OpenShift Commons

kasten by Veeam

Backup and DR for databases on OpenShift with Kasten by Veeam

Michael Courcy
Soln. Arch., Kasten by Veeam

Gaurav Rishi
VP Product, Kasten by Veeam

Content

Introducing Kasten by Veeam

- Use cases and architecture
- Customer benefits

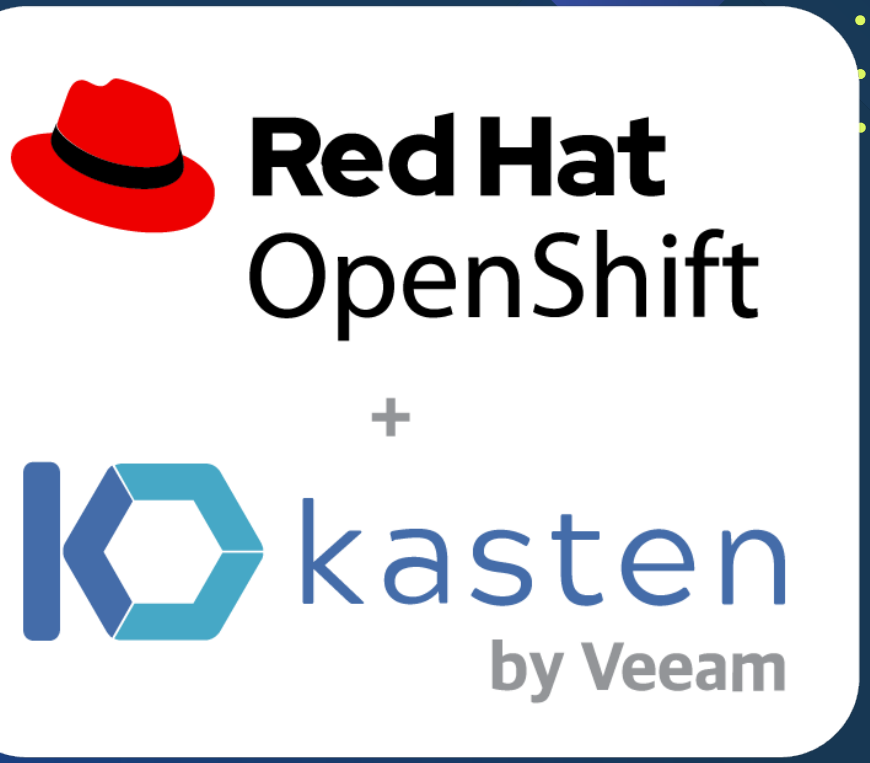
Demonstrating Kasten K10 on Red Hat OpenShift

- Protecting applications – including databases
- Backup and DR

kasten by Veeam

"We are pleased to collaborate with Kasten by Veeam so organizations across the globe can use Kasten K10 data management platform with Red Hat OpenShift to rollout, upgrade, and protect their cloud-native applications."

- Joe Fernandes, VP and GM, Core Cloud Platforms at Red Hat



Kasten K10 by Veeam

Use Cases

Backup and Restore

Multi-tenancy, RBAC, Scale, Performance, Polyglot Persistence

Disaster Recovery

Cross-AZ and Region, Multi and Hybrid Cloud, Storage Transforms

Application Mobility

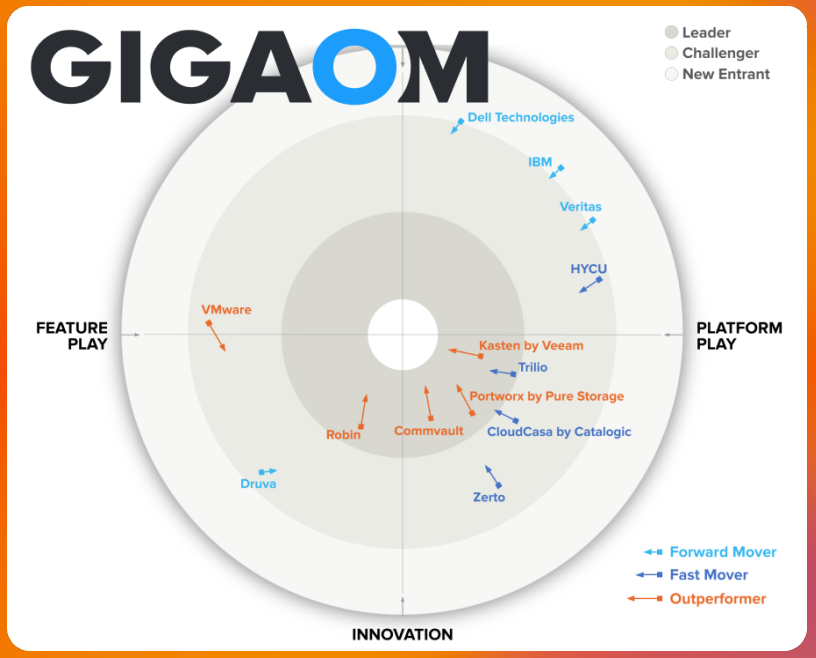
Cluster Upgrades, Application Transforms, Test/Dev Clusters

kasten by Veeam

Kasten: Kubernetes Data Protection Leader - Again!

#1 Kubernetes Backup

ANALYST REPORT



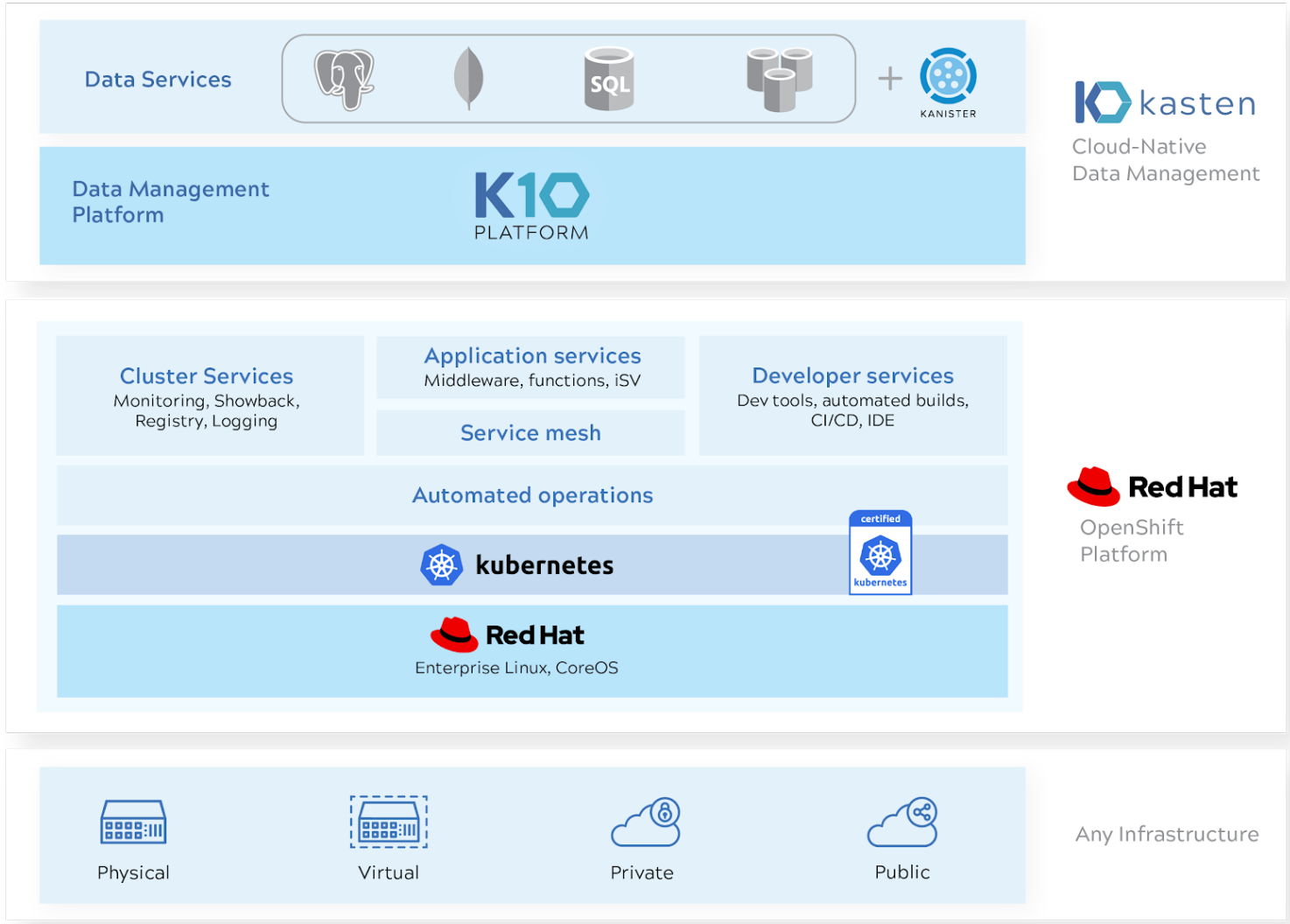
"Kasten is a Kubernetes-native, mature solution that's very suitable for self-hosted, self-managed use cases. Its architecture scales well and is especially well-suited for edge deployments. Its RBAC features and centrally managed policy model are well aligned with large enterprise and self-service requirements."

- Enrico Signoretti, Senior Data Storage Analyst, GigaOm

kasten by Veeam

Kasten K10 on Red Hat OpenShift

Kubernetes backup and mobility

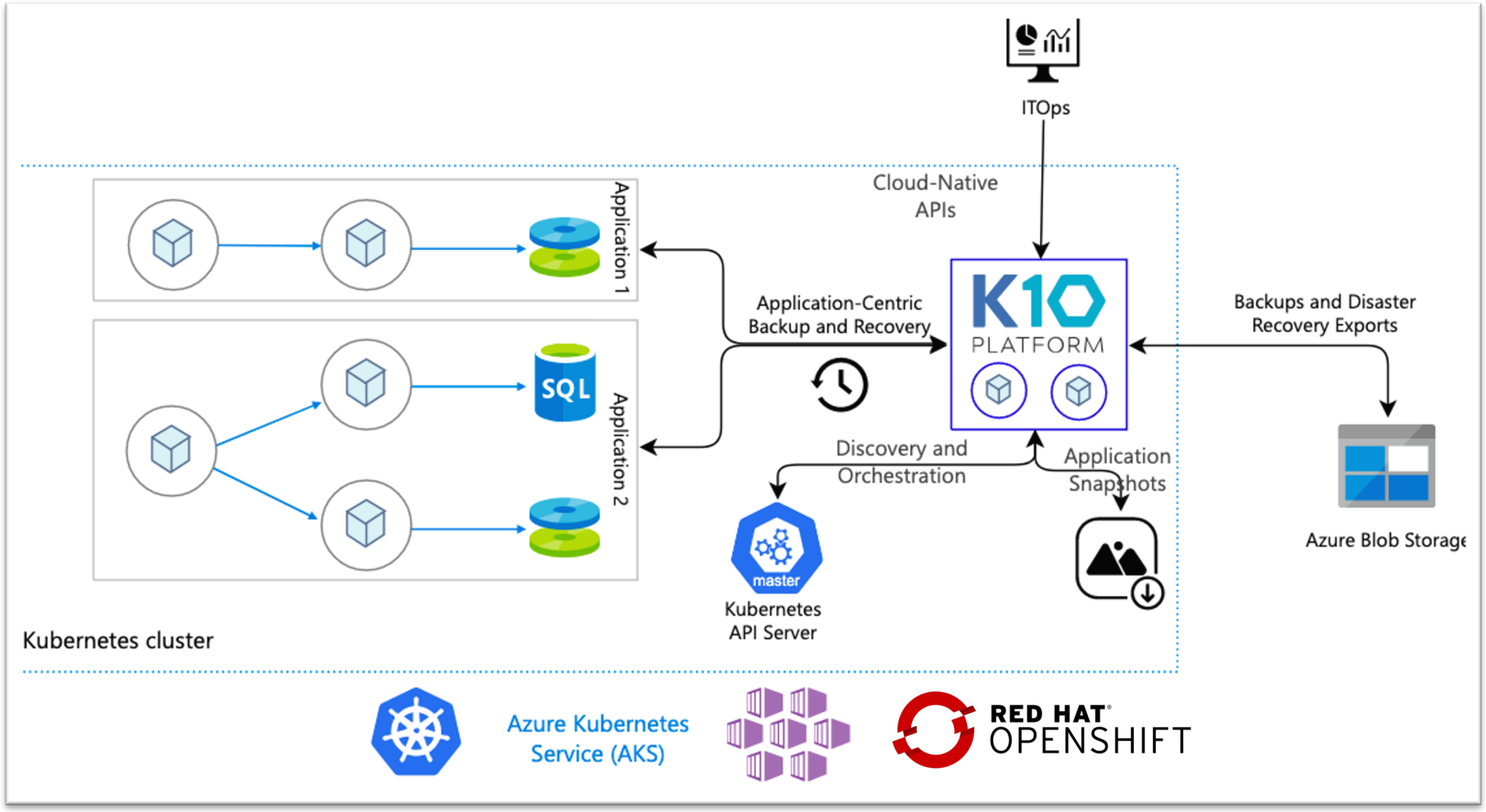


1. Application-Centric
2. Backup, DR & Mobility
3. Red Hat Operator Certified
4. Red Hat Marketplace (including transactions)
5. Freedom of Choice (including OCS)

kasten by Veeam

e.g., Kasten K10 on Azure Red Hat OpenShift (ARO)

Seamless operations



- Application Discovery
- Policy-driven Automation
- End-to-End Security
- Ease of Use, Simple UX

kasten by Veeam

kasten by Veeam

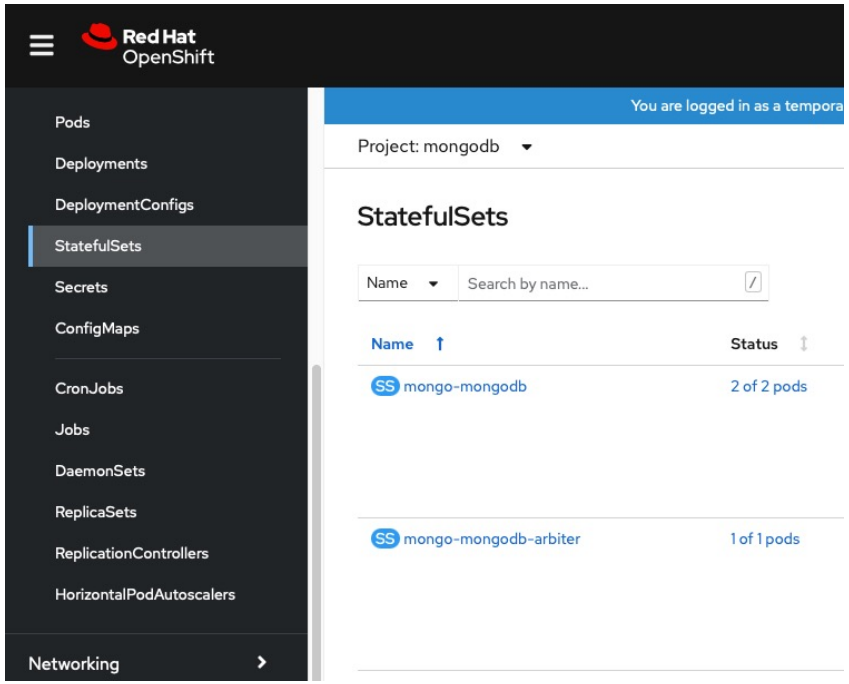
Kasten K10 with Demo

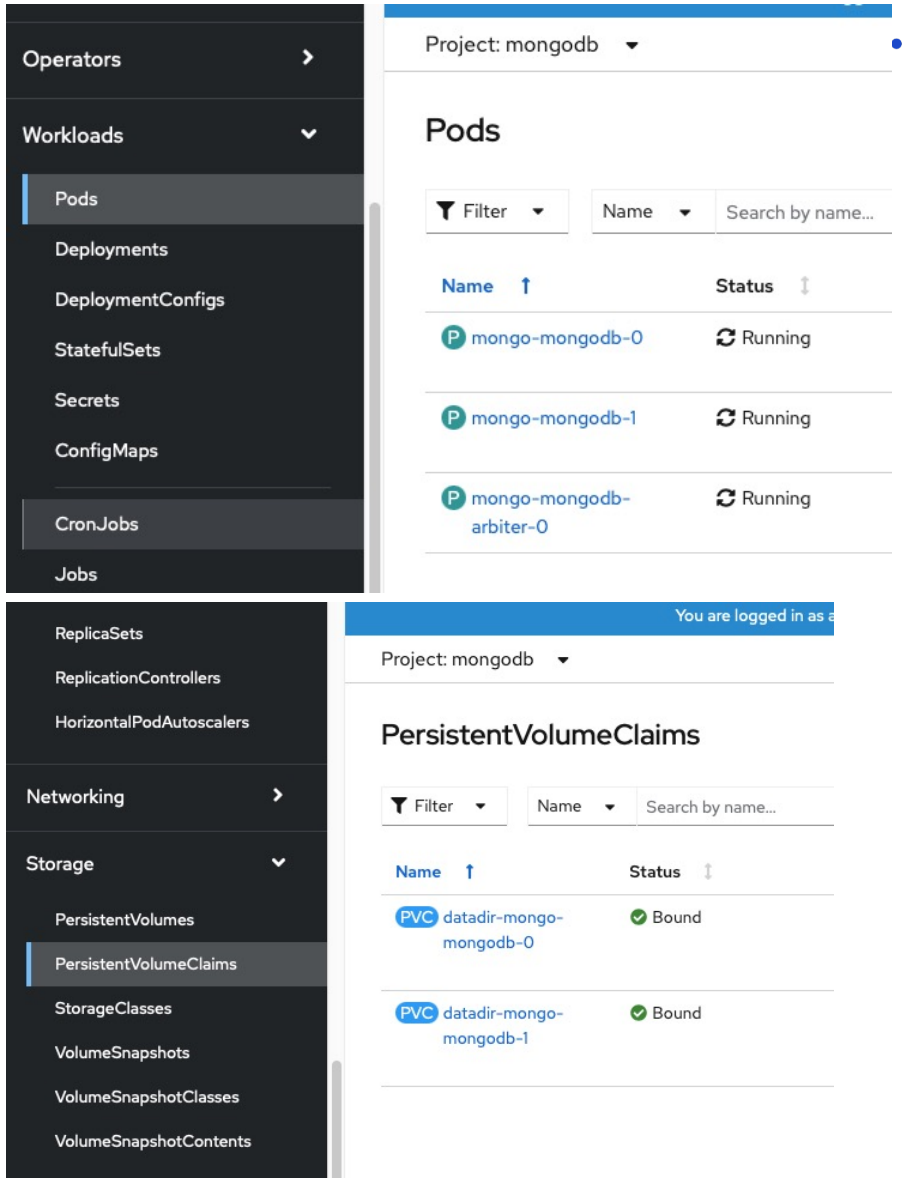
Backing up MongoDB : a complete example

MongoDB deployed with an helm chart

2 Replicas, one Arbiter

Openshift 4.8 on Azure





kasten by Veeam

Backing up MongoDB: Questions ??

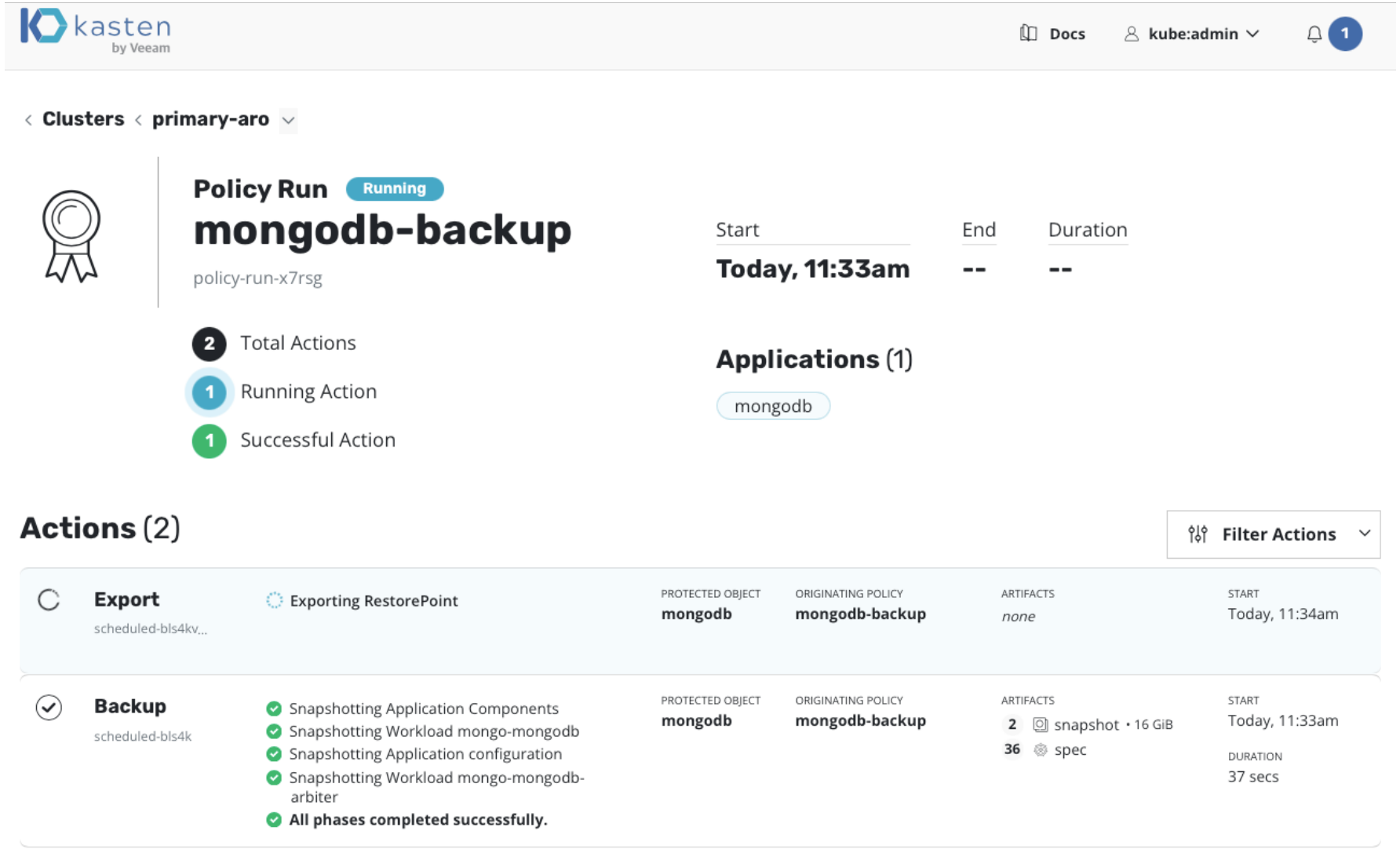
What's the best approach to backup this workload ?

- Capture each PVC ?
- Capture a snapshot of each PVC ?
- Capture a mongodump against the mongo service ?
- Use fsyncLock() on all the replicas and capture a snapshot of each PVC ?

What options can Kasten support?

kasten by Veeam

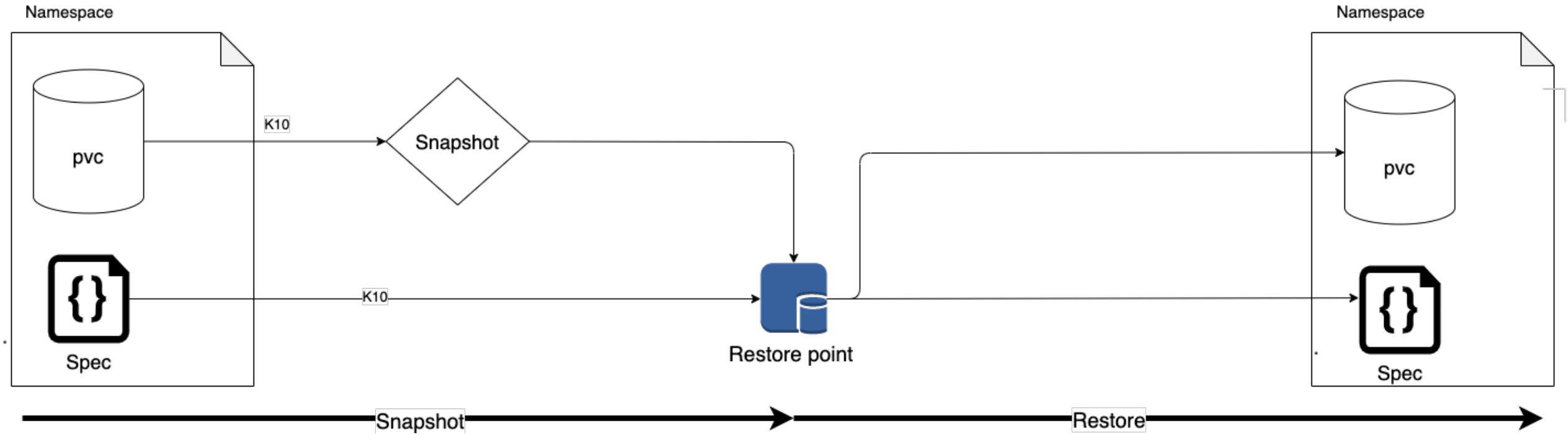
Demo ...



kasten by Veeam

Just capture a snapshot

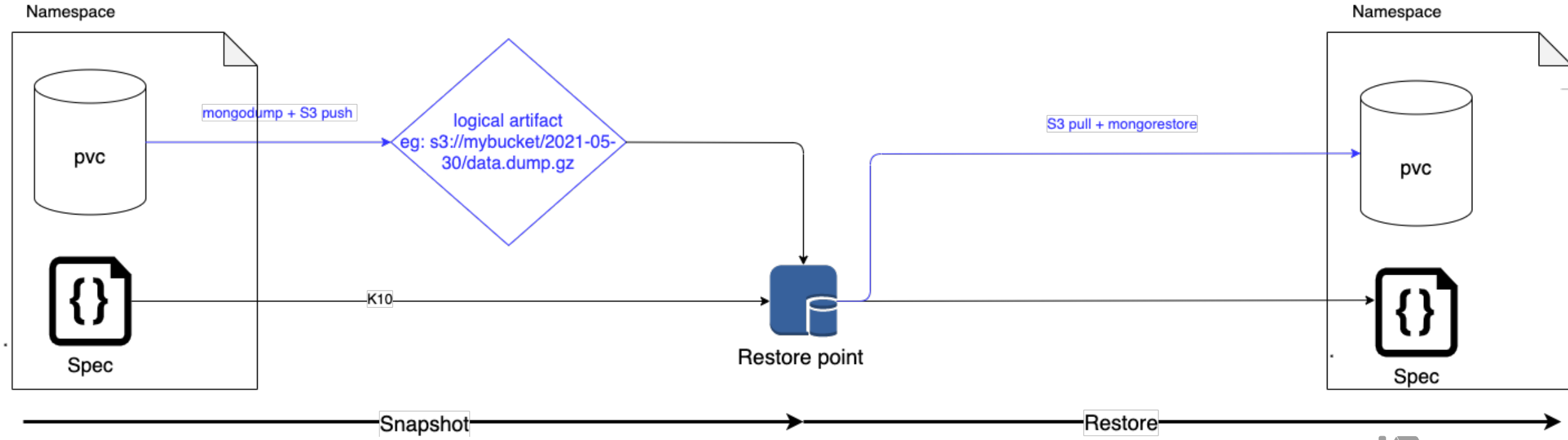
- Snaphshot are « crash consistent » and Kasten leverage cloud API or CSI API
- All the objects in the namespace are also captured, it’s a namespace snap
- However storage snapshot are not « application consistent »



kasten by Veeam

Capture a mongodump

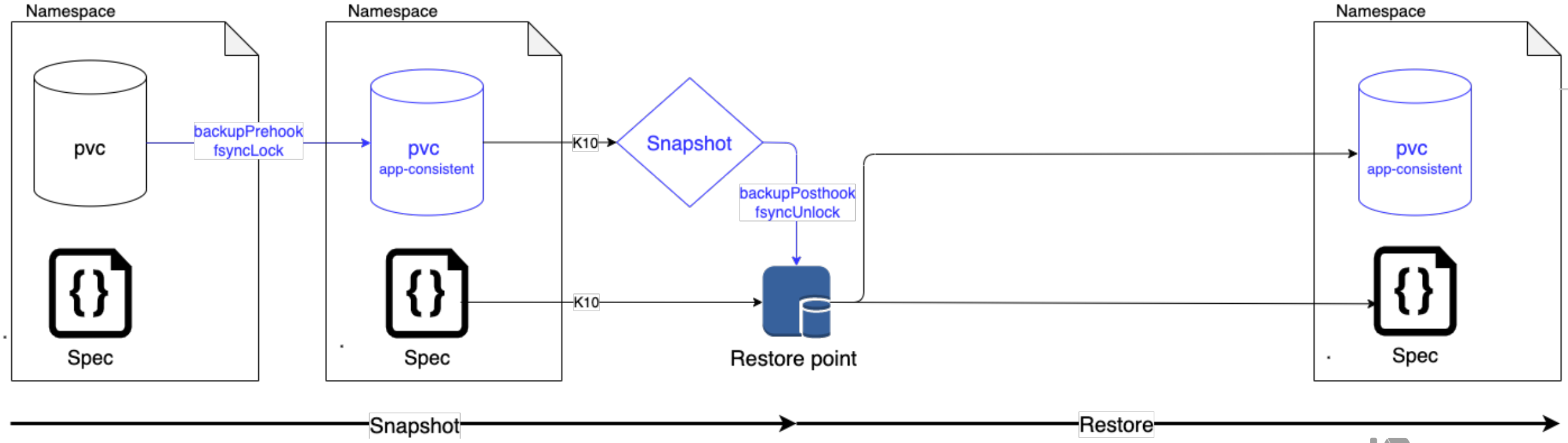
- We don’t capture the PVC anymore
- When we backup we call mongodump
- When we restore we call mongorestore
- We keep on capturing all the objects in the namespace
- The backup is « app consistent » but is not incremental …



kasten by Veeam

Use fsynclock and capture a snapshot

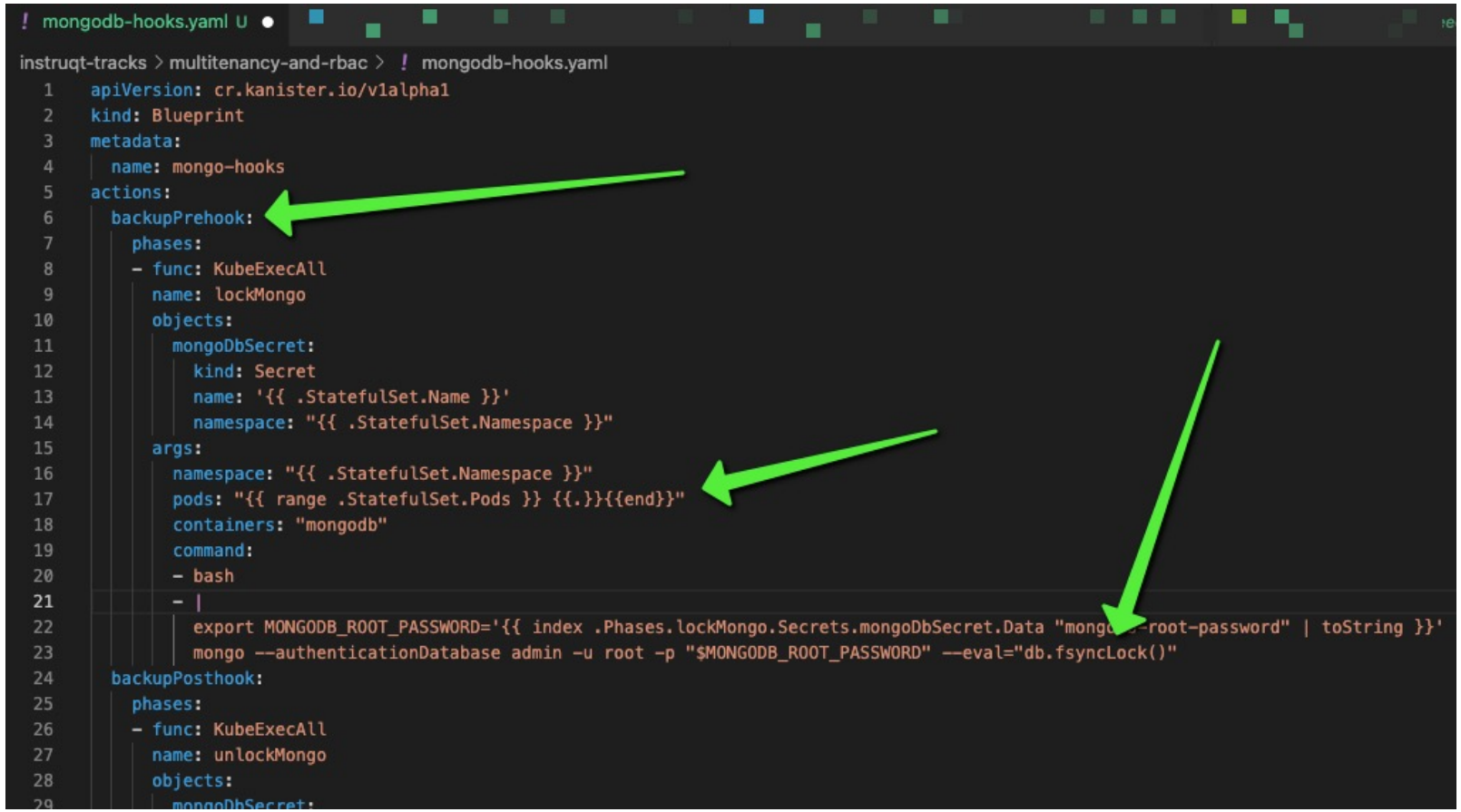
- Before capturing the snapshot we call fsyncLock
- After capturing we call fsyncUnlock
- Now backup are « app consistent » and incremental !!



kasten by Veeam

How does it work ?

We annotate the workload with a Kanister blueprint name
Kasten detects the annotation and applies the blueprint action



kasten by Veeam

About blueprints and database operators ...

- Blueprints are an extension mechanism
  - Gets you that remaining functionality to make it a perfect backup!
- Can be shared across all the databases you deployed a particular way
  - i.e. With the same operator or with the same helm chart
- Blueprint can also deal with **database operators**
  - By calling « backup api » exposed by the operator
- Let’s see another example with the well known « Etcd Operator »

kasten by Veeam

Blueprint can leverage database operator for backup

Example : Etcd operator
Backup API : EtcdBackup
Triggered by annotating the Etcd CR

backup.yaml

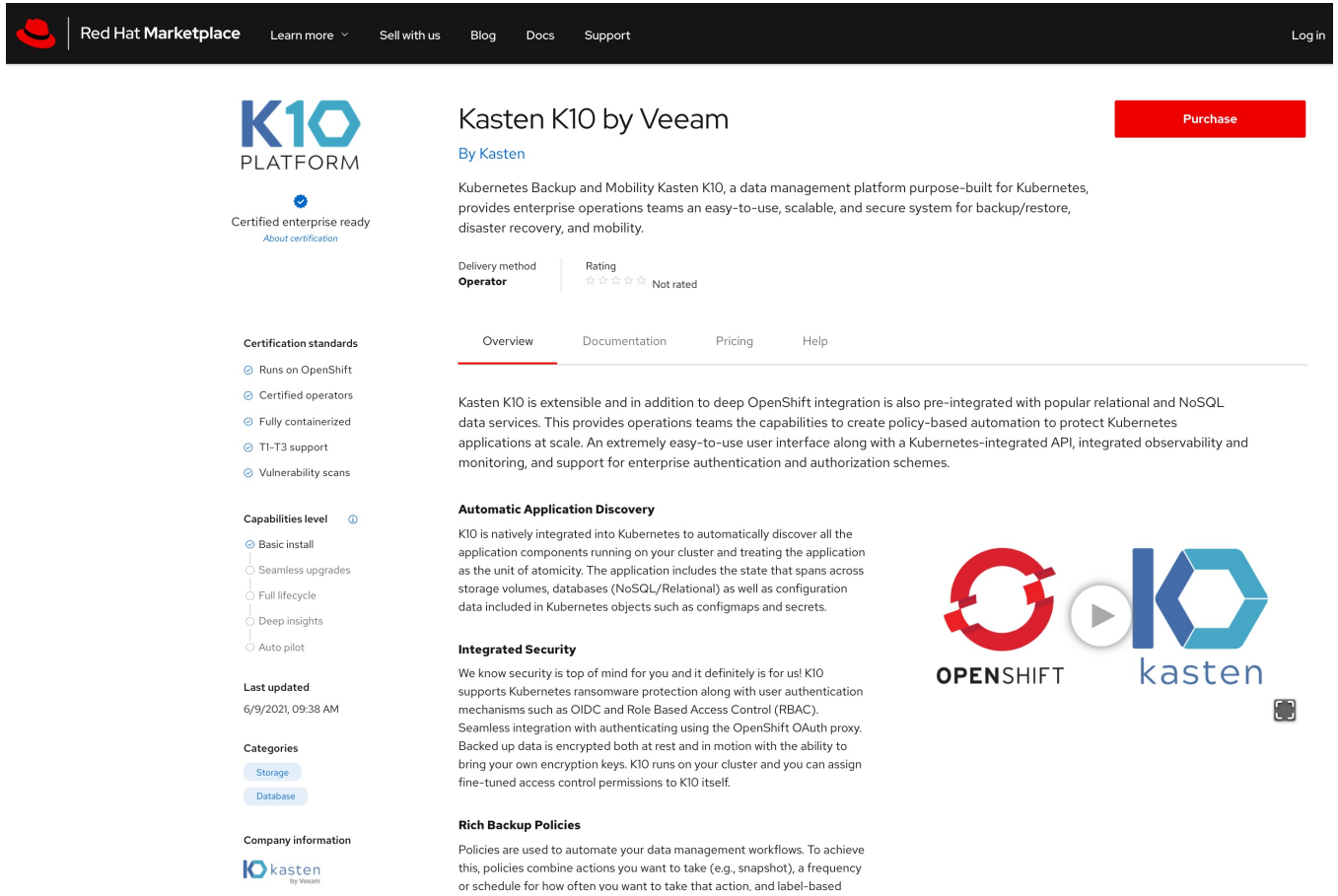
```yaml
apiVersion: "etcd.database.coreos.com/v1beta2"
kind: "EtcdBackup"
metadata:
  name: example-etcd-cluster-backup
spec:
  etcdEndpoints:
  - "http://example-client:2379"
  storageType: S3
  s3:
    path: "michael-bucket/demo-cluster/etcd-example/backup-test"
    awsSecret: aws
```

etcd-bp.yaml

```yaml
apiVersion: cr.kanister.io/v1alpha1
kind: Blueprint
metadata:
  name: etcd-bp
  namespace: kasten-io
actions:
  backup:
    outputArtifacts:
      cloudObject:
        keyValue:
          backupLocation: "{{ .Phases.takeEtcdBackup.Output.backupLocation }}"
    phases:
    - func: KubeTask
      name: takeEtcdBackup
      args:
        image: ghcr.io/kanisterio/kanister-kubectl:1.18
        namespace: '{{ index .Object.metadata "namespace" | toString }}'
        command:
          - sh
          - -o
          - errexit
          - -o
          - pipefail
          - -c
          - |
            BUCKET="{{ index .Object.metadata.annotations "bucket" | toString }}"
            ETCD_CLUSTER_NAME="{{ index .Object.metadata "name" | toString }}"
            BACKUP_TIME="{{ toDate "2006-01-02T15:04:05.999999999Z07:00" .Time | date "2006-01-
            NAMESPACE="{{ index .Object.metadata "namespace" | toString }}"
            BACKUP_LOCATION=${BUCKET}/${NAMESPACE}/${ETCD_CLUSTER_NAME}/${BACKUP_TIME}/backup
            BACKUP_NAME=${ETCD_CLUSTER_NAME}-${BACKUP_TIME}
            cat <<EOF | kubectl create -f -
            apiVersion: "etcd.database.coreos.com/v1beta2"
            kind: "EtcdBackup"
            metadata:
              name: ${BACKUP_NAME}
              namespace: ${NAMESPACE}
            spec:
              etcdEndpoints:
              - "http://${ETCD_CLUSTER_NAME}:2379"
              storageType: S3
              s3:
                path: "${BACKUP_LOCATION}"
                awsSecret: aws
            EOF
            kando output backupLocation ${BACKUP_LOCATION}
  restore:
```


kasten by Veeam

Kasten K10 – Get started today!



kasten.io/try-kasten-k10

https://marketplace.redhat.com/en-us/products/kasten-k10

kasten by Veeam