Kubernetes Event-driven Autoscaling

Application autoscaling made simple

Pedro Ibáñez Requena
Ecosystem Field Engineering Telco 5g Team

Index

- What is KEDA?
- How it works?
- Features
- Setup
- Demo
- References

What is KEDA?

What is KEDA?

KEDA is a Kubernetes-based **Event Driven Autoscaler**. With KEDA, you can drive the scaling of any container in Kubernetes based on the number of events needing to be processed.

KEDA is a **single-purpose** and **lightweight** component that can be added into any Kubernetes cluster (can scale to 0 to save resources).

KEDA works alongside standard Kubernetes components like the Horizontal Pod Autoscaler and can extend functionality without overwriting or duplication. With KEDA you can explicitly map the apps you want to use event-driven scale, with other apps continuing to function. This makes KEDA a **flexible and safe** option to run alongside any number of any other Kubernetes applications or frameworks.

CNCF Sandbox project.

How it works?

How it works?

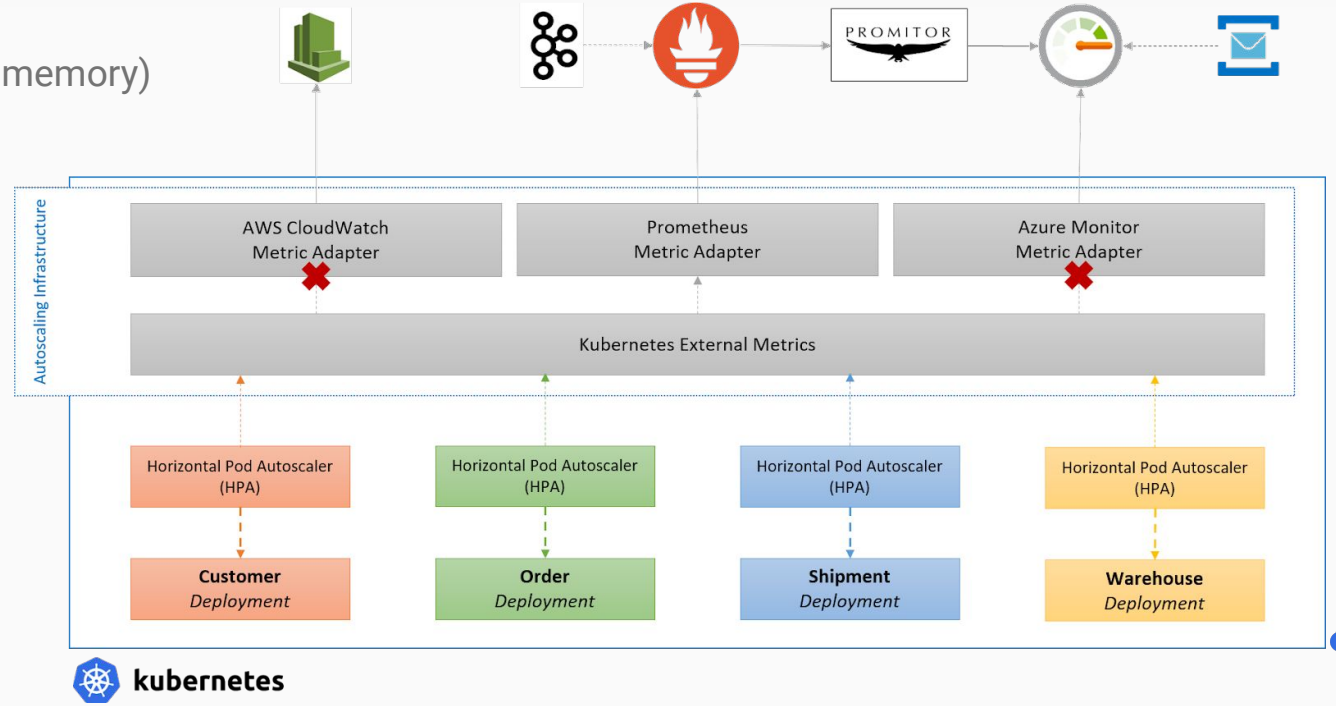
Agent — KEDA activates and deactivates Kubernetes Deployments to scale to and from zero on no events. This is one of the primary roles of the keda-operator container that runs when you install KEDA.

Metrics — KEDA acts as a Kubernetes metrics server that exposes rich event data like queue length or stream lag to the Horizontal Pod Autoscaler to drive scale out. It is up to the Deployment to consume the events directly from the source. This preserves rich event integration and enables gestures like completing or abandoning queue messages to work out of the box. The metric serving is the primary role of the keda-operator-metrics-apiserver container that runs when you install KEDA.

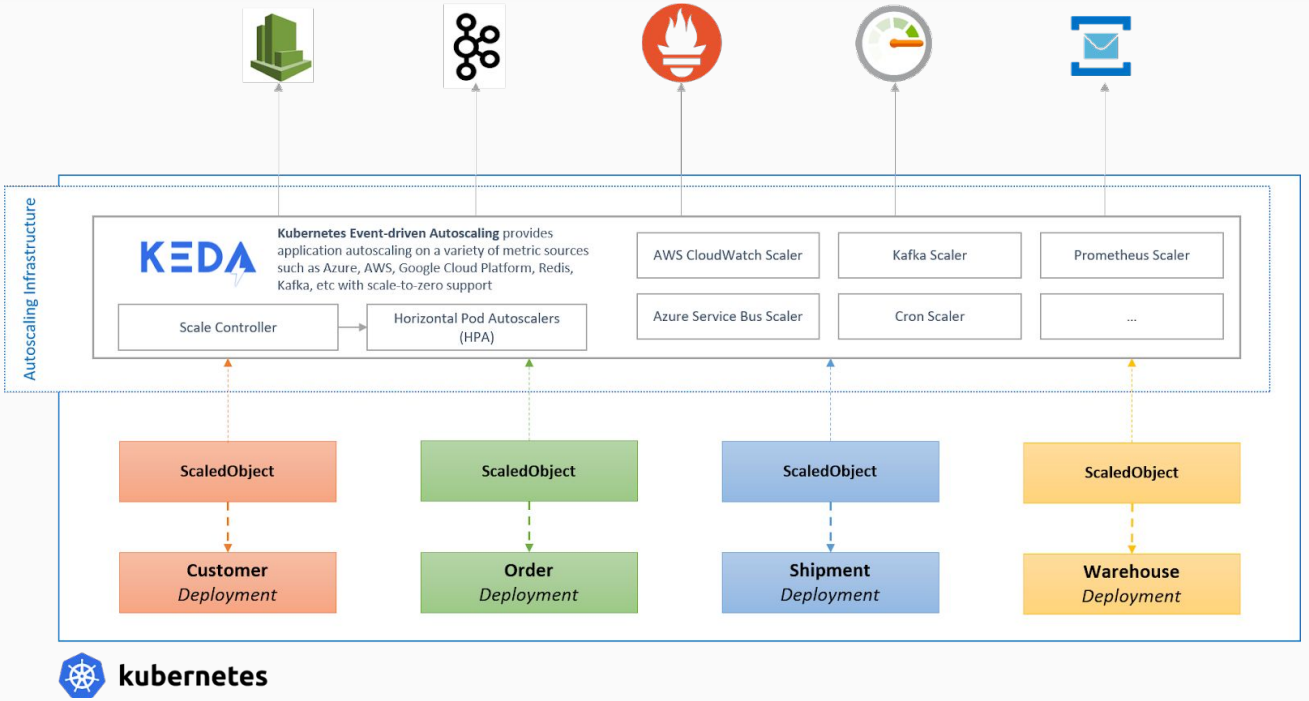
Scalers — can both detect if a deployment should be activated or deactivated, and feed custom metrics for a specific event source.

How it works?

HPA limitations (CPU & memory)



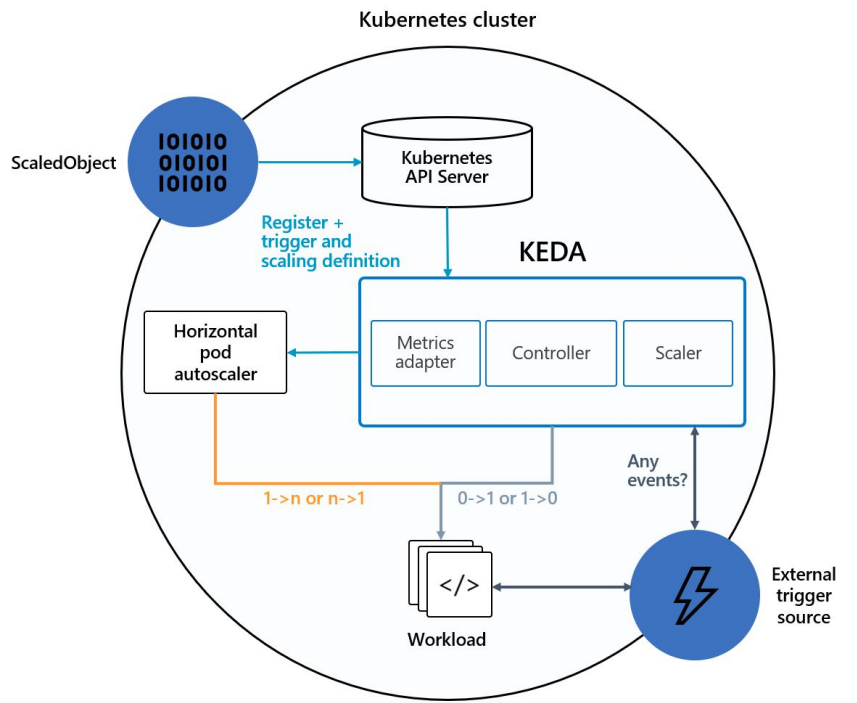
How it works?



kubernetes

KEDA

How it works?


Kubernetes cluster
ScaledObject
Kubernetes API Server
Register + trigger and scaling definition
KEDA
Horizontal pod autoscaler
Metrics adapter
Controller
Scaler
Any events?
1->n or n->1
0->1 or 1->0
</>
Workload
External trigger source

Features

Features

- **Event-driven**: scales an event-driven application
- **Simple autoscaling**: rings rich scaling to every workload in a Kubernetes cluster
- **Built-in Scalers**: out-of-the-box scalers for various vendors, databases, messaging systems, telemetry systems, CI/CD, etc.
- **Multiple Workload Types**: support for variety of workload types such as deployments, jobs & custom resources with /scale sub-resource
- **Vendor-Agnostic**: support for triggers across variety of cloud providers & products
- **Azure Functions Support**: run and scale the Azure Functions on Kubernetes in production workloads
- **Integration with prometheus**: The KEDA Metrics Adapter exposes Prometheus metrics which can be scraped on port 9022 at /metrics

Features

Currently available scalers for KEDA

- ActiveMQ Artemis
- Apache Kafka
- AWS CloudWatch
- AWS Kinesis Stream
- AWS SQS Queue
- Azure Blob Storage
- Azure Event Hubs
- Azure Log Analytics
- Azure Monitor
- Azure Pipelines
- Azure Service Bus
- Azure Storage Queue
- Cassandra
- CPU
- Cron
- Elasticsearch
- External
- External Push
- Google Cloud Platform Pub/Sub
- Graphite
- Huawei Cloudeye
- IBM MQ
- InfluxDB
- Kubernetes Workload
- Liiklus Topic
- Memory
- Metrics API
- MongoDB
- MSSQL
- MySQL
- NATS Streaming
- OpenStack Metric
- OpenStack Swift
- PostgreSQL
- Prometheus
- RabbitMQ Queue
- Redis Lists
- Redis Lists (supports Redis Cluster)
- Redis Lists (supports Redis Sentinel)
- Redis Streams
- Redis Streams (supports Redis Cluster)
- Redis Streams (supports Redis Sentinel)
- Selenium Grid Scaler
- Solace PubSub+ Event Broker

Your own here

Setup

Setup

🚫 KEDA requires Kubernetes cluster version 1.16 and higher

- **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

Setup

YAML declarations - installation of keda

kubectl apply -f https://github.com/kedacore/keda/releases/download/v2.5.0/keda-2.5.0.yaml

kubectl delete -f https://github.com/kedacore/keda/releases/download/v2.5.0/keda-2.5.0.yaml

Setup

YAML declarations - installation of metrics server

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

If necessary edit the deployment and add the parameter --kubelet-insecure-tls

Setup - ScaledObject Example

```
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:
          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:
        address: redis.keda-demo.svc.cluster.local:6379
        listName: mylist
        listLength: "2"
```

Deployment to scale

Redis server to monitor

Setup - ScaledJob Example

```
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 to monitor

Demo

References

References

Official web: https://keda.sh

Source code: 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

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

References

Near term KEDA Tech Preview - What's Next in OpenShift Q4CY2021:
https://docs.google.com/presentation/d/1yYxjIw6Xwy4BWmXMwixrHGfMjj4xdZXJMIbEuS61zVc/edit#slide=id.gb81442103c_0_2462

Thanks!