Introduction to KEDA

Kubernetes Event-Driven Autoscaling

Kunal Das

Date: 18-05-2024

Venue: Kubetools Day 2.0 "Show-n-Tell“, Bengaluru

Kunal Das

Sr DevOps Engineer @ Cyncly

HashiCorp Certified Terraform Associate

Microsoft Certified DevOps Expert

Agenda

- Overview of KEDA
- Key Features of KEDA
- How KEDA Works
- Use Cases
- Hands-on Demo
- Challenges & Future
- Q&A Session

What is KEDA?

Definition of KEDA (Kubernetes Event-Driven Autoscaling)

Brief history and background

Importance of autoscaling in Kubernetes

What is KEDA?

Definition of KEDA

Brief history and background

Importance of autoscaling in Kubernetes

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. 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.

Ref : https://keda.sh/

What is KEDA?

Definition of KEDA

Brief history and background

Importance of autoscaling in Kubernetes

KEDA was created through a collaborative effort between Microsoft and Red Hat. The project was initiated in 2019 and has since evolved significantly.

Accepted as a Cloud Native Computing Foundation (CNCF) sandbox project in 2020.

The community started growing as well, the project governance matured and KEDA became a CNCF Incubation project in 2021.

August 22, 2023 CNCF has accepted to move KEDA to a CNCF Graduated project next to projects such as Kubernetes, Prometheus, Istio and others!

Ref : https://keda.sh/

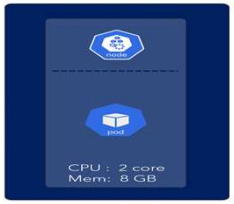
What is KEDA?

Definition of KEDA

HPA

- ✓ Automatically adjusts pod replicas.
- ✓ Scales based on CPU utilization.
- ✓ Enhances application responsiveness.
- ✓ Optimizes resource utilization

Brief history and background



VPA

- ✓ Adjusts CPU and memory resources.
- ✓ Resource right-sizing.
- ✓ Supports 1,000 pods per cluster
- ✓ Stores up to eight days of historical data.

Importance of autoscaling in Kubernetes

KEDA

- ✓ Enables event-driven scaling.
- ✓ Supports multiple event sources like RabbitMQ, Azure Functions etc.
- ✓ Provides fine-grained scaling based on custom metrics.
- ✓ Reduces resource consumption during idle periods by scaling to zero.

Ref : https://keda.sh/

What is KEDA?

Definition of KEDA

Brief history and background

Importance of autoscaling in Kubernetes

- ✓ **Elastic responsiveness:** Autoscaling adapts to resource demand, ensuring optimal performance.
- ✓ **Efficient resource utilization:** Scales up or down based on CPU and memory needs.
- ✓ **Application reliability:** Maintains consistent service during traffic spikes or lulls.
- ✓ **Cost optimization:** Avoids overprovisioning by dynamically adjusting resources.
- ✓ **Streamlined management:** Automates scaling, freeing administrators from manual adjustments.

Ref : https://keda.sh/

Features

Autoscaling Made Simple

Bring rich scaling to every workload in your Kubernetes cluster

Event-driven

Intelligently scale your event-driven application

Built-in Scalers

Catalog of 50+ built-in scalers for various cloud platforms, databases, messaging systems, telemetry systems, CI/CD, and more

Multiple Workload Types

Support for variety of workload types such as deployments, jobs & custom resources with `/scale` sub-resource

Reduce environmental impact

Build sustainable platforms by optimizing workload scheduling and scale-to-zero

Extensible

Bring-your-own or use community-maintained scalers

Vendor-Agnostic

Support for triggers across variety of cloud providers & products

Azure Functions Support

Run and scale your Azure Functions on Kubernetes in production workloads

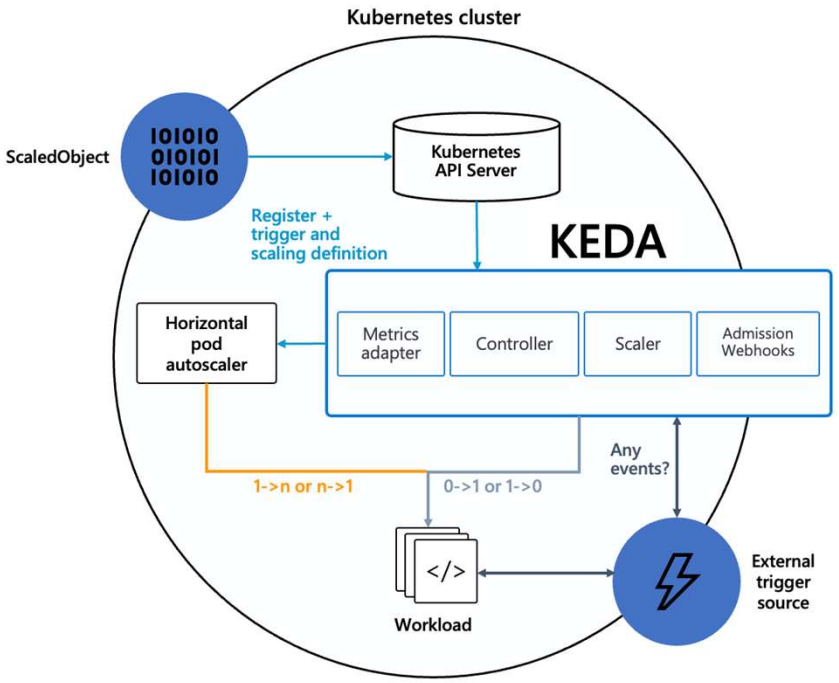
Ref : https://keda.sh/
https://keda.sh/docs/2.14/scalers/

How KEDA Works?

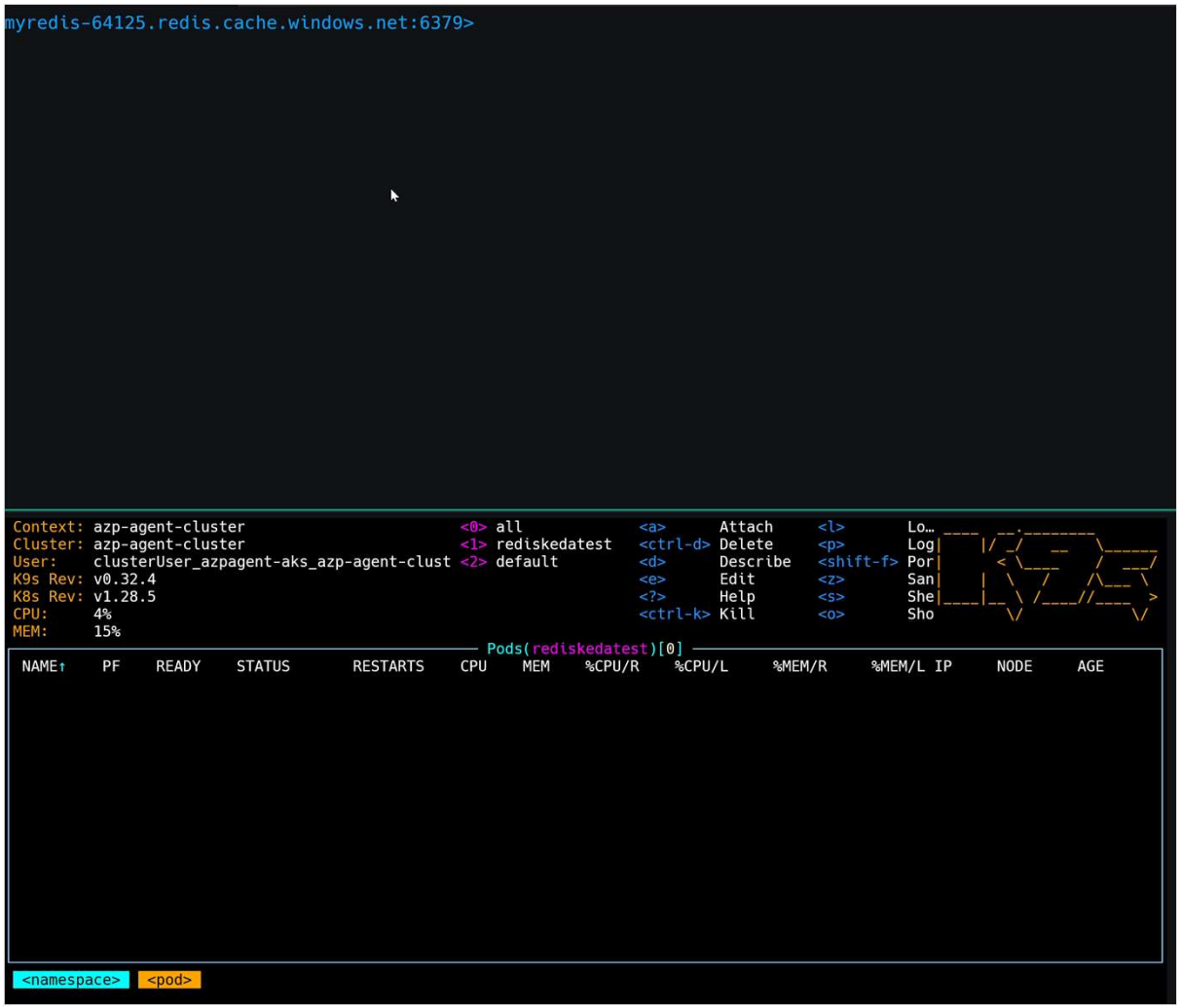
- scaledobjects.keda.sh
- scaledjobs.keda.sh
- triggerauthentications.keda.sh
- clustertriggerauthentications.keda.sh

Ref : https://keda.sh/

KEDA Architecture



Hands-on Demo



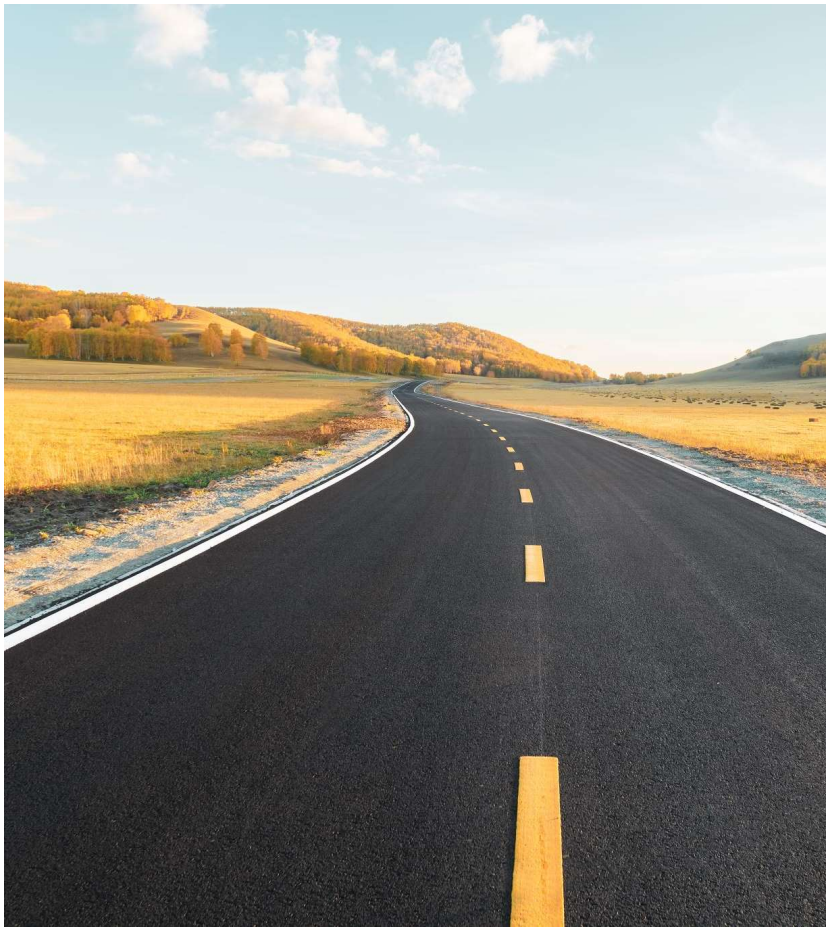
Ref : https://github.com/kedacore/samples

Challenges and Limitations

- Cold Start Latency
- Limited Scaler Support
- Configuration Overhead

Future of KEDA

- ✓ Enhanced Scaler Support
- ✓ PredictKube
- ✓ Improved Monitoring



Ref : https://github.com/kedacore/keda/blob/main/ROADMAP.md | https://dysnix.com/predictkube

Thank You

Reach out to me in case any doubts, suggestions or feedback !!