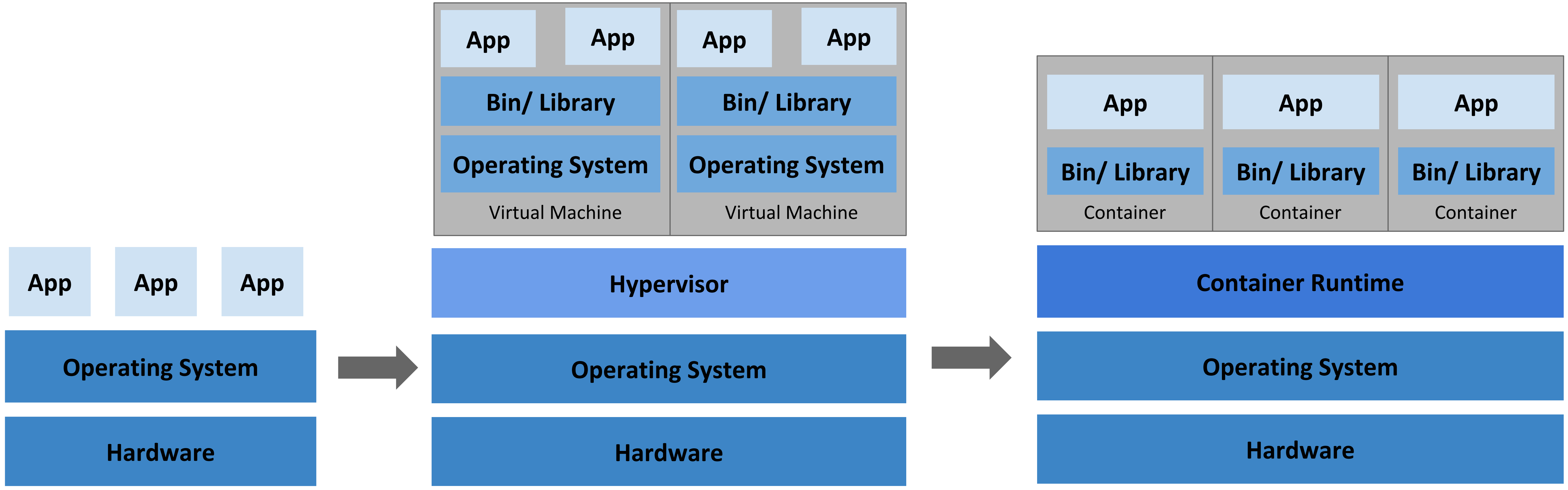
# K8s and Go

*Managing Your Cluster with Operators*

# Kubernetes



App App App

Operating System

Hardware

App App App App

Bin/ Library Bin/ Library

Operating System Operating System

Virtual Machine Virtual Machine

Hypervisor

Operating System

Hardware

App App App

Bin/ Library Bin/ Library Bin/ Library

Container Container Container

Container Runtime

Operating System

Hardware

**Traditional Deployment** **Virtualized Deployment** **Container Deployment**

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

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery.

https://kubernetes.io/

# K8s Apps Management

1. Stateless applications
2. Stateful applications

# Deployments

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app-container
        image: my-app:1.0
```

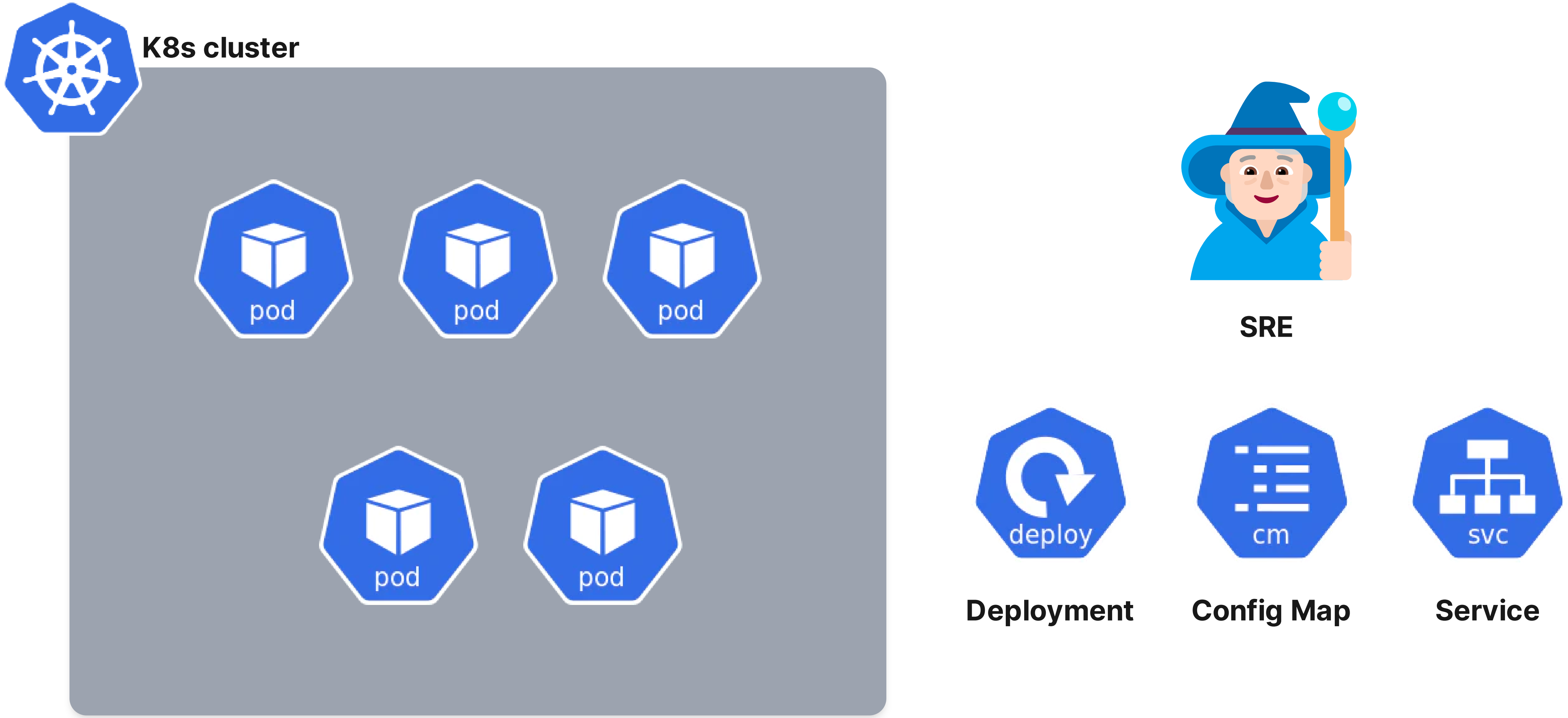
# Config Maps

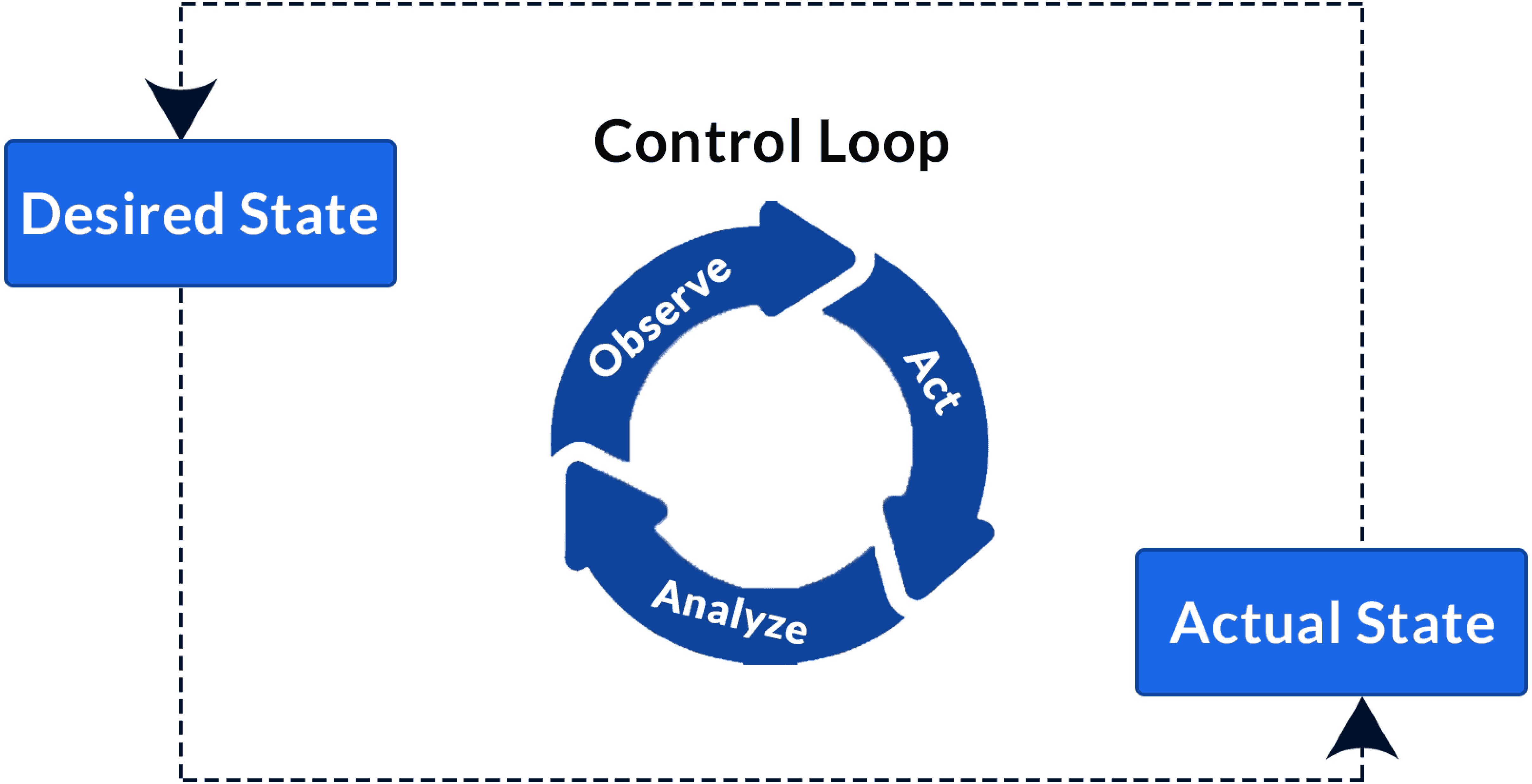
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-config
data:
  DATABASE_URL: "mysql://db-server:3306/mydb"
  API_KEY: "my-api-key"
```

# Services

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

# Stateless application





Control Loop

Desired State

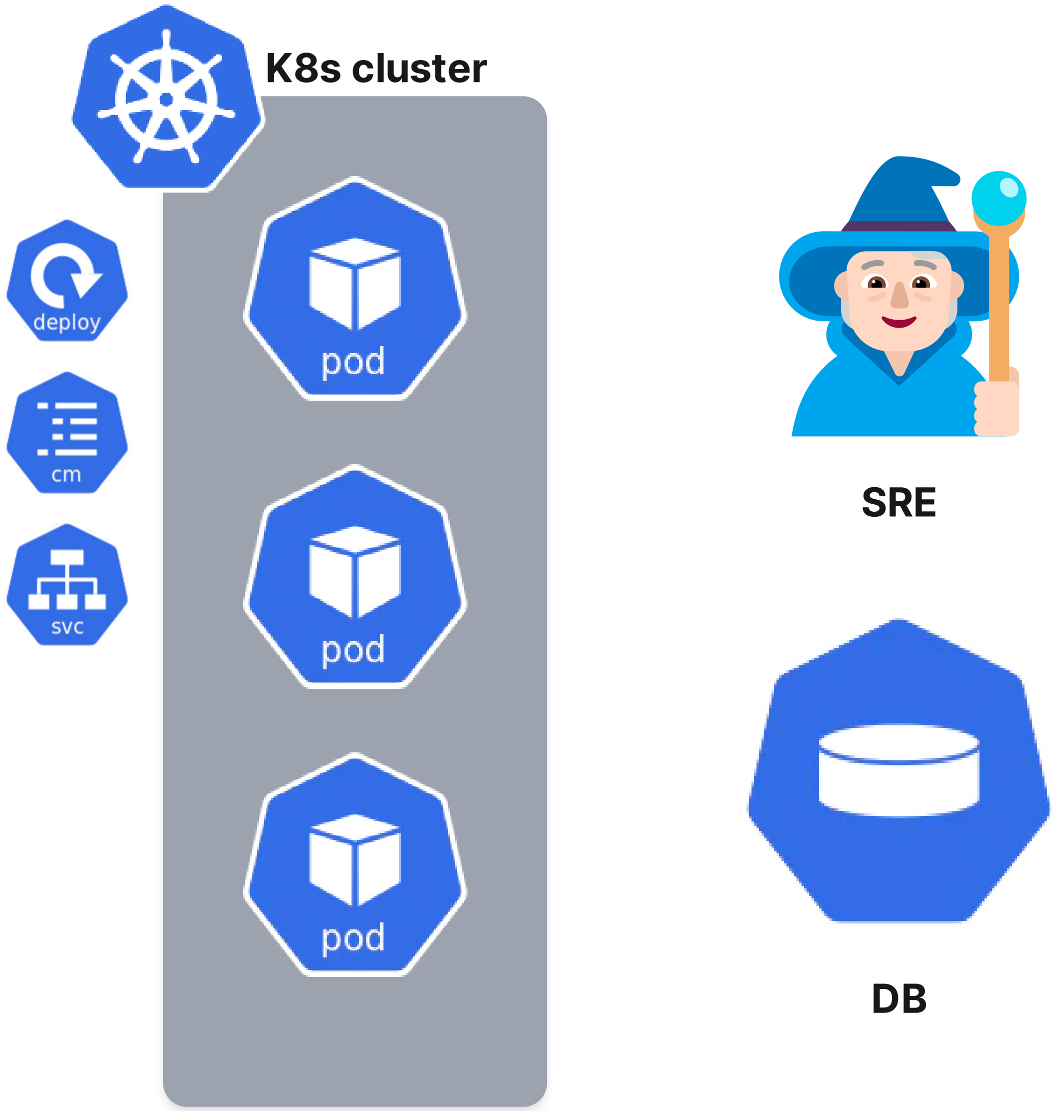
Observe

Act

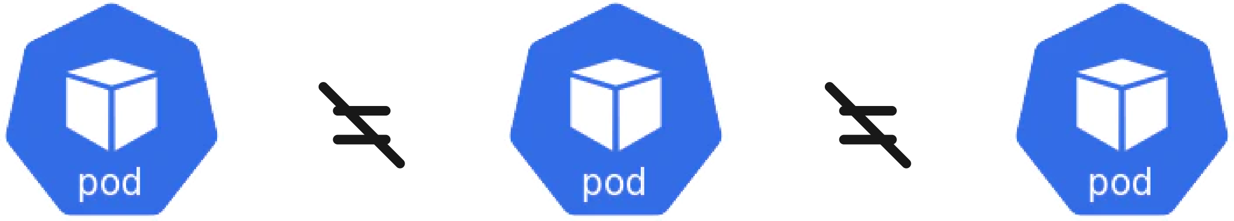
Analyze

Actual State

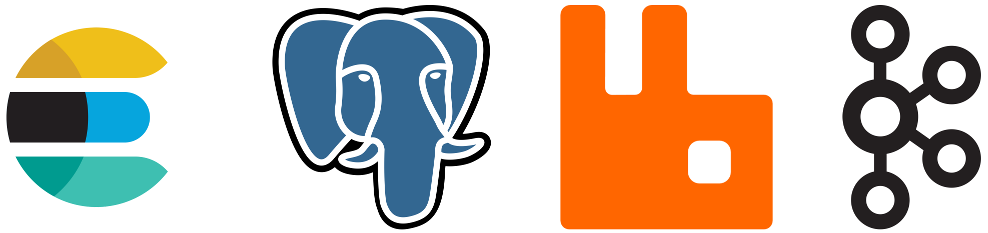
# Stateful application



# Stateful application



PostgreSQL replica 1 PostgreSQL replica 2 PostgreSQL replica 3



# Human Operator

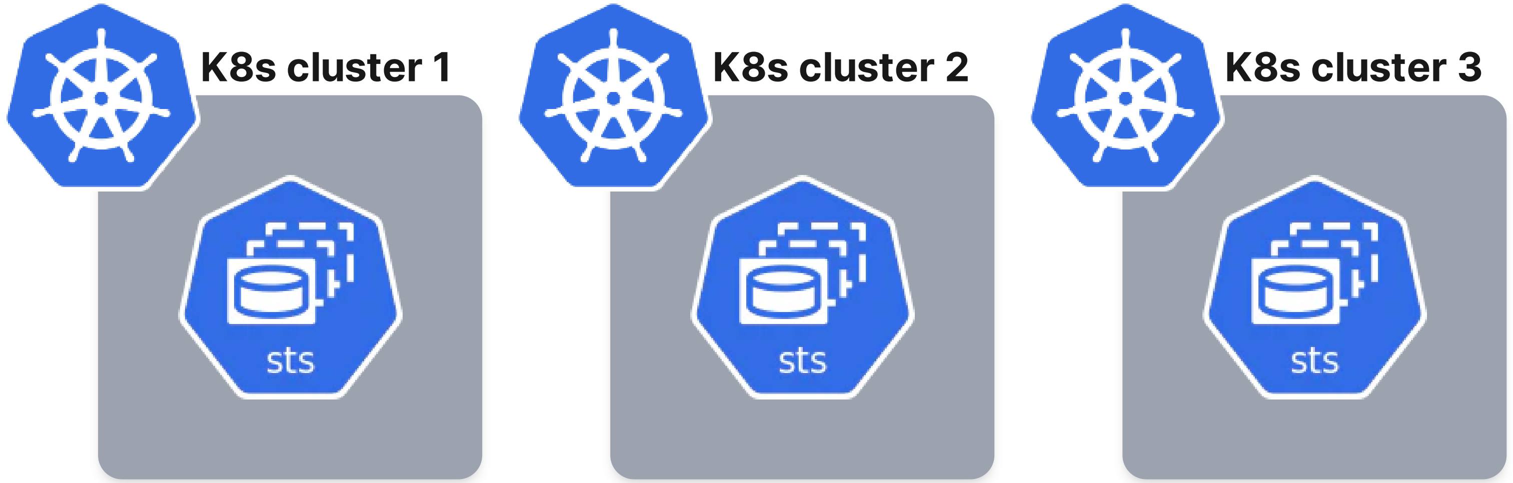
🧙 📜

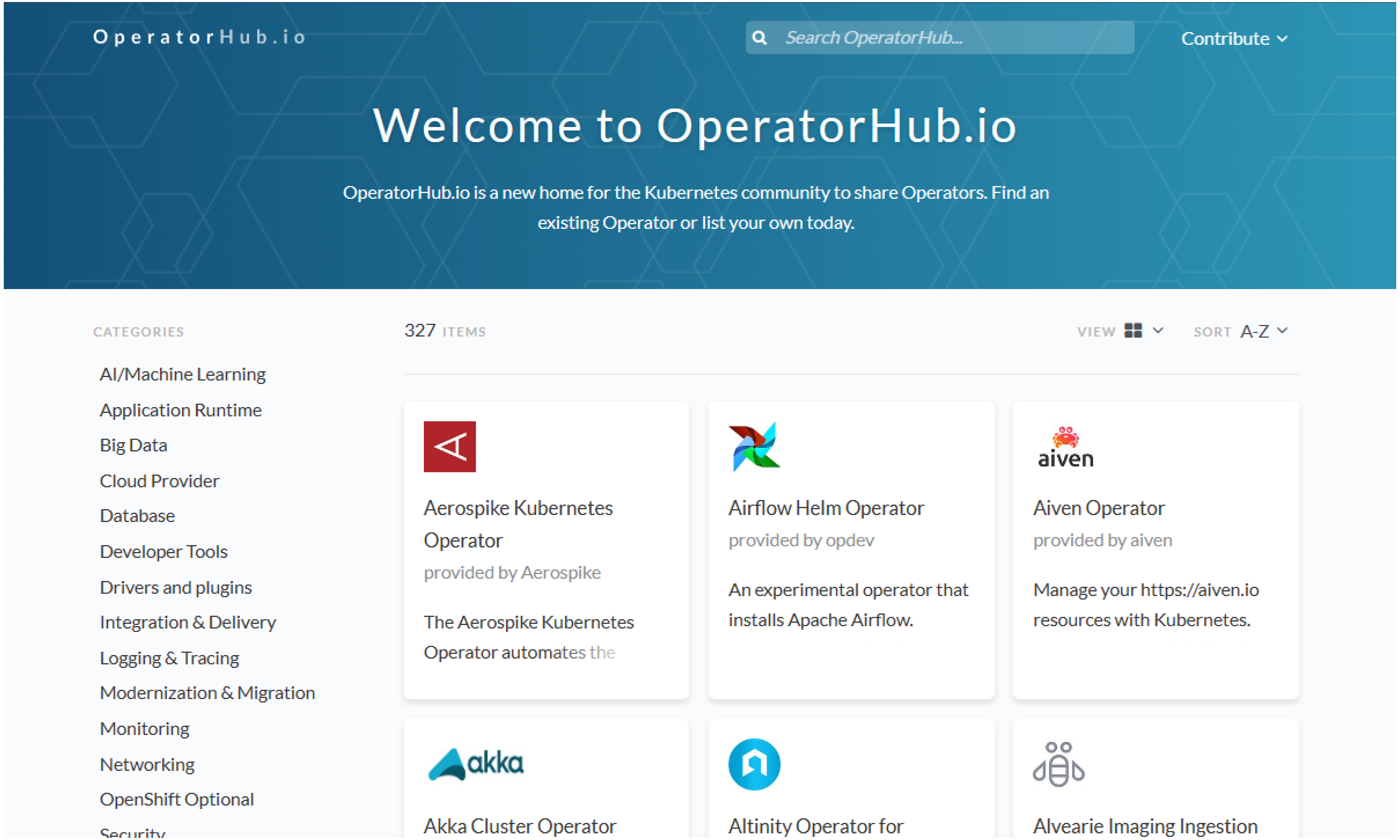
```
wise SRE'gician appears with his Elder Scripts
```

# K8s Operator

🤖

# Multiple clusters





OperatorHub.io

Search OperatorHub...

Contribute

# Welcome to OperatorHub.io

OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.

CATEGORIES

- AI/Machine Learning
- Application Runtime
- Big Data
- Cloud Provider
- Database
- Developer Tools
- Drivers and plugins
- Integration & Delivery
- Logging & Tracing
- Modernization & Migration
- Monitoring
- Networking
- OpenShift Optional
- Security

327 ITEMS

VIEW

SORT A-Z

Aerospike Kubernetes Operator
provided by Aerospike
The Aerospike Kubernetes Operator automates the

Airflow Helm Operator
provided by opdev
An experimental operator that installs Apache Airflow.

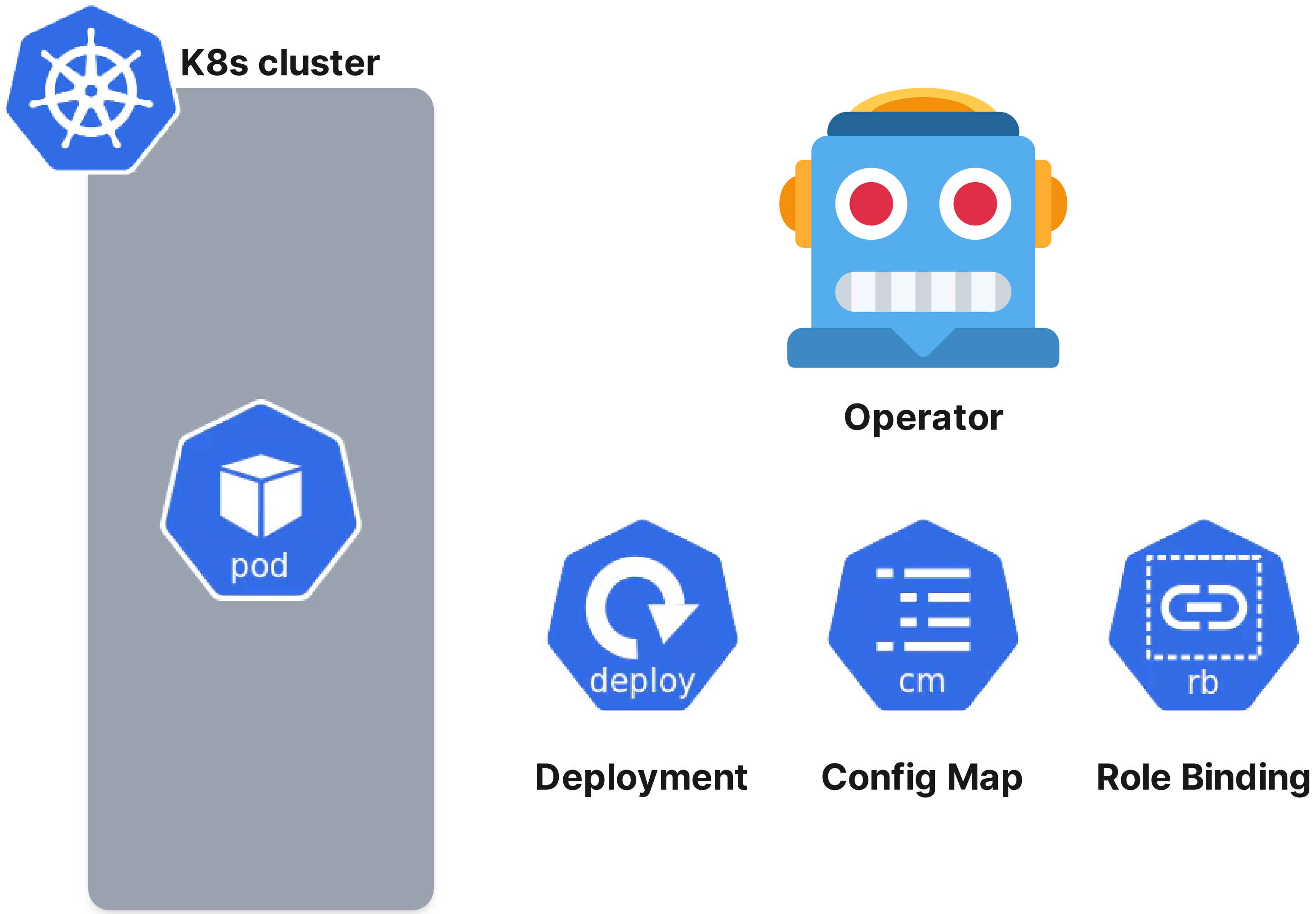
aiven
Aiven Operator
provided by aiven
Manage your https://aiven.io resources with Kubernetes.

akka
Akka Cluster Operator

Altinity Operator for

Alvearie Imaging Ingestion

# How does K8s Operator work?



K8s cluster

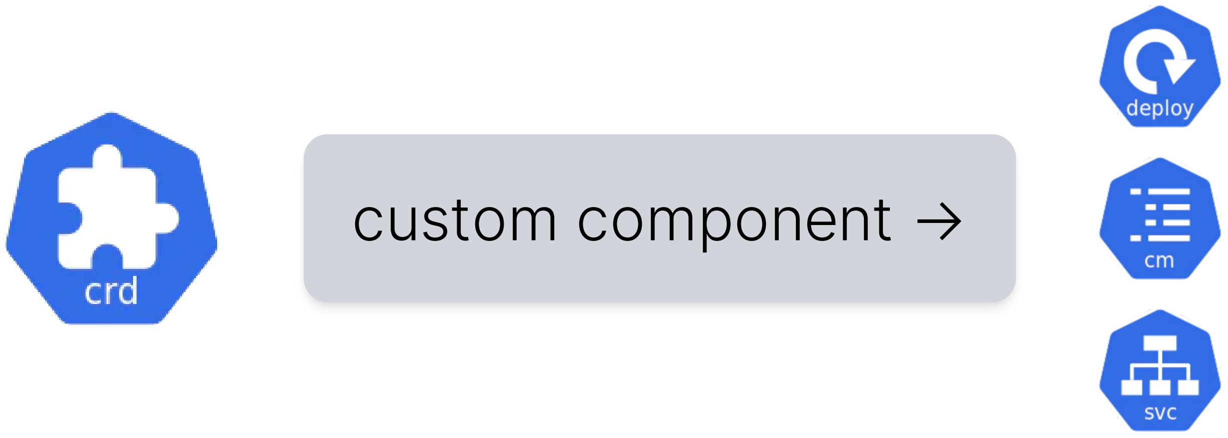
Operator

Deployment

Config Map

Role Binding

# Custom Resource Definitions



custom component →

# Custom Resource Definition (CRD)

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: preserves.wildlife.com
spec:
  group: wildlife.com
  names:
    kind: WildlifePreserve
    plural: preserves
    singular: preserve
  scope: Namespaced
  version: v1
  subresources:
    status: {}
    . . .
```

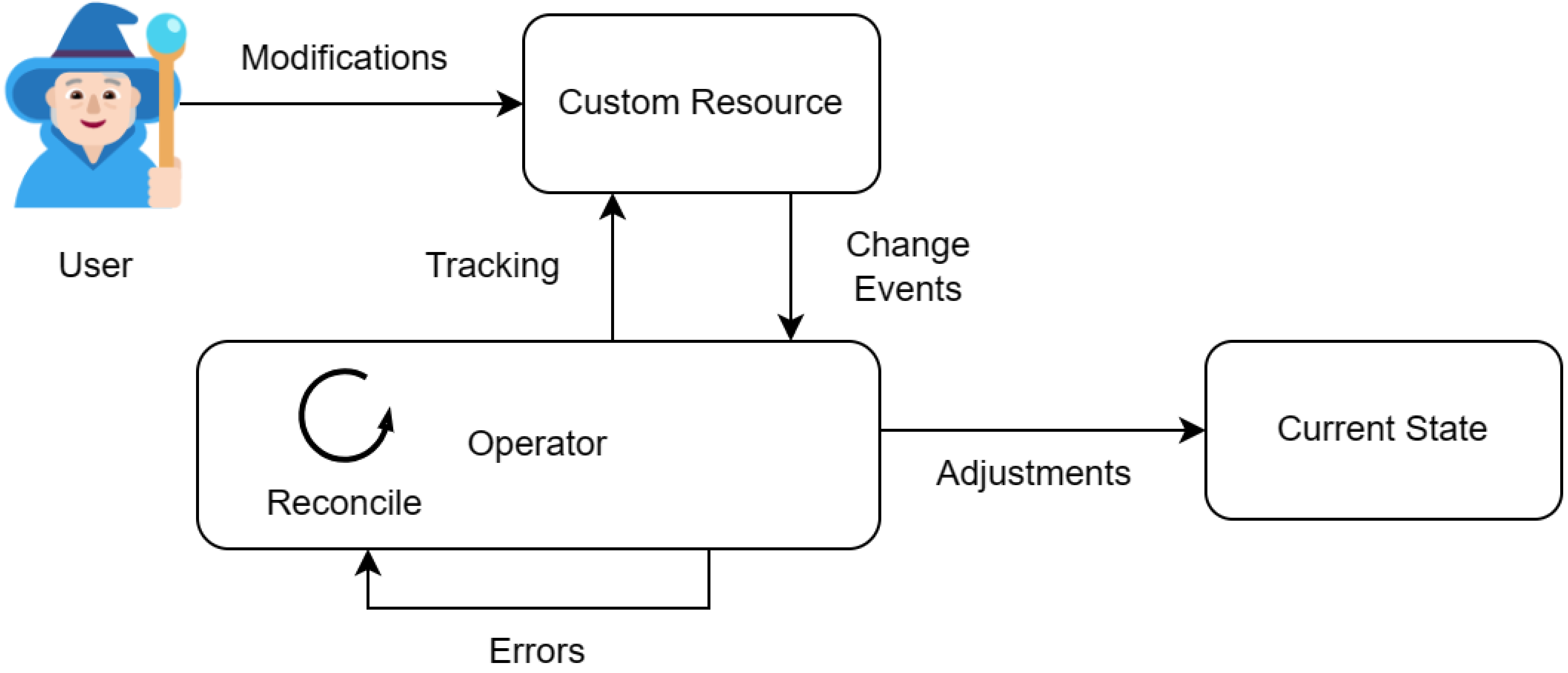
# Custom Resource Definition (CRD)

```yaml
. . .
validation:
  openAPIV3Schema:
    type: object
    properties:
      spec:
        type: object
        properties:
          name:
            type: string
          location:
            type: string
          capacity:
            type: integer
            minimum: 1
            maximum: 1000
```

# Custom Resource (CR)

```yaml
apiVersion: wildlife.com/v1
kind: WildlifePreserve
metadata:
  name: bialowieza-forest-preserve
spec:
  name: Białowieża Forest Preserve
  location: Białowieża Forest
  capacity: 500
```

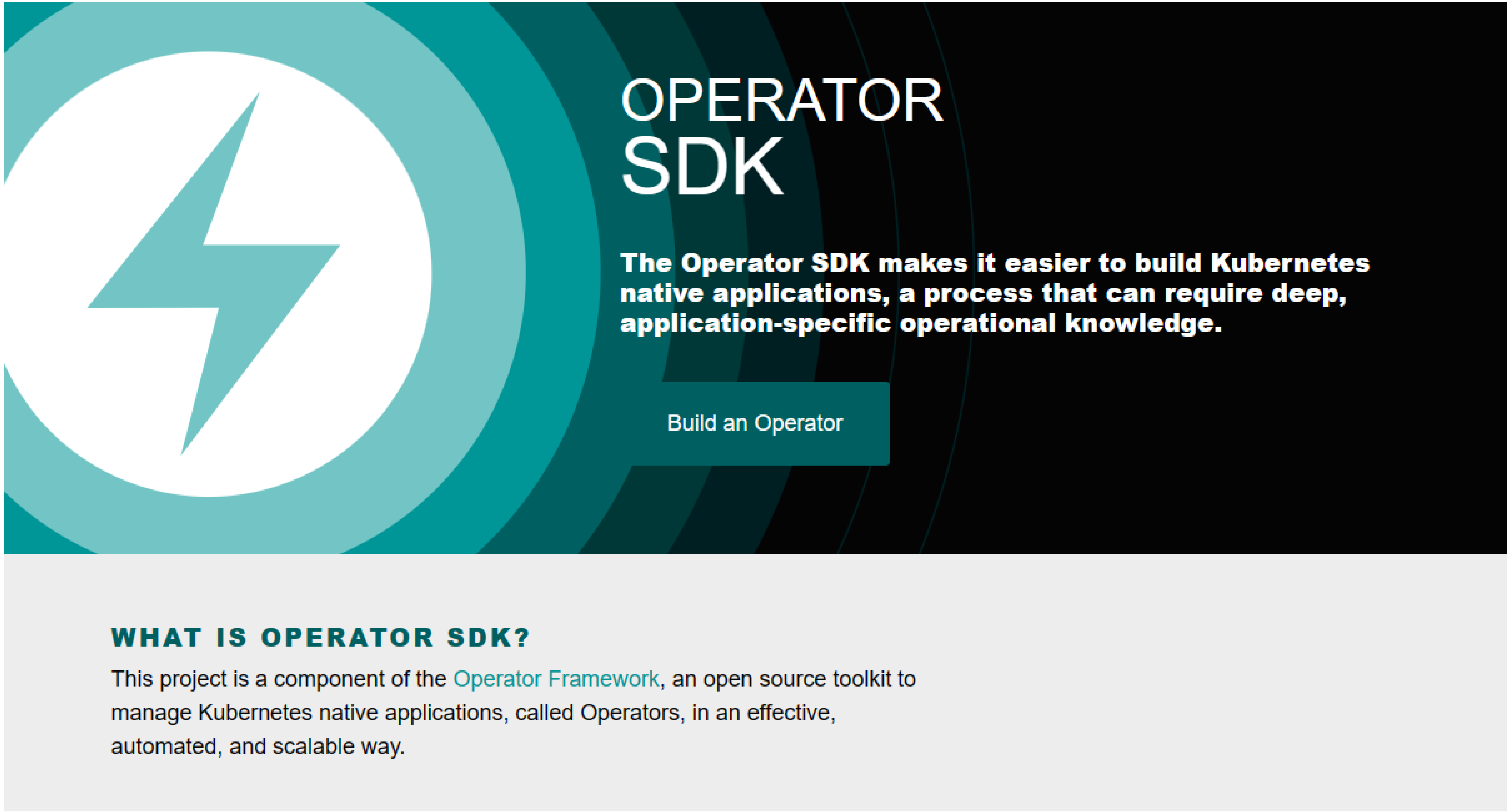
# Custom Controllers


Modifications
Custom Resource
User
Tracking
Change Events
Reconcile
Operator
Current State
Adjustments
Errors

# Kubernetes Operator technologies

```
Operators are usually implemented in one of three main technologies:
1. Golang
2. Ansible
3. Helm
```

# The Operator SDK



The Operator SDK makes it easier to build Kubernetes native applications, a process that can require deep, application-specific operational knowledge.

WHAT IS OPERATOR SDK?

This project is a component of the Operator Framework, an open source toolkit to manage Kubernetes native applications, called Operators, in an effective, automated, and scalable way.

# Creating a Custom Operator

# Creating a Custom Operator

## 1. Initialize a New Operator Project

```
operator-sdk init wildlife-preserve-operator \
--repo=github.com/jakubpieta/wildlife-preserve-operator
```

## 2. Create a Custom Resource Definition (CRD)

```
operator-sdk create api
--group wildlife.preserves --version v1alpha1 \
--kind WildlifePreserve --resource=true --controller=true
```

# Creating a Custom Operator

```go
// api/v1alpha1/wildlifepreserve_types.go

// WildlifePreserveSpec defines the desired state of WildlifePreserve
type WildlifePreserveSpec struct {
  Name            string `json:"name"`
  Location        string `json:"location"`
  Replicas        int32  `json:"replicas"`
  VolumeMountPath string `json:"volumeMountPath"`
}
```

# Creating a Custom Operator

## 3. Implement the controller

```go
// controllers/wildlifepreserve_controller.go

// Reconcile handles WildlifePreserve resource reconciliation
func (r *WildlifePreserveReconciler) Reconcile(
  ctx context.Context, req ctrl.Request) (ctrl.Result, error) {
    // Fetch the WildlifePreserve resource
    preserve := &wildlifev1alpha1.WildlifePreserve{}
    if err := r.Get(ctx, req.NamespacedName, preserve); err != nil {
        return ctrl.Result{}, client.IgnoreNotFound(err)
    }

    // Handle pod, volume and volume claim creation and deletion
    return ctrl.Result{}, nil
}
```

# Creating a Custom Operator

## 4. Build, install CRDs and deploy the operator

```
make docker-build IMG=jakubpieta/wildlife-preserve-operator:v0.0.1
make docker-push IMG=jakubpieta/wildlife-preserve-operator:v0.0.1
make install
make deploy IMG=docker.io/jakubpieta/wildlife-preserve-operator:v0.0.1

$ kubectl get pods -n wildlife-preserve-operator-system
NAME                                                             READY   STATUS    RESTARTS   AGE
wildlife-preserve-operator-controller-manager-65c758445f-ss2k4   2/2     Running   0          32s
```

# Creating a Custom Operator

4. Build, install CRDs and deploy the operator

```
namespace/wildlife-preserve-operator-system created
customresourcedefinition.apiextensions.k8s.io/wildlifepreserves.wildlife.preserves.jakubpieta configured
serviceaccount/wildlife-preserve-operator-controller-manager created
role.rbac.authorization.k8s.io/wildlife-preserve-operator-leader-election-role created
clusterrole.rbac.authorization.k8s.io/wildlife-preserve-operator-manager-role created
clusterrole.rbac.authorization.k8s.io/wildlife-preserve-operator-metrics-reader created
clusterrole.rbac.authorization.k8s.io/wildlife-preserve-operator-proxy-role created
rolebinding.rbac.authorization.k8s.io/wildlife-preserve-operator-leader-election-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/wildlife-preserve-operator-manager-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/wildlife-preserve-operator-proxy-rolebinding created
service/wildlife-preserve-operator-controller-manager-metrics-service created
deployment.apps/wildlife-preserve-operator-controller-manager created
```

# Creating a Custom Operator

## 5. Create WildlifePreserve Custom Resources

```yaml
apiVersion: wildlife.preserves.jakubpieta/v1alpha1
kind: WildlifePreserve
metadata:
  name: bialowieza-wildlife-preserve
  namespace: wildlife
spec:
  name: "Bialowieza"
  volumeMountPath: "/animals-storage"
  replicas: 3
  location: "Poland"
```

```
$ kubectl apply -f wildlifepreserve.yaml
wildlifepreserve.wildlife.preserves.jakubpieta/bialowieza-wildlife-preserve created
```

# Creating a Custom Operator

6. Watch the Operator Logs and observe changes on the cluster

```
$ kubectl logs -n wildlife-preserve-operator-system wildlife-preserve-operator-controller-manager-78b6d9969-4fbjn

$ kubectl get pods -n wildlife
NAME                                                READY   STATUS    RESTARTS   AGE
bialowieza-wildlife-preserve-deployment-5c84b54646-7gmfc   1/1     Running   0          96s
bialowieza-wildlife-preserve-deployment-5c84b54646-d2n8f   1/1     Running   0          96s
bialowieza-wildlife-preserve-deployment-5c84b54646-q48lp   1/1     Running   0          96s
```

# Summary

# Q&A

🧙 ☸ 🤖

```
Questions?
```