STATEFUL KUBERNETES
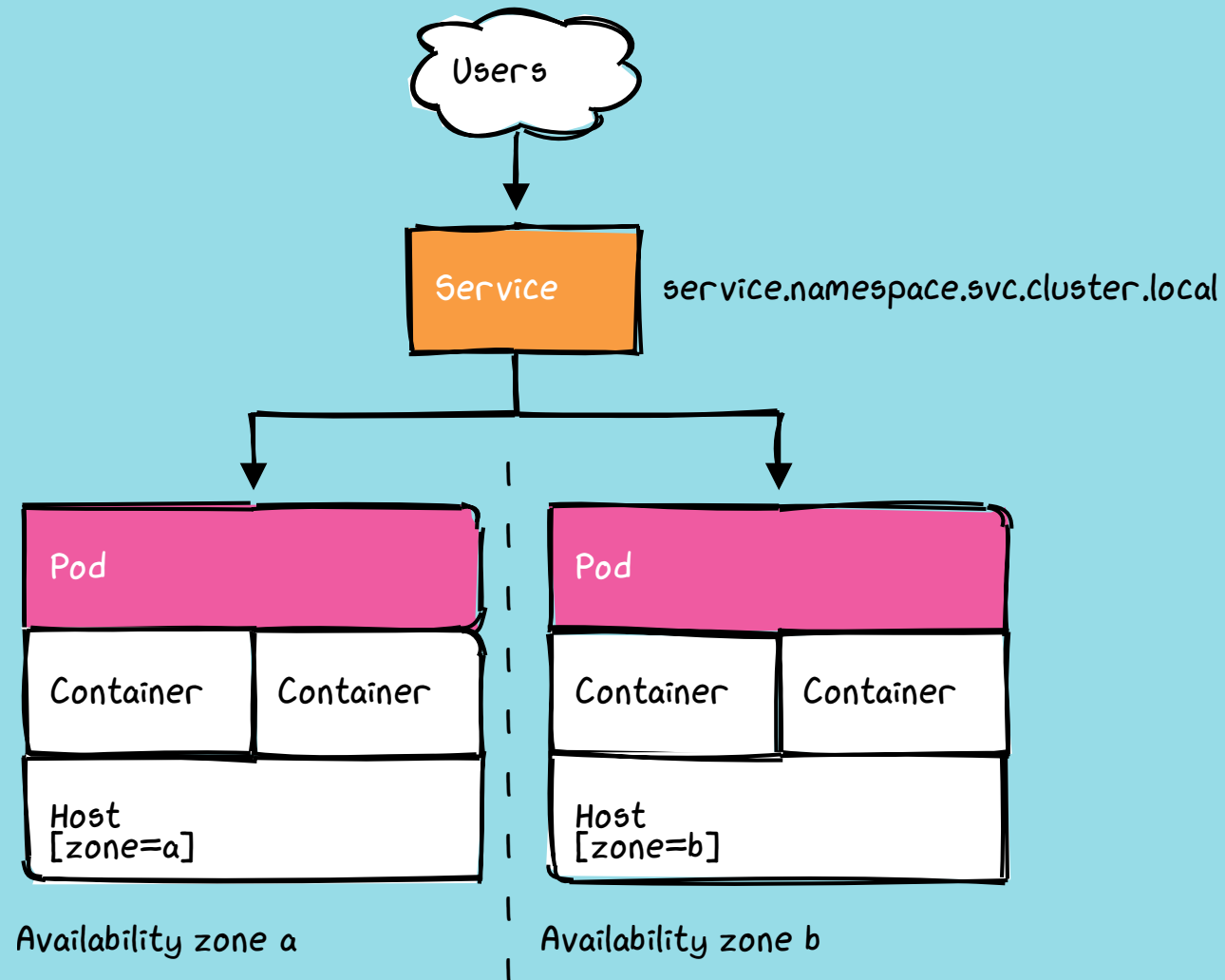
PAST, PRESENT, FUTURE
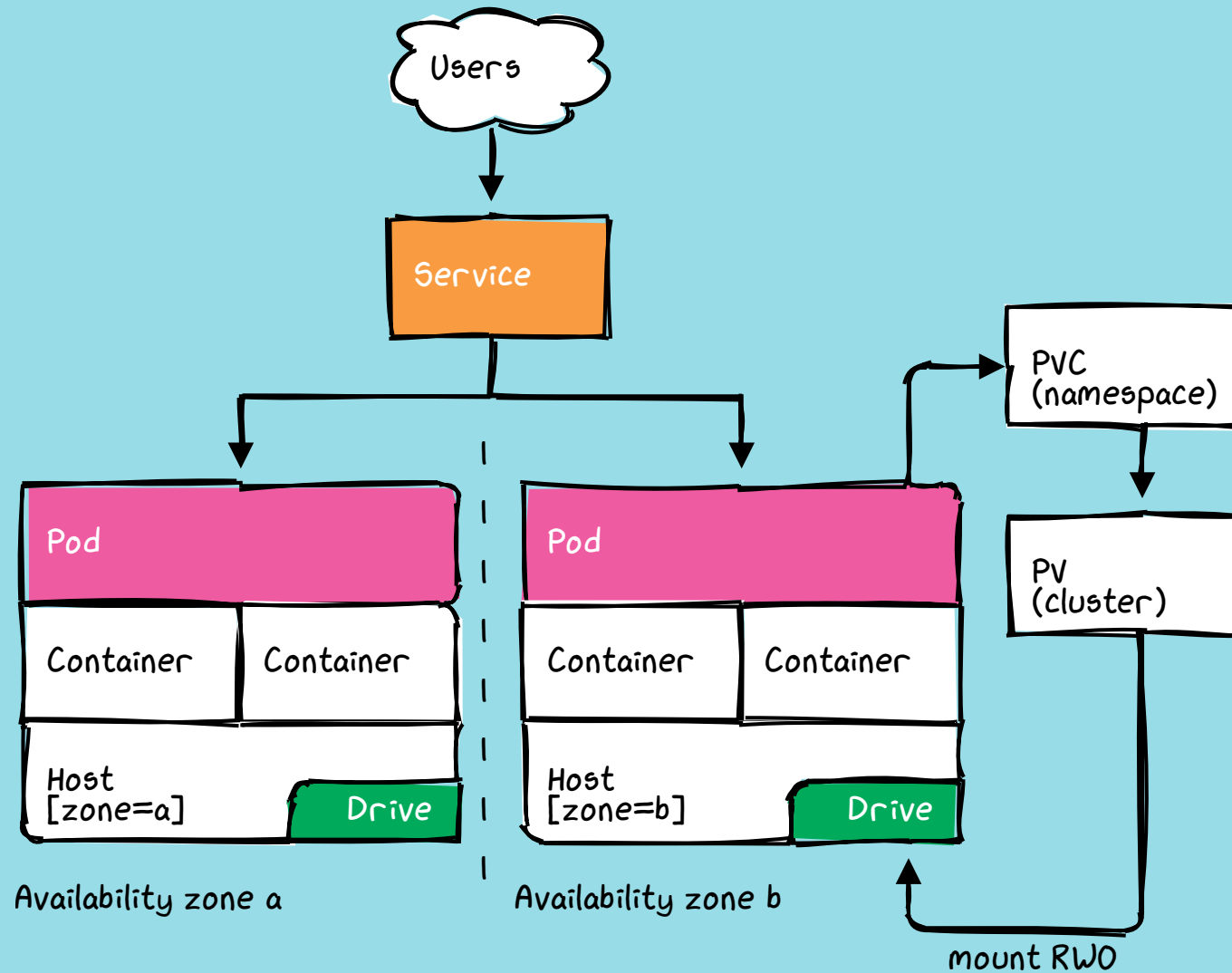
STATELESS KUBERNETES
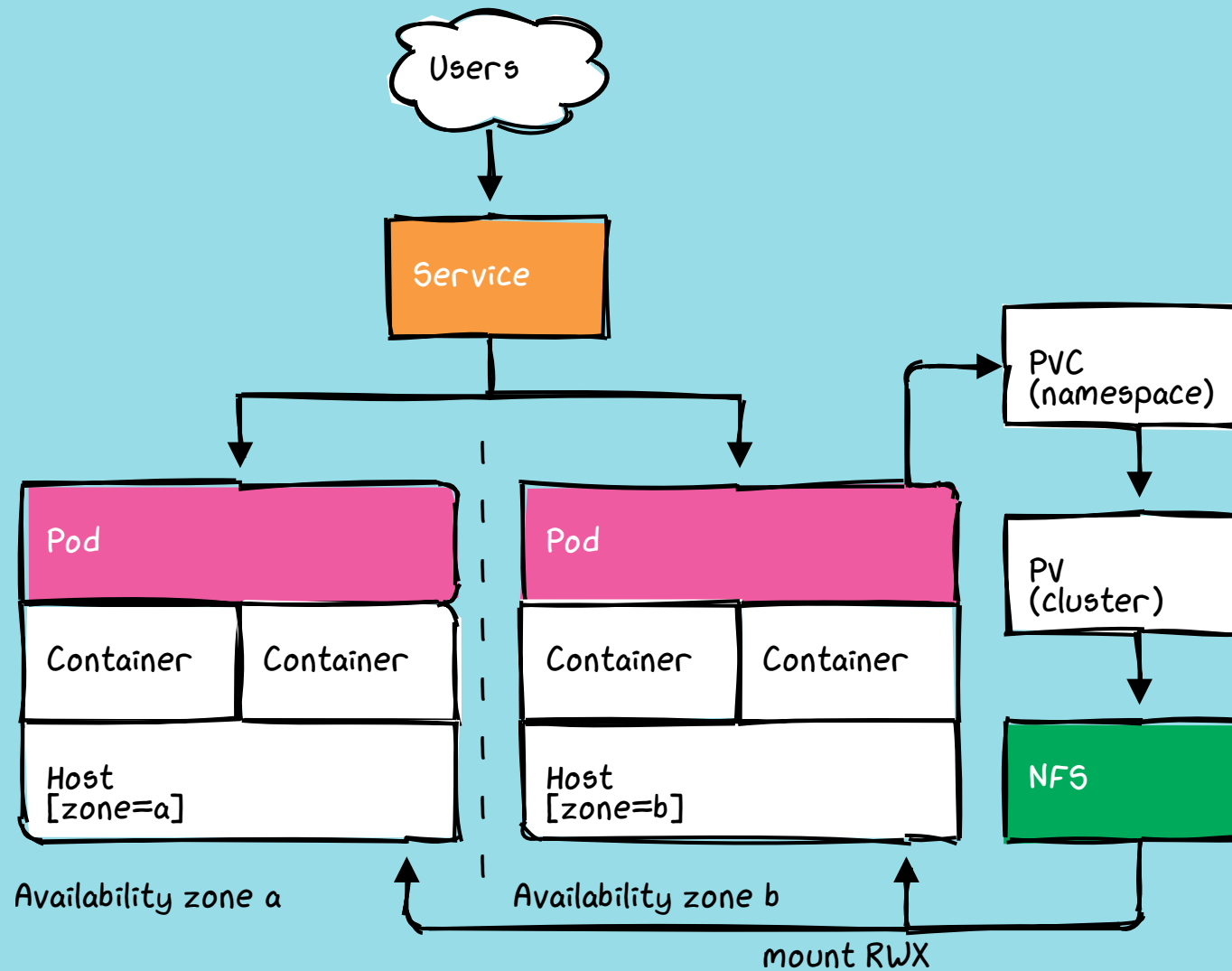

Users

Service   service.namespace.svc.cluster.local

Pod

Container	Container

Host
[zone=a]

Pod

Container	Container

Host
[zone=b]

Availability zone a   Availability zone b

CONTAINER STORAGE FABLE
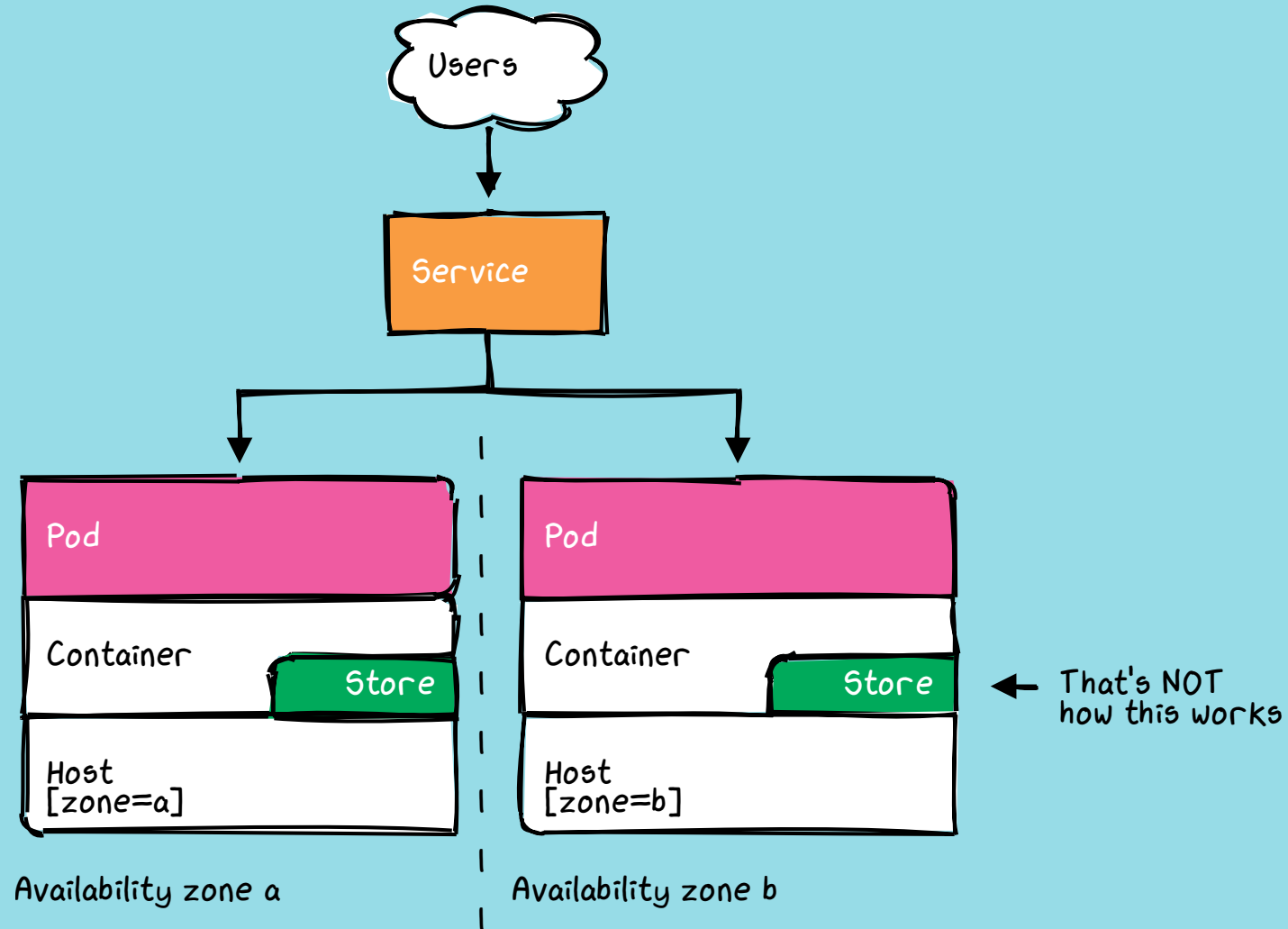
CONTAINER BLOCK STORAGE

CONTAINER NETWORK FILE STORAGE

Users

Service

PVC
(namespace)

Pod

Container	Container

Host
[zone=a]

Availability zone a

Pod

Container	Container

Host
[zone=b]

Availability zone b

PV
(cluster)

NFS

mount RWX

TIMELINES

IDENTITY, NETWORK, STORAGE AND STATE

IDENTITY

Kubernetes   OpenShift

attribute-based access control (ABAC) →

← roles and cluster-roles

role-based access control (RBAC) →

Kubernetes   OpenShift

NETWORK

Kubernetes   OpenShift

flat network →

← multitenant plugin

← prototype network policies

network policies →

Kubernetes   OpenShift

STORAGE AND STATE

Kubernetes   Red Hat OpenShift   CoreOS Tectonic

beware stateful applications →

← application templates (2015)

stateful applications still hard →

← etcd operator with third-party resources (2016)

Custom Resource Definitions →

← Operator Framework and SDK (2018)

merges with (2019) ←

Kubernetes   Red Hat OpenShift   CoreOS Tectonic

REBUILDING THE DATACENTRE

Kubernetes

Role-based access control

| Container runtime interface | Container network interface | Container storage interface |
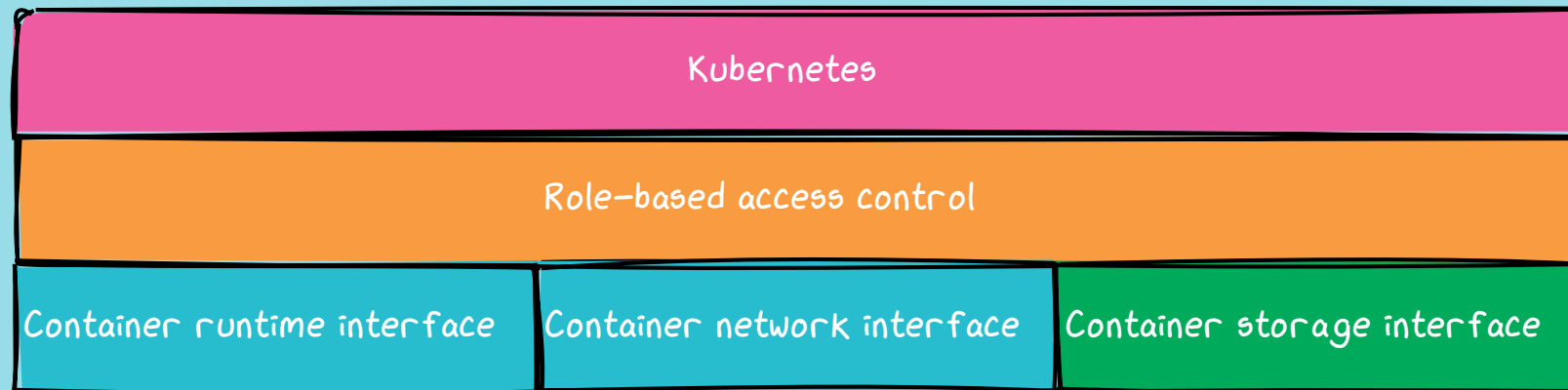
ECOSYSTEM RESILIENCE

SHARED OWNERSHIP

Microsoft

IBM
Red Hat
CoreOS

Google

VMware

Alibaba

Cloud Native Computing Foundation (CNCF)
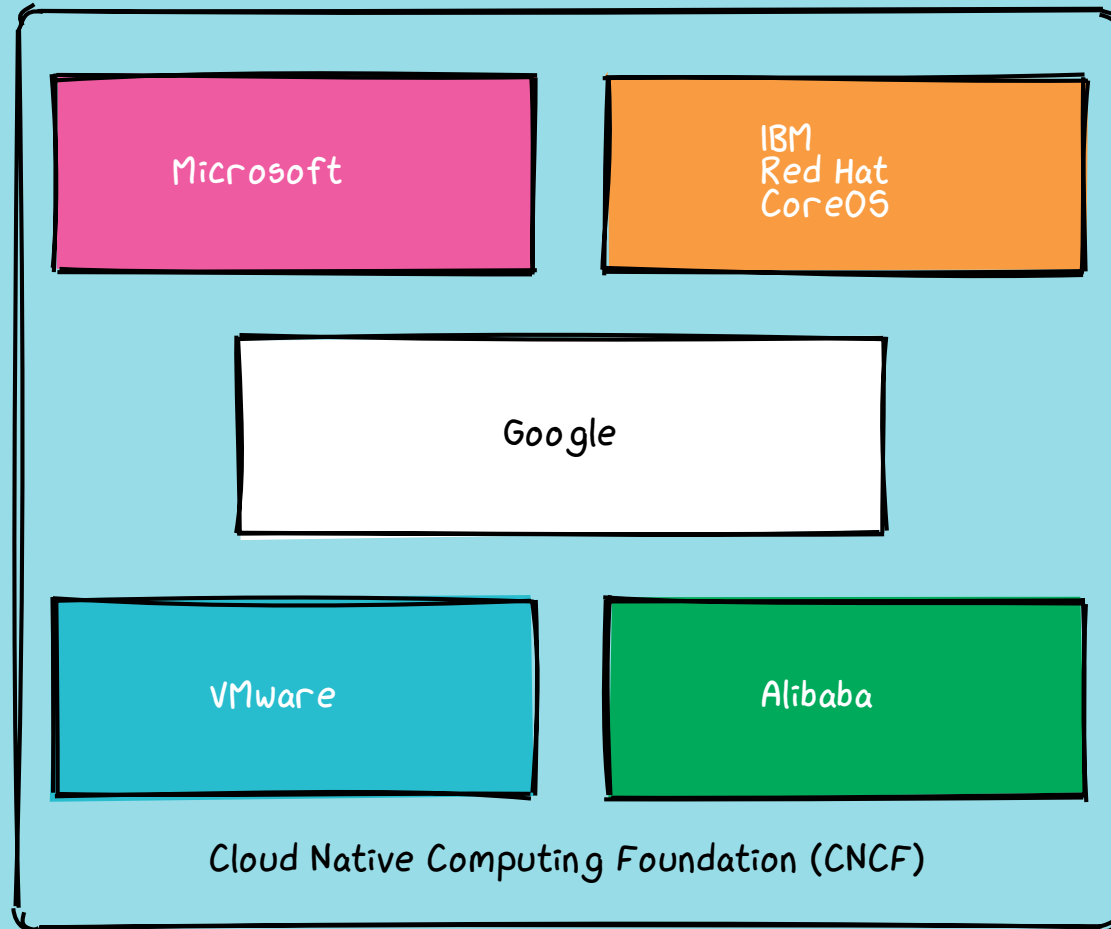
PRECONDITIONS

open source
neutral IP ownership
extensibility

PRECONDITIONS

open source — Apache

neutral IP ownership

extensibility

A suitably permissive license is a necessary but not sufficient precondition.

PRECONDITIONS

open source – Apache

neutral IP ownership – CNCF

extensibility

PRECONDITIONS

K8s would *not* be what it is today without neutral IP ownership... K8s has spawned an entire ecosystem *because* it can be used by consuming projects/products without fear.

— Matt Klein

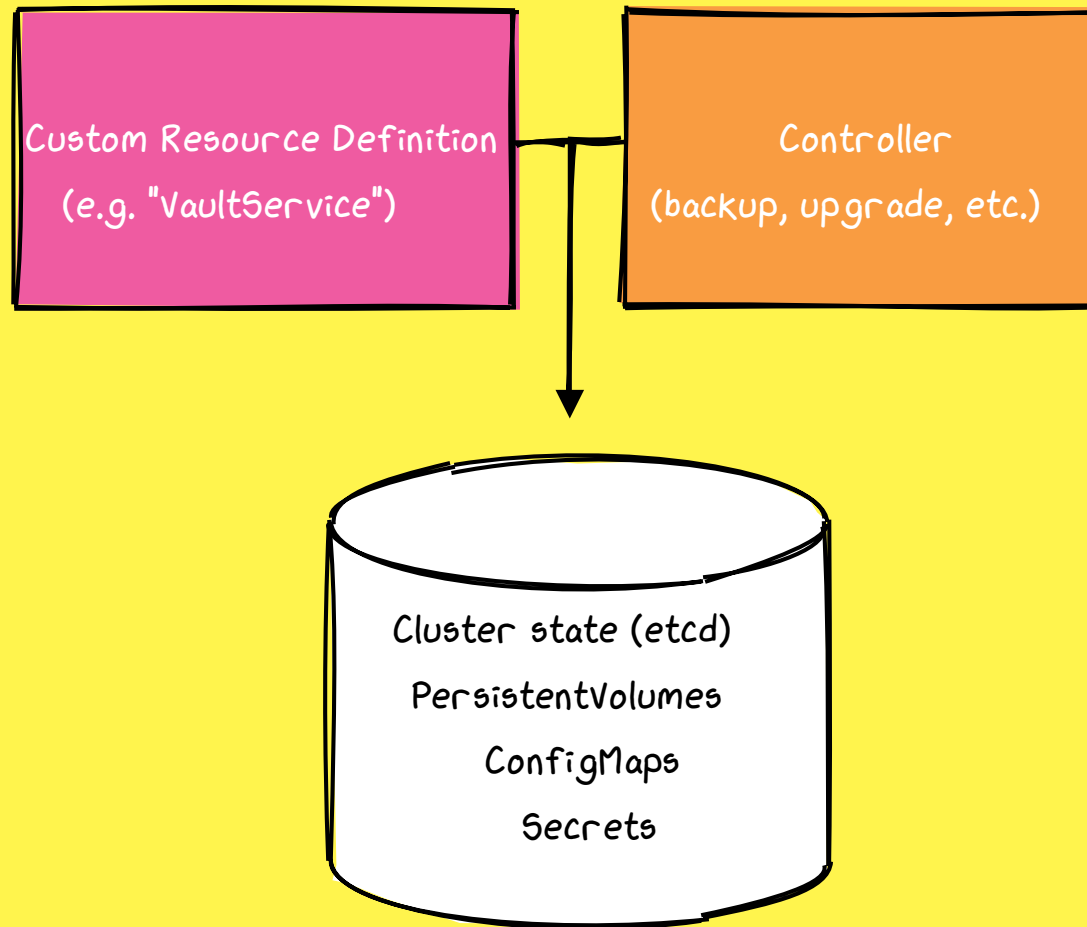Source: @mattklein123 on Twitter, 17 February 2020.

PRECONDITIONS

open source — Apache

neutral IP ownership — CNCF

extensibility — controllers and operators
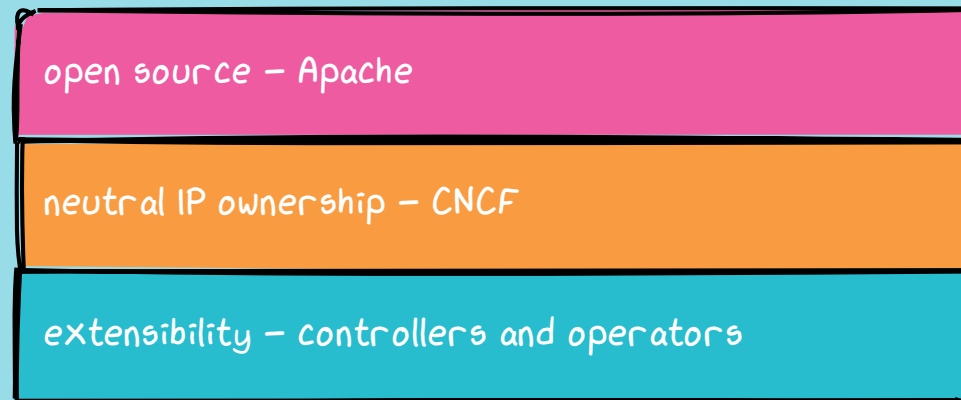
Controllers add custom processing to the core reconciliation loop.

When paired with custom resource definitions, they are known as **operators**.

THE OPERATOR LIFE

OPERATORS

Custom Resource Definition
(e.g. "VaultService")

Controller
(backup, upgrade, etc.)

Cluster state (etcd)
PersistentVolumes
ConfigMaps
Secrets

CORPORATE SPONSORS

Vault operator

Core OS

MySQL operator

ORACLE®

PostgreSQL operator
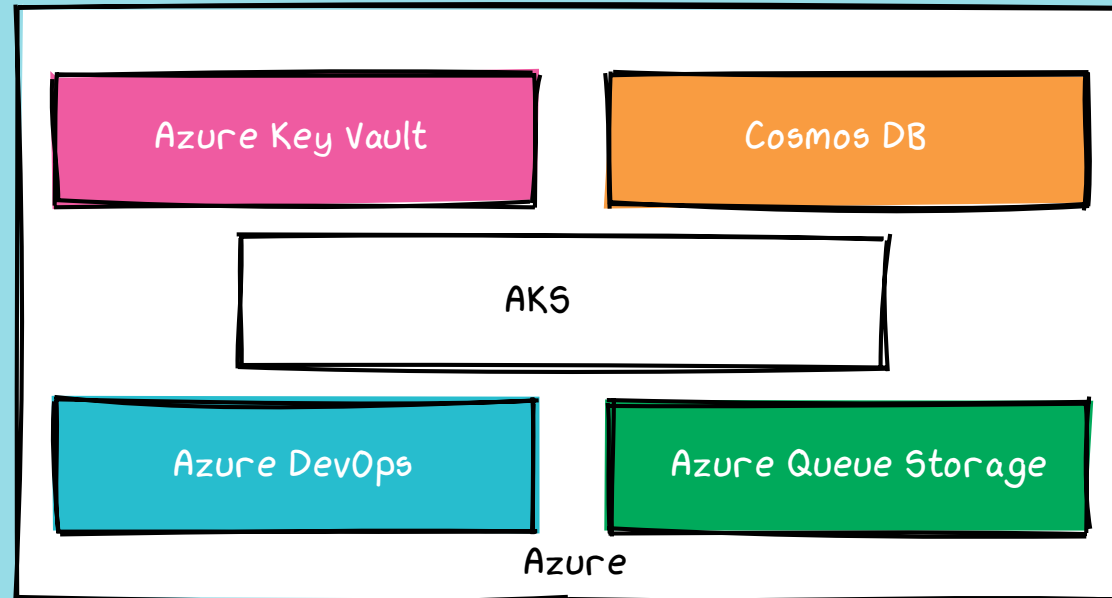
zalando

STATEFUL WORKLOADS

Stateless is Easy, Stateful is Hard.

— Brandon Philips (2016)

Source: coreos.com/blog/introducing-operators.html

# SOUL-SEARCHING AT THE BAZAAR

FALLING OUT OVER STATE MANAGEMENT

Cloud vendors perfected the use of proprietary services based on open source products, putting companies creating those products on notice.

In 2016 MongoDB Inc. responded with the Server Side Public License.

A PORTABLE STACK

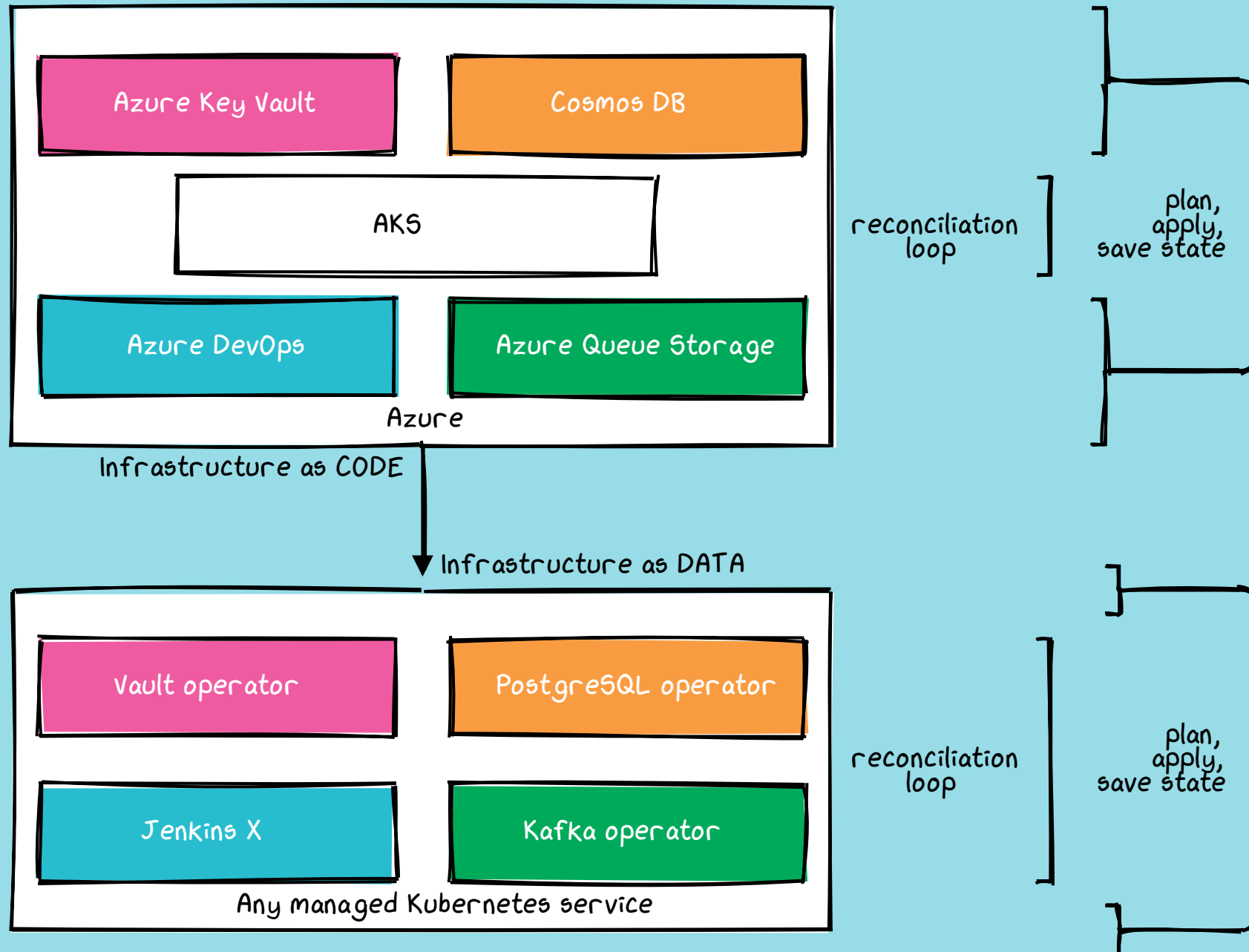| Vault operator | PostgreSQL operator |
| Jenkins X | Kafka operator |

Any managed Kubernetes service
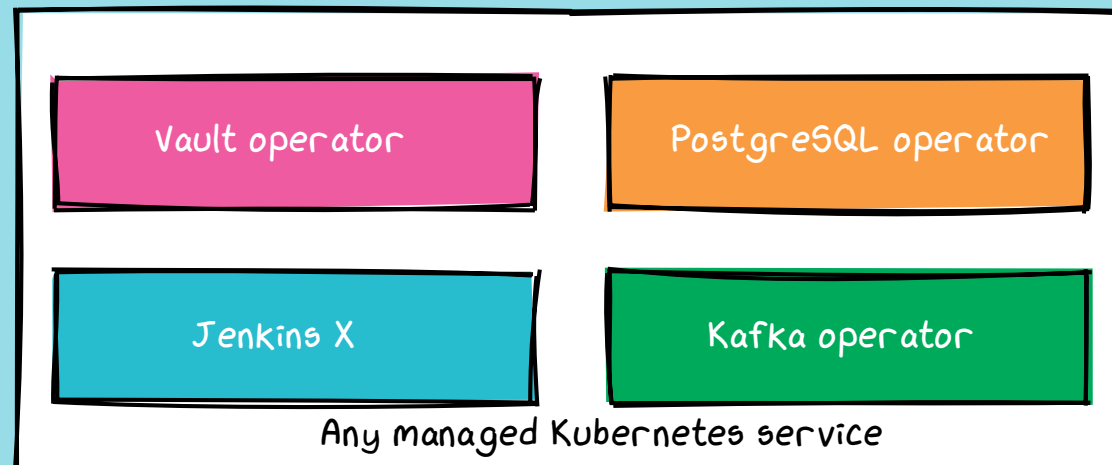
INFRASTRUCTURE AS ~~CODE~~ DATA

> Declarative configuration is about treating infrastructure as data, which is more portable than code, and enables workflows that manipulate desired state based on policy.
>
> — Kelsey Hightower (2019)

Source: @kelseyhightower on Twitter, 21 August 2019.

DESIRED STATE

Azure (Infrastructure as CODE)

- Azure Key Vault
- Cosmos DB
- AKS
- Azure DevOps
- Azure Queue Storage

Azure

Infrastructure as CODE

reconciliation loop → plan, apply, save state

Any managed Kubernetes service (Infrastructure as DATA)

Infrastructure as DATA

- Vault operator
- PostgreSQL operator
- Jenkins X
- Kafka operator

Any managed Kubernetes service

reconciliation loop → plan, apply, save state

TRADE-OFFS

Azure Key Vault	**Cosmos DB**

AKS

Azure DevOps	**Azure Queue Storage**

Azure

Infrastructure as CODE

open source
less sticky
less mature

proprietary
sticky
mature

Infrastructure as DATA

Vault operator	**PostgreSQL operator**
Jenkins X	**Kafka operator**

Any managed Kubernetes service
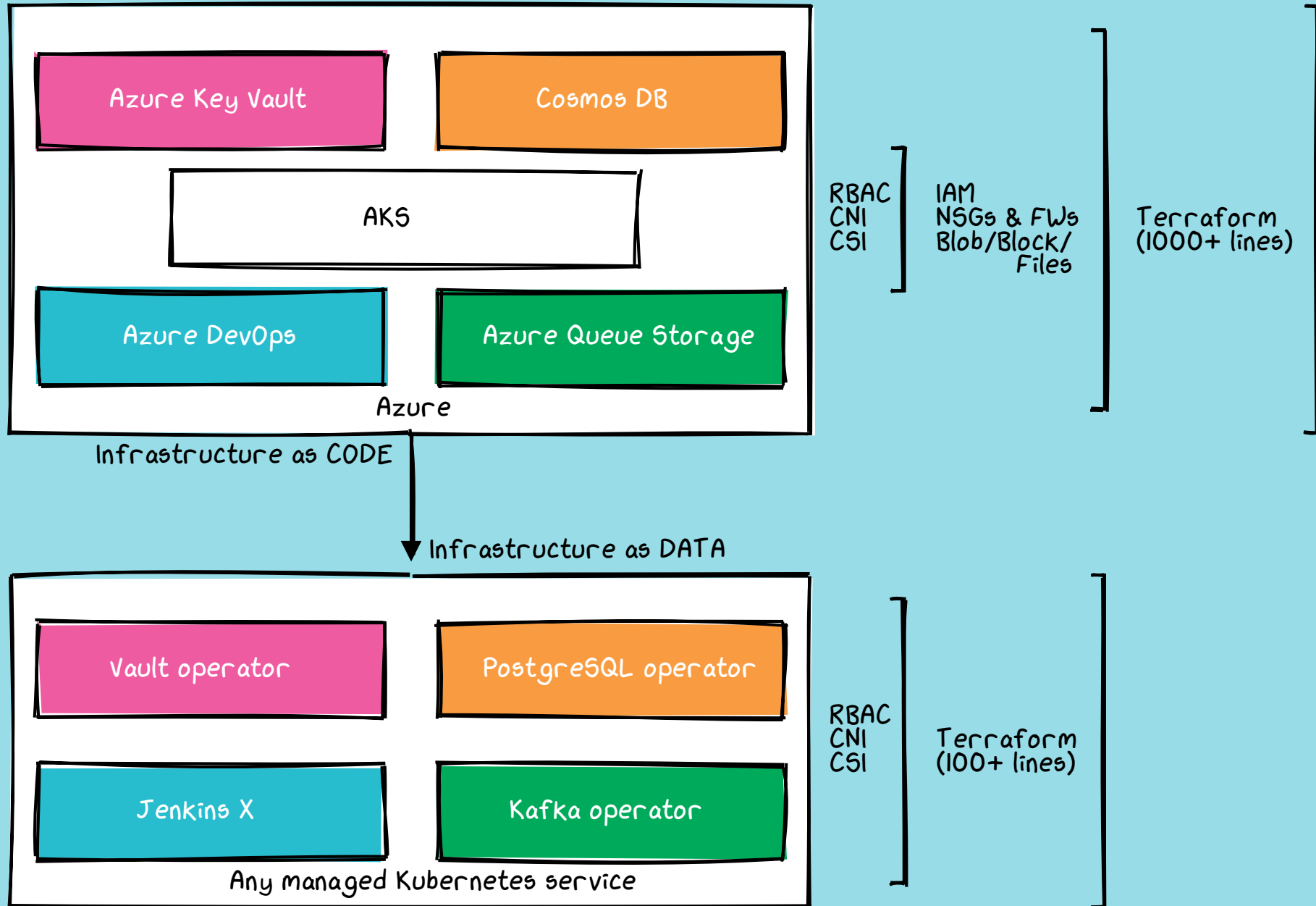
open source
less sticky
less mature

proprietary
sticky
mature

CONFIGURATION

NATIVE RESOURCES

Can I trust custom resources to create and manage the lifecycle of objects native to the platform?

Yes, this is something resources like PersistentVolumeClaim and Service have done for a long time, dynamically creating storage volumes and load balancers respectively.

SERVICE LEVEL

Can the PostgreSQL operator match the availability and durability guarantees of managed SQL Server DBs?

Not today, no. It is worth considering, though, that:

✴ RDS and the operator use the same Azure storage primitives

✴ Until this improves, there's Microsoft's Azure Service Operator for Kubernetes (EventHub, Azure SQL, CosmosDB, Storage Accounts)

Source: cloudblogs.microsoft.com/opensource/2020/06/25/announcing-azure-service-operator-kubernetes/

A PLAUSIBLE FUTURE, BUT NOT A DONE DEAL

The CoreOS Vault operator depends on their etcd operator

Two choices: Crunchy and Zalando's Spilo
MySQL has a first-party operator by Oracle
There is also CNCF graduate project Vitess

Vault operator	PostgreSQL operator
Jenkins X	Kafka operator

Any managed Kubernetes service
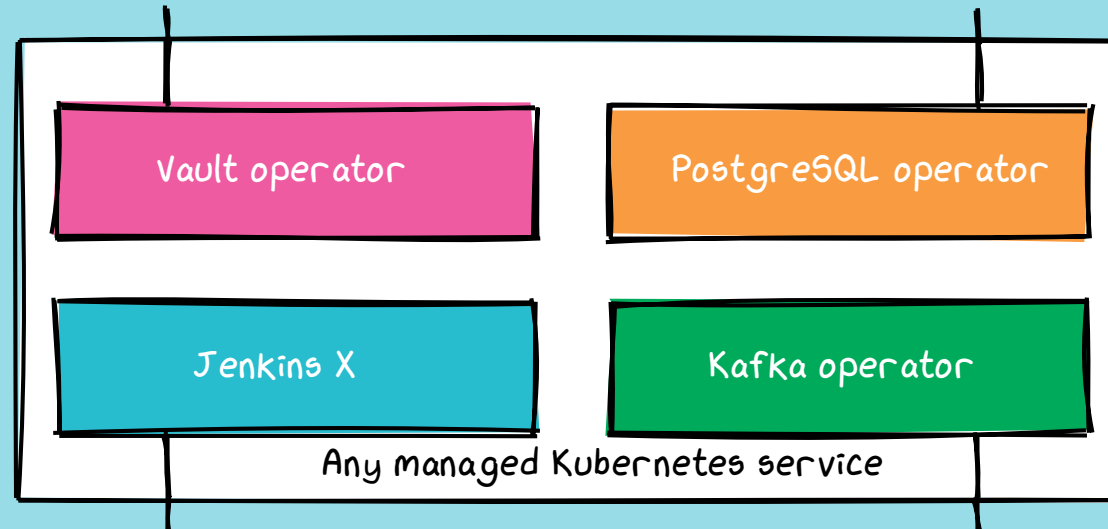
Note that this is not "Hudson"
UI requires VMware Octant with octant-jx plugin

First-party operator by Confluent
One promising alternative is the RabbitMQ Cluster Operator

WHAT'S IN IT FOR US?

As with multicloud, the value proposition for the consumer can seem lacklustre, but there are tangible benefits

Create one path to production for all workloads (not three separate ones for containers, functions and VMs)

Enabling zero trust and mutual TLS is much easier, and again you only do it once

Continuous reconciliation based on policies you define beats setting desired state at the start and hoping for the best

SUMMARY

Persistent cloud storage is just as reliable when claimed by a pod

The core Kuberenetes interfaces will continue to mature because everyone in the industry has a stake in it

The operator pattern is set to thrive because software companies and second-tier cloud vendors depend on it

Of all the large, distributed systems found in modern application architectures, only Kubernetes promises a significant reduction in complexity in return

THANK YOU

Slides built with Markdeck
GitHub gerald1248/stateful-kubernetes-slides
Twitter @03spirit

ThoughtWorks®