Operadores de Kubernetes: el poder de la automatización

Edith Puclla
@Percona

Edith Puclla (Edi)

- Technology Evangelist at Percona
- Visa de Talento Global UK
- Embajadora de CNCF
  - KCD, Lima Peru
- Docker Captain
  - Meetups en Ayacucho
- Colaborador de código abierto:
  - Apache Airflow, Kubernetes Website

PERCONA

Edith Puclla

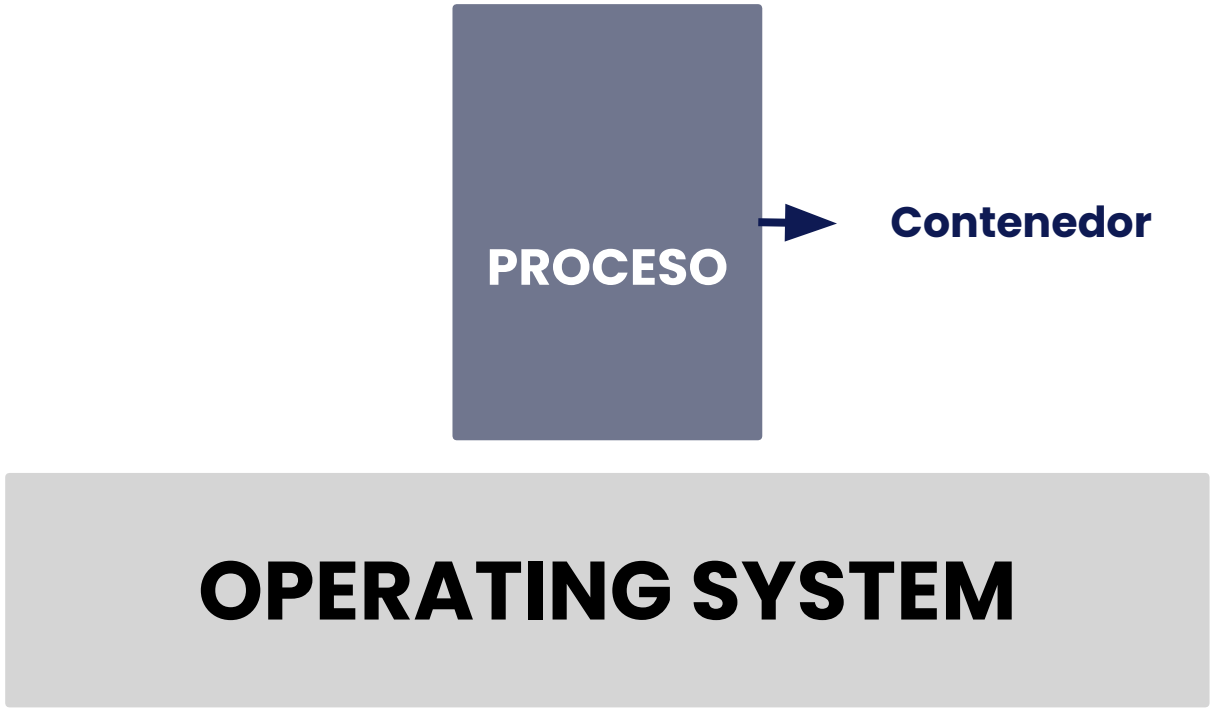
@edithpuclla

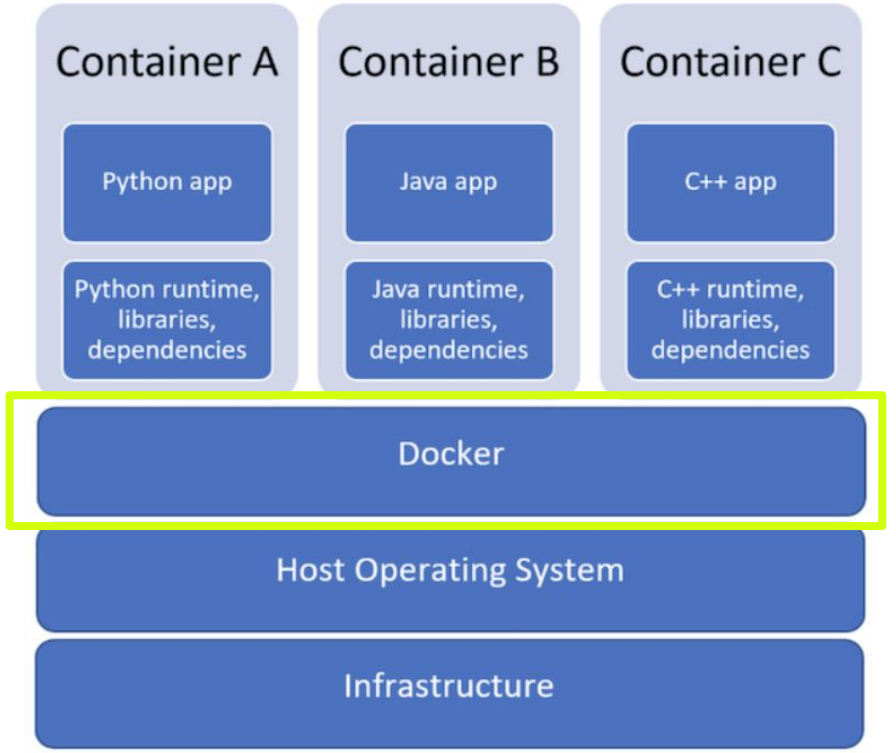
Agenda

- Kubernetes
  - Deploying an Application
  - Default Resources
  - Limitations
- Kubernetes Operators
  - Components
    - CRD, CR, OLM, Controllers
  - Operator framework, Operator Hub, Capability Models


Kubernetes
k8s

Contenedores

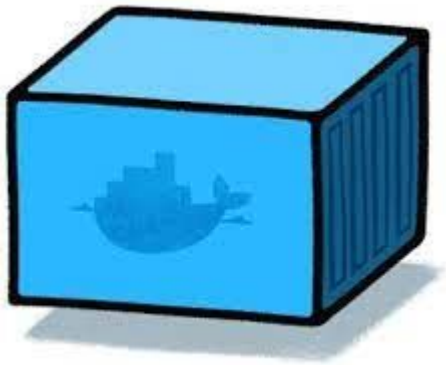




PERCONA

Desafíos con contenedores a escala

- Orquestación
- Seguridad
- Monitoreo y registro
- Escalabilidad
- Almacenamiento y Persistencia de Datos



Ventajas de Kubernetes

- Automatización de despliegues
- Escalado en función a demanda
- Portabilidad de aplicaciones
- Auto-curación
- Buena opción para microservicios
- Comunidad activa
- Amplia adopción

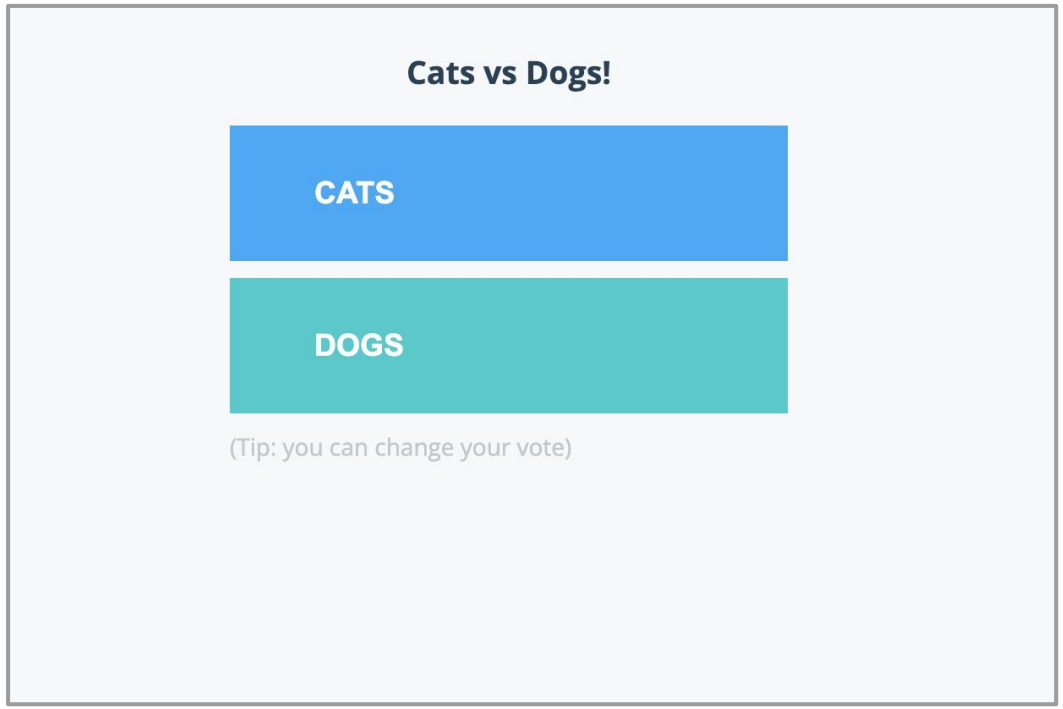
kubernetes

Terminología Kubernetes

- Pods
  - Contenedores
    - Red y almacenamiento
- Deployments
  - Implementación de aplicaciones
  - Estado deseado
  - Replicas
- Services
  - Acceso a Pods

Ejemplo: Application de Voto

voting-app

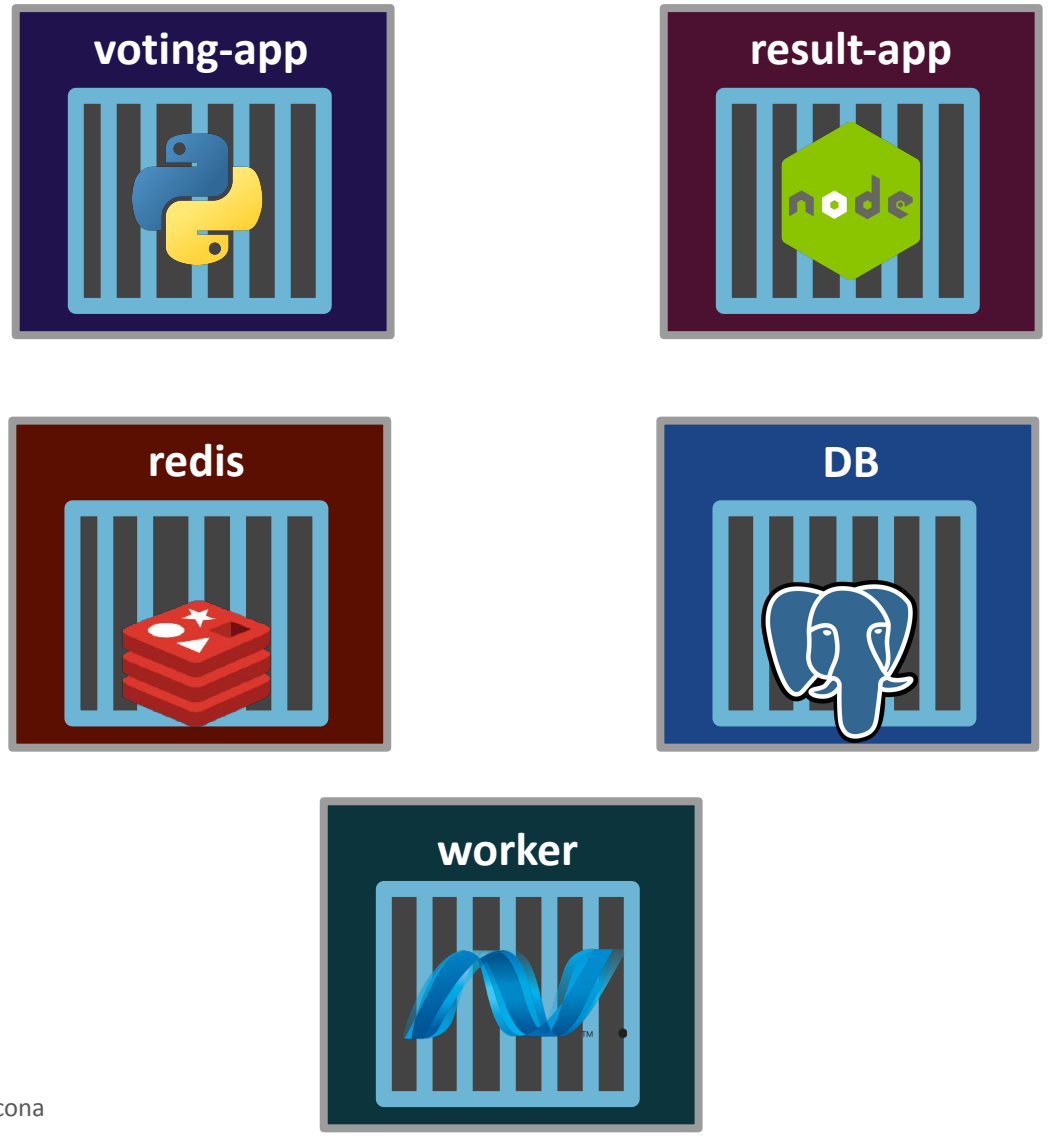


result-app

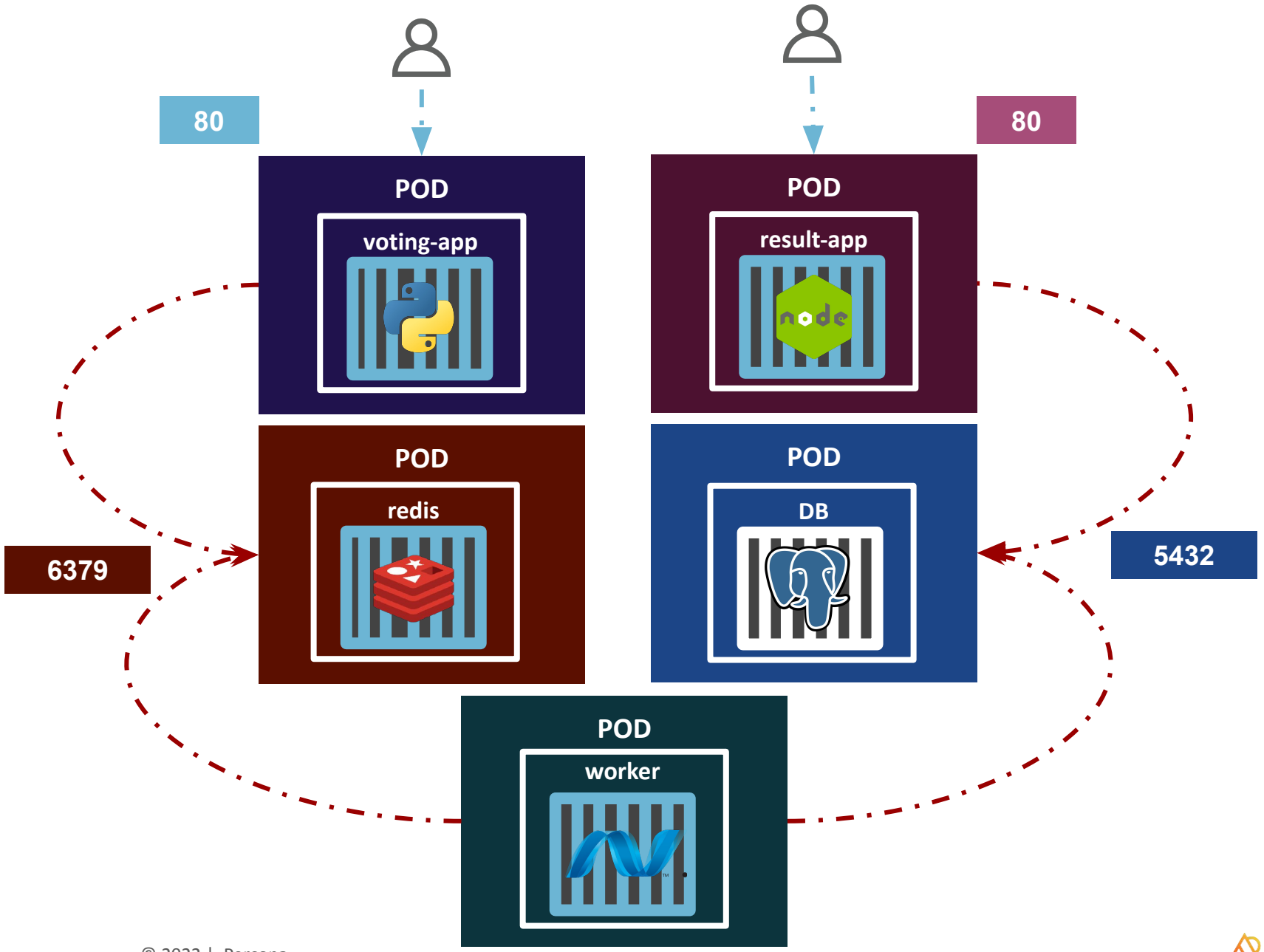
KodeKloud: youtube.com/watch?v=XuSQU5Grv1g

PERCONA

Contenedores

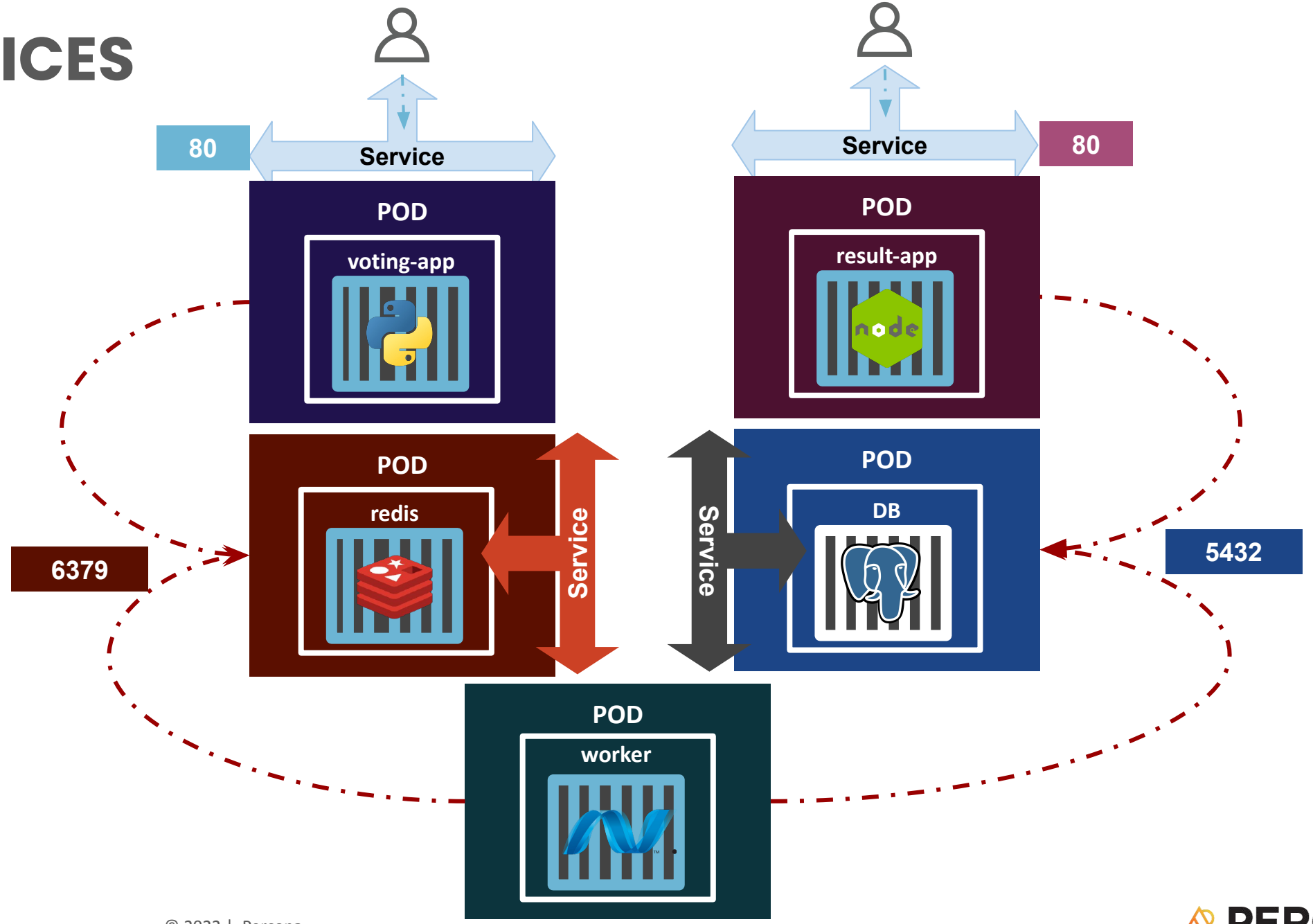


PODS

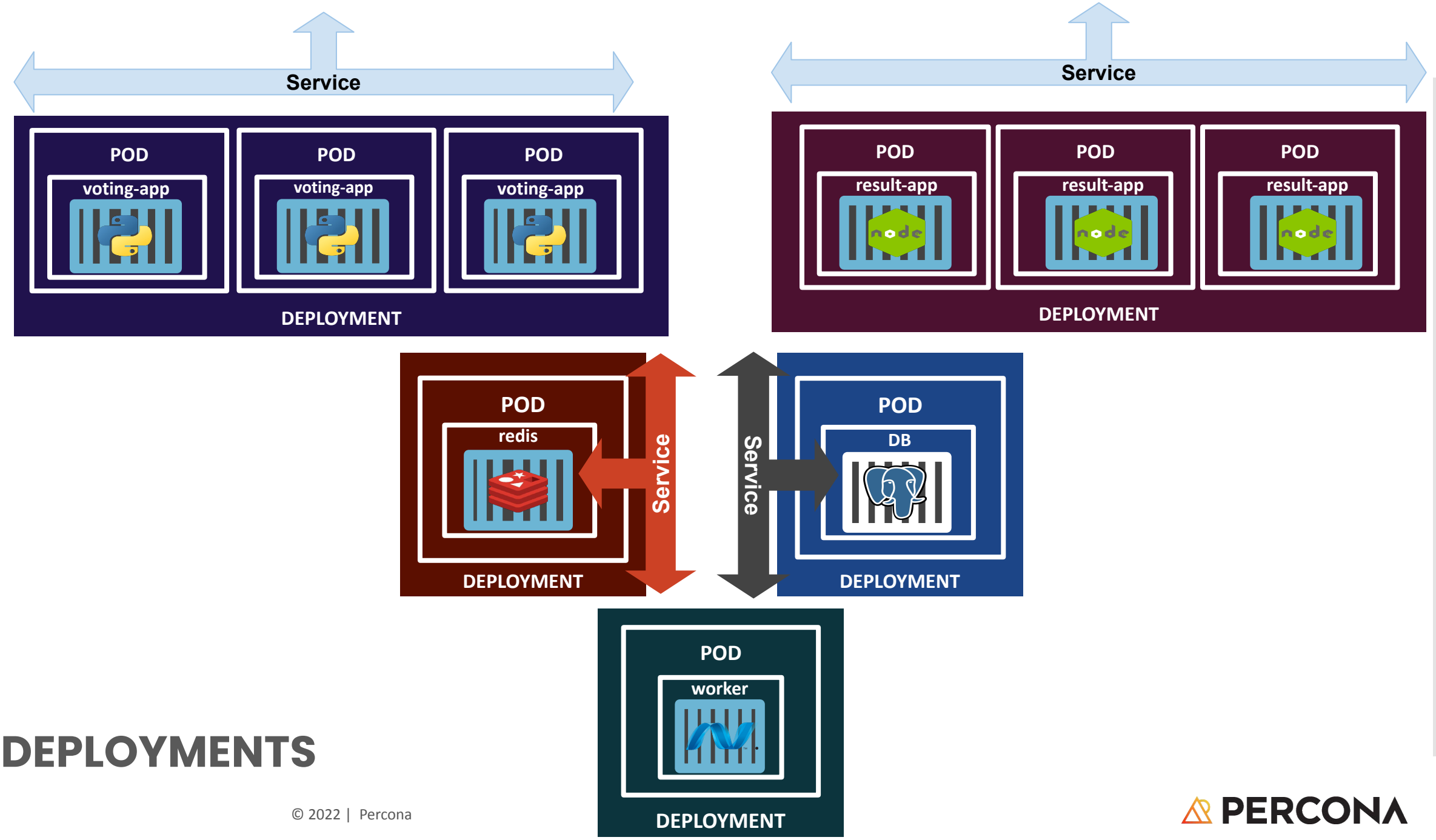


PERCONA

SERVICES



PERCONA



DEPLOYMENTS

YAML

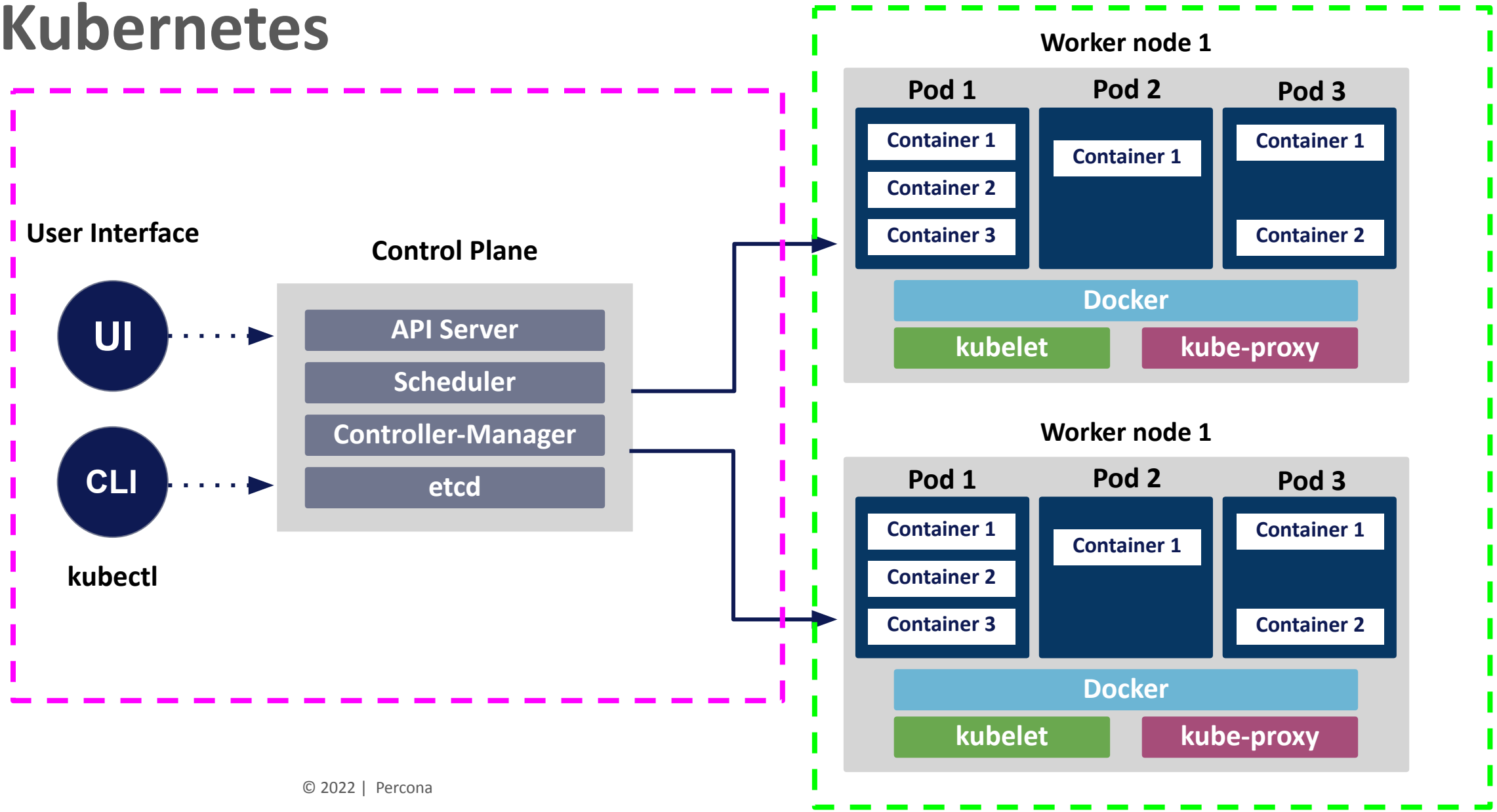
File: **voting-app-deploy.yaml**

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: voting-app-deploy
  labels:
    name: voting-app-deploy
    app: demo-voting-app
spec:
  replicas: 3
  selector:
    matchLabels:
      name: voting-app-pod
      app: demo-voting-app

  template:
    metadata:
      name: voting-app-pod
      labels:
        name: voting-app-pod
        app: demo-voting-app
    spec:
      containers:
        - name: voting-app
          image: kodekloud/examplevotingapp_vote:v1
          ports:
            - containerPort: 80
```

PERCONA

Arquitectura de Kubernetes



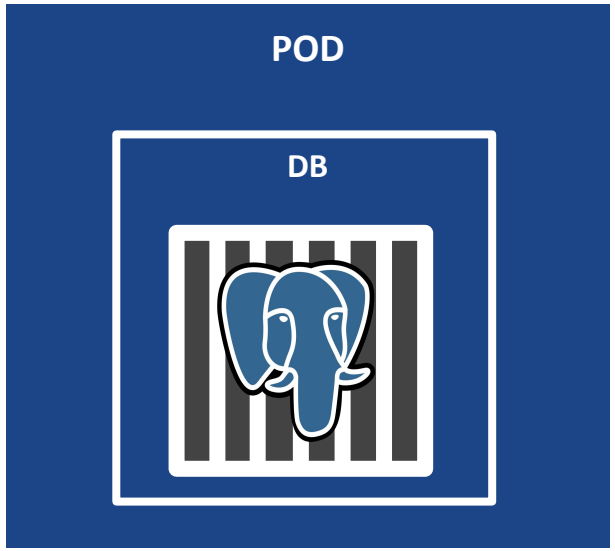
Kubernetes Operators

Escalado de aplicaciones sin estado: fácil

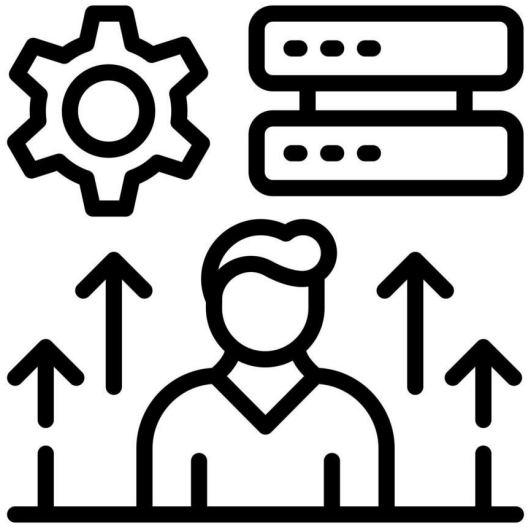
kubectl scale staticweb --replicas=4

¿Qué pasa con las aplicaciones que almacenan datos?

“Desplegar” una base de datos: fácil



Ejecutar una base de datos a lo largo del tiempo es lo más difícil



Operador Humanos

Conocimiento

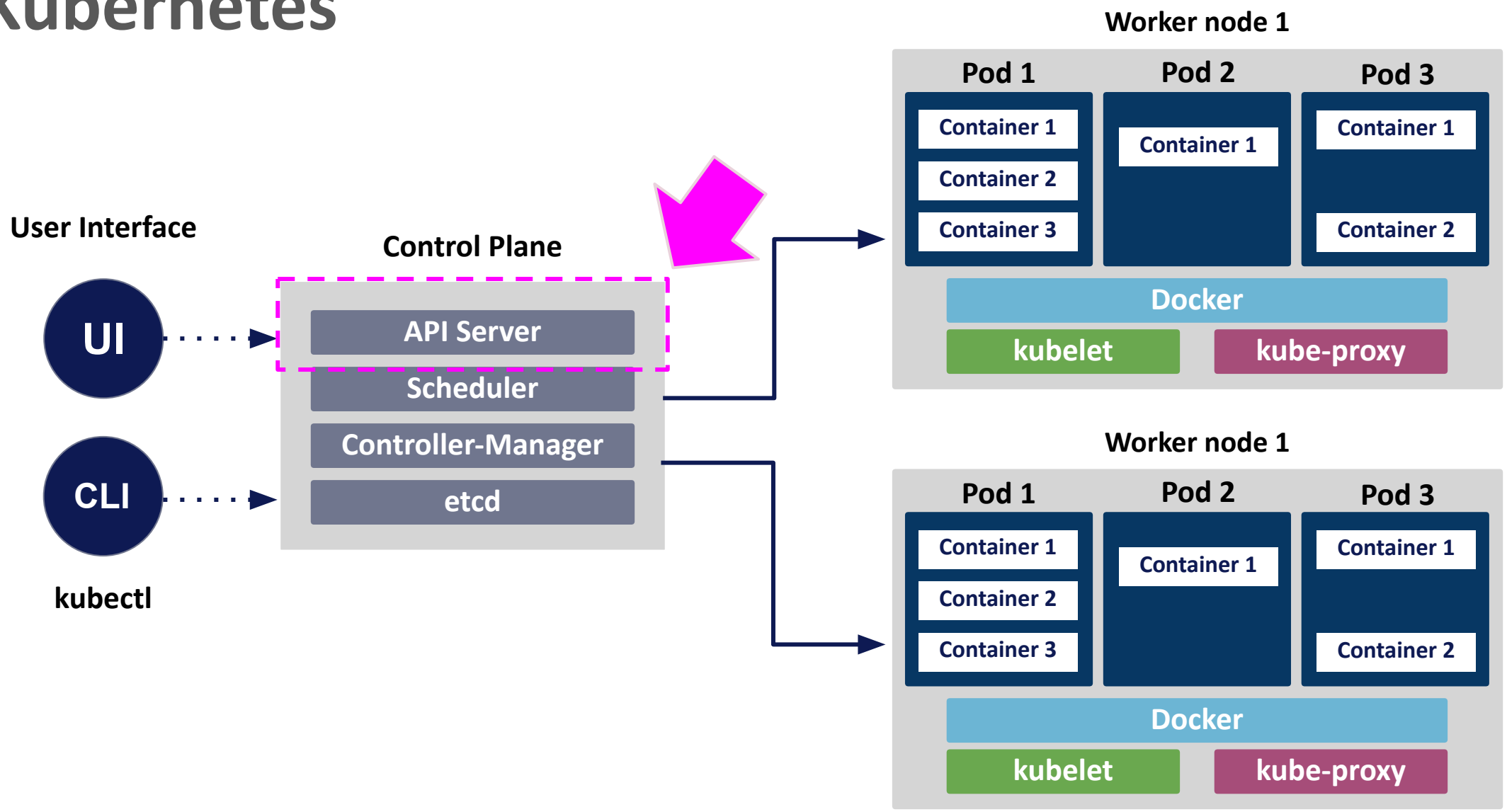
kubernetes

Ciclo de vida de la aplicación Kubernetes

Operadores de Kubernetes:

Extender la **API** de Kubernetes

Arquitectura de Kubernetes



Cómo se Extiende la API:

- Definiciones de recursos personalizados **(Custom Resource Definitions - CRD)**
- Controladores personalizados para aplicaciones específicas **(custom controllers)**

Custom Resource Definitions (CRD)

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  # name must match the spec fields below, and be in the form: <plural>.<group>
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of versions supported by this CustomResourceDefinition
  versions:
    - name: v1
      # Each version can be enabled/disabled by Served flag.
      served: true
      # One and only one version must be marked as the storage version.
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                cronSpec:
                  type: string
                image:
                  type: string
                replicas:
                  type: integer
  # either Namespaced or Cluster
  scope: Namespaced
  names:
    # plural name to be used in the URL: /apis/<group>/<version>/<plural>
    plural: crontabs
    # singular name to be used as an alias on the CLI and for display
    singular: crontab
    # kind is normally the CamelCased singular type. Your resource manifests use this.
    kind: CronTab
    # shortNames allow shorter string to match your resource on the CLI
    shortNames:
    - ct
```

CRD example

PERCONA

Custom Object

my-crontab.yaml

```yaml
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-new-cron-object
spec:
  cronSpec: "* * * * */5"
  image: my-awesome-cron-image
```

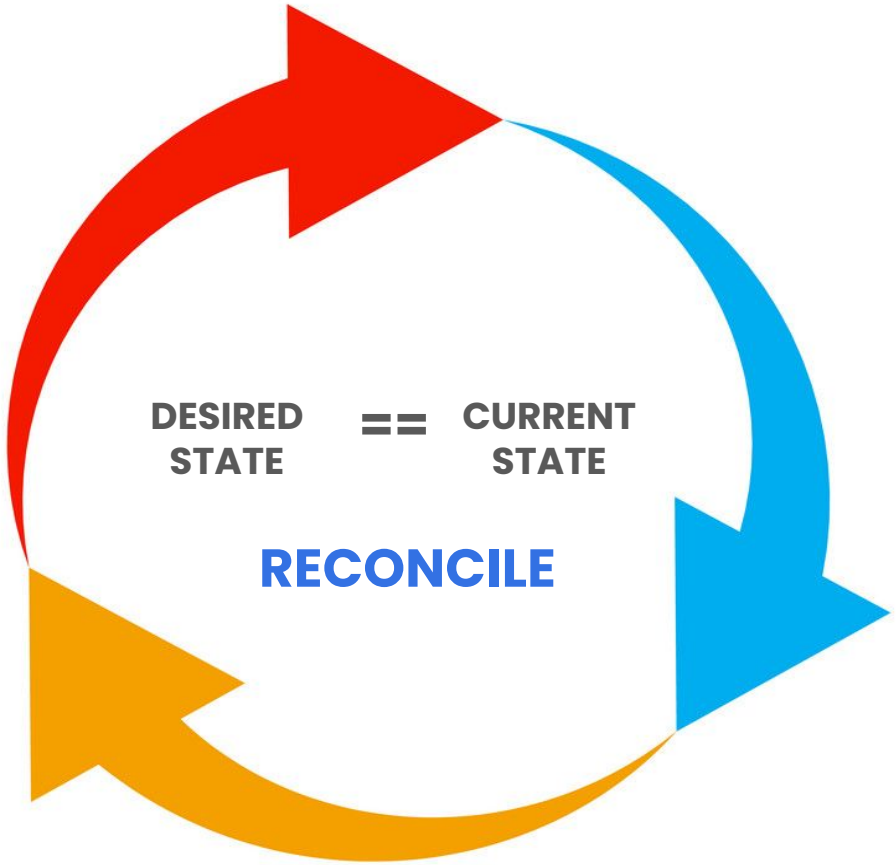
```
kubectl apply -f my-crontab.yaml
```

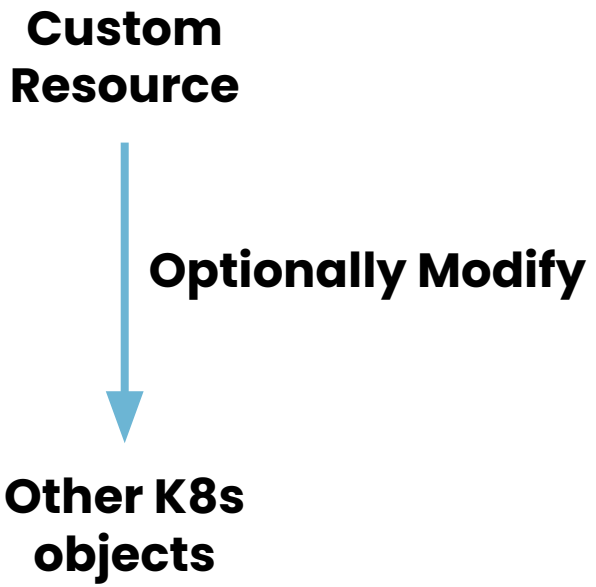
```
kubectl get crontab
```

```
NAME                 AGE
my-new-cron-object   6s
```

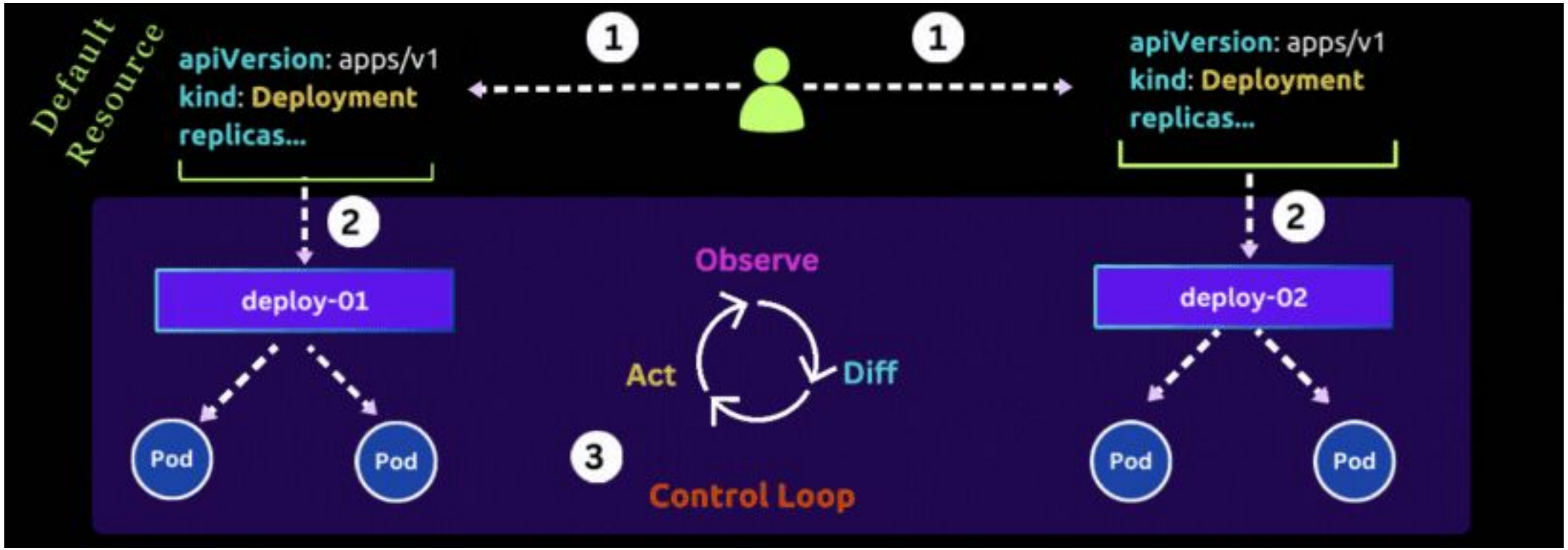
Controladores personalizados (Custom Controller)

Observa cambios en recursos definidos en el **CRD** en el clúster de Kubernetes, detecta cambios y reacciona para nivelarlo.

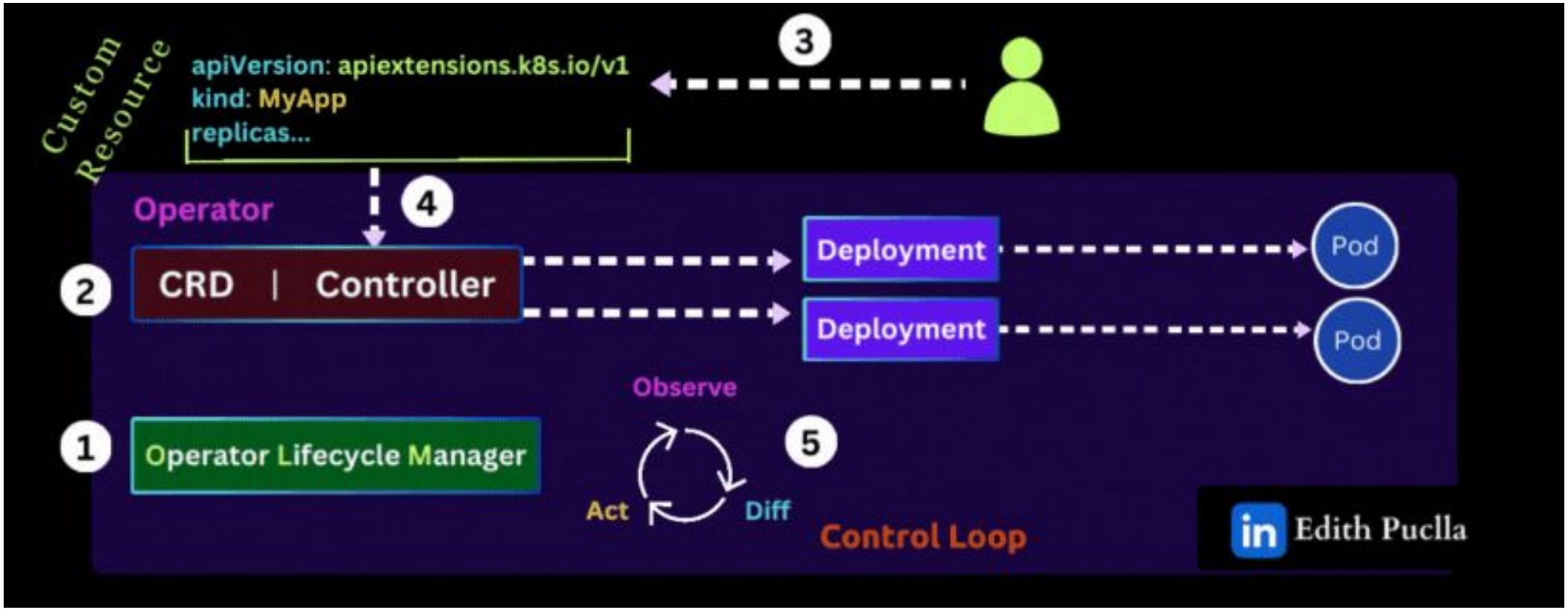




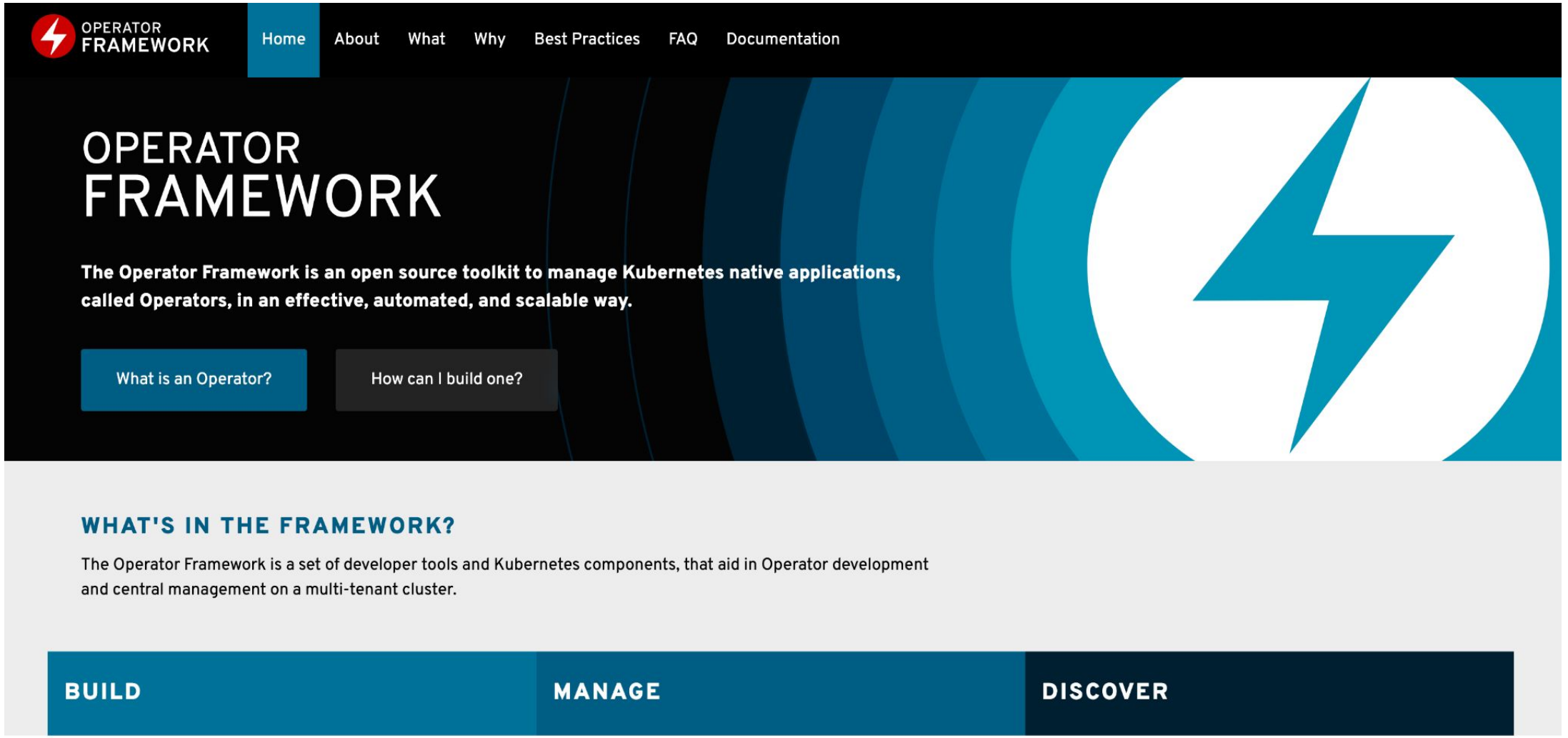
Kubernetes sin Operadores



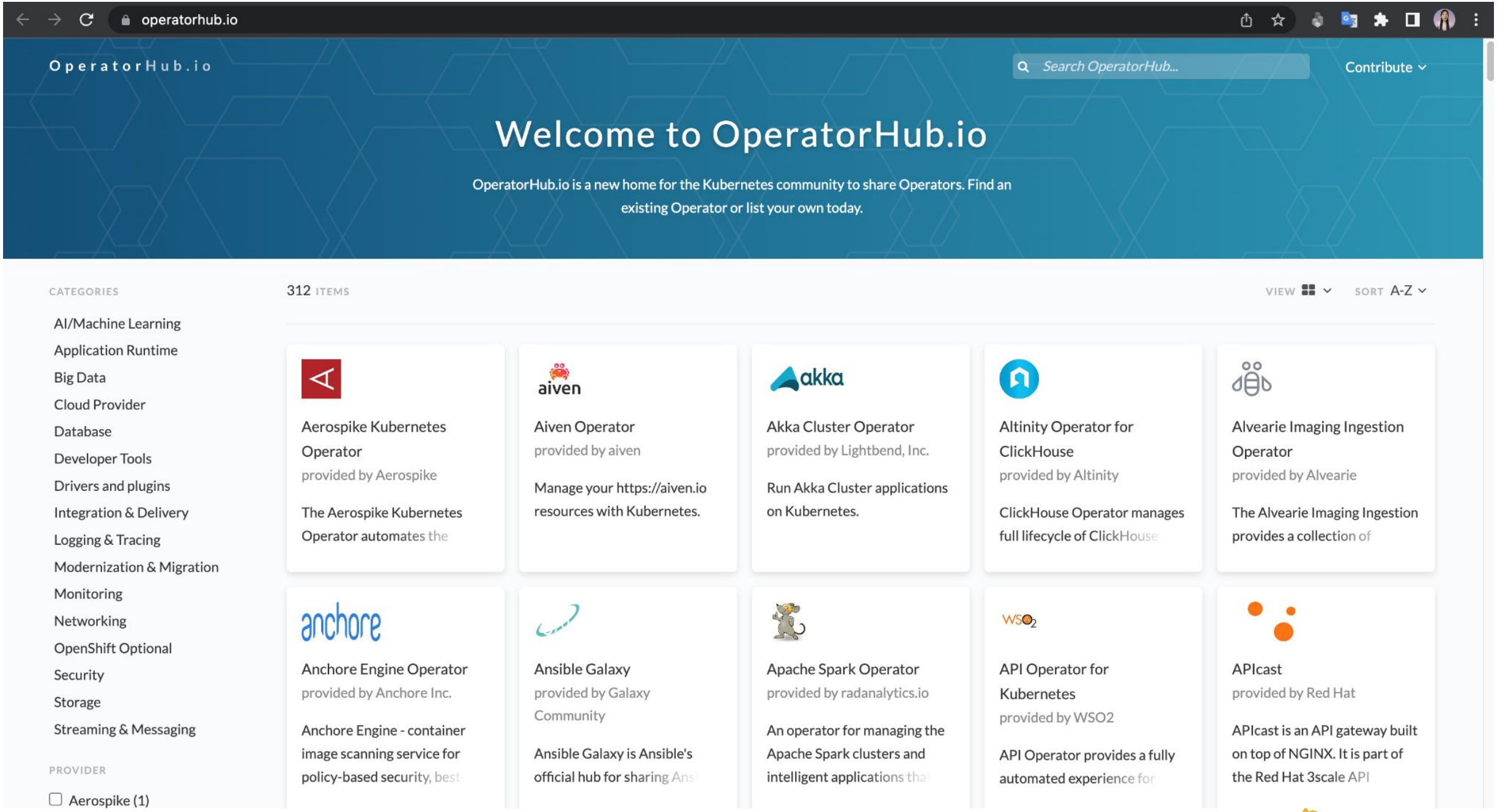
Kubernetes con Operadores



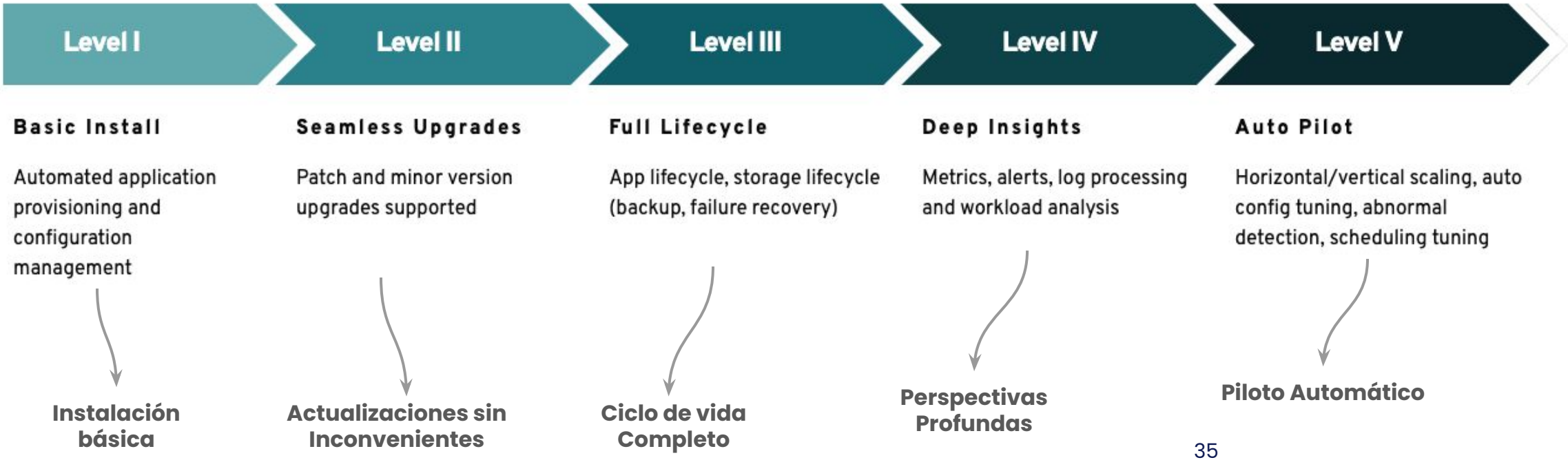
¿Cómo creo los operadores?



[Operator Framework]



Modelo de capacidad

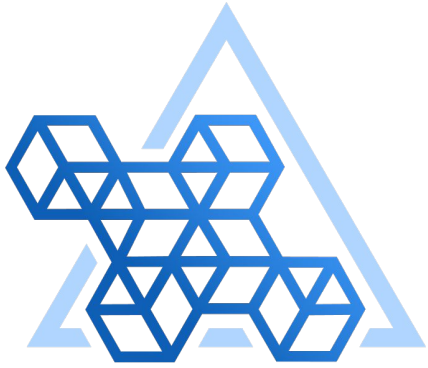


PERCONA

Los operadores de Percona simplifican la configuración y el mantenimiento de clústeres MySQL, PostgreSQL y MongoDB robustos y de nivel empresarial en Kubernetes

Try Percona Operators:

- ➔ Percona Operator for MySQL
- ➔ Percona Operator for MongoDB
- ➔ Percona Operator for PostgreSQL



PERCONA

Kubernetes Operators

Open Source

Percona Operator for MySQL based on Percona XtraDB Cluster

Percona Operator for MySQL based on Percona XtraDB Cluster manages the lifecycle of Percona XtraDB cluster instances.

Percona Operator for MySQL based on Percona XtraDB Cluster

Install

CHANNEL
stable

VERSION
1.12.0 (Current)

CAPABILITY LEVEL
- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

PROVIDER
Percona

LINKS

Percona is Cloud Native

Percona Operator for MySQL based on Percona XtraDB Cluster is an open-source drop in replacement for MySQL Enterprise with synchronous replication running on Kubernetes. It automates the deployment and management of the members in your Percona XtraDB Cluster environment. It can be used to instantiate a new Percona XtraDB Cluster, or to scale an existing environment.

Consult the documentation on the Percona Operator for MySQL based on Percona XtraDB Cluster for complete details on capabilities and options.

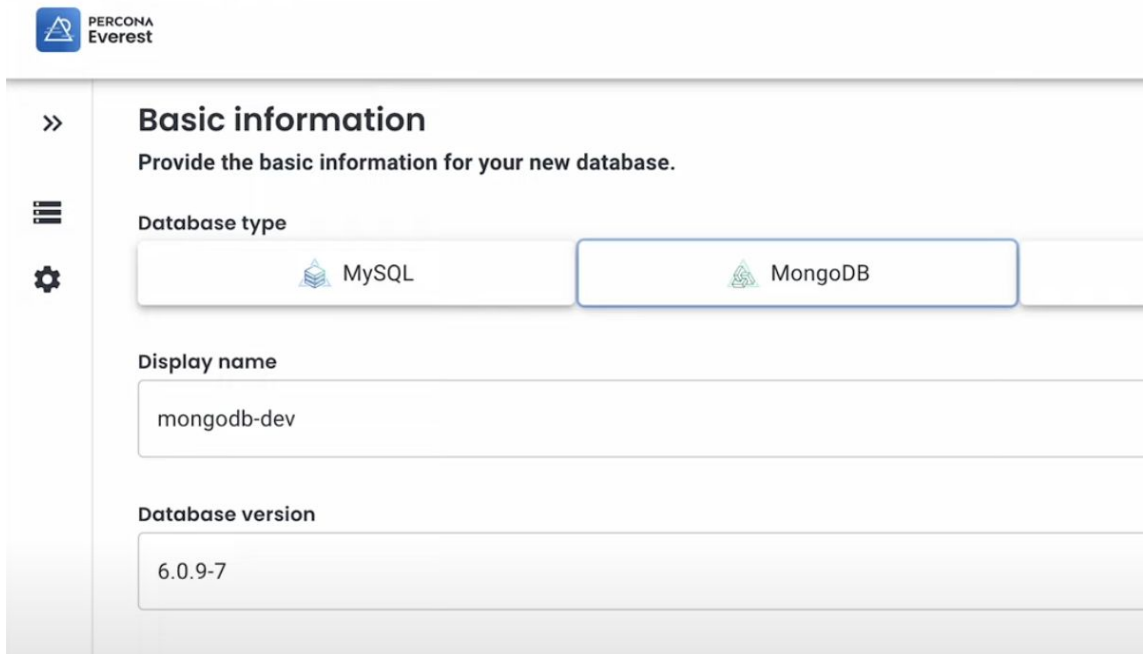
Supported Features

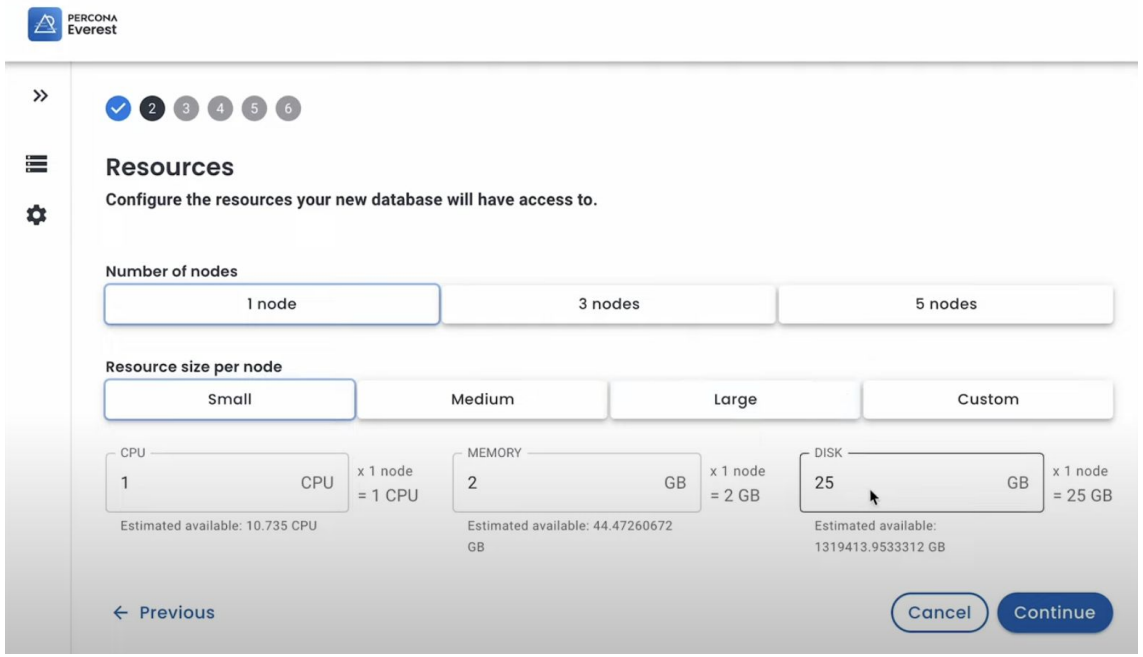
- **Scale Your Cluster** change the `size` parameter to add or remove members of the cluster. Three is the minimum recommended size for a functioning cluster.
- **Manage Your Users** add, remove, or change the privileges of database users
- **Automate Your Backups** configure cluster backups to run on a scheduled basis. Backups can be stored on a persistent volume or S3-compatible storage. Leverage Point-in-time recovery to reduce RPO/RTO.
- **Proxy integration** choose HAProxy or ProxySQL as a proxy in front of the Percona XtraDB Cluster. Proxies are deployed and configured automatically with the

Percona Everest open source, cloud-native database platform

→ docs.percona.com/everest

PERCONA Everest





Open Source

Feedback:

- ➔ percona.community
- ➔ forums.percona.com
- ➔ github.com/percona

percona.com

Twitter: @Percona, @PerconaBytes

LinkedIn: Percona

vacantes

Questions?


Edith Puclla


@edithpuclla


edithturn