

# Tech - Ep 5 - We're back, let's see some KEDA!

**Kubernetes Event Driven Autoscaler**

Pedro Ibáñez Requena  
Principal Software Engineer



# Index

- Que es KEDA?
- Como funciona?
- Funcionalidades
- Instalación
- Demo
- Links de interés



Que es KEDA?

# Que es KEDA?

KEDA es un autoescaler basado en eventos. Con KEDA se puede gestionar el escalado de cualquier contenedor en Kubernetes basado en un amplio número de eventos diferentes.

KEDA es un componente ligero con único propósito, que puede ser añadido a cualquier cluster de Kubernetes (puede escalar a 0 para ahorrar recursos).

KEDA trabaja complementando componentes estándar de Kubernetes como el [Horizontal Pod Autoscaler](#) y extiende sus funcionalidades sin sobre escribirlas o duplicarlas. Con KEDA puedes mapear las aplicaciones que quieres para usar el escalado basado en eventos, mientras otras aplicaciones siguen funcionando de forma normal. KEDA es una opción flexible y fiable para poder ser ejecutada con cualquier tipo de aplicaciones o frameworks.

CNCF Incubating project.



Como funciona?

# Como funciona?

**Agent** — KEDA activa y desactiva los deployments de Kubernetes para escalar hacia o desde cero cuando se producen eventos. Este es uno de los roles primarios del operador de KEDA que se ejecuta al instalar KEDA.

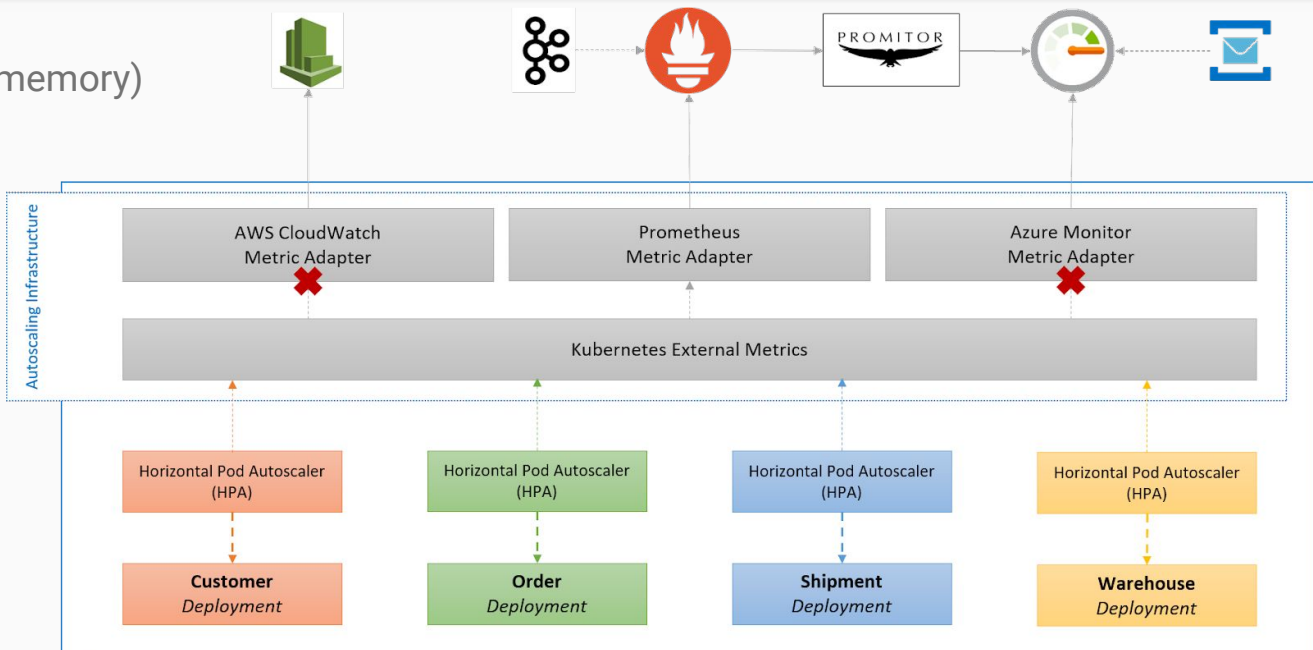
**Metrics** — KEDA actúa como servidor de métricas de Kubernetes que expone datos ricos en eventos como la longitud de las colas o el retardo de las colas del Horizontal Pod Autoscaler para poder escalar hacia abajo. Servir métricas es el rol primario del contenedor keda-operator-metrics-apiserver que se ejecuta al instalar KEDA.

**Scalers** — detectan si un deployment debería ser activado o desactivado y al mismo tiempo alimenta las métricas para un evento específico.



# Como funciona?

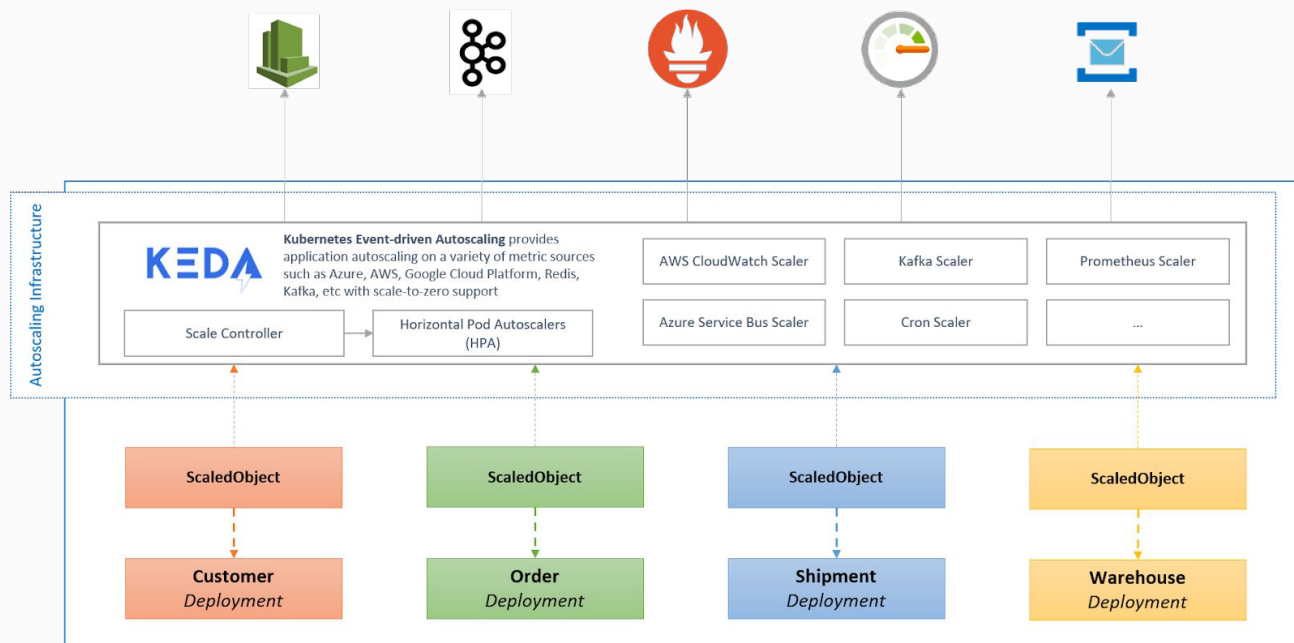
HPA limitations (CPU & memory)



kubernetes



# Como funciona?

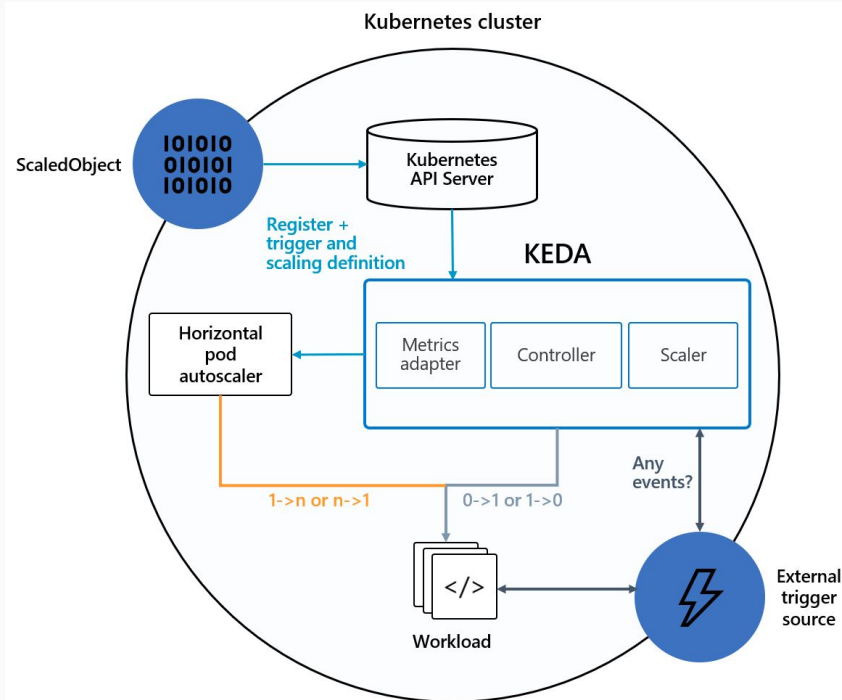


kubernetes





# Como funciona?



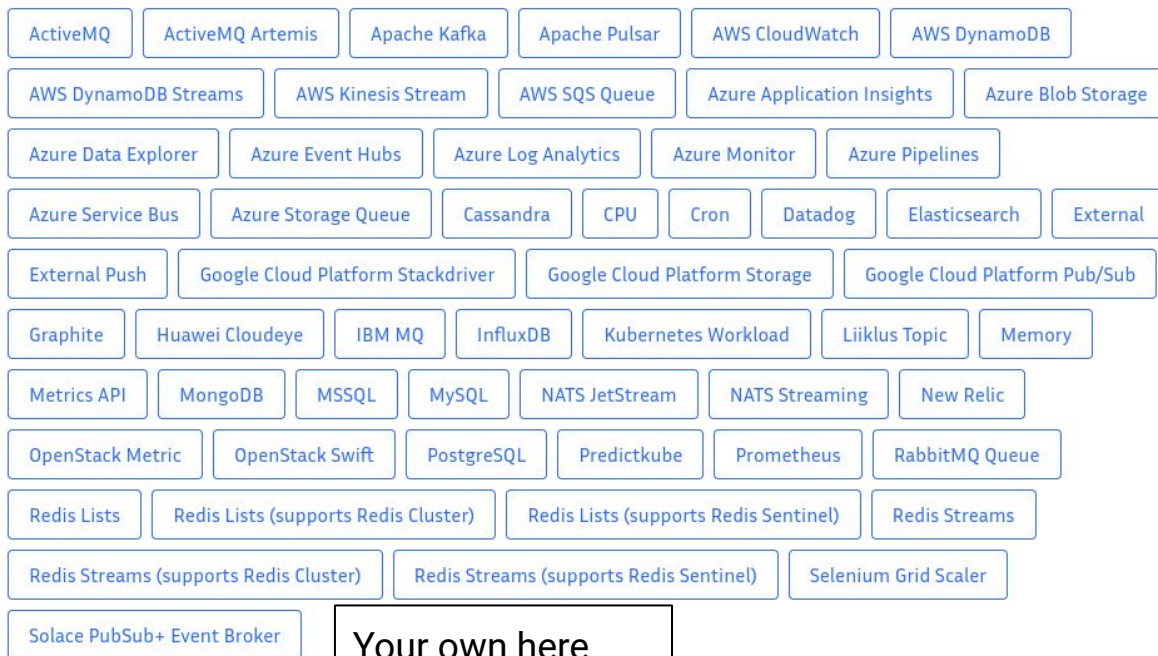
# Funcionalidades

# Funcionalidades

- **Event-driven:** escalado basado en eventos de aplicaciones.
- **Simple autoscaling:** escalado simple de cargas de trabajo.
- **Built-in Scalers:** lista de *scalars* preparados para ser usados, diferentes proyectos, bases de datos, sistemas de mensajes, sistemas de telemetría, CI/CD, etc.
- **Multiple Workload Types:** soporte para diferentes tipos de cargas como deployments, jobs y custom resources.
- **Vendor-Agnostic:** soporte para disparadores en diferentes cloud providers y productos.
- **Azure Functions Support:** compatible con las funciones de Azure en Kubernetes en producción.
- **Integration with prometheus:** El adaptador de métricas de KEDA expone métricas de Prometheus que pueden ser *scrapeadas* en el puerto 9022 en /metrics.
- **Pause the scalation:** para ventanas de mantenimiento se puede pausar el autoescalado.



# Funcionalidades



# Instalación

# Instalación

⊘ KEDA necesita un cluster de Kubernetes version 1.16 o superior

- **Helm charts:** <https://keda.sh/docs/2.5/deploy/#helm>
- **Operator Hub:** <https://keda.sh/docs/2.5/deploy/#operatorhub>
- **YAML declarations:** <https://keda.sh/docs/2.5/deploy/#yaml>



# Instalación

## Archivos YAML - instalación del servidor de métricas

```
kubectl apply -f
```

```
https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

Necesitarás añadir el parámetro `--kubelet-insecure-tls` si no tienes certificados propios.



# Instalación

## Archivos YAML - instalación de keda

kubectl apply -f <https://github.com/kedacore/keda/releases/download/v2.8.0/keda-2.8.0.yaml>

kubectl delete -f <https://github.com/kedacore/keda/releases/download/v2.8.0/keda-2.8.0.yaml>





# Instalación - ScaledObject

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: mysql-scaledobject
  namespace: default
spec:
  advanced:
    horizontalPodAutoscalerConfig:
      behavior:
        scaleDown:
          stabilizationWindowSeconds: 60
        policies:
          - type: Pods
            value: 1
            periodSeconds: 3
        scaleUp:
          stabilizationWindowSeconds: 0
        policies:
          ...
```

```
pollingInterval: 5
cooldownPeriod: 5
minReplicaCount: 1
maxReplicaCount: 10
```

```
scaleTargetRef:
  apiVersion: apps/v1
  kind: Deployment
  name: example-workload
```

Deployment  
a escalar

```
triggers:
```

```
- type: mysql
```

```
metadata:
```

```
  queryValue: "1"
```

```
  activationQueryValue: "2"
```

```
  query: "SELECT CEIL(COUNT(*) / 2) FROM onlineusers
WHERE status='online'"
```

```
authenticationRef:
```

```
  name: keda-trigger-auth-mysql-secret
```

Evento SQL



# Instalación - ScaledObject

apiVersion: keda.sh/v1alpha1

kind: ScaledObject

metadata:

name: redis-so

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: **example-workload**

pollingInterval: 3

cooldownPeriod: 10

minReplicaCount: 1

maxReplicaCount: 10

advanced:

horizontalPodAutoscalerConfig:

Deployment  
a escalar

behavior:

scaleDown:

stabilizationWindowSeconds: 10

policies:

- type: Pods

value: 1

periodSeconds: 3

scaleUp:

stabilizationWindowSeconds: 0

policies:

- type: Pods

value: 1

periodSeconds: 3

triggers:

- type: redis

metadata:

Redis server que  
se monitoriza

address: redis.keda-demo.svc.cluster.local:6379

listName: mylist

listLength: "2"



# Instalación - ScaledJob

apiVersion: keda.sh/v1alpha1

kind: ScaledJob

metadata:

name: redis-job

spec:

**jobTargetRef:**

parallelism: 1

completions: 1

activeDeadlineSeconds: 30

backoffLimit: 6

template:

spec:

containers:

- image: alpine:3.13.5

name: alpine

command: ['echo', 'hello world']

restartPolicy: Never

pollingInterval: 3

successfulJobsHistoryLimit: 5

failedJobsHistoryLimit: 5

maxReplicaCount: 10

scalingStrategy:

strategy: "accurate"

triggers:

- type: redis

**metadata:**

**address: redis.keda-demo.svc.cluster.local:6379**

**listName: myotherlist**

**listLength: "1"**

Redis server que  
se monitoriza



Demo

Links de interés

# Links de interés

Web oficial: <https://keda.sh>

Código fuente: <https://github.com/kedacore/keda>

Scalers: <https://github.com/kedacore/keda/tree/main/pkg/scalers>

Getting started: <https://github.com/kedacore/keda#getting-started>

Kubecon EU 2021: [https://www.youtube.com/watch?v=H5eZEq\\_wqSE](https://www.youtube.com/watch?v=H5eZEq_wqSE)

Alibaba Cloud usa KEDA para autoescalar aplicaciones:

<https://www.cncf.io/blog/2021/03/30/why-alibaba-cloud-uses-keda-for-application-autoscaling/>



# Links de interés

Ask an OpenShift Admin (Ep 85) | New and Improved Pod Autoscaling:

<https://youtu.be/4cExHjdKtkY>

HPA: <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>

Lab howto: <https://gist.github.com/ptrnull/6a0370a674f10d0f51964f4014bcf093>

Lab files: <https://github.com/ptrnull/keda.sh-demo/tree/main/keda-mysql>

Deck file:

<https://github.com/ptrnull/presentations/blob/main/KEDA.sh%20Autoescalado%20de%20Kubernetes%20basado%20en%20eventos.pdf>



Gracias!

