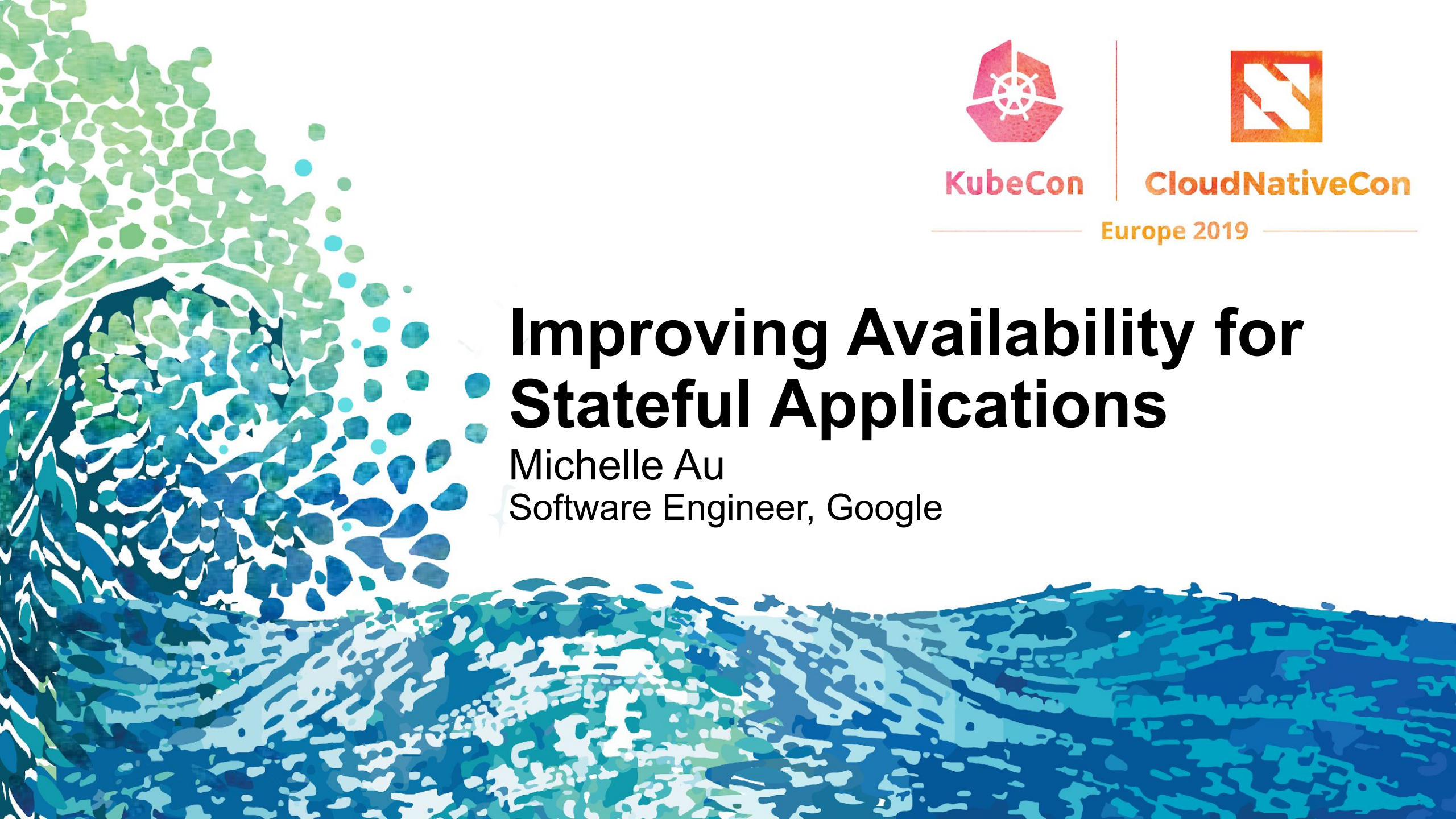


KubeCon | CloudNativeCon Europe 2019

Improving Availability for Stateful Applications

Michelle Au
Software Engineer, Google

Agenda

Persistent storage options

Building highly available stateful applications

- Failure domain spreading
- Demo
- Pod downtime and recovery

Persistent Storage Options

Supported Storage Systems

In-tree Drivers

- https://kubernetes.io/docs/concepts/storage/#types-of-volumes
- Over 15!

CSI Drivers

- https://kubernetes-csi.github.io/docs/drivers.html
- Over 35!

Wide range of characteristics

- Local vs remote, cloud vs appliance vs software-defined, distributed vs hyper-converged, etc.

Storage Characteristics

Accessibility

- At what granularity does your app have to be co-located with storage?

Availability

- At what granularity is storage still available during an outage?

Durability

- Under what conditions could my data be lost?

Access Mode

- How many nodes can access the volume concurrently?

Storage Characteristics

Performance

- Read/write/mixed IOPS and throughput

Cost

- Including operation, maintenance

Examples

Example	Accessibility	Availability	Durability	Access Mode	Performance	Cost
Local disk	Single node	Single node	Single disk*	Single node	Best	$

* Most cloud local disks are not durable beyond VM

Examples

Example	Accessibility	Availability	Durability	Access Mode	Performance	Cost
Local disk	Single node	Single node	Single disk*	Single node	Best	$
Cloud disk	Single zone	Single zone	3x	Single node	Better	$$

* Most cloud local disks are not durable beyond VM

Examples

Example	Accessibility	Availability	Durability	Access Mode	Performance	Cost
Local disk	Single node	Single node	Single disk*	Single node	Best	$
Cloud disk	Single zone	Single zone	3x	Single node	Better	$$
Replicated cloud disk	Multi zone	Multi zone	3x	Single node	Good	$$$

* Most cloud local disks are not durable beyond VM

Examples

Example	Accessibility	Availability	Durability	Access Mode	Performance	Cost
Local disk	Single node	Single node	Single disk*	Single node	Best	$
Cloud disk	Single zone	Single zone	3x	Single node	Better	$$
Replicated cloud disk	Multi zone	Multi zone	3x	Single node	Good	$$$
Single NFS	Global	Single server	Varies	Multi node	Good	$$$

* Most cloud local disks are not durable beyond VM

Examples

Example	Accessibility	Availability	Durability	Access Mode	Performance	Cost
Local disk	Single node	Single node	Single disk*	Single node	Best	$
Cloud disk	Single zone	Single zone	3x	Single node	Better	$$
Replicated cloud disk	Multi zone	Multi zone	3x	Single node	Good	$$$
Single NFS	Global	Single server	Varies	Multi node	Good	$$$
Scaleout/HA Filer	Global	Global	Varies	Multi node	Varies	$$$$

* Most cloud local disks are not durable beyond VM

Building Highly-Available Stateful Applications

Pod Anti-Affinity

Spread replicas across failure domains

```
affinity:
   podAntiAffinity:
     requiredDuringSchedulingIgnoredDuringExecution:
     - topologyKey: failure-domain.beta.kubernetes.io/zone
       labelSelector:
         matchExpressions:
         - key: app
           operator: In
           values:
           - my-app
```

12 Factor Model

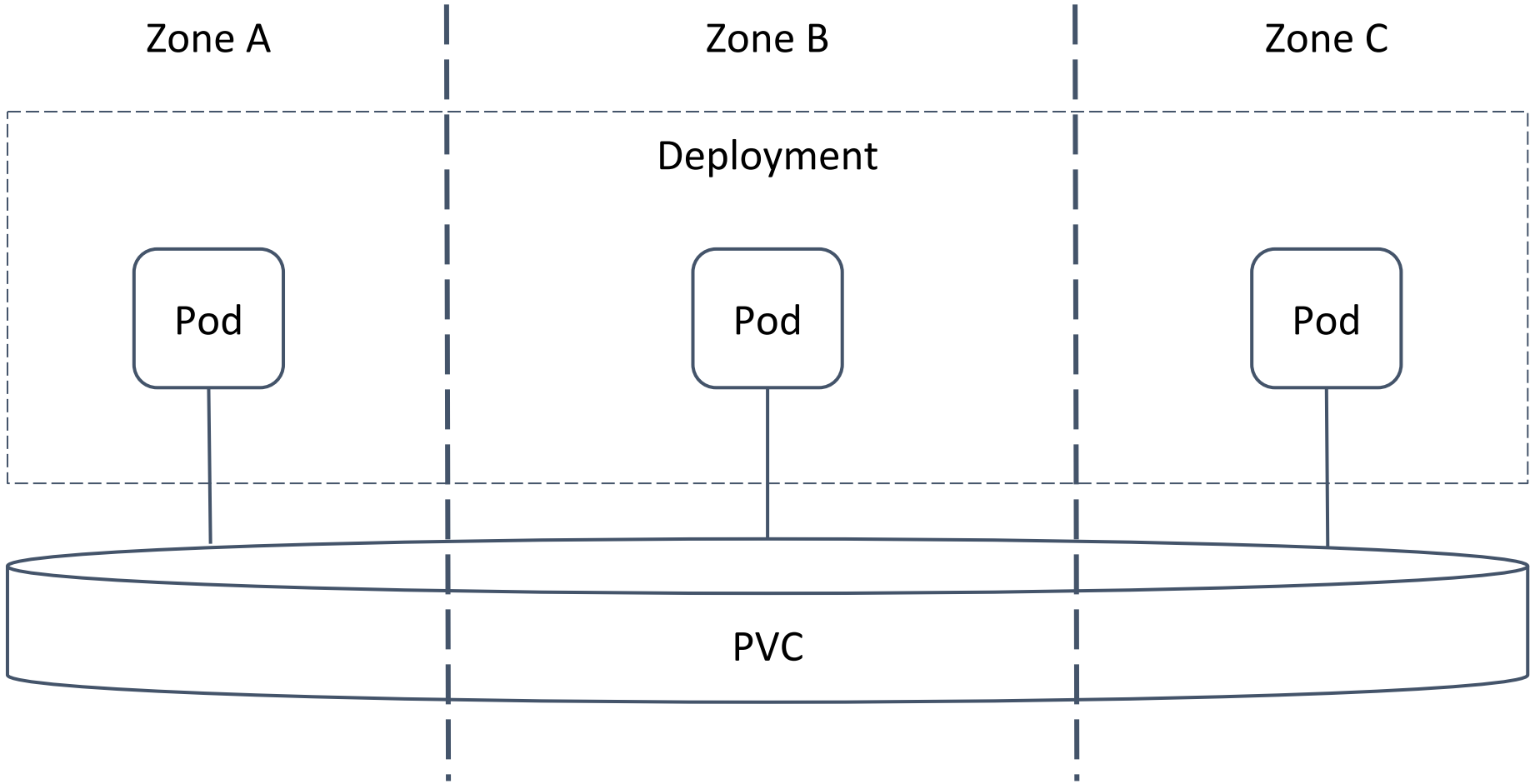
All replicas share the same data

- Example: Content Management Systems (CMS)

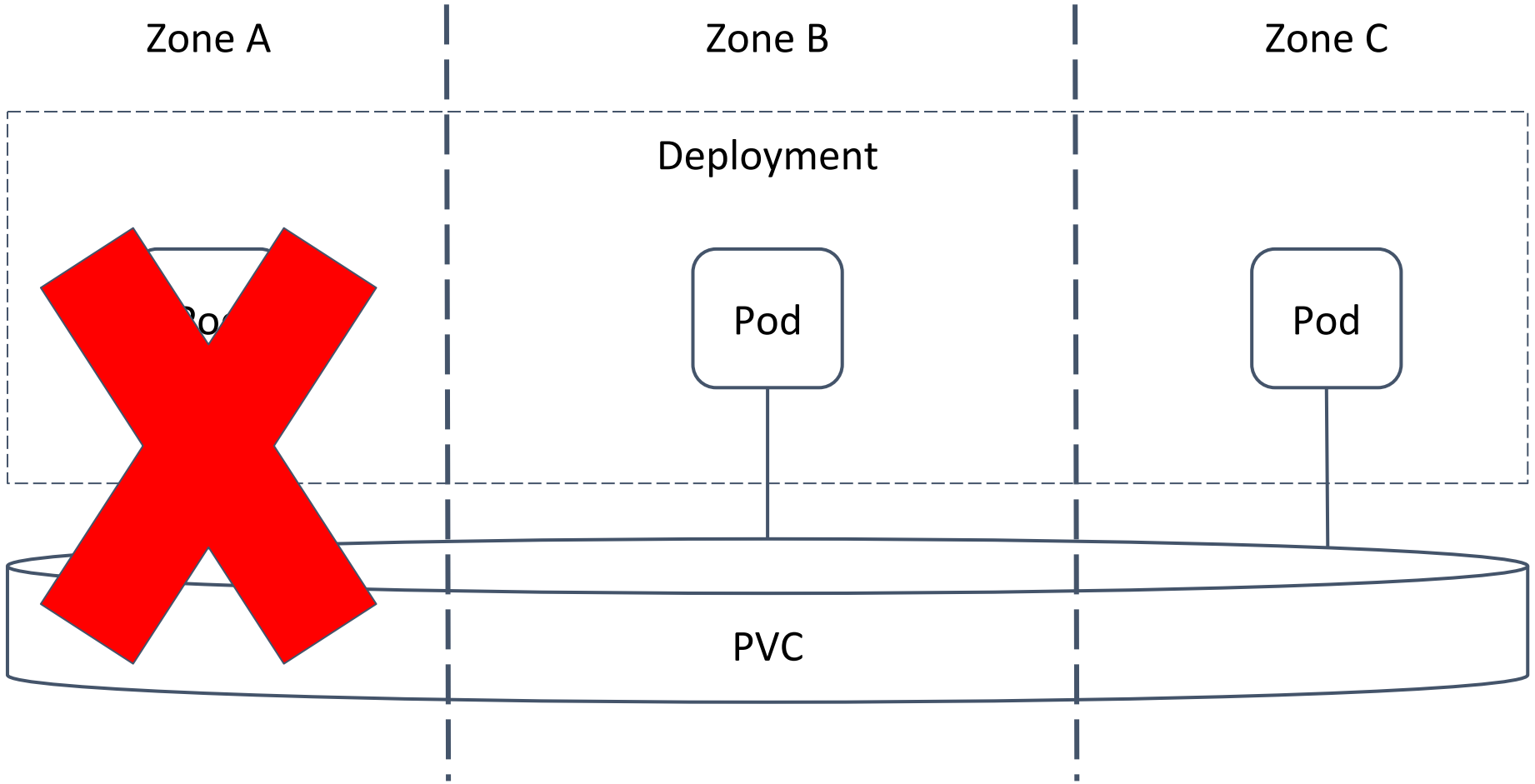
Need high availability at storage layer

- Multi-writer
- Globally accessible and available
- Example: Scaleout/HA filer

Deployment



Deployment



Distributed Model

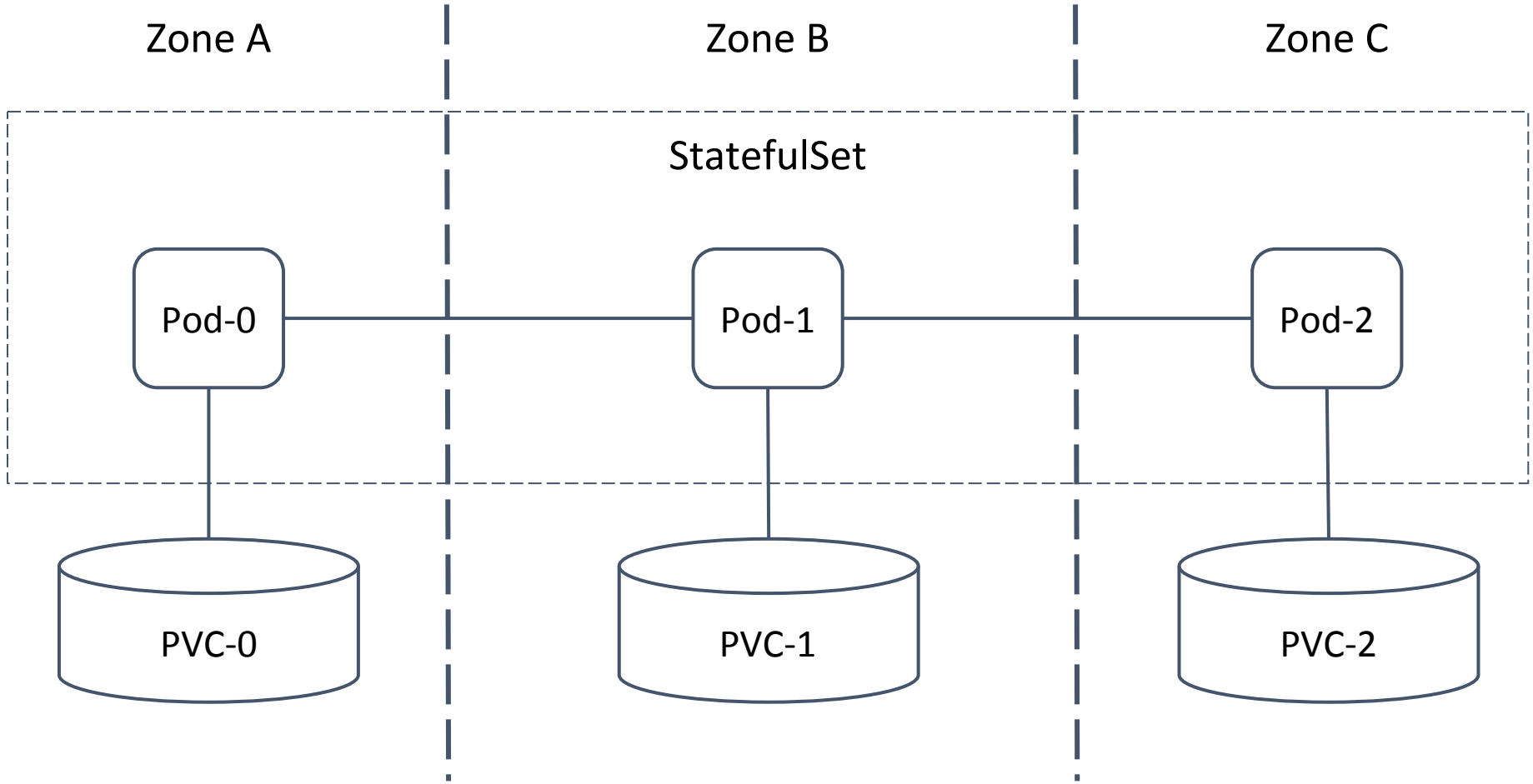
Shard and replicate data between pods

- Example: Cassandra, MongoDB

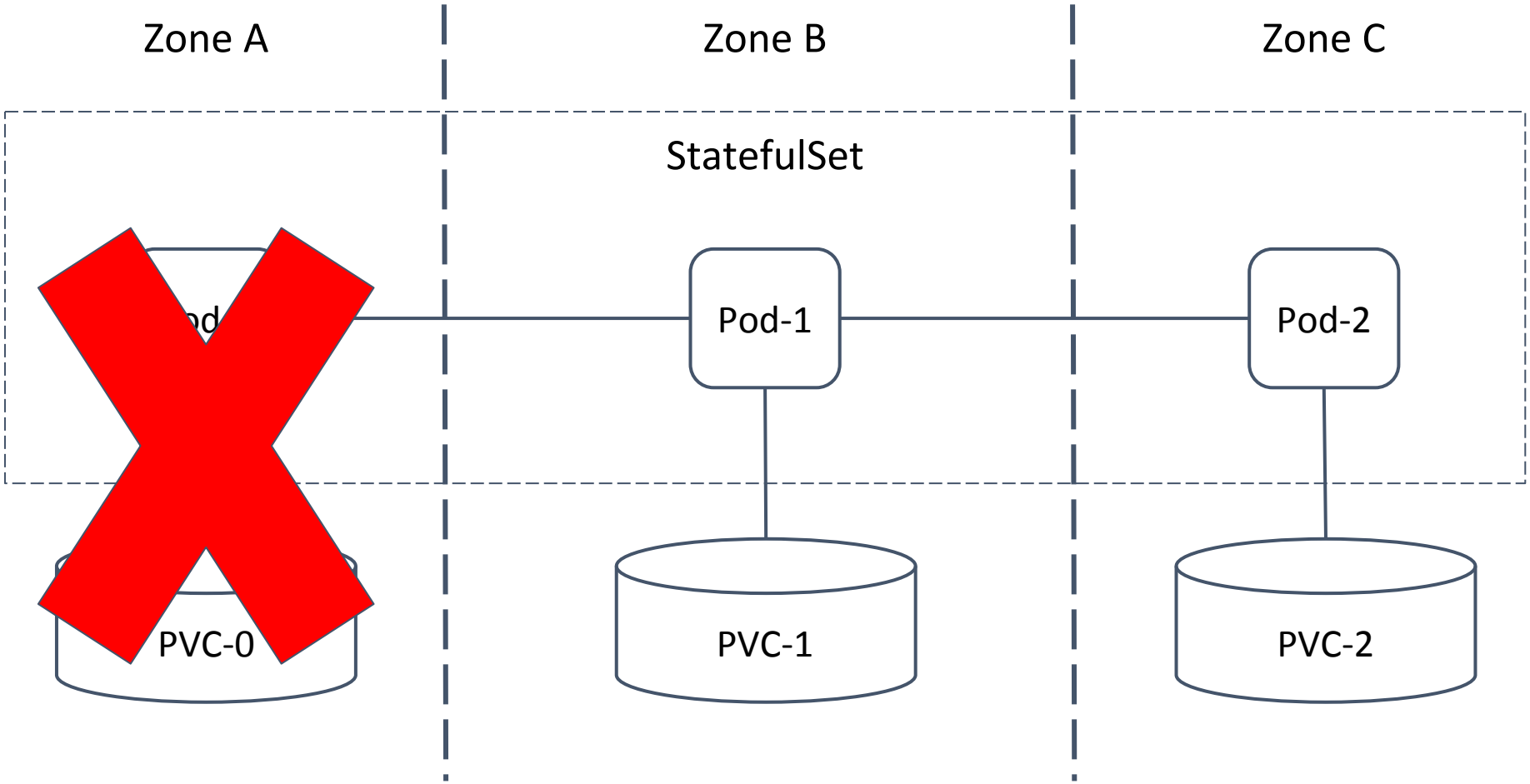
Do not need high-availability at storage layer

- Single writer
- Non-global accessibility and availability
- Example: Local disks, cloud disks

StatefulSet



StatefulSet



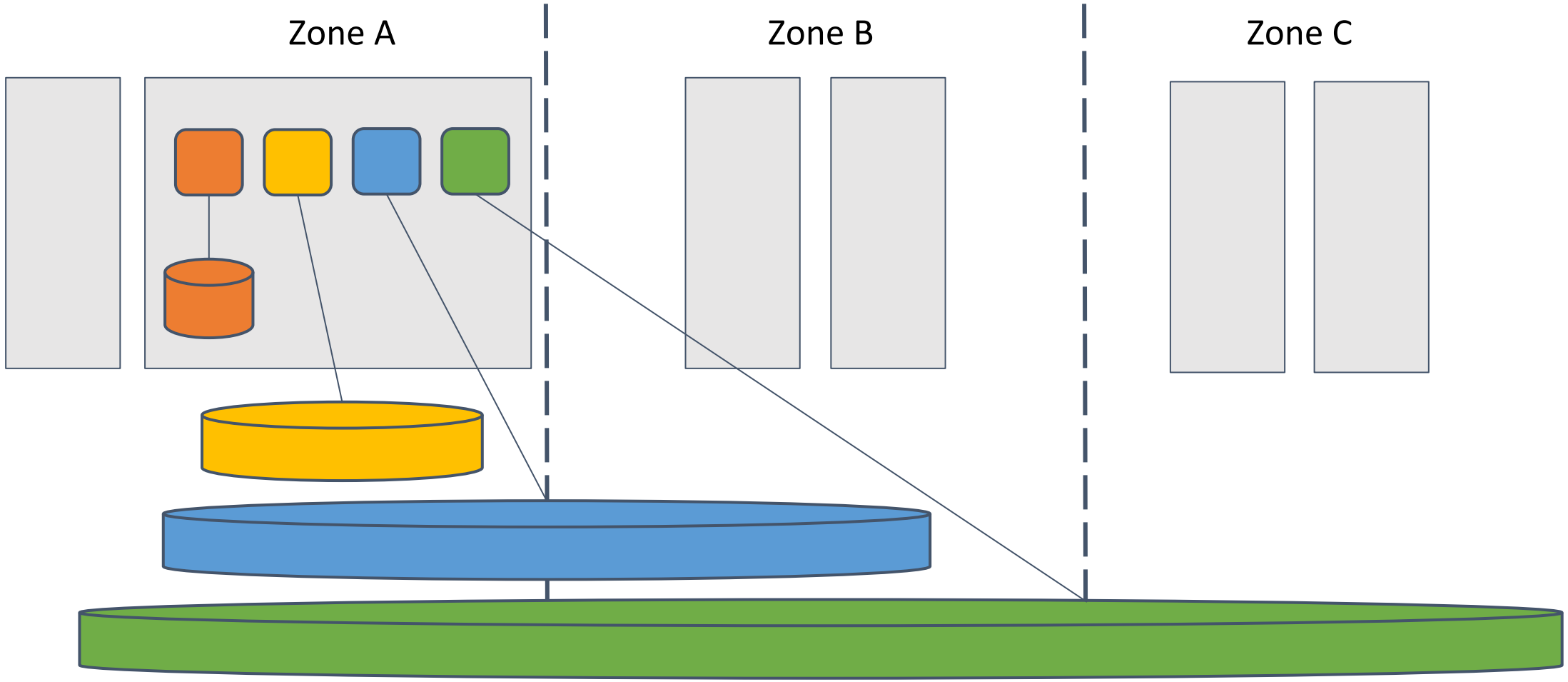
Volume Topology

Scheduler understands volume accessibility constraints

- No user configuration needed
- Storage driver provides topology

Auto-scale replicas and dynamically provision volumes across zones (except local)

Demo



Downtime

Time to detect failure

+

Time to replace pod

StatefulSet Caveat

Stateful applications may require exactly-once semantics

- Two containers cannot write to the same volume

During split brain, replacement Pod cannot be started

- Node fencing can help

StatefulSet pod recovery can be long

- Minutes: automated
- Hours: manual

Summary

Kubernetes features for high-availability

- Volume topology, pod anti-affinity, node taints

Stateful application models with pod anti-affinity

- Deployment vs Statefulset
- Storage redundancy vs application redundancy

Design for redundancy and account for downtime

Additional Resources

Deployments and StatefulSets

Pod anti-affinity

Even pod spreading design pod proposal

Volume topology blog post

Node taints and tolerations

Node fencing discussions

Get Involved

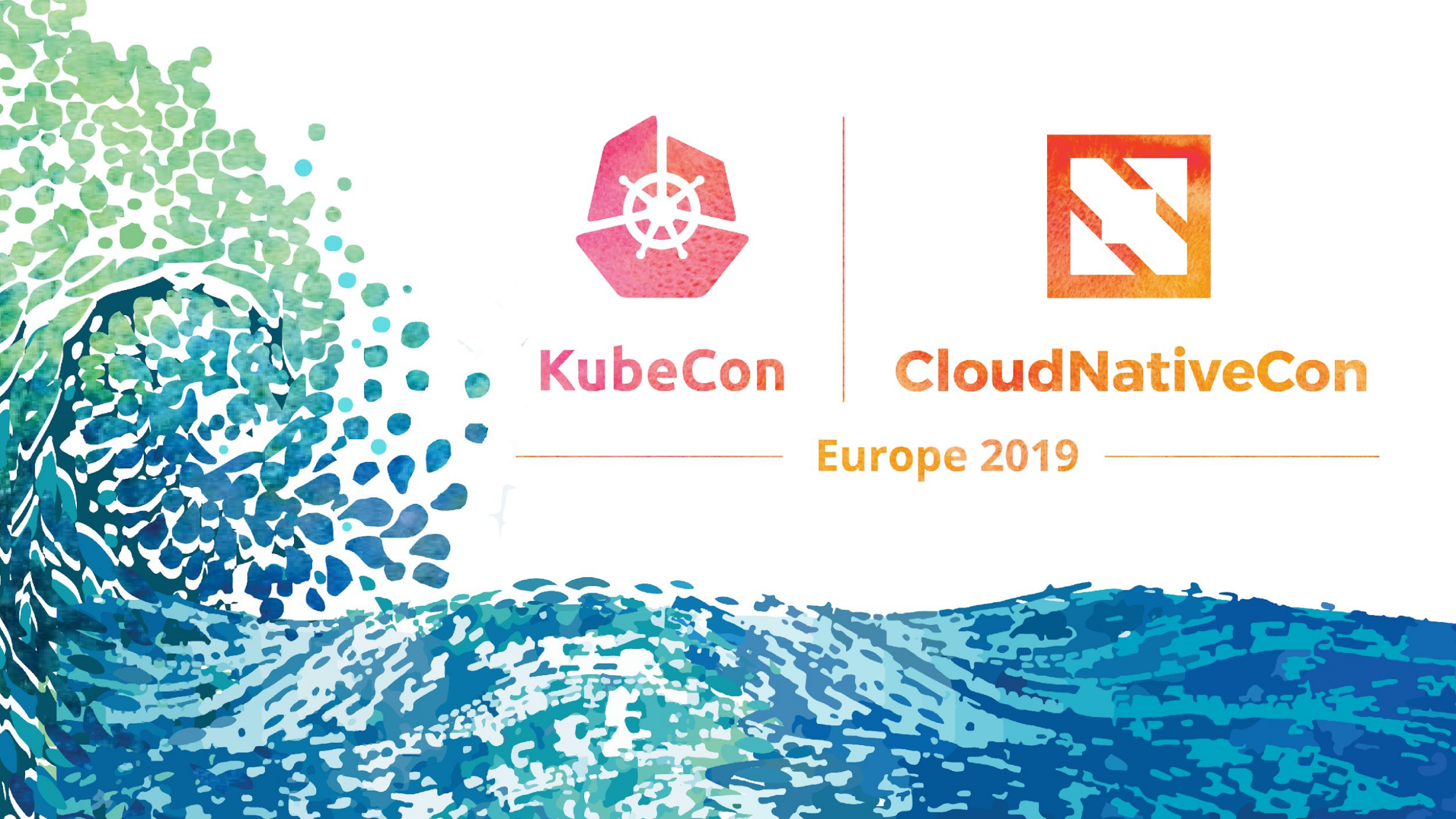
Kubernetes Special Interest Groups (SIGs)

- sig-storage, sig-apps, sig-node, sig-scheduling
- Community meetings, slack

Me

- Github/Slack: msau42
- Twitter: _msau42_

Questions?


KubeCon
CloudNativeCon
Europe 2019