DATAVISION
by soda

Kanister: Application-Level Data Operations on Kubernetes

Michael Cade
Global Field CTO
Veeam Software

Michael Cade

Michael Cade is a seasoned technology executive with over 20 years of experience in the data protection space. Currently serving as the Global Field CTO for Veeam Software.

vZilla

90DaysOfDevOps

@MichaelCade1

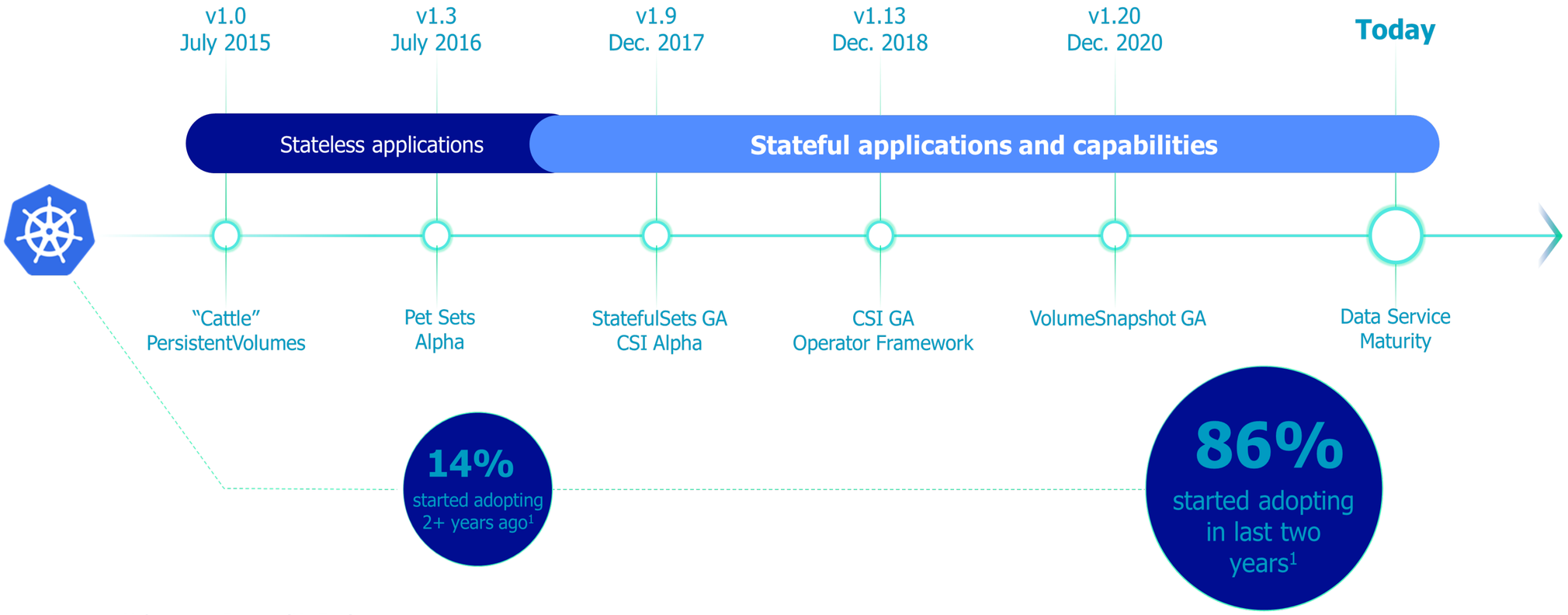
YouTube

LinkedIn

Michael Cade
Field CTO, Cloud Native
Veeam Software

DATAVISION by soda

From Stateless to Stateful

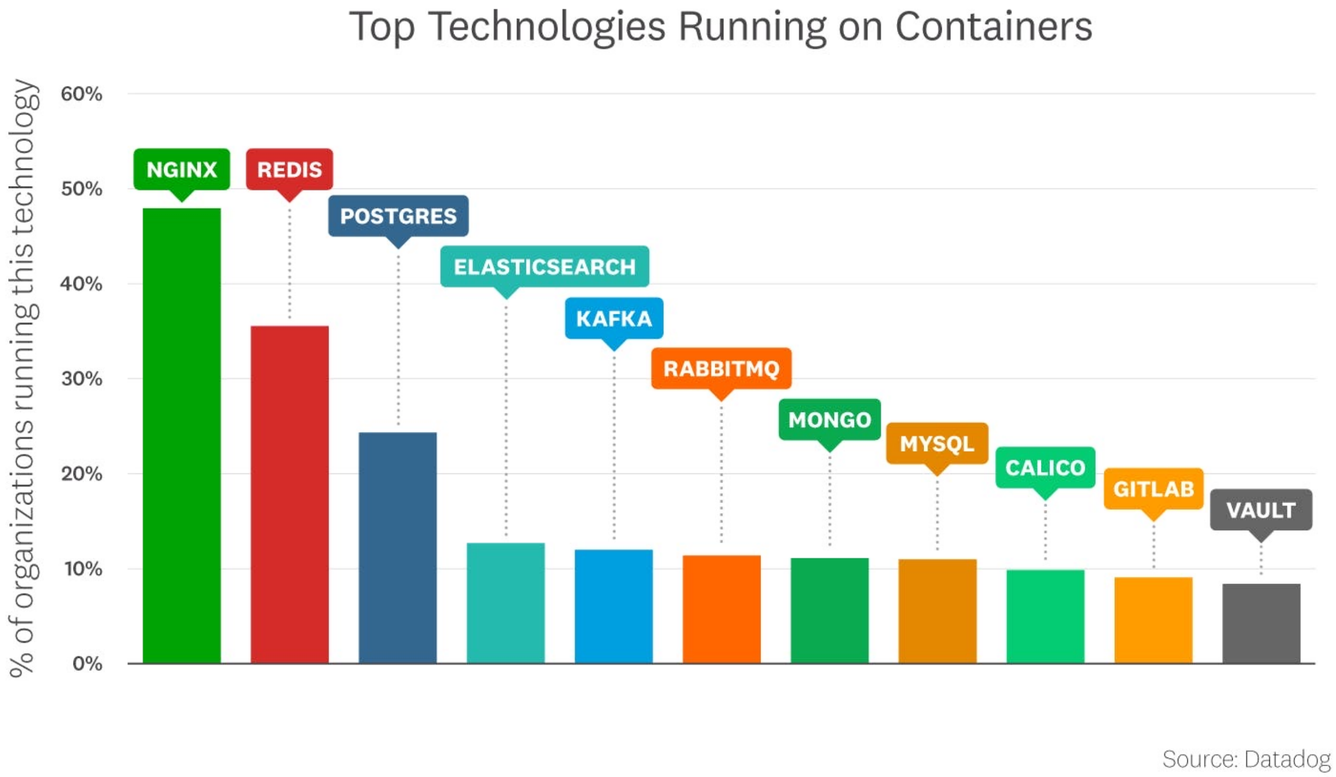


[1] 2021 Data on Kubernetes Report (N=502)

DATAVISION by soda

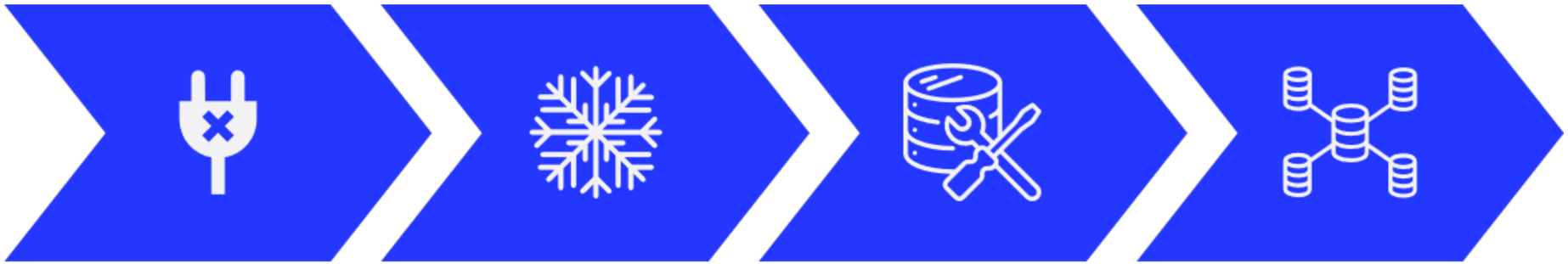
As Kubernetes has matured

Stateful Applications are common



Datadog Real-World Container Use Report – Nov 2022

Application Consistency Spectrum



Crash Consistent

- Storage snapshots

“App” Consistent

- Freeze data service
- Storage snapshot
- Unfreeze data service

Logical Backup

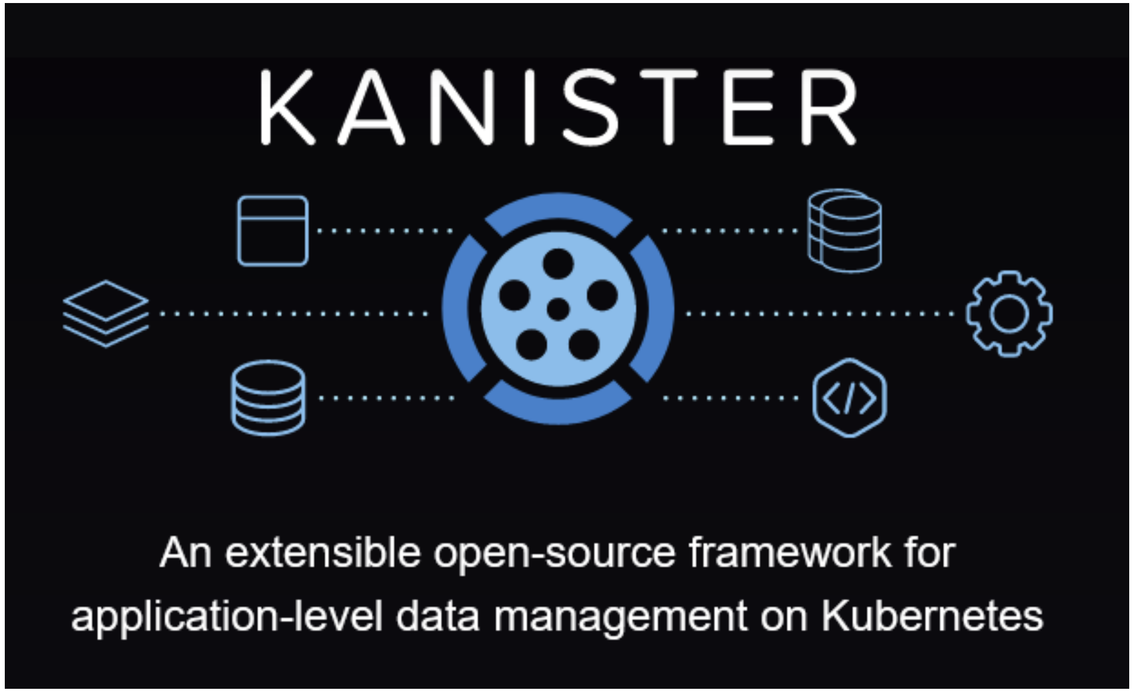
- Logical dumps via data service-specific tool (e.g., `elasticdump`)

System Consistent

- Full app capture
- Combination of tools across data and storage layers

PostgreSQL MySQL mongoDB elastic cassandra FoundationDB redis Couchbase

Introducing Kanister



KANISTER allows domain experts to capture application-specific data management tasks in blueprints which can be easily shared and extended.

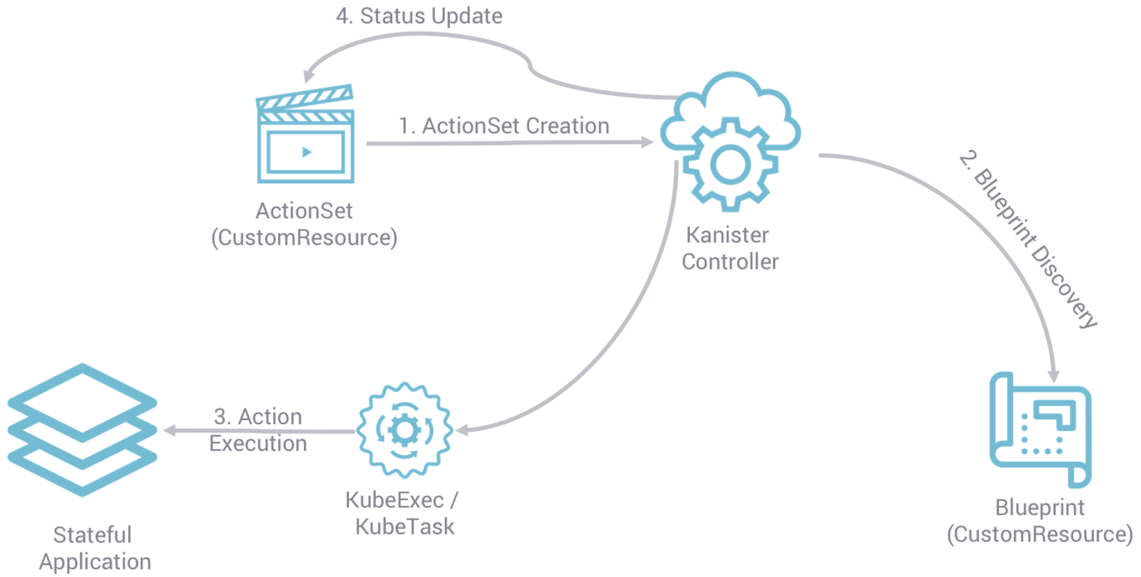
DATAVISION by soda

How Kanister.io Orchestrates Data Protection

A cloud native, extensible framework for app consistent operations

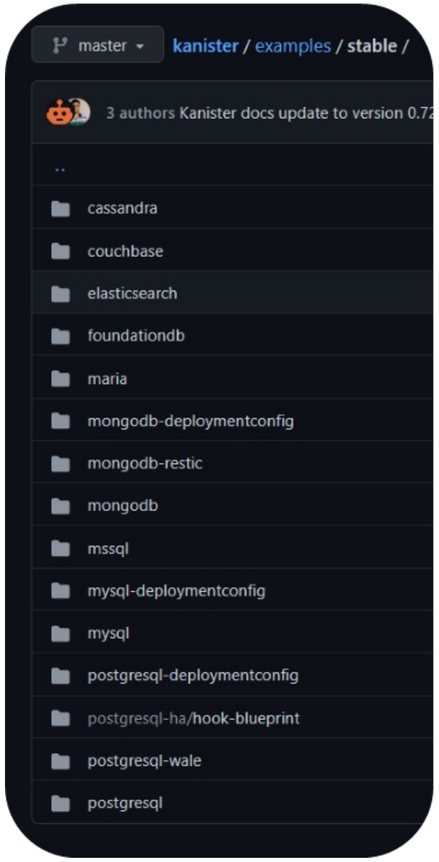
A Kanister controller, installed by Helm chart,
provides new Custom Resource Definition

- ActionSet: an instance of a Blueprint operation
- Blueprint: orchestrated actions for backup and recovery
- Profile: infrastructure endpoint config

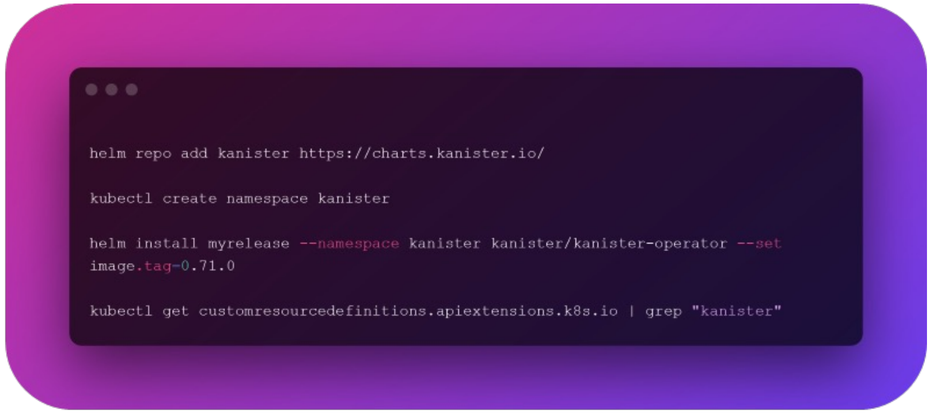


Execution Walkthrough

Kanister Deployment & Blueprints



Controller



```
helm repo add kanister https://charts.kanister.io/

kubectl create namespace kanister

helm install myrelease --namespace kanister kanister/kanister-operator --set
image.tag=0.71.0

kubectl get customresourcedefinitions.apiextensions.k8s.io | grep "kanister"
```

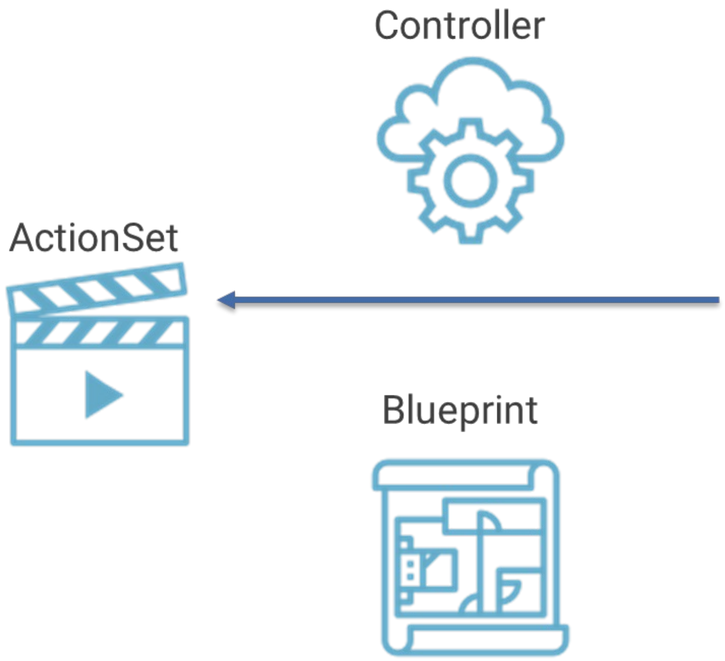
Blueprint

Database Workload

DATAVISION by soda

Execution Walkthrough

ActionSets

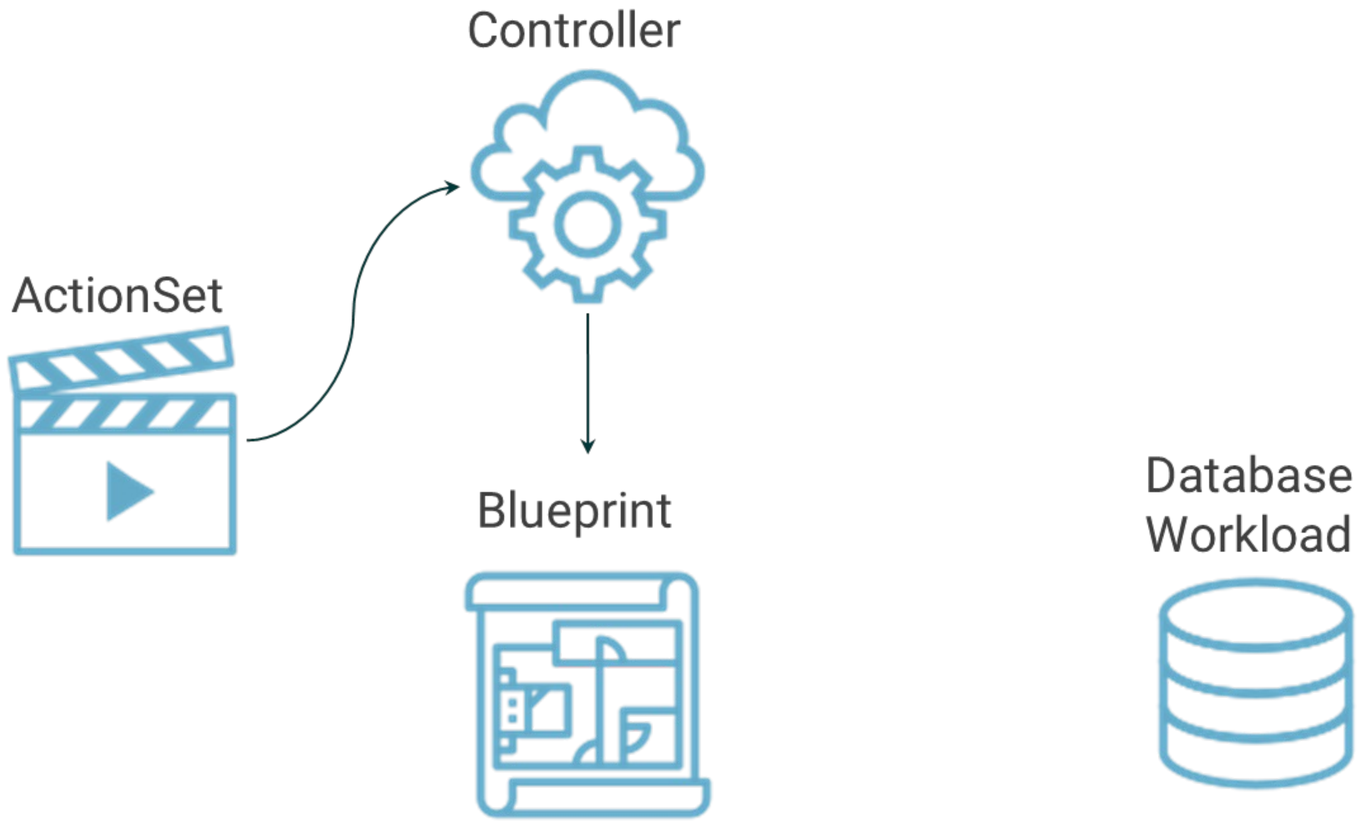


create an actionset on the project

```yaml
apiVersion: cr.kanister.io/v1alpha1
kind: ActionSet
metadata:
  name: $backup_name
  namespace: kanister
spec:
  actions:
  - blueprint: mysql-blueprint
    name: backup
    object:
      kind: statefulset
      name: mysql
      namespace: $NAMESPACE
    profile:
      name: $PROFILE
      namespace: kanister
    secrets:
      mysql:
        name: mysql
        namespace: $NAMESPACE
```

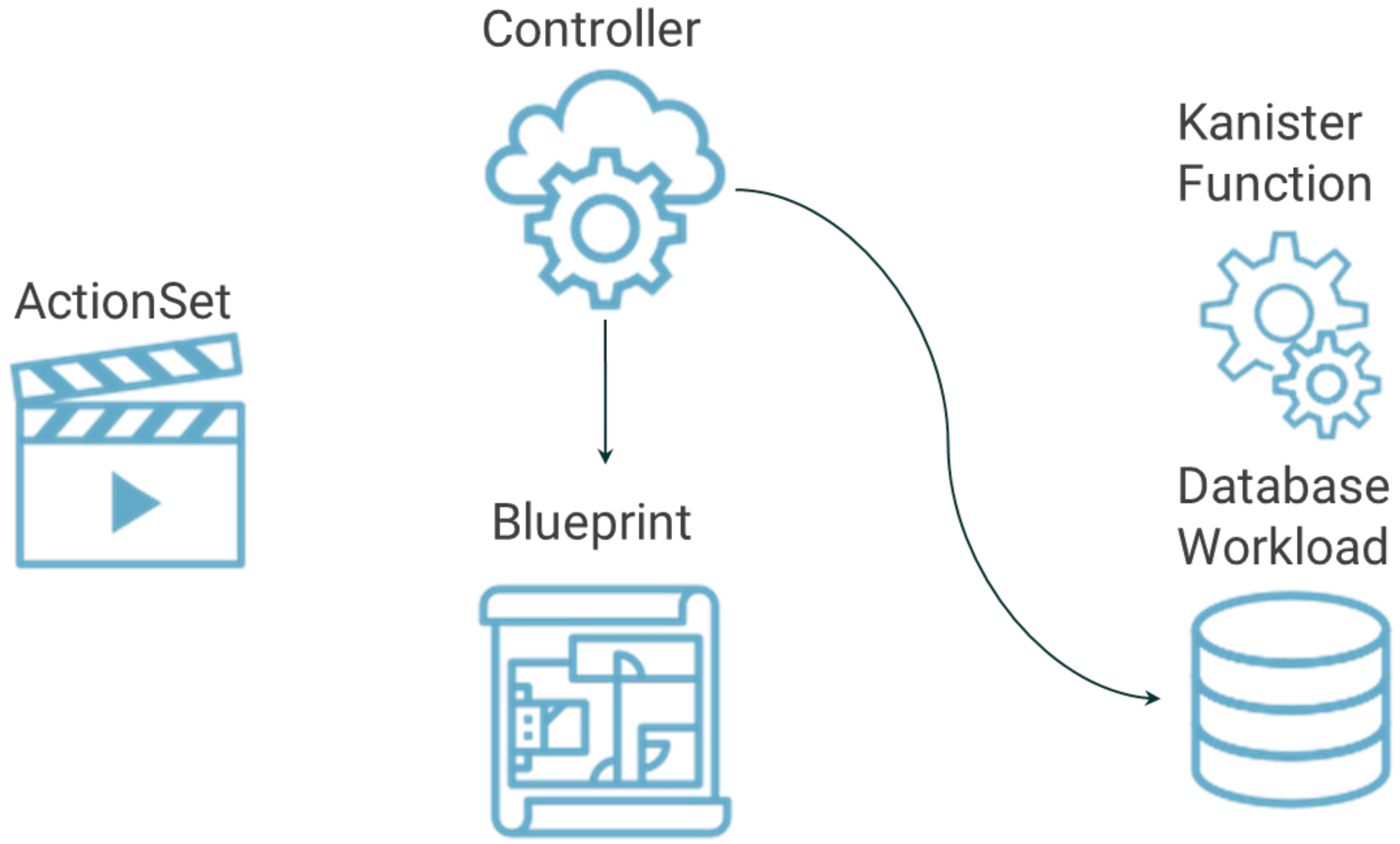
Execution Walkthrough

ActionSet is started, which then calls the Blueprint



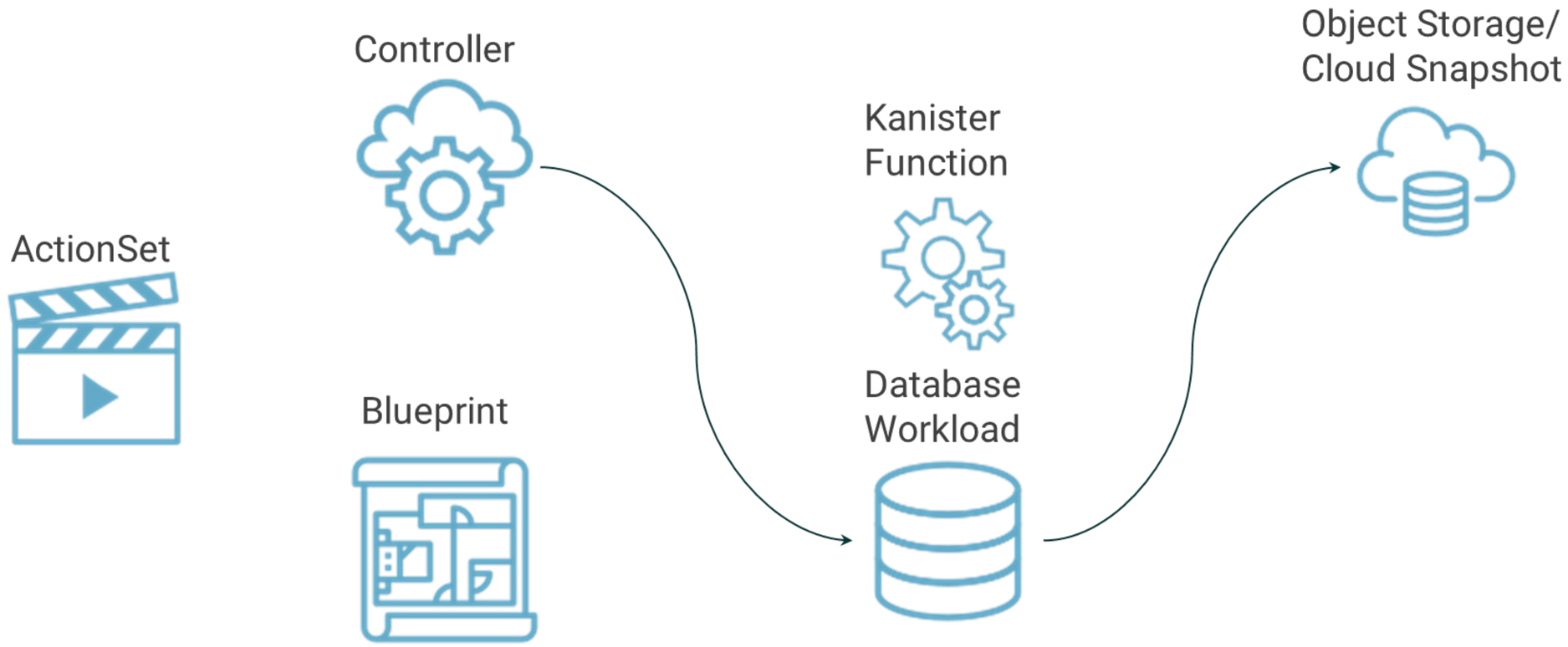
Execution Walkthrough

Controller then calls Kanister function to interact with Database workload



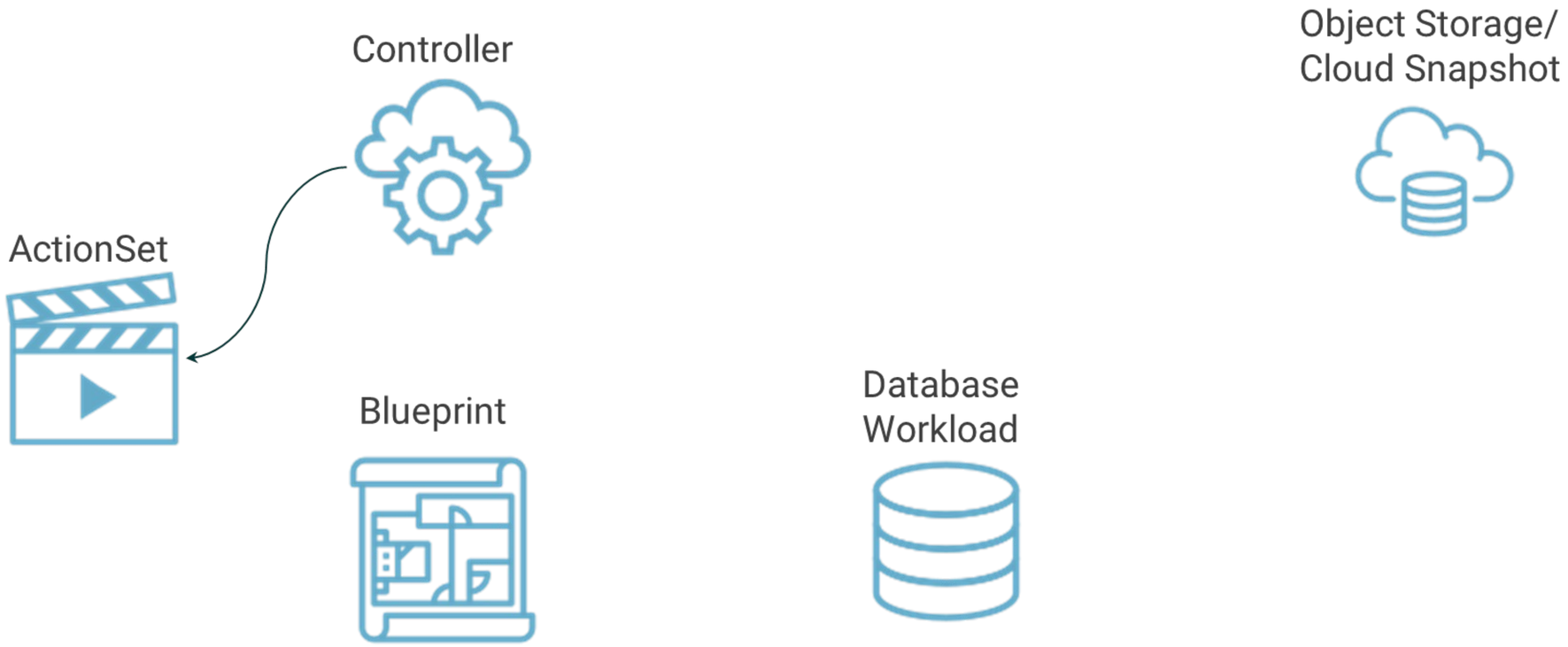
Execution Walkthrough

When consistency is reached a copy is then sent to backup location



Execution Walkthrough

Report back status to ActionSet



DATAVISION
by soda

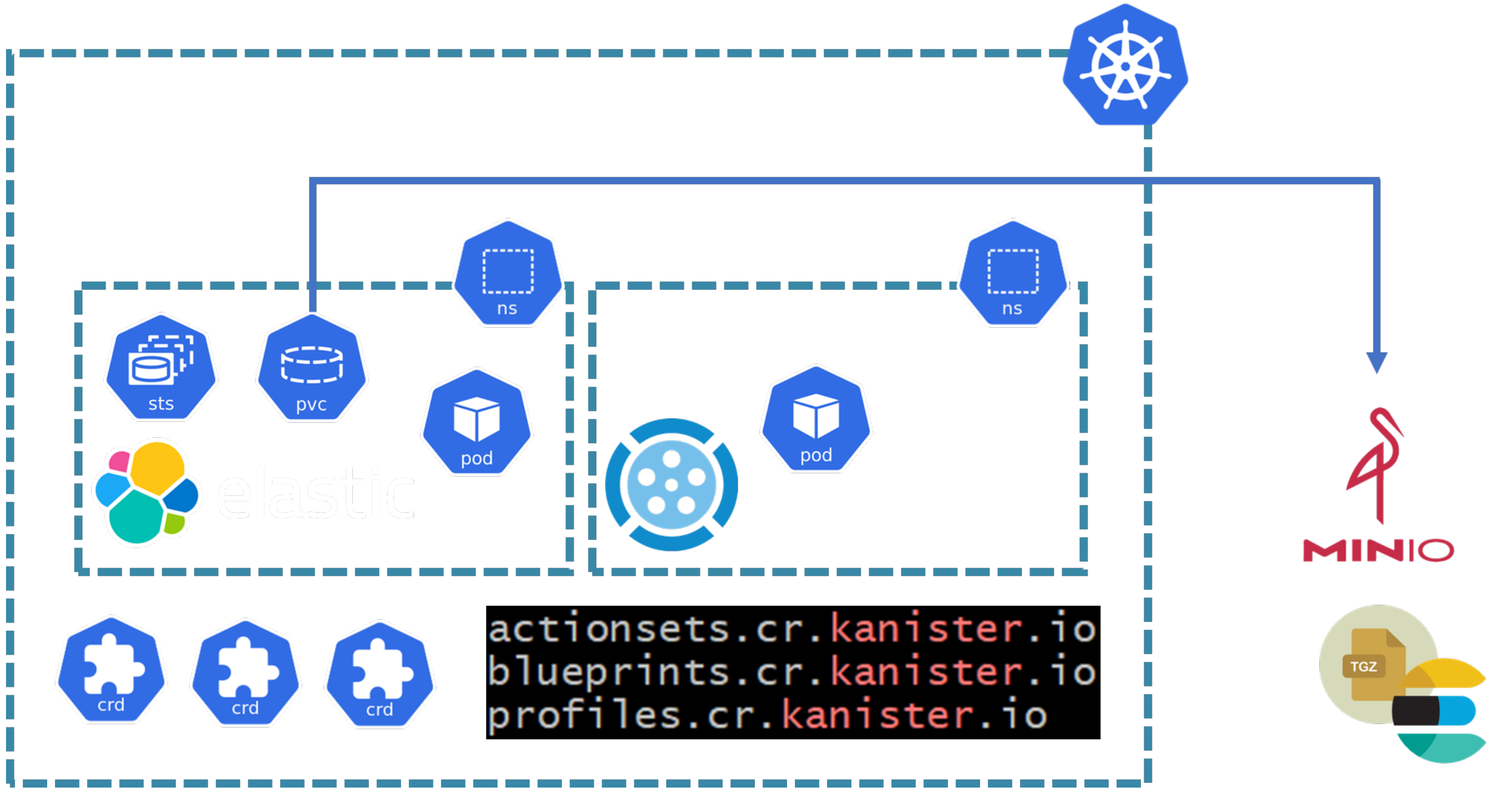
Demo
ElasticSearch with Kanister

KANISTER

DATAVISION by soda

Demo Architecture

Show you what I am about to show you



Kahu – What could happen?

Kahu: Container Data Protection

Kahu is a cloud native tool to backup and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes.

KAHU
Data Protection

KANISTER

- Snapshot & Backup of your Kubernetes resources
- Scheduling & Orchestration
- Application-level management
- Blueprint mechanism for custom applications

DATAVISION by soda

Resources

Kanister.io is a cloud-native, open source, extensible framework for Kubernetes data protection!

Please adopt and join us at:

Community:
biweekly Zoom meetings and Slack

Tutorial on https://KubeCampus.io

Example blueprints:
https://github.com/kanisterio/kanister/tree/master/examples

https://Kanister.io

Getting Started with Kanister

Operators & Developers welcome!

Kanister is available as an open-source project today!

- github.com/kanisterio/kanister
- @kanisterio
- #kanisterio
- tiny.cc/kanisterio

```
$ git clone git@github.com:kanisterio/kanister.git
# install Kanister operator controller
$ kubectl apply -f bundle.yaml
# install your application
$ kubectl apply -f examples/mongo-sidecar/mongo-cluster.yaml
# use an existing blueprint, tweak one, or create one yourself
$ kubectl apply -f examples/mongo-sidecar/mongo-blueprint.yaml
# perform operations (requires setting secrets and configmap)
$ kubectl create -f examples/mongo-sidecar/backup-actionset.yaml
```

Enterprise Orchestrated Data Management

Free up to **5 Nodes**
Fully Featured

aws
OPENSHIFT
HELM

*Subscription terms vary

KASTEN by Veeam

THANK YOU!