DataStax

How to put a Database in K8s

Jeff Carpenter - Software Engineer @ DataStax

October 14, 2021

How to put a Database in K8s

The idea of running a stateful workload in Kubernetes can be intimidating, especially if you haven't done it before. How do you deploy a database? Where is the actual storage? How is the storage mapped to the database or application that uses it? In this talk, we'll demystify the deployment of databases and stateful workloads in K8s by showing that databases are just applications composed of compute, network and storage. We can deploy them like any other Kubernetes application and take advantage of resources that K8s provides including Storage Classes, Persistent Volumes, Persistent Volume Claims, and Stateful Sets. We will demonstrate how to make it all work by deploying a relational database (MySQL) and a NoSQL database (Apache Cassandra).

Who am I?

- Developer
- Architect
- Author
- Advocate

- Defense
- Hospitality
- R&D

- Distributed Systems
- Large Scale
- Cassandra

Wait, you can run databases on K8s?

Kubernetes is everywhere. Transactional apps, video streaming services and machine learning workloads are finding a home on this ever-growing platform. But what about databases? If you had asked me this question five years ago, the answer would have been a resounding "**No!**" — based on my experience in development and operations. In the following years, as more resources emerged for stateful applications, my answer would have changed to "*Maybe,*" but always with a qualifier: "It's fine for development or test environments..." or "If the rest of your tooling is Kubernetes-based, and you have extensive experience..."

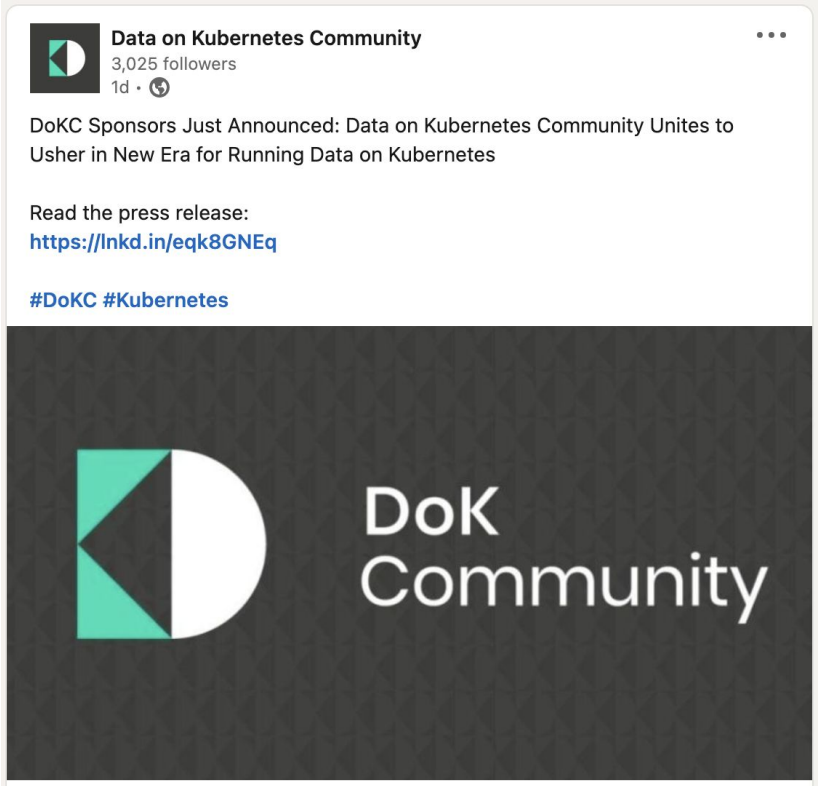
But how about today? Should you run a database on Kubernetes? With complex operations and the requirements of persistent, consistent data, let's retrace the stages in the journey to my current answer: "In a cloud native environment? **Yes!**"

Christopher Bradford

Christopher has a passion for enabling efficiency through automation. From promoting effortless scaling via Cassandra to DevOps pipelines with infrastructure automation and containers, he is here to get work done and enable operators to rest easy.

https://thenewstack.io/a-case-for-databases-on-kubernetes-from-a-former-skeptic/

Data on Kubernetes Community – https://dok.community



“90% of respondents believe K8s is ready for stateful workloads, and a large majority (70%) are running them in production with databases topping the list. ”

Data on Kubernetes 2021 Research Report

https://dok.community/wp-content/uploads/2021/10/DoK_Report_2021.pdf

How to put a database on Kubernetes

(in 4 easy steps?)

01

Learn the Kubernetes primitives

02

Pick a storage provider

03

Pick a database

04

Pick an operator

How to put a database on Kubernetes

01

Learn the Kubernetes primitives

02

Pick a storage provider

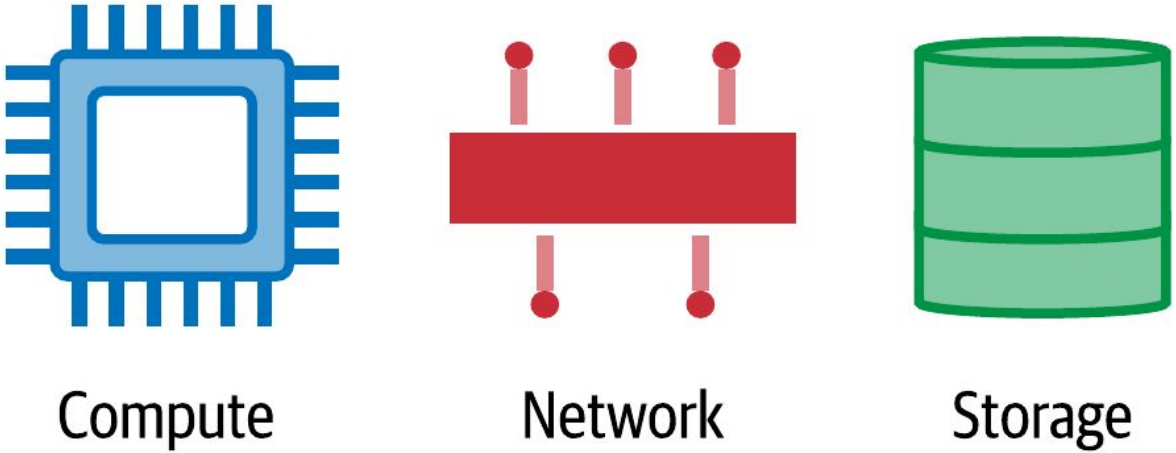
03

Pick a database

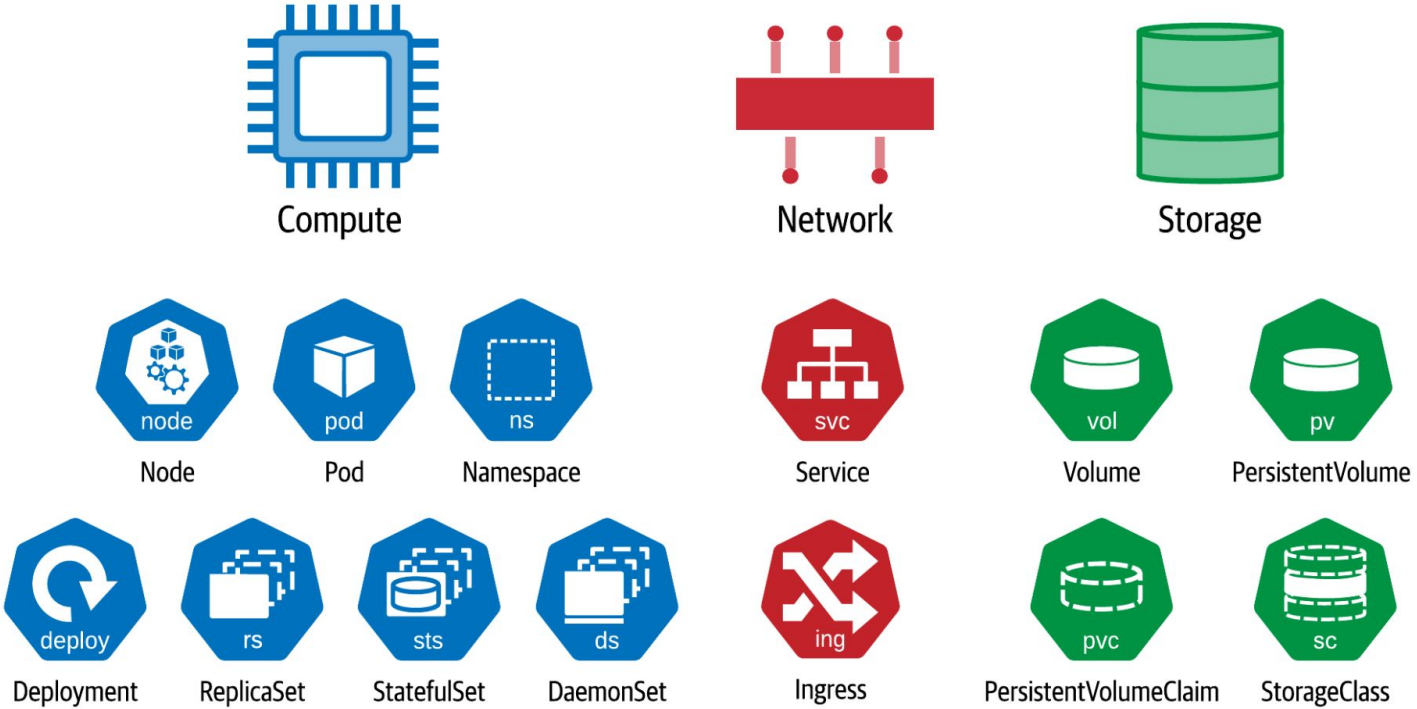
04

Pick an operator

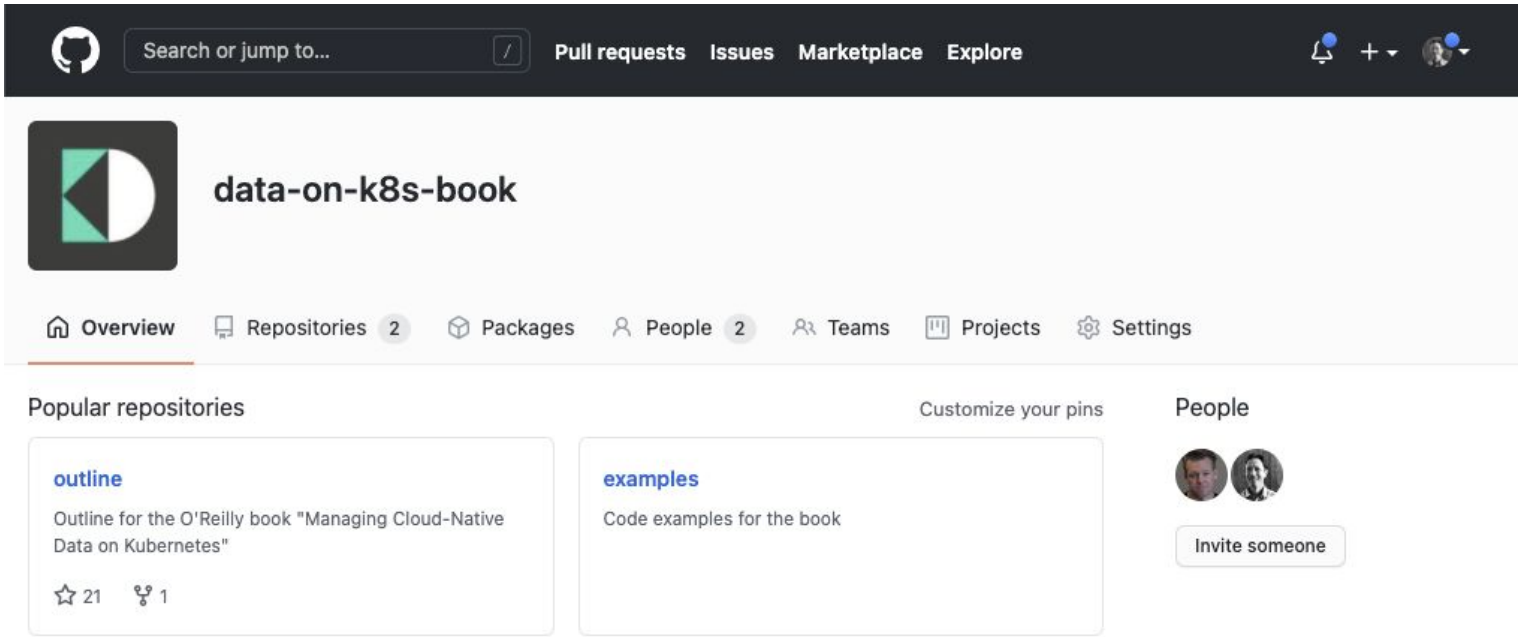
Fundamental Resources



Kubernetes Primitives

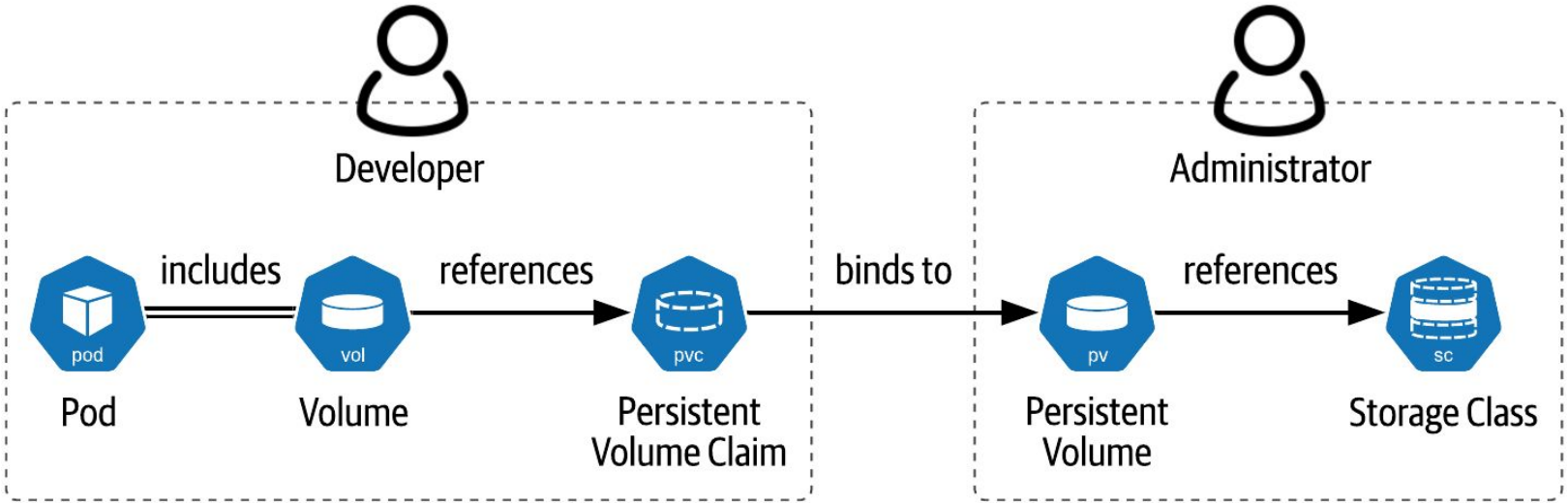


Example code

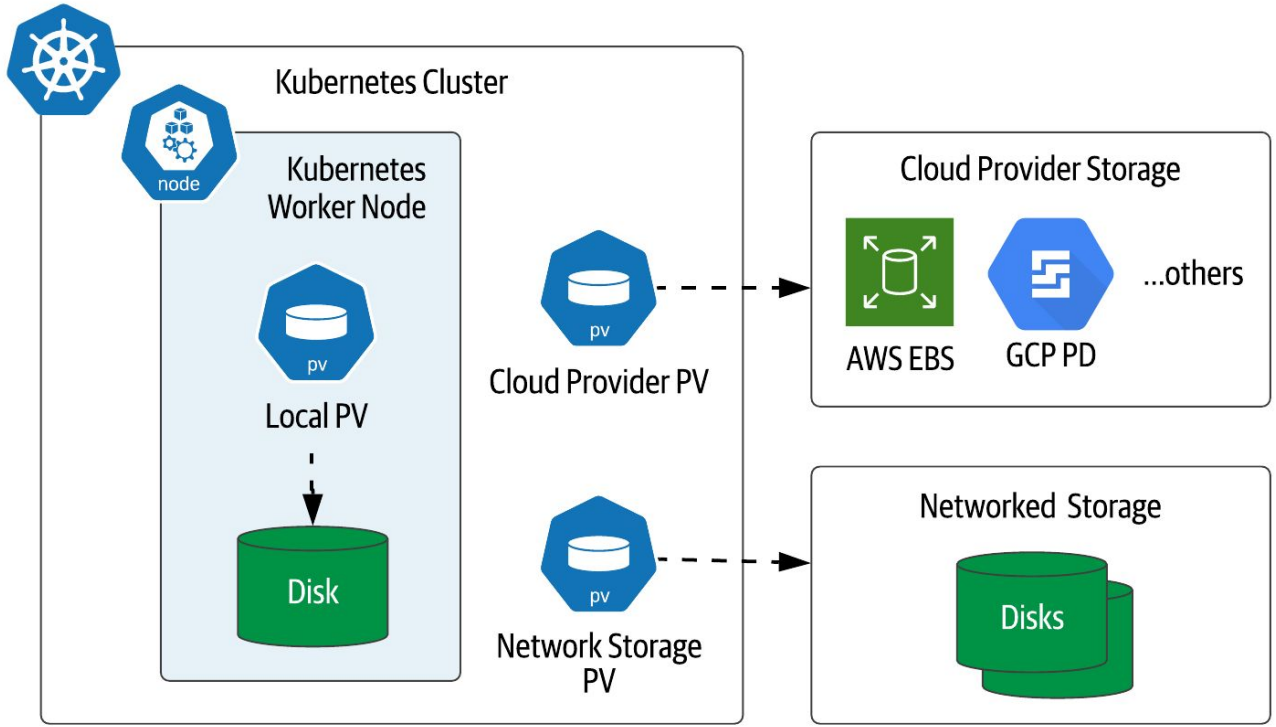


https://github.com/data-on-k8s-book/examples

Kubernetes Persistent Volume Subsystem



Persistent Volumes



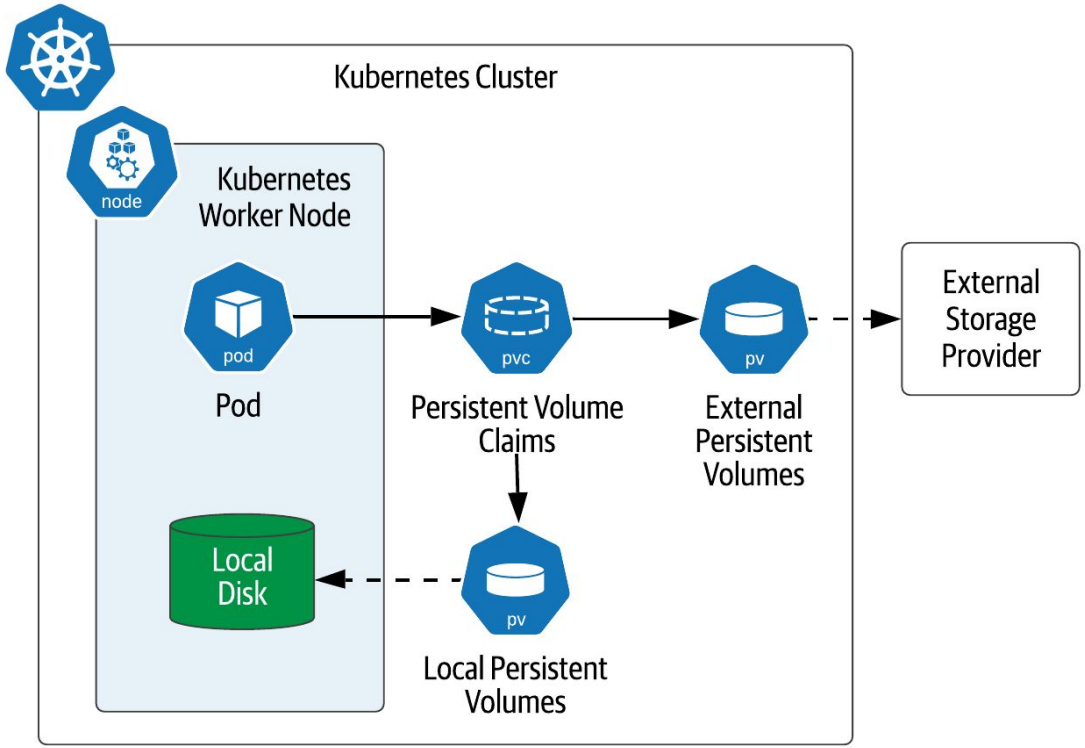
Persistent Volumes

Example local volume using node affinity

Default storage class

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-volume
spec:
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteOnce
  local:
    path: /app/data
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - node1
```

Persistent Volume Claims



Persistent Volume Claims

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-claim
spec:
  storageClassName: ""
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - mountPath: "/app/data"
      name: my-volume
  volumes:
  - name: my-volume
    persistentVolumeClaim:
      claimName: my-claim
```

How to put a database on Kubernetes

01

Learn the Kubernetes primitives

02

Pick a storage provider

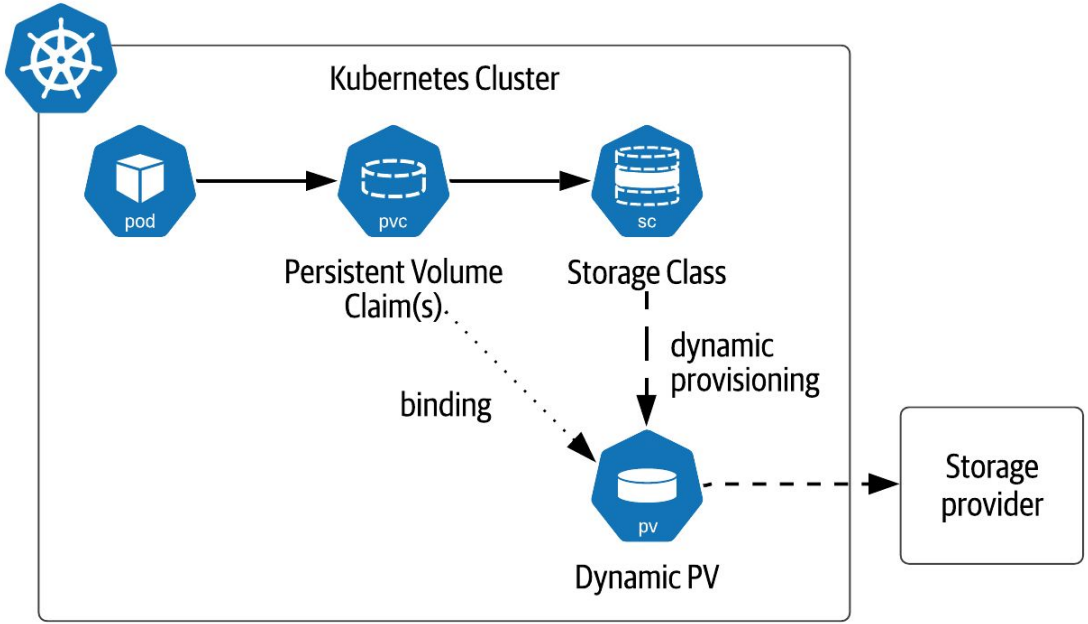
03

Pick a database

04

Pick an operator

Storage Class



Storage Classes

Many options available

- Local Storage
- Networked Storage
- Object Storage

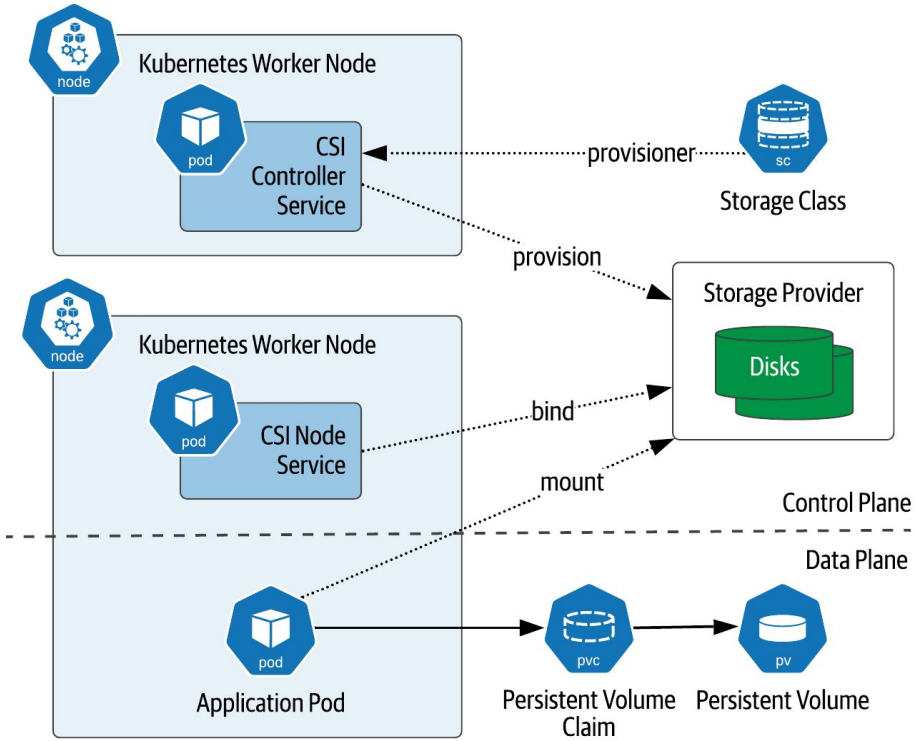
Many providers

- Public clouds
- CSI-compliant vendors

https://kubestr.io/

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-path
provisioner: rancher.io/local-path
volumeBindingMode: WaitForFirstConsumer
reclaimPolicy: Delete
```

Container Storage Interface (CSI)



How to put a database on Kubernetes

01

Learn the Kubernetes primitives

02

Pick a storage provider

03

Pick a database

04

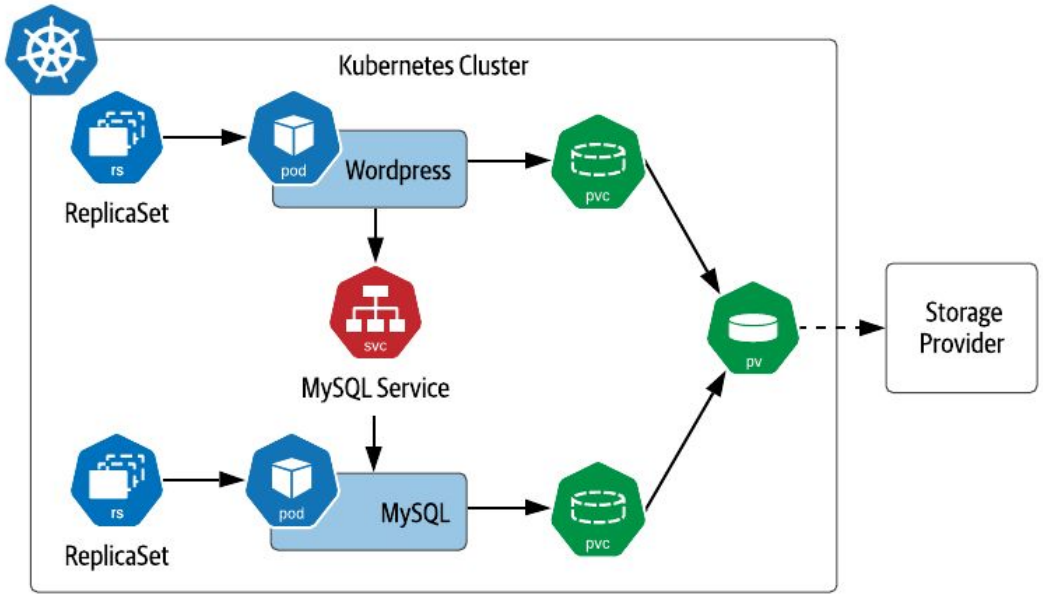
Pick an operator

Let's try a database

Relational, Single Node

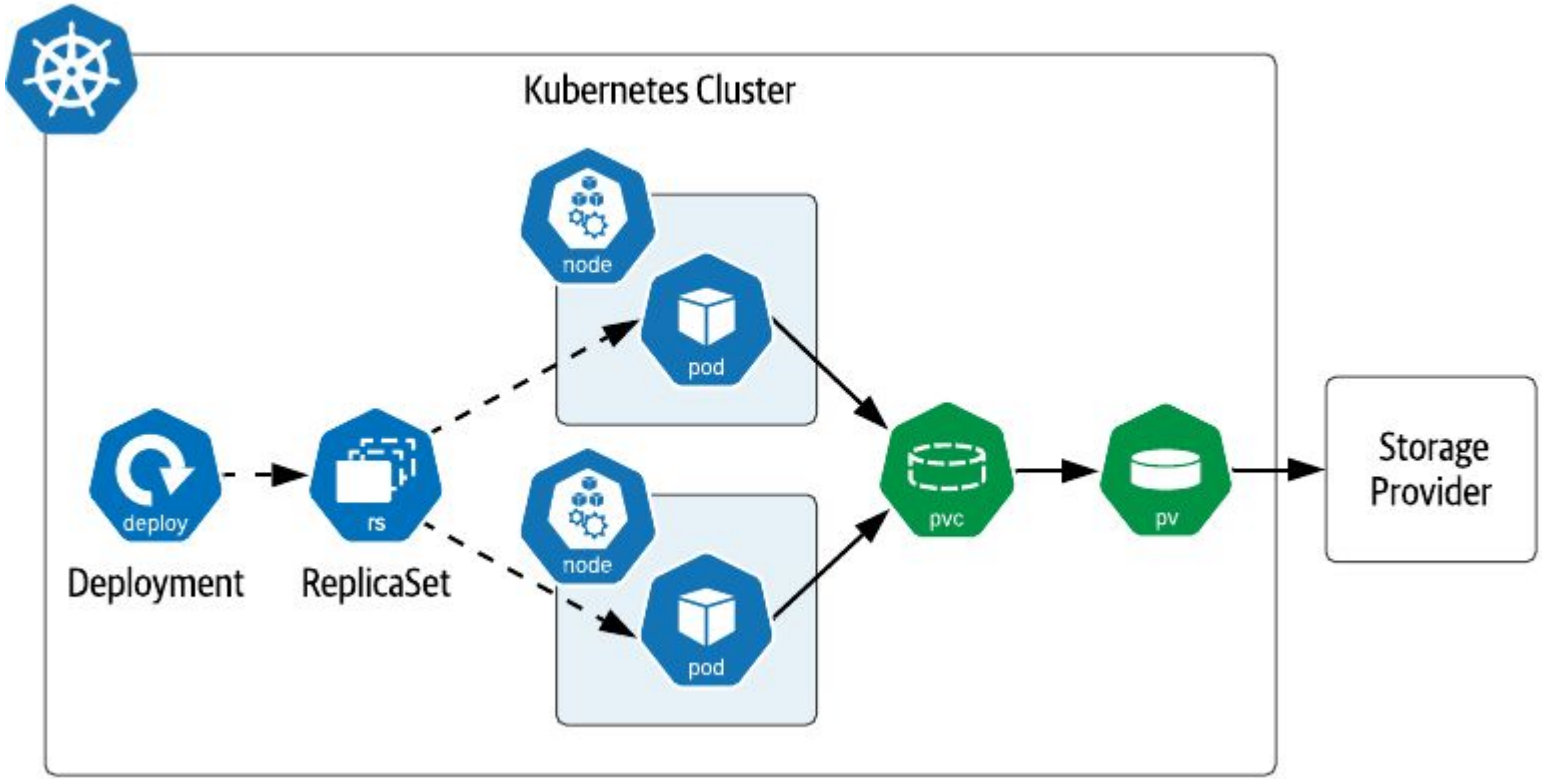
NoSQL, Multiple Nodes

Deploying MySQL on K8s



Inspiration: https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/
Update: https://github.com/data-on-k8s-book/examples/tree/main/ch3-mysql

Deployments and Replica Sets



Secret and PVC – MySQL

```
kubectl create secret generic
mysql-root-password
--from-literal=password=<your password>
```

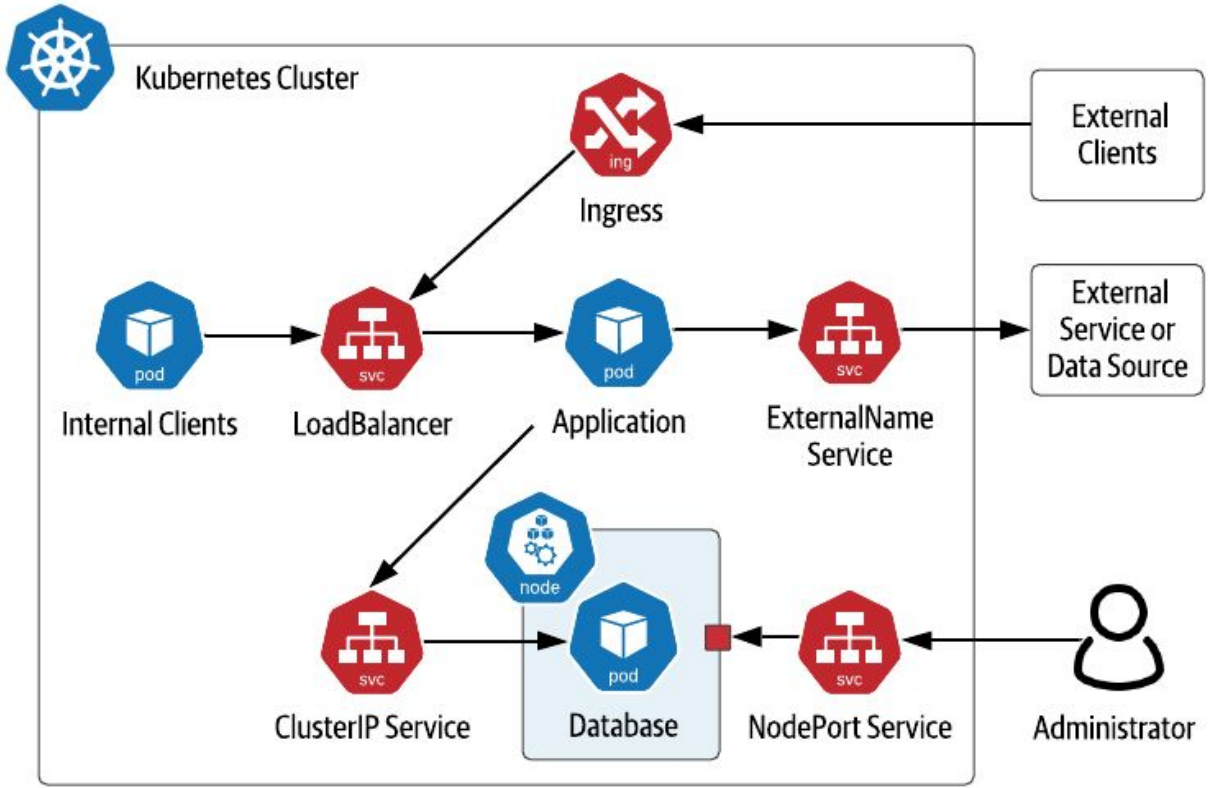
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Deployment – MySQL

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
```

```
  spec:
    containers:
    - image: mysql:5.7
      name: mysql
      env:
      - name: MYSQL_ROOT_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mysql-root-password
            key: password
      ports:
      - containerPort: 3306
        name: mysql
      volumeMounts:
      - name: mysql-persistent-storage
        mountPath: /var/lib/mysql
    volumes:
    - name: mysql-persistent-storage
      persistentVolumeClaim:
        claimName: mysql-pv-claim
```

Services

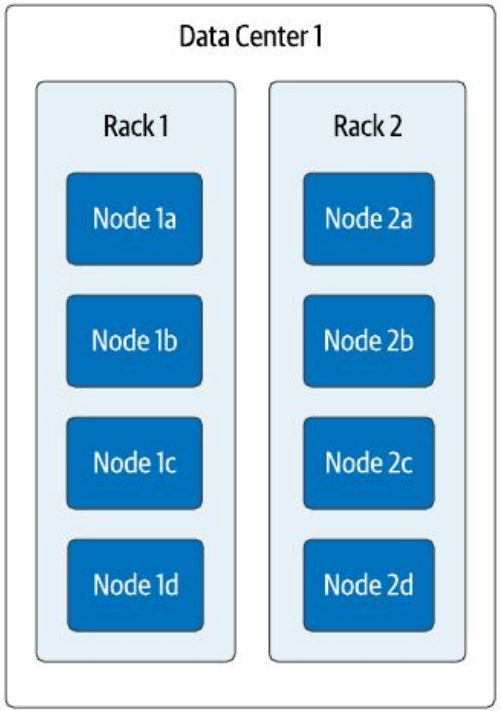

Kubernetes Cluster
ing
Ingress
External Clients
pod
Internal Clients
svc
LoadBalancer
pod
Application
svc
ExternalName Service
External Service or Data Source
node
pod
Database
svc
ClusterIP Service
svc
NodePort Service
Administrator

Service – MySQL

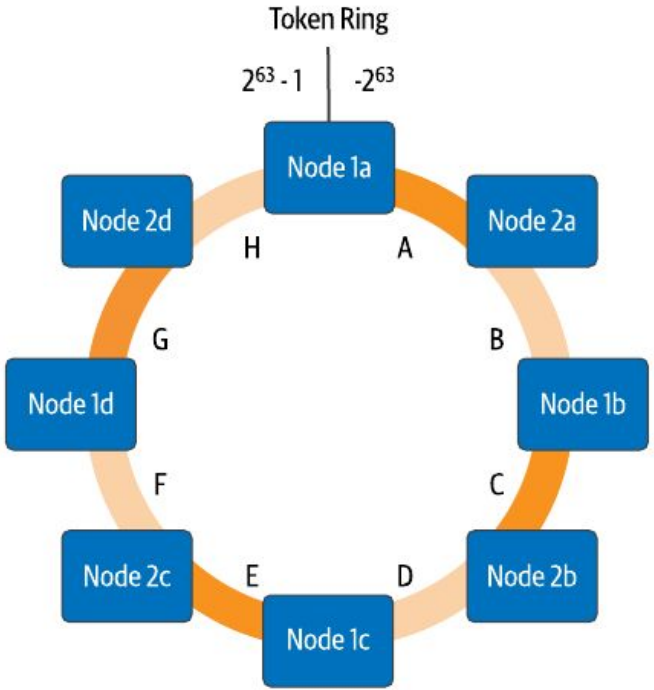
Headless service for access within cluster

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
```

Cassandra – Logical and Physical Views

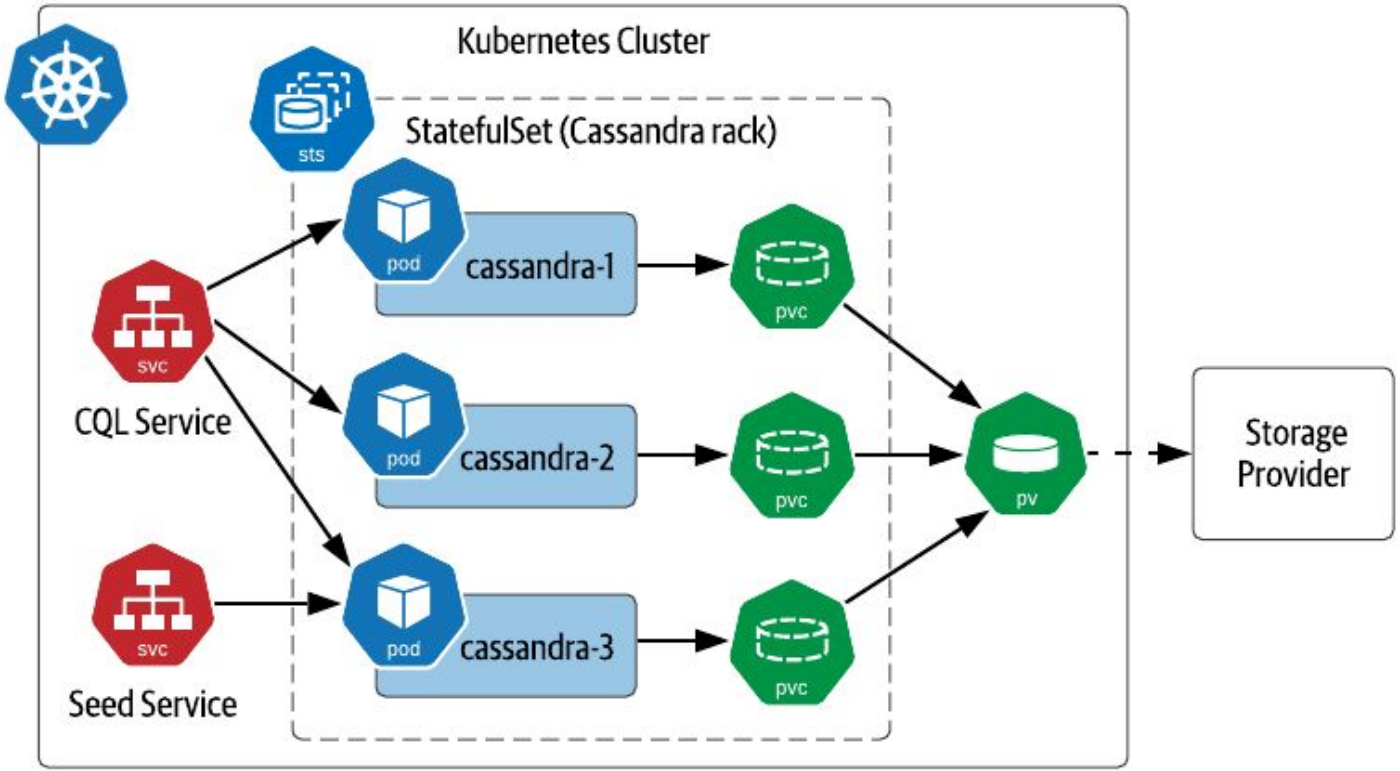


Physical View (Topology)



Logical View (Ring)

Deploying Cassandra on K8s



Service – Cassandra

Headless service for access within cluster

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: cassandra
  name: cassandra
spec:
  clusterIP: None
  ports:
  - port: 9042
  selector:
    app: cassandra
```

StatefulSet – Cassandra

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: cassandra
  labels:
    app: cassandra
spec:
  serviceName: cassandra
  replicas: 3
  podManagementPolicy: OrderedReady
  updateStrategy: RollingUpdate
  selector:
    matchLabels:
      app: cassandra
  template:
    metadata:
      labels:
        app: cassandra
```

```
  spec:
    containers:
    - name: cassandra
      image: cassandra
      ports:
      - containerPort: 7000
        name: intra-node
      - containerPort: 7001
        name: tls-intra-node
      - containerPort: 7199
        name: jmx
      - containerPort: 9042
        name: cql
      lifecycle:
        preStop:
          exec:
            command:
            - /bin/sh
            - -c
            - nodetool drain
```

StatefulSet – Cassandra (cont.)

```
        env:
          - name: CASSANDRA_CLUSTER_NAME
            value: "cluster1"
          - name: CASSANDRA_DC
            value: "dc1"
          - name: CASSANDRA_RACK
            value: "rack1"
          - name: CASSANDRA_SEEDS
            value:
"cassandra-0.cassandra.default.svc.cluster.l
ocal"
        volumeMounts:
        - name: cassandra-data
          mountPath: /var/lib/cassandra
```

```
volumeClaimTemplates:
- metadata:
    name: cassandra-data
  spec:
    accessModes: [ "ReadWriteOnce" ]
    storageClassName: standard-rwo
    resources:
      requests:
        storage: 1Gi
```

How to put a database on Kubernetes

01

Learn the Kubernetes primitives

02

Pick a storage provider

03

Pick a database

04

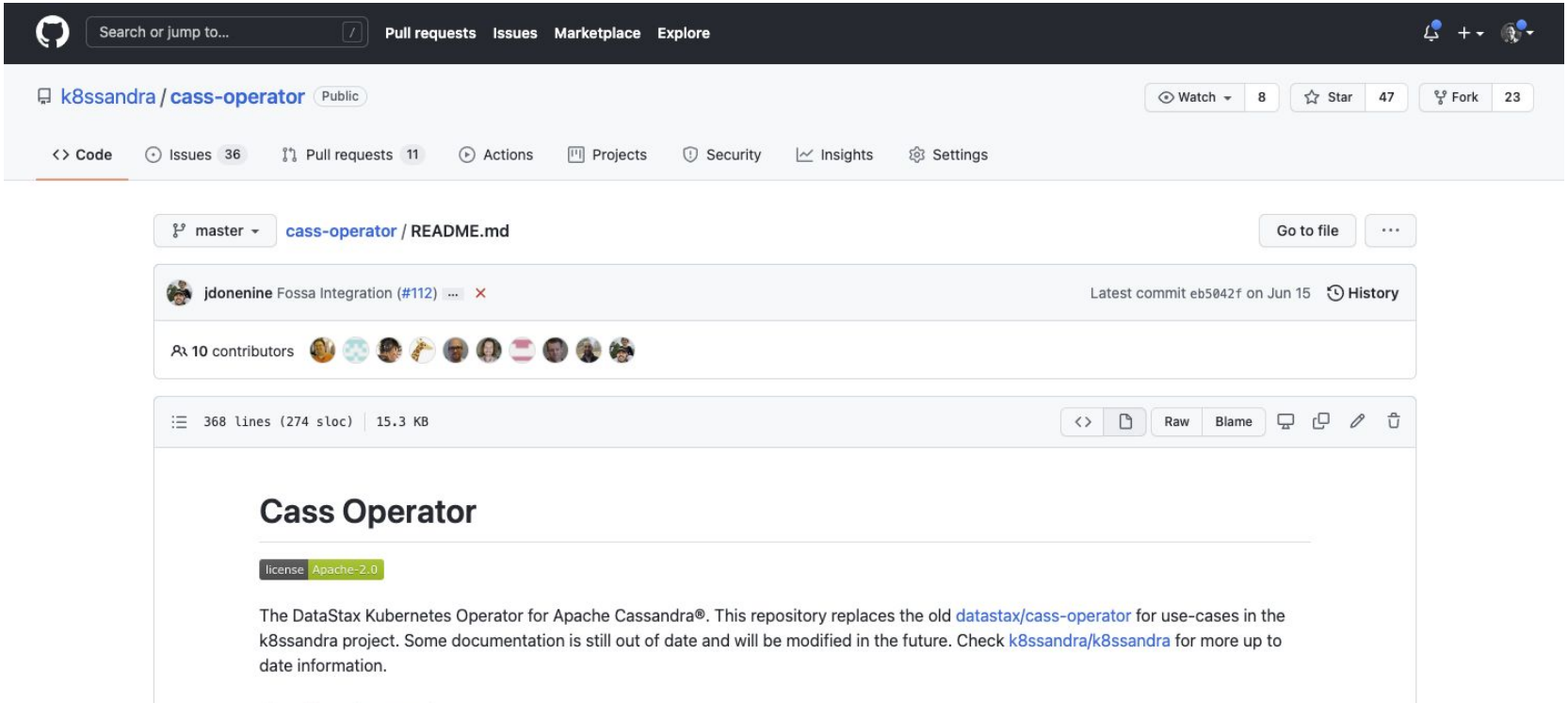
Pick an operator

“If you’re running a database on K8s, you should be using an operator. ”

Rick Vasquez, Western Digital

DoK Community Day - Kubecon NA 2021

Cass-operator – https://github.com/k8ssandra/cass-operator



Deploying Cassandra on K8s – What's Missing?

Affinity / Anti-Affinity

Resource requests

Availability / PodDisruptionBudgets

Backup and Restore

Secure provisioning of access credentials

Expanding volumes

K8SSANDRA

Cloud native, scalable data tier with administration tools and easy data access

Jeff Carpenter, Software Engineer, DataStax | @jscarp | jscarp.medium.com

K8ssandra – A Complete Cassandra Ecosystem for K8s

Data Gateway providing REST, GraphQL, Document APIs

Scalable cloud-native database managed via cass-operator

Reaper
Medusa

Cassandra utilities for repair and backup/restore

Metrics aggregation and visualization

Packaged and delivered via Helm charts

Kubecon Talk on Thursday, October 14

2:30 PM - 3:05 PM

Investigating multi-cluster deployments for data (Intermediate level)

https://sched.co/lV2m

New Book!

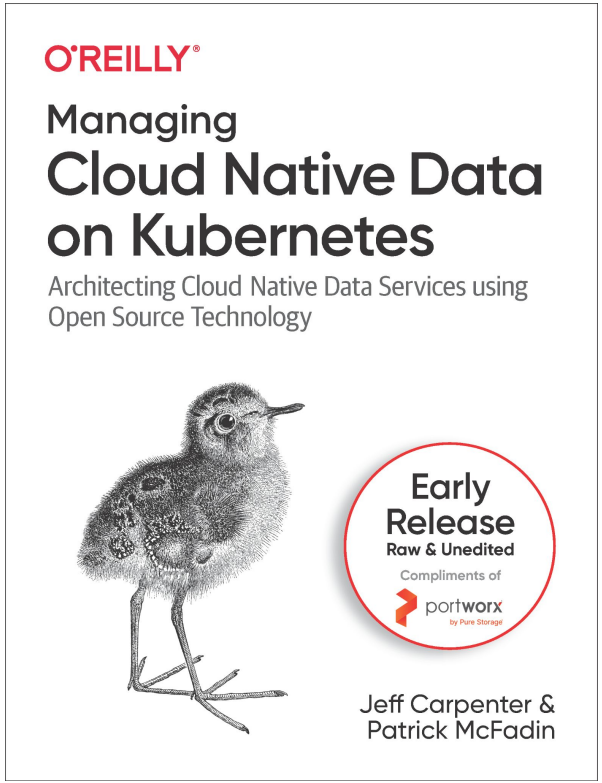
Early release now available at: https://learning.oreilly.com and https://portworx.com/resource/ebook-managing-cloud-native-data-on-kubernetes/

Chapter 1: Introduction to Cloud Native Data Infrastructure

Chapter 2: Managing Data Storage on Kubernetes

Chapter 3: Running Databases on Kubernetes (the hard way)

More info: https://github.com/data-on-k8s-book



DataStax Booth P15

Visit us at Booth P15
for K8ssandra demos

AND

your chance to
Win an in-home
Dream Arcade!

We'll pay for
The shipping
to your home!