Cloud Native Deployment Models
Operators, Protection & Mobility

Michael Cade

Field CTO

Veeam Software

Gunnar Grosch

Principal Developer Advocate

AWS

Agenda

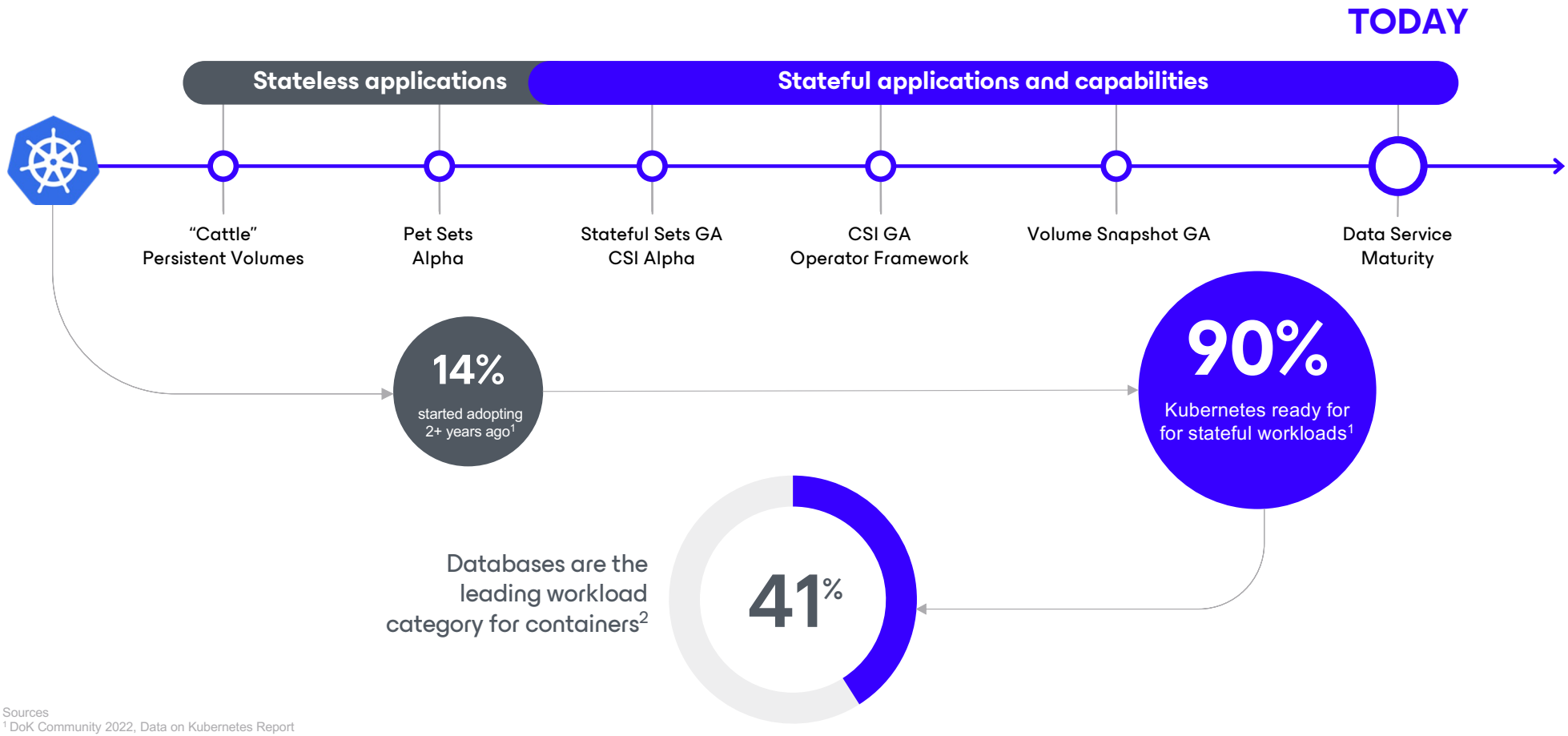
Why run Databases in Cloud Native?

Running Databases on Kubernetes

*How to run Cloud Native Databases

Maturing Kubernetes

From stateless to stateful

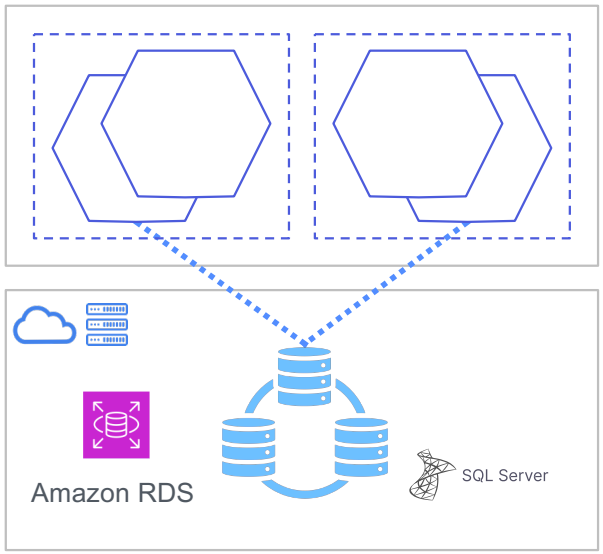


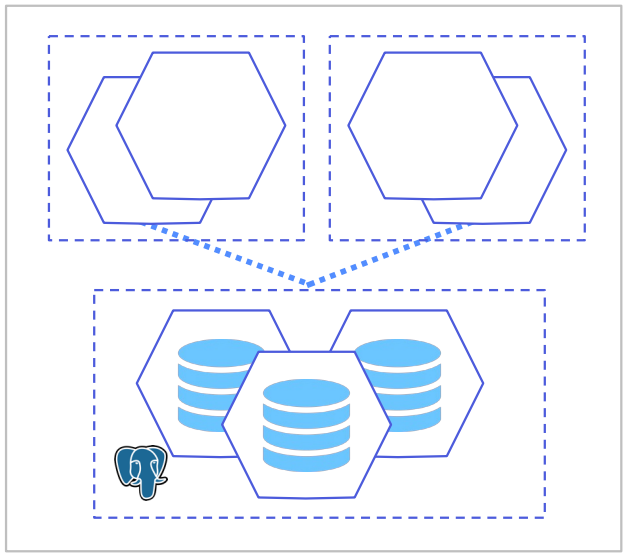
Sources
[1] DoK Community 2022, Data on Kubernetes Report
2 Datadog 2023, 10 Insights on Real-World Container Use

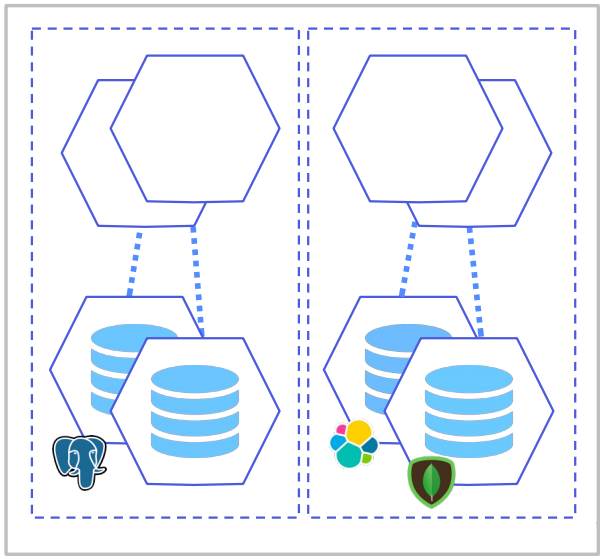
Cloud Native Deployment Patterns

Multiple paths to providing data persistence

Application using data services outside of Kubernetes	Data services in Kubernetes – separate from Application	Application includes data services – all in Kubernetes







KASTEN K10

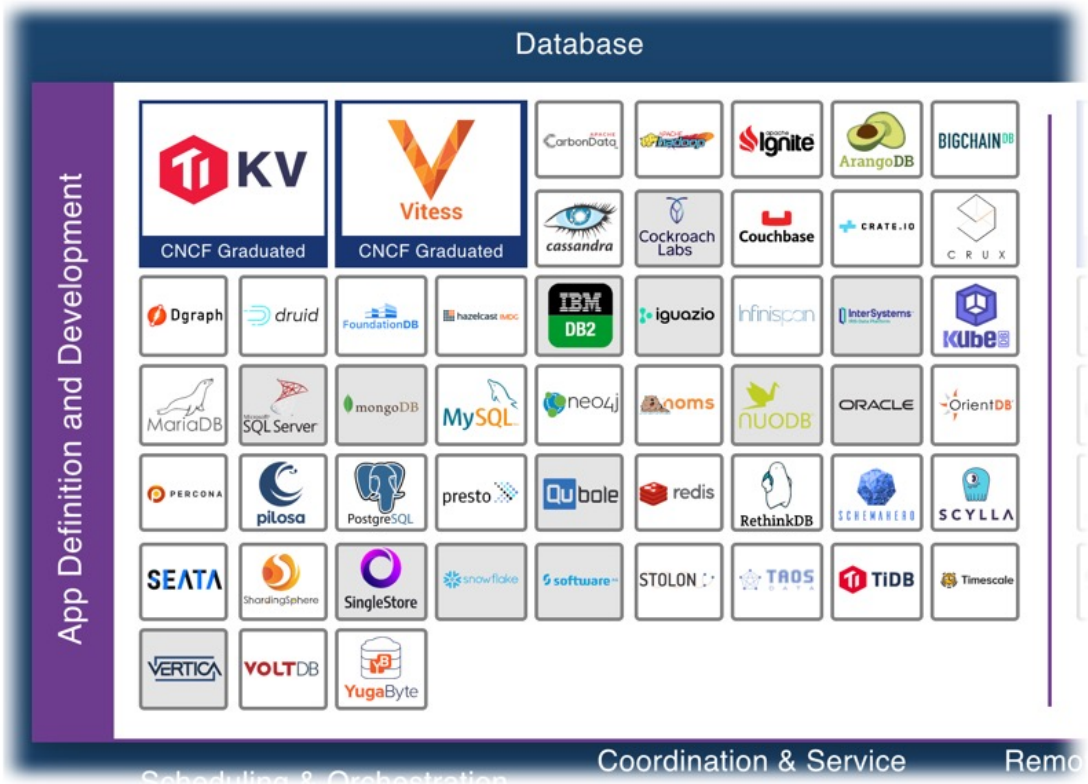
Kanister.io

veeam

What is a Cloud Native Database?

A cloud native database is a database that is designed to take full advantage of cloud technology and distributed systems.

Database Overload...



CLOUD NATIVE DATABASES

- **DESIGNED**

  to take full advantage of cloud technology and distributed systems.

- **CLOUD READY VS CLOUD NATIVE**

  Just because databases can run in the cloud, there is a difference between cloud-ready and cloud native.

- **SERVICES, SOFTWARE, APIS**

  databases that are architected and built to be run and deployed on the cloud while benefiting from features that the cloud native systems offers.

- **CONTAINERS**

  Cloud Native = Containers, Our Databases need to be able to run in containers

- **KUBERNETES**

  run on cloud native technologies such as Kubernetes to deliver a flexible and scalable data storage and querying solution.

TRADITIONAL DATABASES

- **SCALABILITY**
- **SECURITY**
- **ACCESSIBILITY**

Even though they can be integrated with the cloud, using these databases in the cloud limits the ability of applications to enjoy the benefits of cloud technology.

STATEFULSET

STARTED WITH A STATEFULSET NOW WE'RE HERE.

The Kubernetes controller can only get us so far around data migration, management and replication.

GOOD

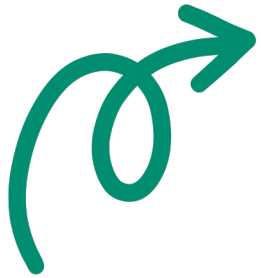
- Ordered Deployment
- Stable Network Identifiers

BAD

- Complex Configuration
- Scaling Limitations
- Rolling Updates

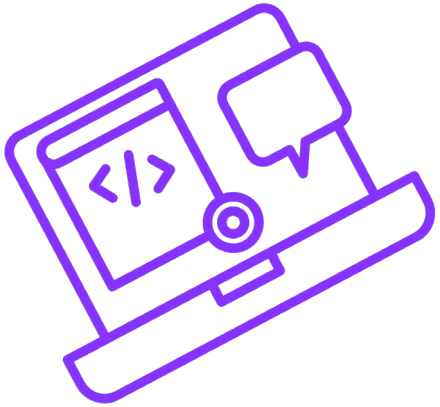
UNKNOWN

- Performance Impact
- Integration with operators is still evolving



DEMO

A STATEFULSET



PostgreSQL

veeam

```
NAME                    READY   STATUS    RESTARTS   AGE
pod/postgresql-db-0     1/1     Running   0          3m24s
pod/postgresql-db-1     1/1     Running   0          2m55s

NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
service/kubernetes        ClusterIP      10.96.0.1       <none>          443/TCP          21m
service/postgres-db-lb    LoadBalancer   10.111.253.4    10.111.253.4    5432:30975/TCP   3m7s

NAME                              READY   AGE
statefulset.apps/postgresql-db    2/2     3m24s
```

Kubernetes PostgreSQL StatefulSet

01-storage-class.yaml

Raw

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
```

02-persistent-volume.yaml

Raw

```yaml
kind: PersistentVolume
apiVersion: v1
metadata:
  name: postgres-pv
  labels:
    app: postgres
    type: local
spec:
  storageClassName: local-storage
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/var/data"

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: postgres-pv-claim
  labels:
    app: postgres
spec:
  storageClassName: local-storage
  capacity:
```

OPERATORS

OPERATORS KNOW BEST

Developed to handle the sophisticated, stateful applications that the default Kubernetes controllers can't handle

An application-specific controller that can help you manage a Kubernetes application.

GOOD

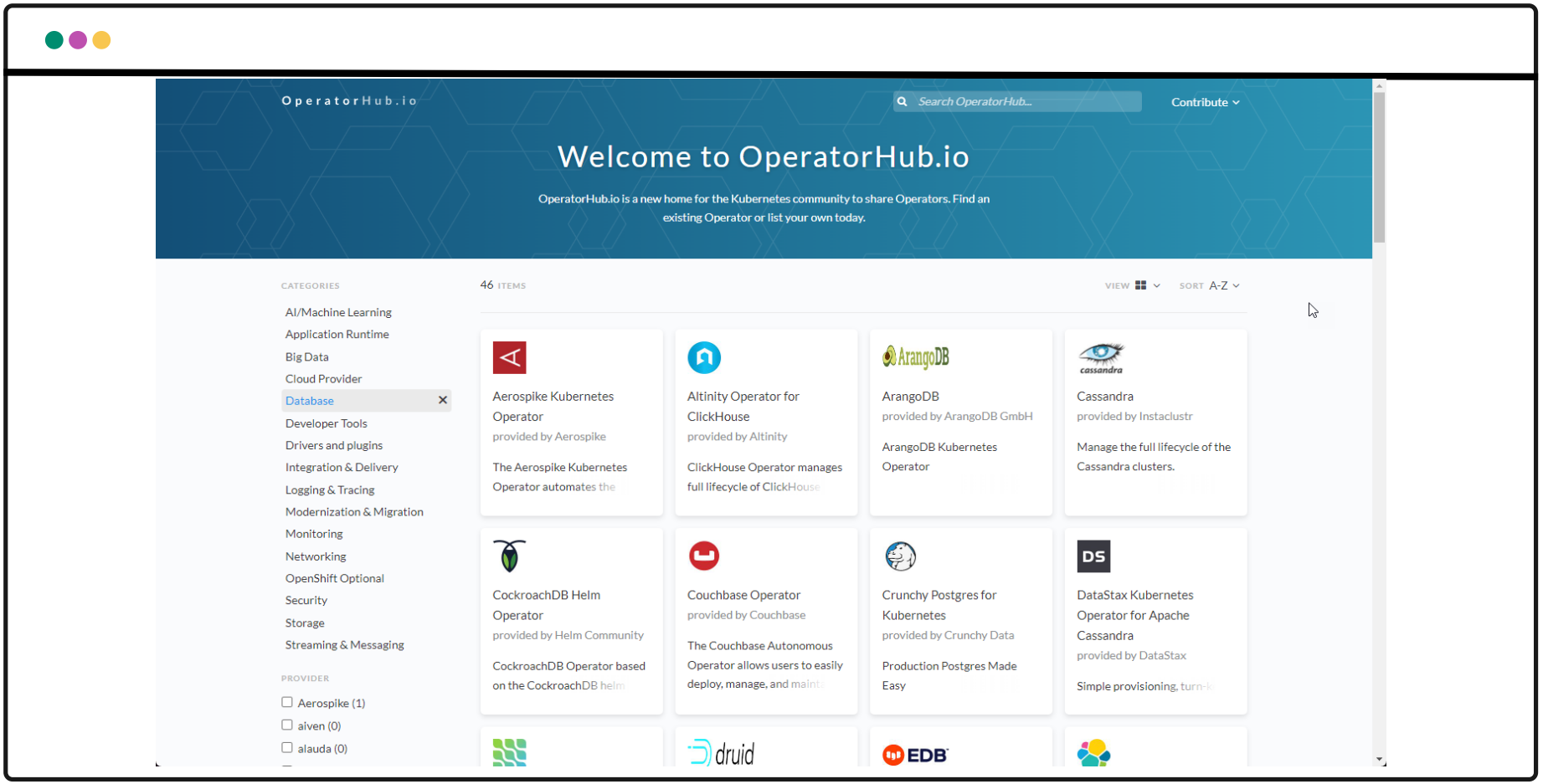
- Automated Operations
- Custom Resources
- Scaling & Self Healing

BAD

- Complexity
- Resource Intensive

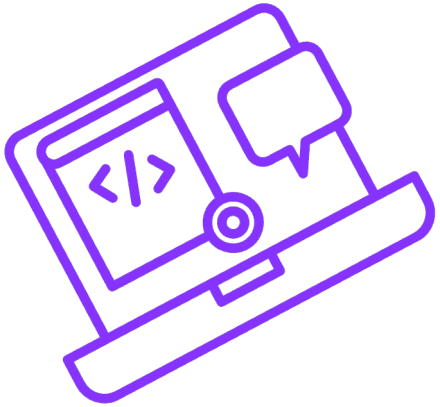
UNKNOWN

- Security Implications – (custom code)
- Operator Ecosystem



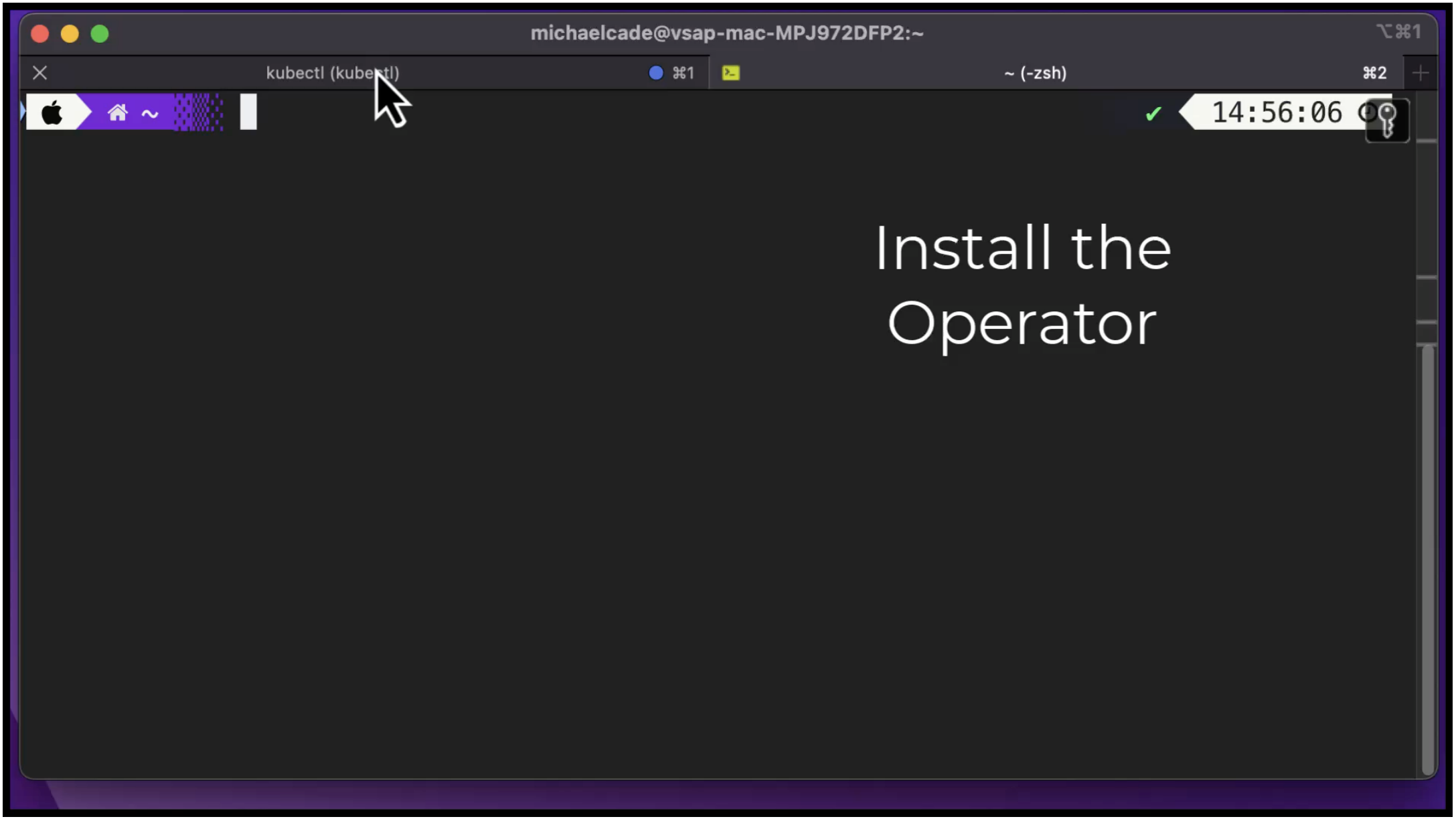
DEMO

THE OPERATOR MODEL



EDB™
POWER TO POSTGRES

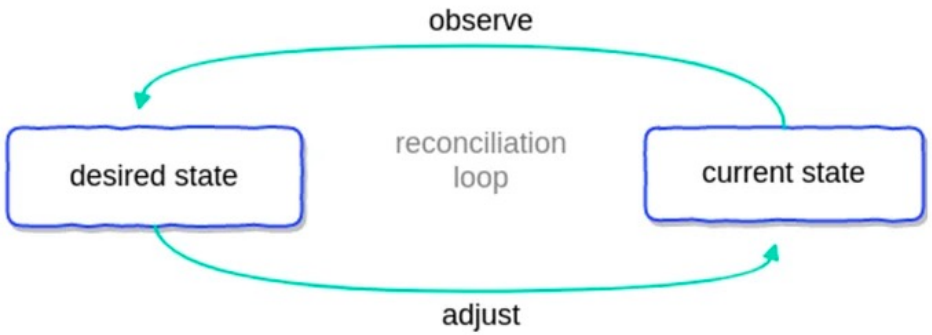
veeam


michaelcade@vsap-mac-MPJ972DFP2:~
kubectl (kubectl)
⌘1
~ (-zsh)
⌘2
14:56:06
Install the
Operator

THE RECONCILIATION LOOP

WATCHING FOR CHANGES

- Continuous Control
- State Synchronization
- Desired State
- Automated Convergence
- Self-Healing Process



EXTERNAL PAAS

PAAS THE OVERHEARD

Amazon Relational Database Service (Amazon RDS)

Google Cloud SQL

MongoDB® Atlas

GOOD

- Ease of use
- Scalability
- Managed Service

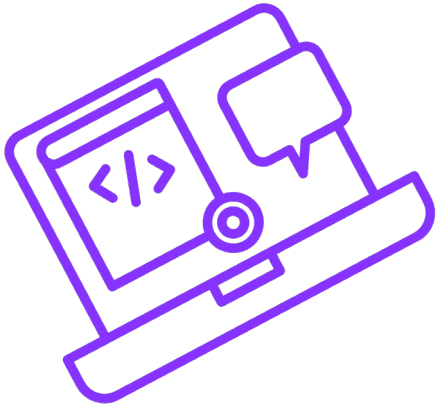
BAD

- Limited Control
- Cost

UNKNOWN

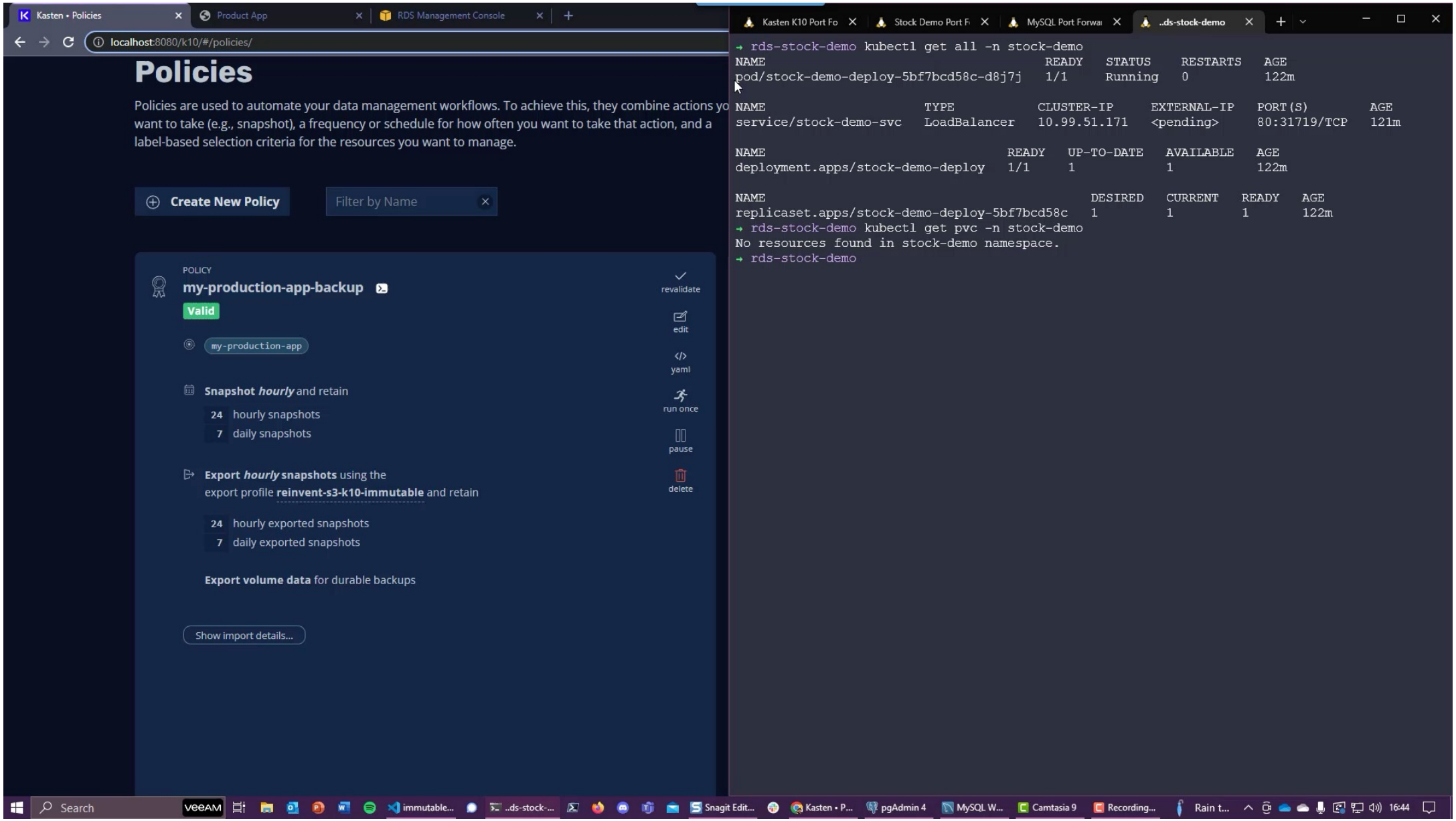
- Data Sovereignty
- Performance at scale

DEMO

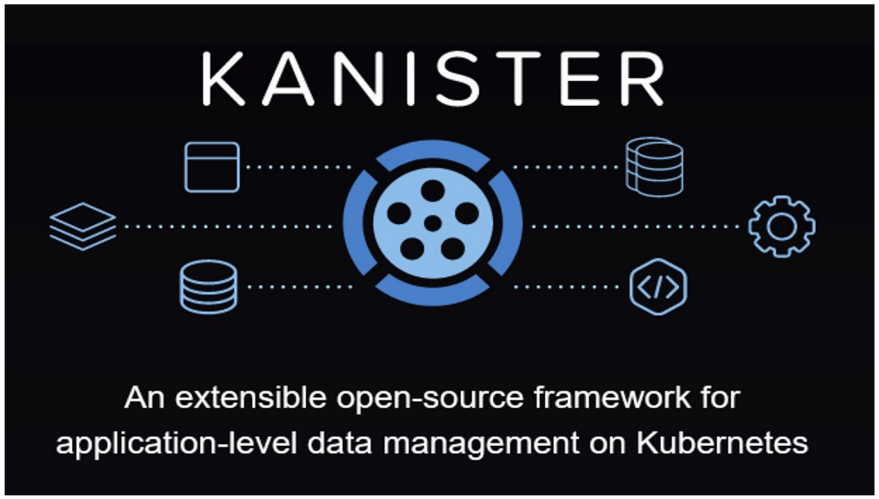


REGARDLESS OF CHOICE WE STILL NEED

BACKUP


Kasten • Policies
Product App
RDS Management Console
localhost:8080/k10/#/policies/
Policies
Policies are used to automate your data management workflows. To achieve this, they combine actions yo
want to take (e.g., snapshot), a frequency or schedule for how often you want to take that action, and a
label-based selection criteria for the resources you want to manage.
Create New Policy
Filter by Name
POLICY
my-production-app-backup
Valid
my-production-app
Snapshot hourly and retain
24 hourly snapshots
7 daily snapshots
Export hourly snapshots using the
export profile reinvent-s3-k10-immutable and retain
24 hourly exported snapshots
7 daily exported snapshots
Export volume data for durable backups
Show import details...
revalidate
edit
yaml
run once
pause
delete
Kasten K10 Port Fo
Stock Demo Port F
MySQL Port Forwai
..ds-stock-demo
→ rds-stock-demo kubectl get all -n stock-demo
NAME READY STATUS RESTARTS AGE
pod/stock-demo-deploy-5bf7bcd58c-d8j7j 1/1 Running 0 122m
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/stock-demo-svc LoadBalancer 10.99.51.171 <pending> 80:31719/TCP 121m
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/stock-demo-deploy 1/1 1 1 122m
NAME DESIRED CURRENT READY AGE
replicaset.apps/stock-demo-deploy-5bf7bcd58c 1 1 1 122m
→ rds-stock-demo kubectl get pvc -n stock-demo
No resources found in stock-demo namespace.
→ rds-stock-demo
Search
Rain t...
16:44

Introducing



https://kanister.io

THANK YOU