Introduction to Kubernetes

Markus Lackner

Product Architect / Dynatrace

Engineering Kiosk 2024-09-26

Agenda

- Create a Cluster
- A Brief History of Orchestration
- Kubernetes Introduction
- Example 1: Deploy an Application
- Example 2: Leader Election
- When (and when not) to use Kubernetes?
- Questions?

Create a Cluster

A Brief History of Orchestration

What is orchestration?

- Different meanings
- Coordination and management of multiple computer systems / applications
- Managing software/applications at larger scale

History Timeline

Bare Metal

VM Ware
2003

Project Borg

2005

puppet labs

KVM
2007

cgroups
2008

2009

CHEF

Vagrant
2010

Apache MESOS

2011

CLOUDFOUNDRY

docker
2013

2012

ANSIBLE

2015

kubernetes

2021

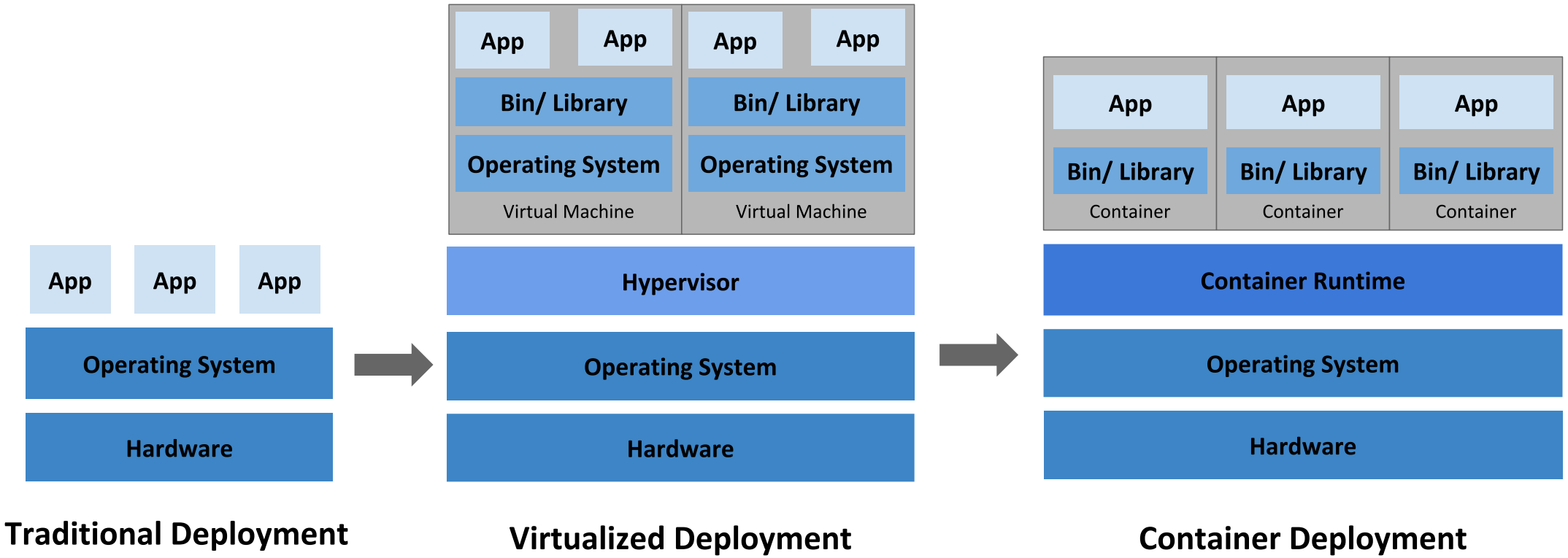
Market Survey

96% using / evaluating

69% in production

CLOUD NATIVE COMPUTING FOUNDATION

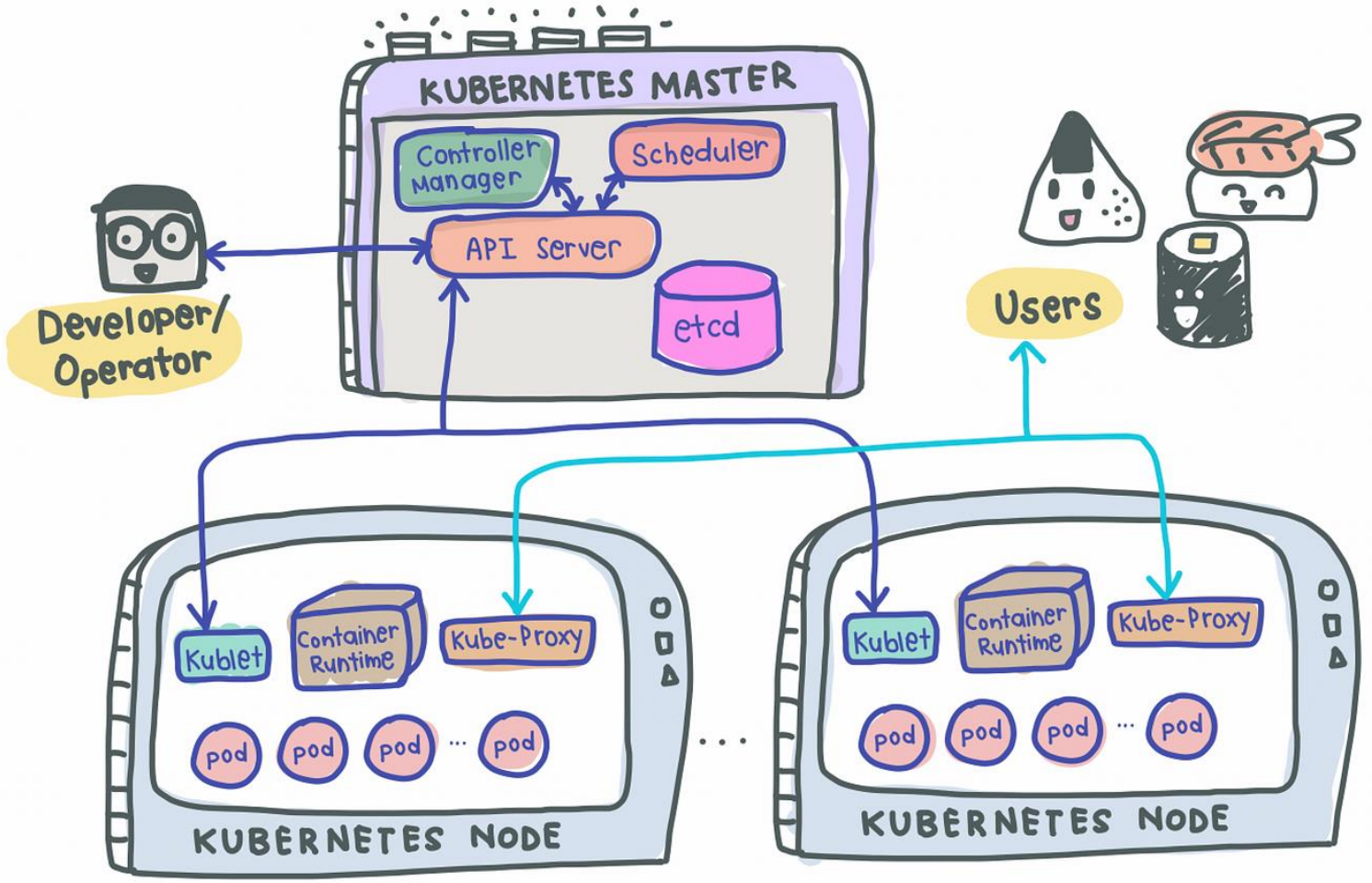
Deployment



https://kubernetes.io/docs/concepts/overview/#going-back-in-time

Kubernetes Introduction

Kubernetes - Introduction



https://towardsdatascience.com/a-beginner-friendly-introduction-to-kubernetes-540b5d63b3d7

Kubernetes - Everything is a yaml

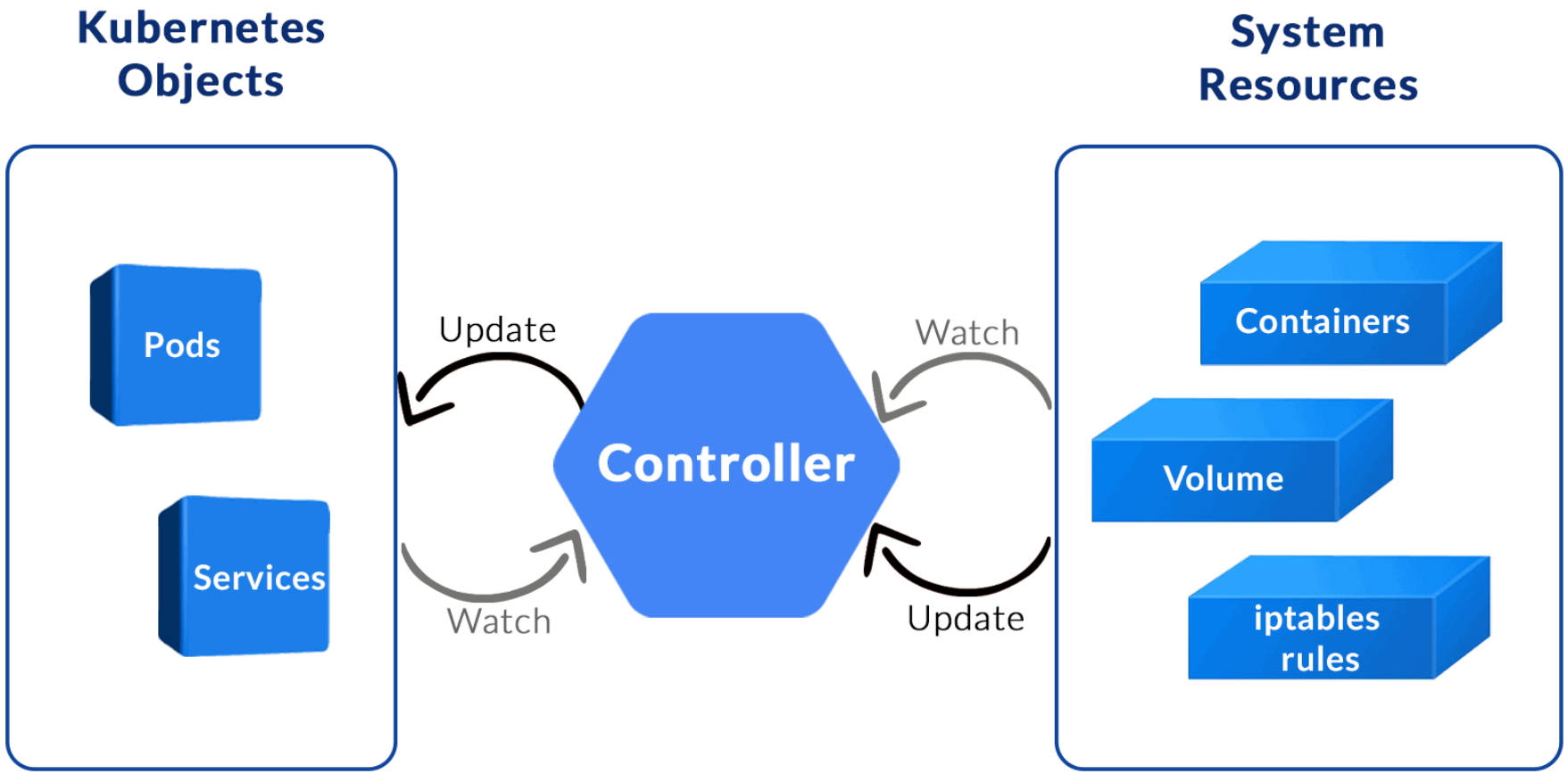

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: foo-pvc
  namespace: foo
spec:
  storageClassName: "" # Empty string
  volumeName: foo-pv
  ...
```

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx-example
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

Kubernetes - Reconciliation Loop



https://k21academy.com/docker-kubernetes/kubernetes-operator/

Example 1: Deploy an Application

Example 2: Leader Election

When (and when not) to use Kubernetes?

When (and when not) to use Kubernetes?

Where Kubernetes shines:

- Multi Cloud
- High Availability
- Service Mesh
- Autoscaling
- 12 Factor Apps

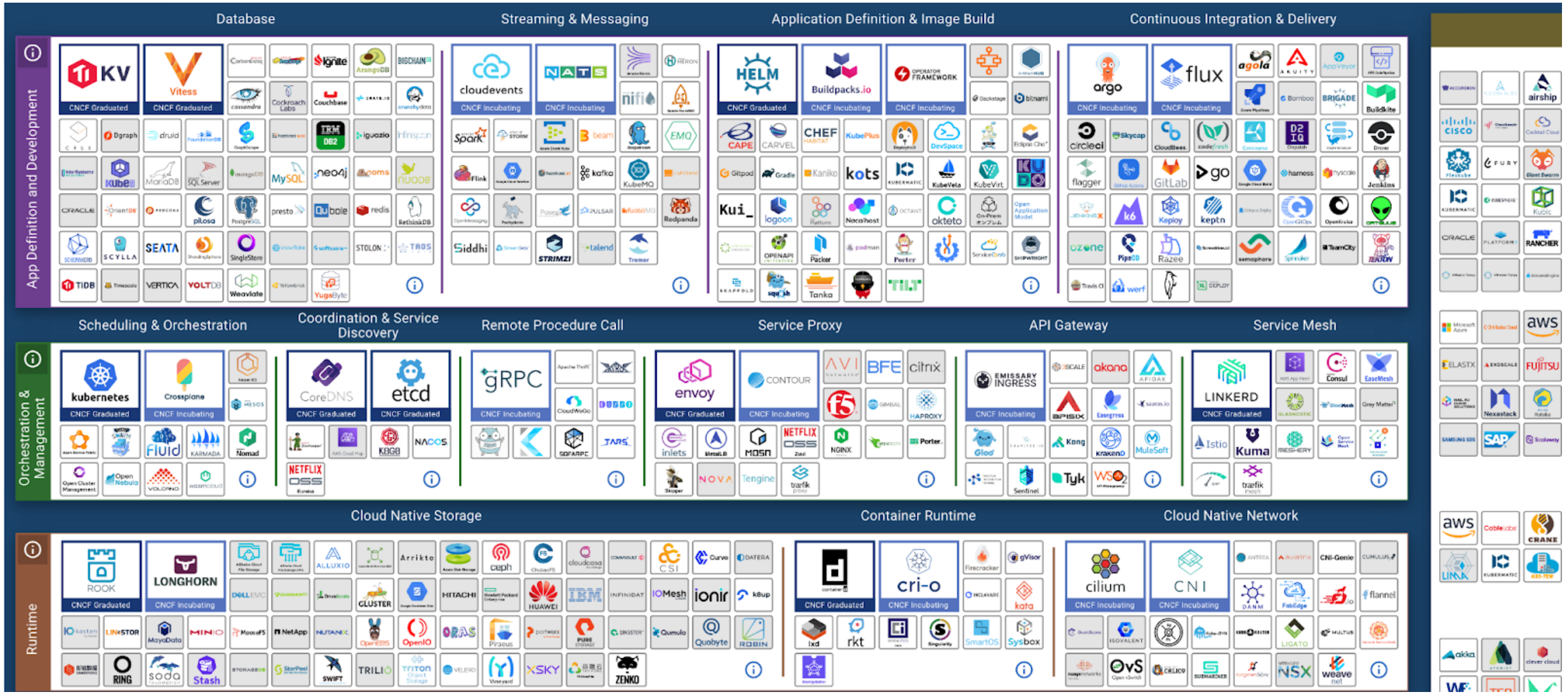
What to consider:

- Total costs
- Complexity / Learning curve
- Simplest solution?

What to avoid:

- Using k8s without reasons
- "Monoliths"
- Legacy code/apps (e.g. java <= 8u121)

CNCF (Cloud Native Computing Foundation) Landscape



Thank You!