

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

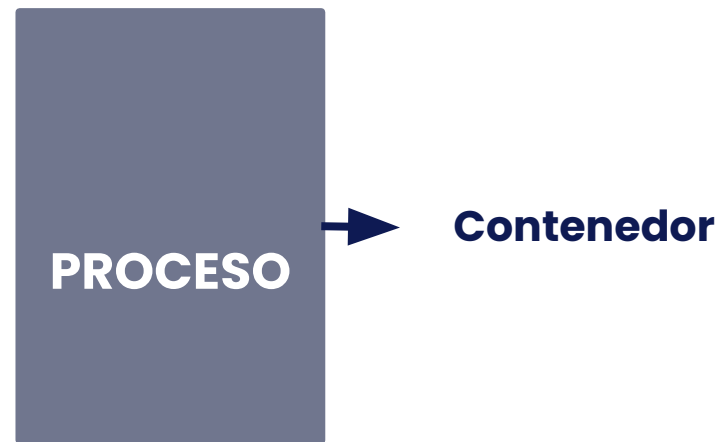
- Kubernetes
 - Deploying an Application
 - Default Resources
 - Limitations
- Kubernetes Operators
 - Components
 - CRD, CR, OLM, Controllers
 - Operator framework, Operator Hub, Capability Models

Kubernetes

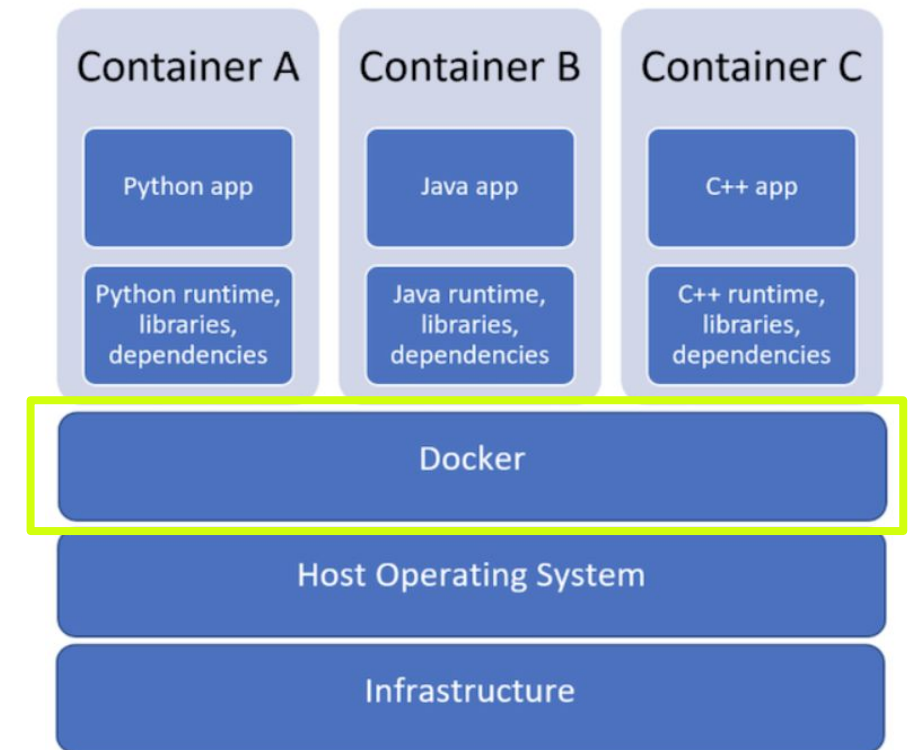
k8s



Contenedores

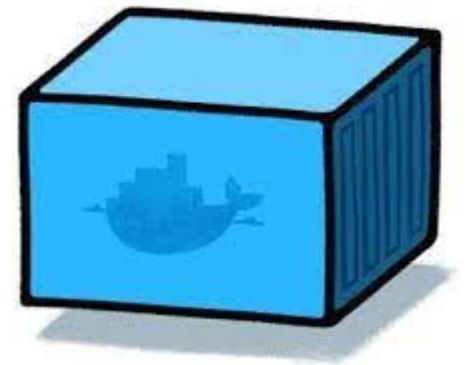


OPERATING SYSTEM



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



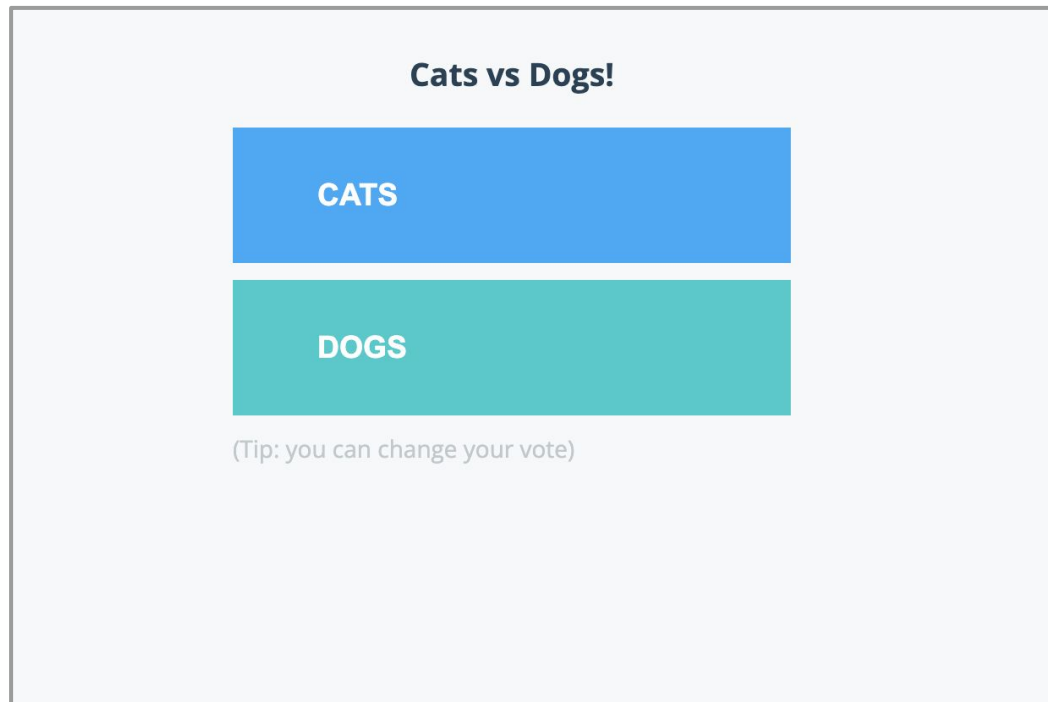
kubernetes

Terminología Kubernetes

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

Ejemplo: Application de Voto

voting-app

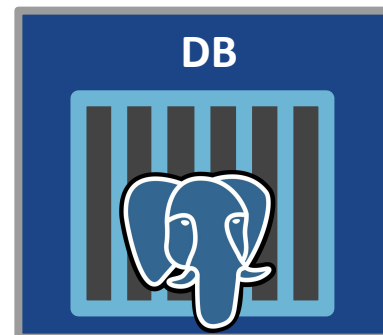
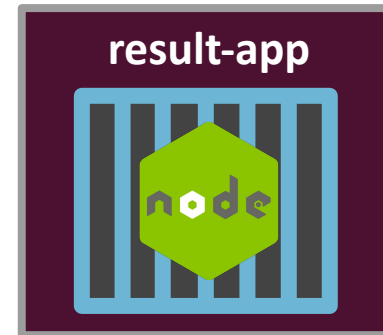


result-app

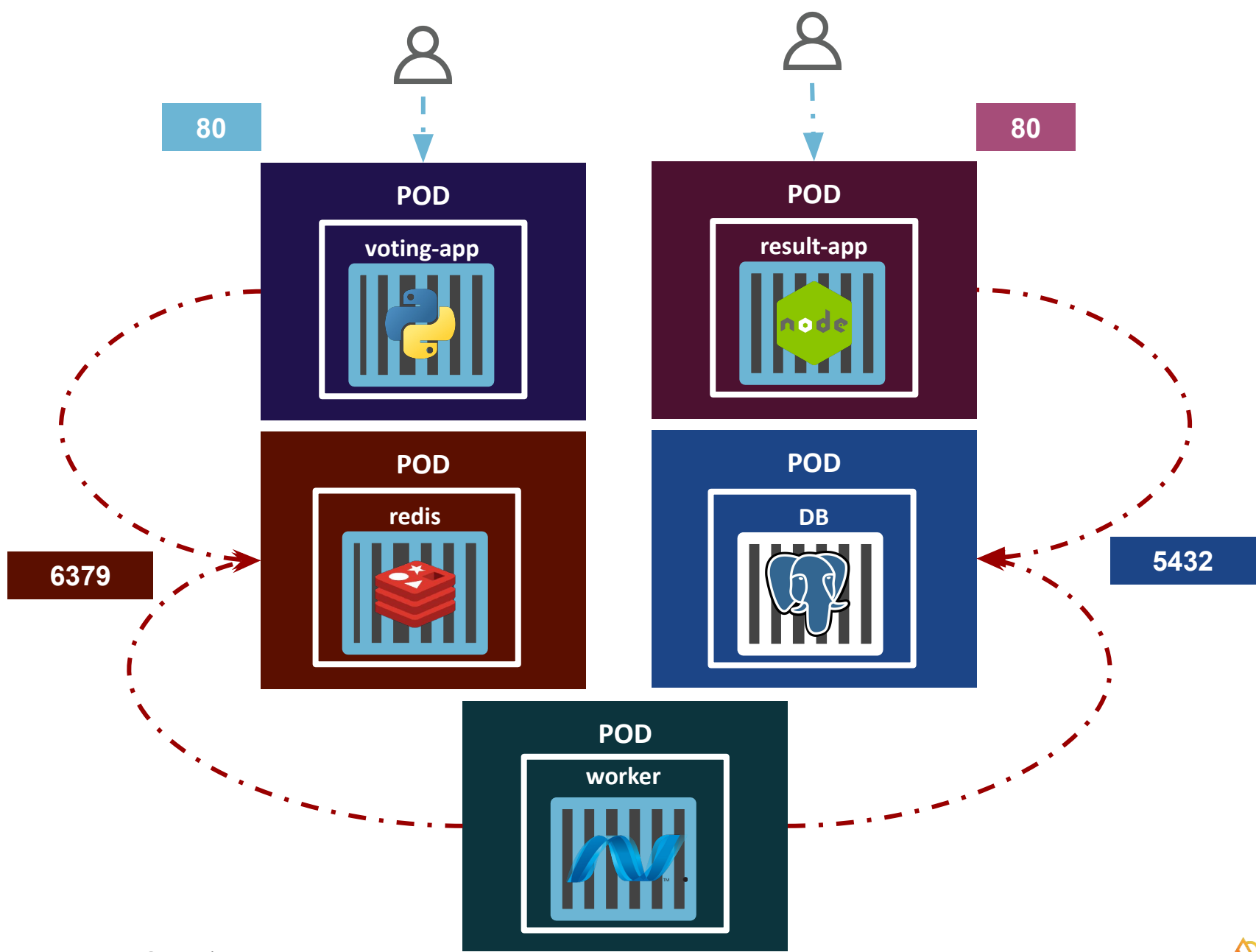


KodeKloud: youtube.com/watch?v=XuSQU5Grvlg

Contenedores

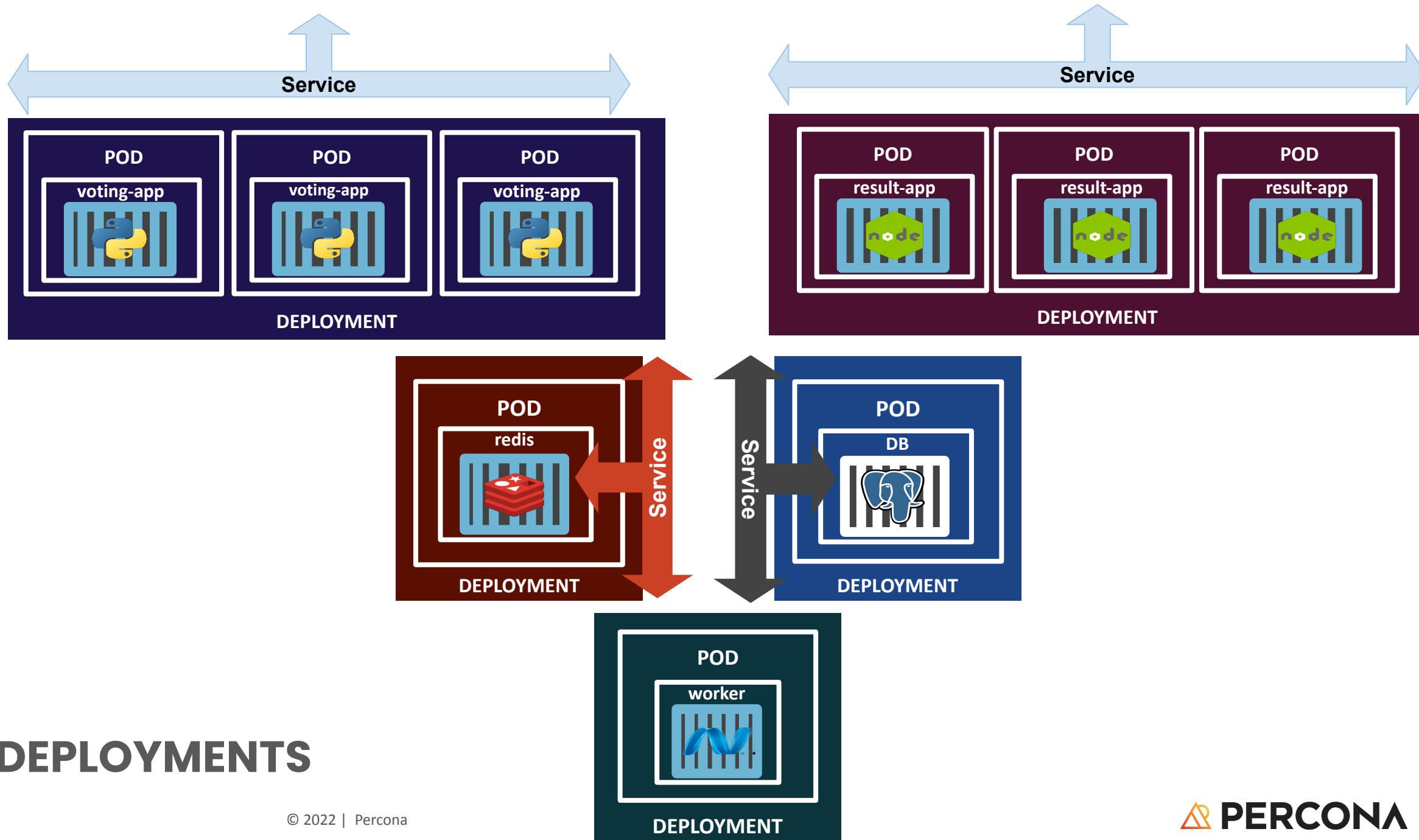


PODS



SERVICES





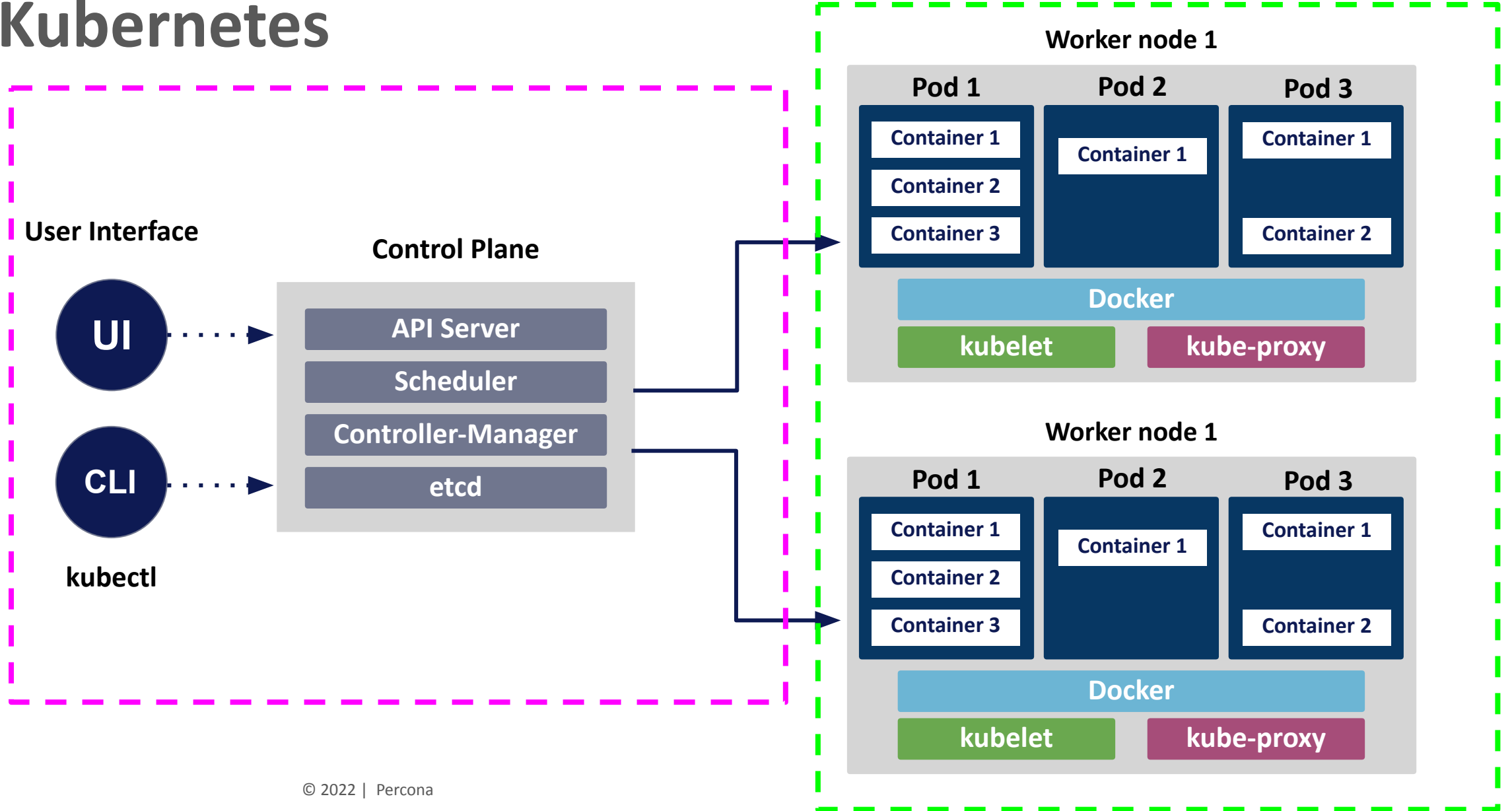
DEPLOYMENTS

YAML

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

Arquitectura de Kubernetes



A photograph of the Machupicchu ruins in Peru, featuring terraced stone buildings and a prominent mountain peak in the background. The image is covered with a semi-transparent blue overlay.

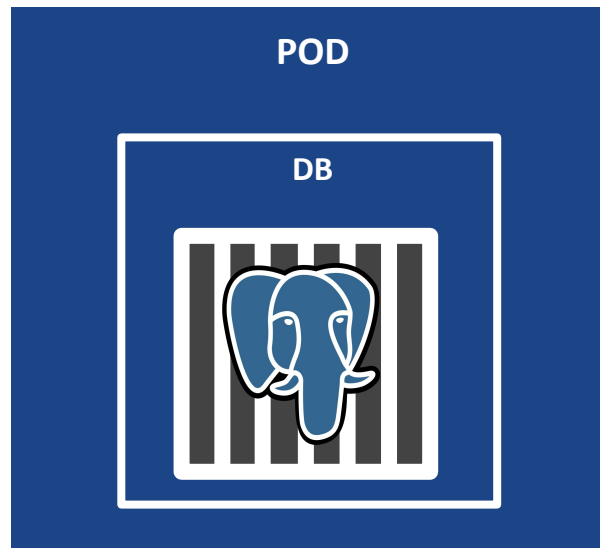
Kubernetes Operators

Escalado de aplicaciones sin estado: **fácil**

kubectl scale staticweb --replicas=4

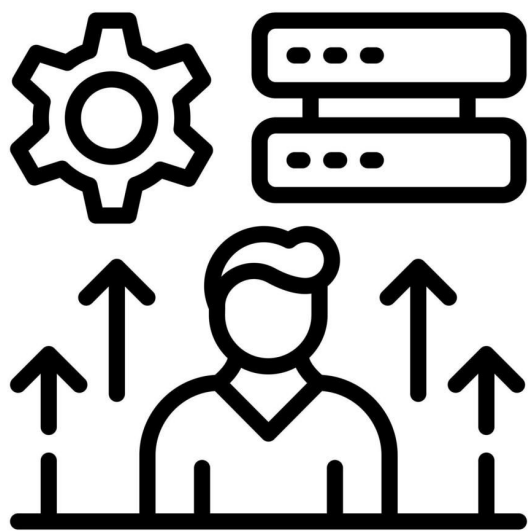
¿Qué pasa con las aplicaciones que almacenan datos?

“Desplegar” una base de datos: **fácil**



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





Operador Humanos

Conocimiento



kubernetes

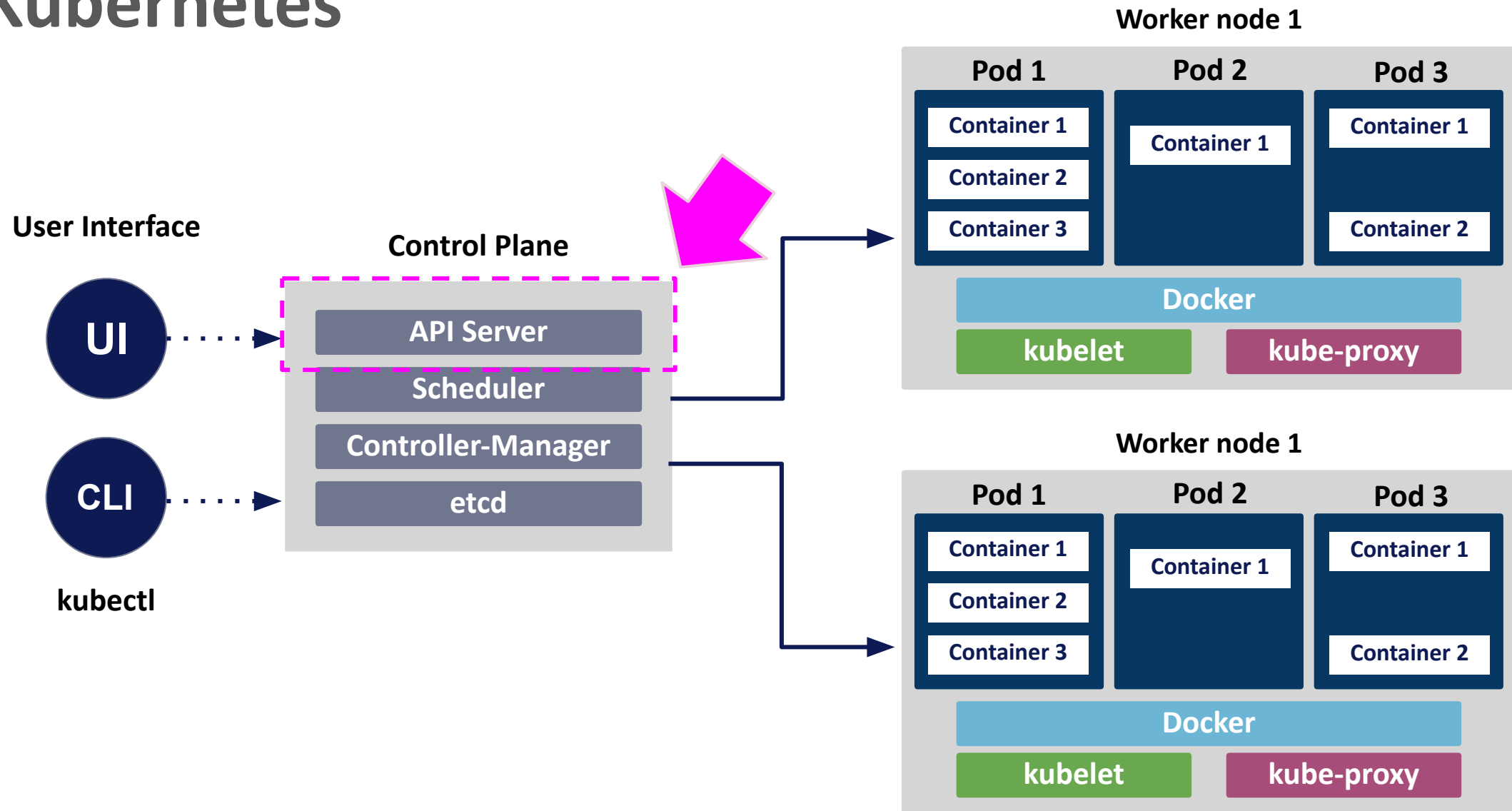
Ciclo de vida de la aplicación Kubernetes



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)

```
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
```

kind: CronTab

CRD example

Custom Object

my-crontab.yaml

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



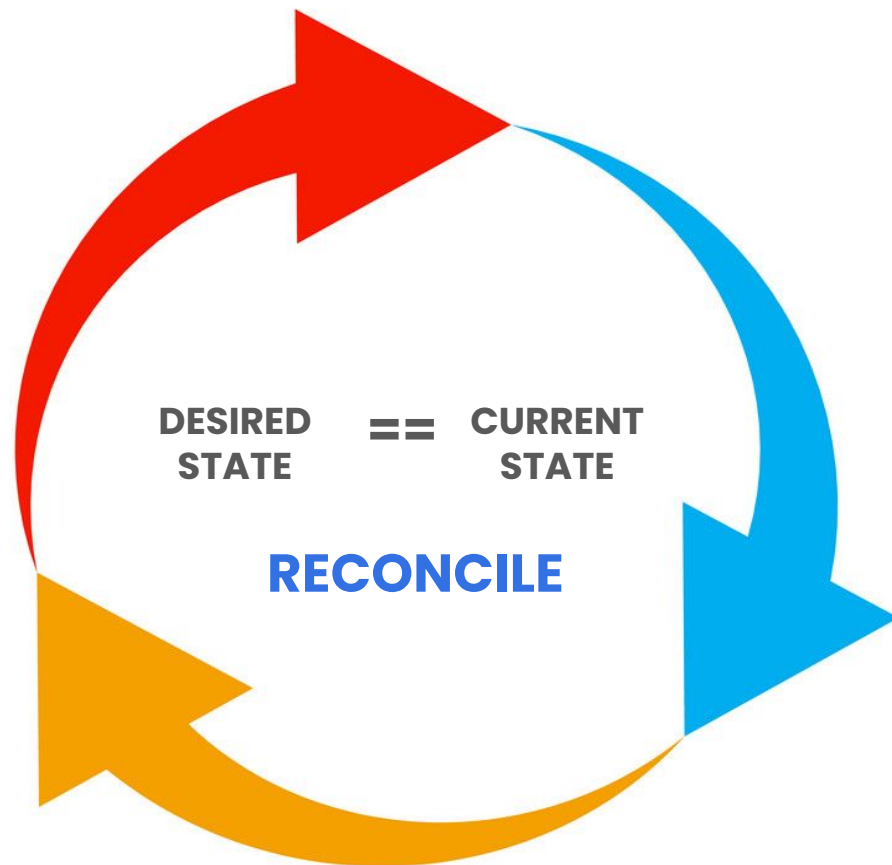
```
kubectl apply -f my-crontab.yaml
```

```
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.

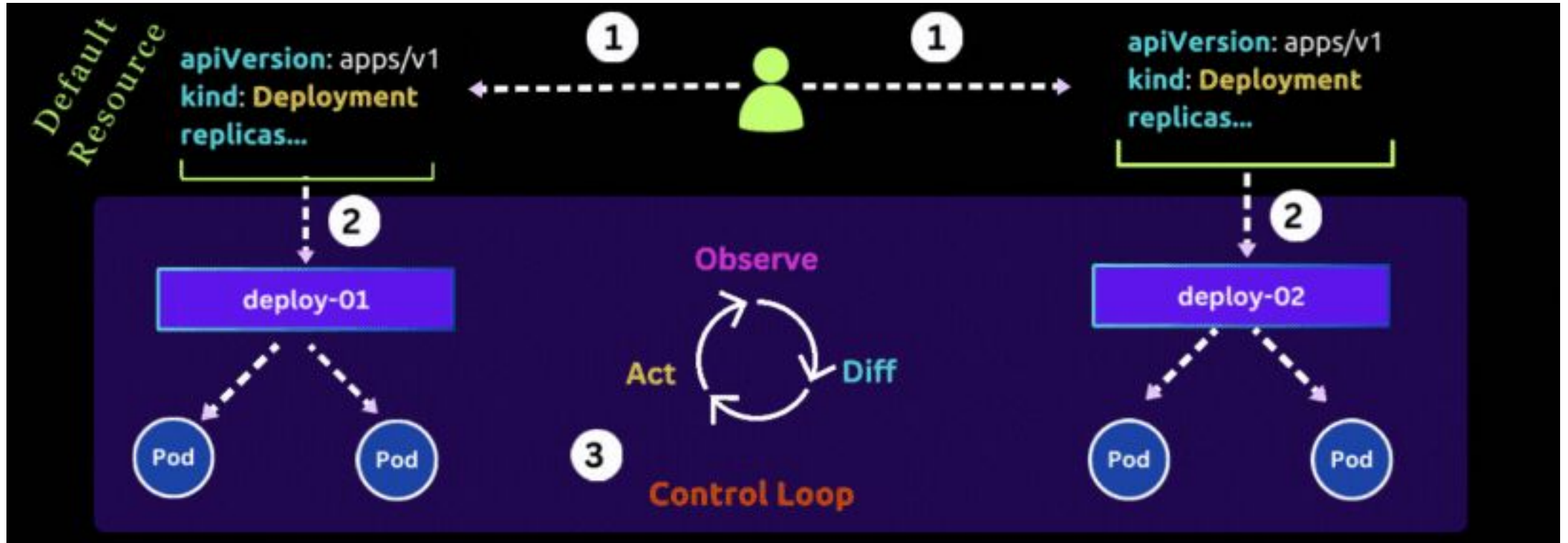


**Custom
Resource**

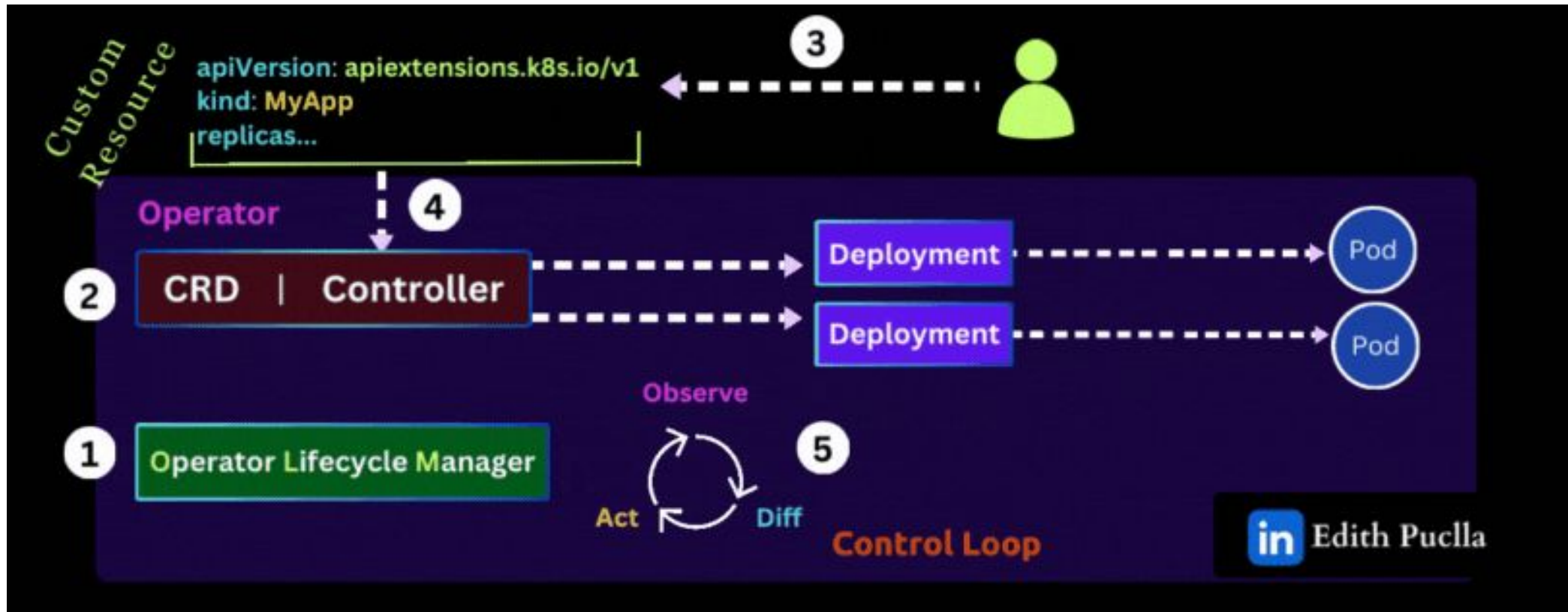
Optionally Modify

**Other K8s
objects**

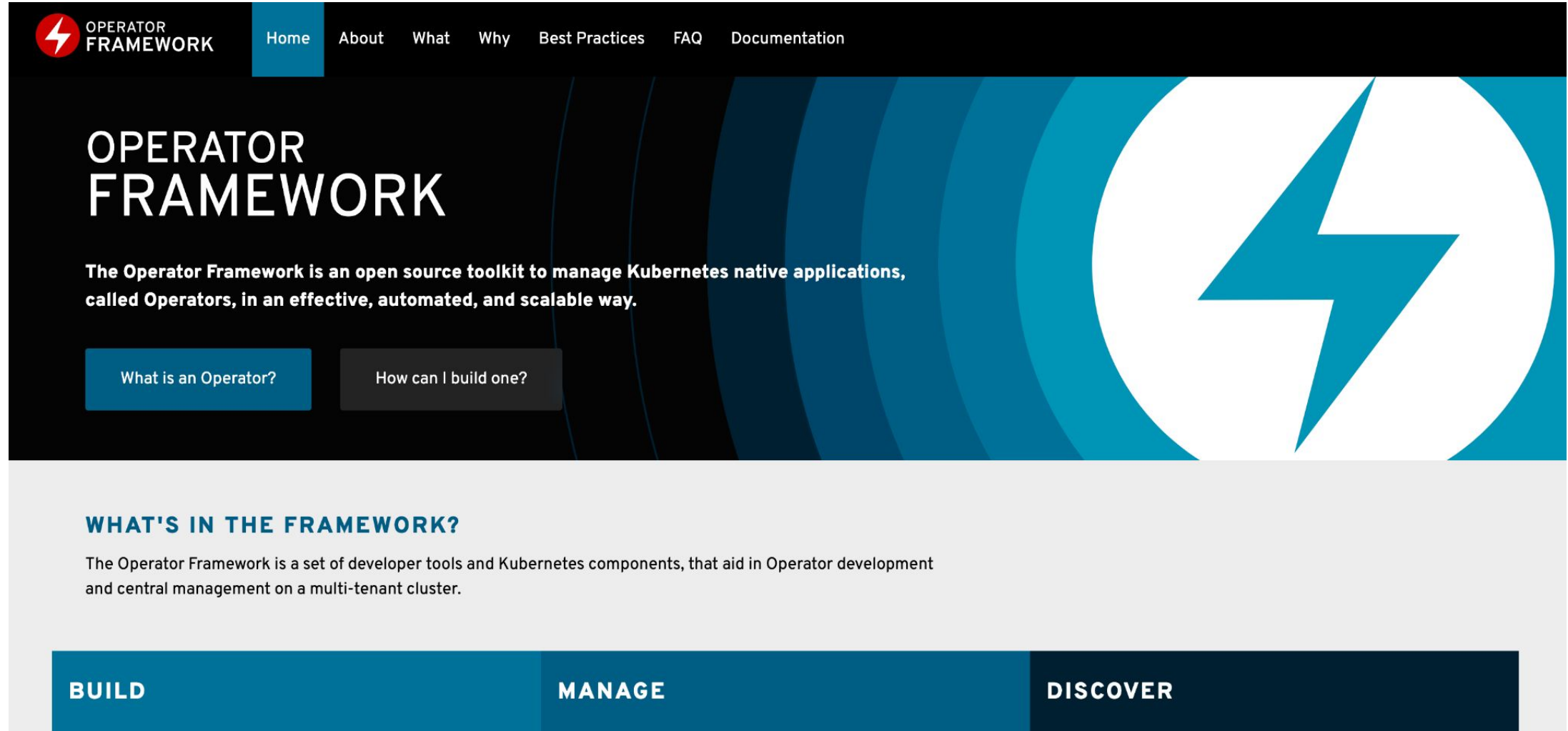
Kubernetes sin Operadores



Kubernetes con Operadores



¿Cómo creo los operadores?

The image is a screenshot of the Operator Framework website. The header is dark with a navigation menu containing links: Home, About, What, Why, Best Practices, FAQ, and Documentation. The 'Home' link is highlighted in blue. The main content area has a dark background with a large blue lightning bolt logo on the right. The text 'OPERATOR FRAMEWORK' is prominently displayed in white. Below it, a paragraph describes the framework as an open source toolkit for managing Kubernetes native applications. Two buttons are present: 'What is an Operator?' (blue) and 'How can I build one?' (grey). A section titled 'WHAT'S IN THE FRAMEWORK?' follows, with a paragraph explaining the framework's components. At the bottom, there is a horizontal bar with three segments: 'BUILD' (blue), 'MANAGE' (blue), and 'DISCOVER' (dark blue).

OPERATOR FRAMEWORK

The Operator Framework is an open source toolkit to manage Kubernetes native applications, called Operators, in an effective, automated, and scalable way.

[What is an Operator?](#) [How can I build one?](#)

WHAT'S IN THE FRAMEWORK?

The Operator Framework is a set of developer tools and Kubernetes components, that aid in Operator development and central management on a multi-tenant cluster.

BUILD **MANAGE** **DISCOVER**

[Operator Framework](#)

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

312 ITEMS

VIEW [📄](#) ▾ SORT [A-Z](#) ▾

- 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
- Storage
- Streaming & Messaging

PROVIDER

☐ Aerospike (1)



Aerospike Kubernetes Operator

provided by Aerospike

The Aerospike Kubernetes Operator automates the



Aiven Operator

provided by aiven

Manage your <https://aiven.io> resources with Kubernetes.



Akka Cluster Operator

provided by Lightbend, Inc.

Run Akka Cluster applications on Kubernetes.



Altinity Operator for ClickHouse

provided by Altinity

ClickHouse Operator manages full lifecycle of ClickHouse



Alvearie Imaging Ingestion Operator

provided by Alvearie

The Alvearie Imaging Ingestion provides a collection of



Anchore Engine Operator

provided by Anchore Inc.

Anchore Engine - container image scanning service for policy-based security, best



Ansible Galaxy

provided by Galaxy Community

Ansible Galaxy is Ansible's official hub for sharing Ans



Apache Spark Operator

provided by radanalytics.io

An operator for managing the Apache Spark clusters and intelligent applications that



API Operator for Kubernetes

provided by WSO2

API Operator provides a fully automated experience for

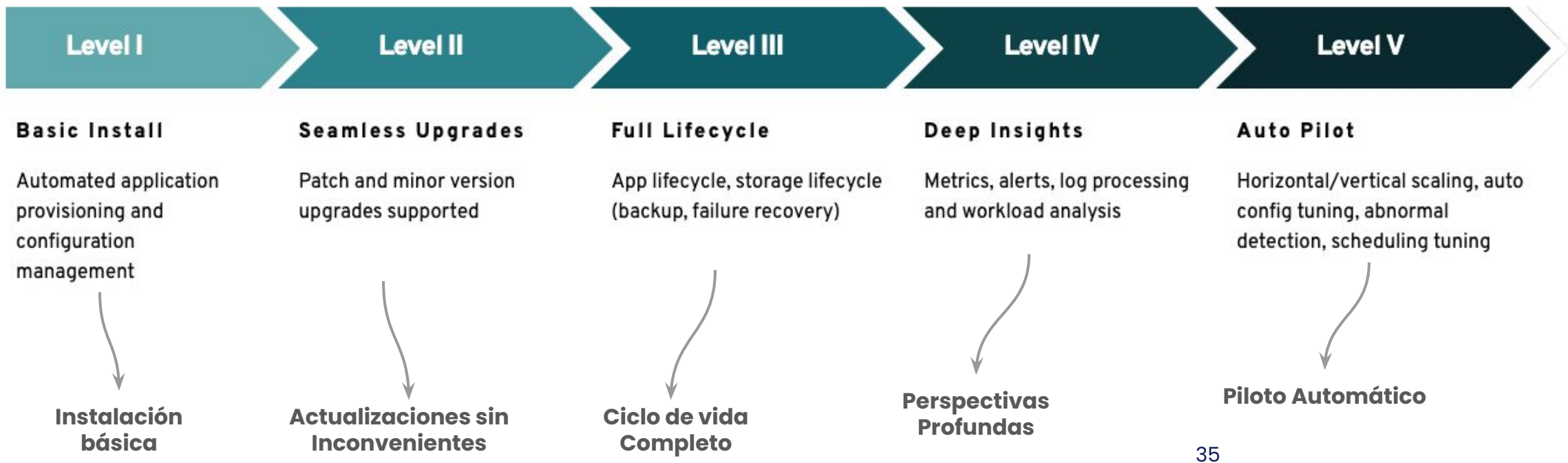


APIcast

provided by Red Hat

APIcast is an API gateway built on top of NGINX. It is part of the Red Hat 3scale API

Modelo de capacidad

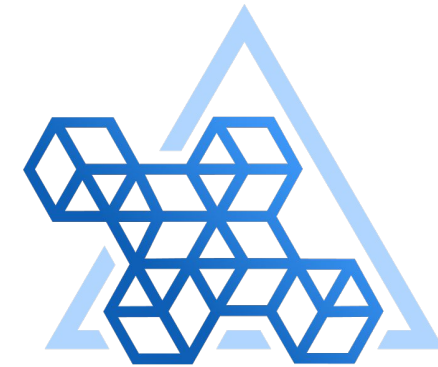


Los operadores de Percona simplifican la configuración y el mantenimiento de clústeres MySQL, PostgreSQL y MongoDB robustos y de nivel empresarial en Kubernetes

Try Percona Operators:

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

Open source



PERCONA

Kubernetes
Operators



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.

[Home](#) > Percona Operator for MySQL based on Percona XtraDB Cluster

Percona Operator for MySQL based on Percona XtraDB Cluster

Install

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

CHANNEL

stable

VERSION

1.12.0 (Current)

CAPABILITY LEVEL ?

- ☒ Basic Install
- ☒ Seamless Upgrades
- ☒ Full Lifecycle
- ☒ Deep Insights
- ☐ Auto Pilot

PROVIDER

Percona

LINKS

Percona Everest open source, cloud-native database platform

→ docs.percona.com/everest



PERCONA Everest

>> **Basic information**
Provide the basic information for your new database.

Database type

☐ MySQL ☒ MongoDB

Display name

mongodb-dev

Database version

6.0.9-7

PERCONA Everest

1 2 3 4 5 6

Resources
Configure the resources your new database will have access to.

Number of nodes

Resource size per node

☒ Small ☐ Medium ☐ Large ☐ Custom

CPU CPU x 1 node = 1 CPU
Estimated available: 10.735 CPU

MEMORY GB x 1 node = 2 GB
Estimated available: 44.47260672 GB

DISK GB x 1 node = 25 GB
Estimated available: 1319413.9533312 GB

[← Previous](#) [Cancel](#) [Continue](#)

Open Source

Feedback:

- [percona.community](https://percona.com/community)
- forums.percona.com
- github.com/percona

percona.com

Twitter: @Percona, @PerconaBytes

LinkedIn: Percona



PERCONA



vacantes

Questions?



Edith Puclla



@edithpuclla



edithturn

