



APACHECON
North America



Event-driven autoscaling through Apache Kafka Source, KEDA, and Knative Integration



Daniel Oh



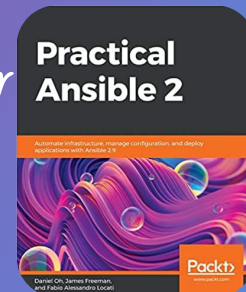
- *Developer Advocate at Red Hat*
 - *Cloud Native Runtimes*
 - *Serverless, Service Mesh, and GitOps Practices*
- *CNCF Ambassador*
- *Advisory Board Member of Global Skill Development Council*
- *Opensource.com Correspondents*
- *Public Speaker & Published Author*



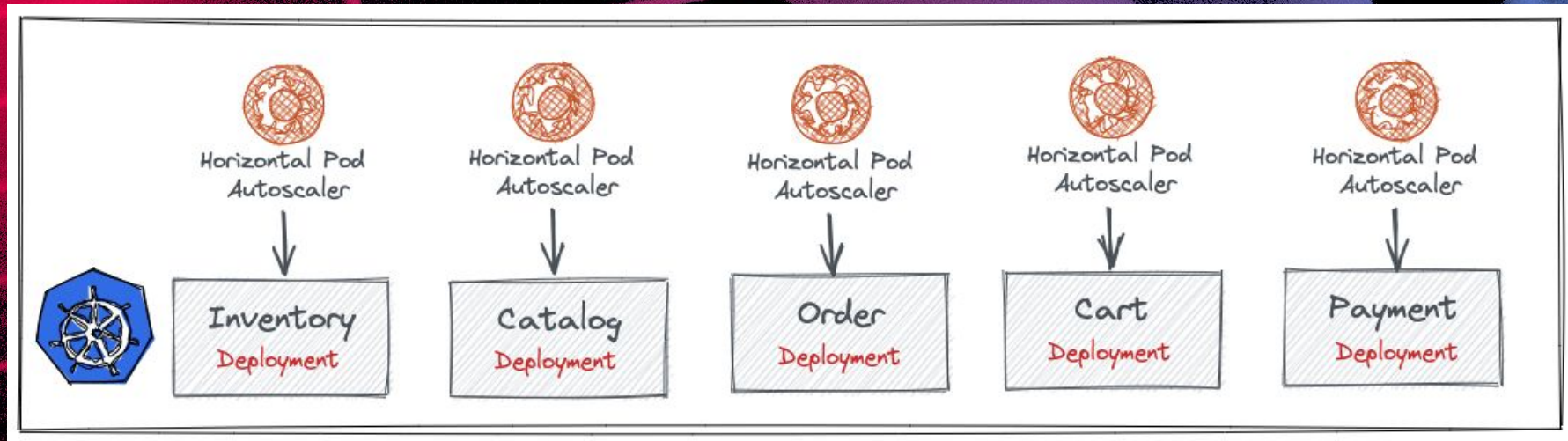
 @danieloh30

 bit.ly/danielohtv

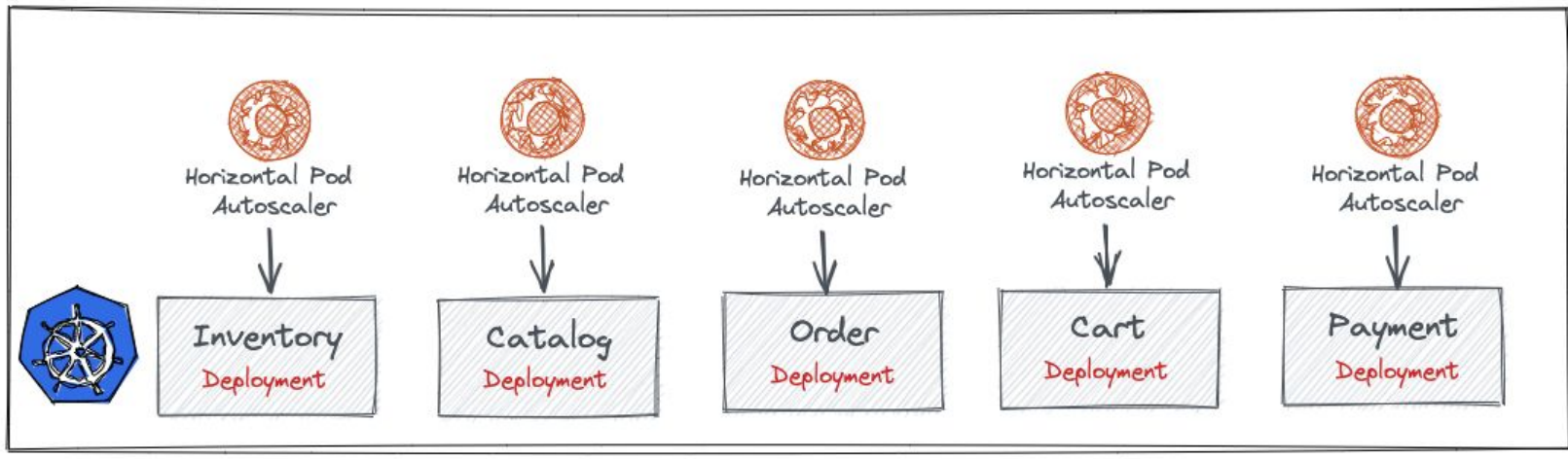
 danieloh30

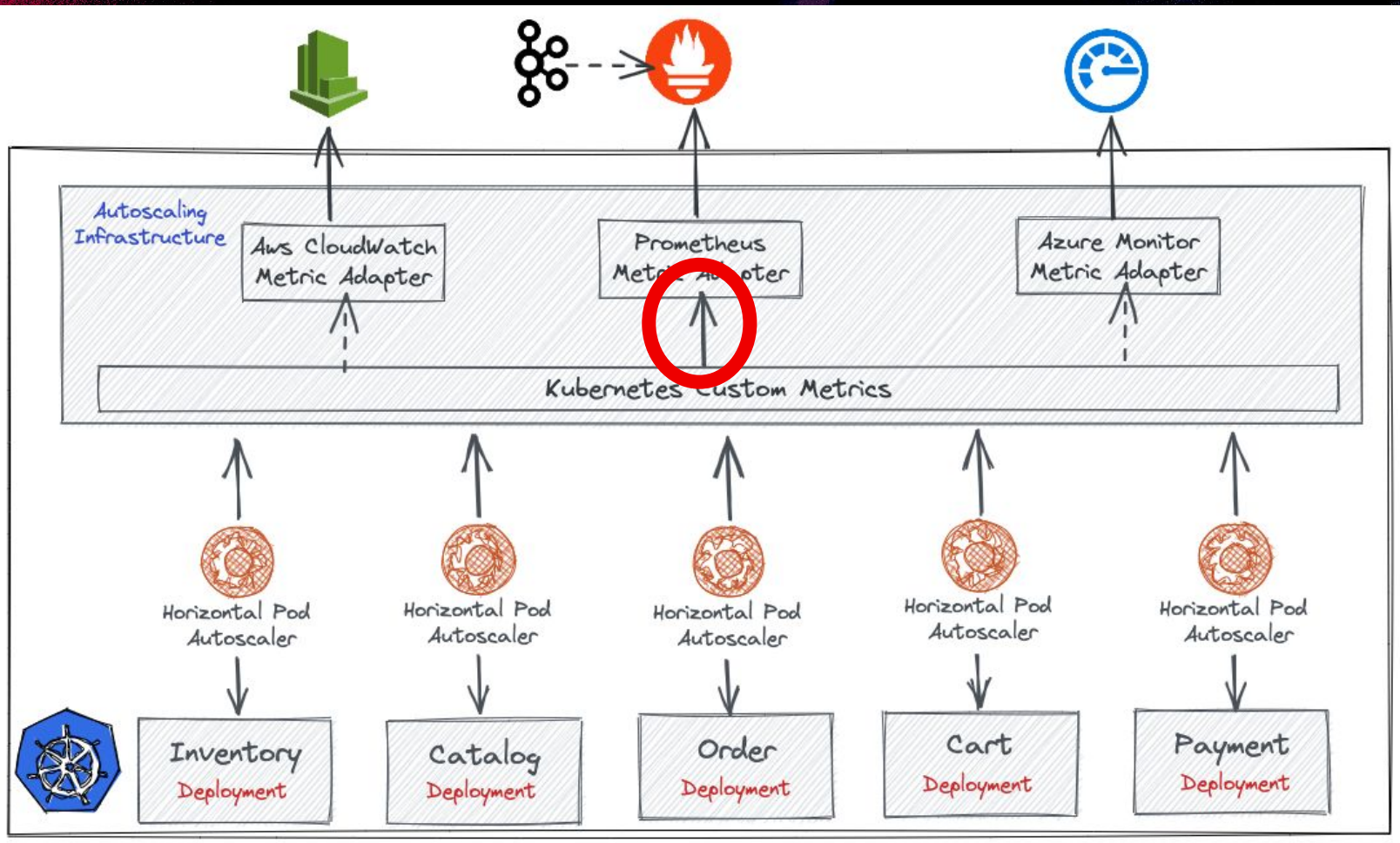



AUTOSCALING ARCHITECTURE ON KUBERNETES



AUTOSCALING ARCHITECTURE ON KUBERNETES W/ EXTERNAL SERVICES







KEDA



KEDA ⚡

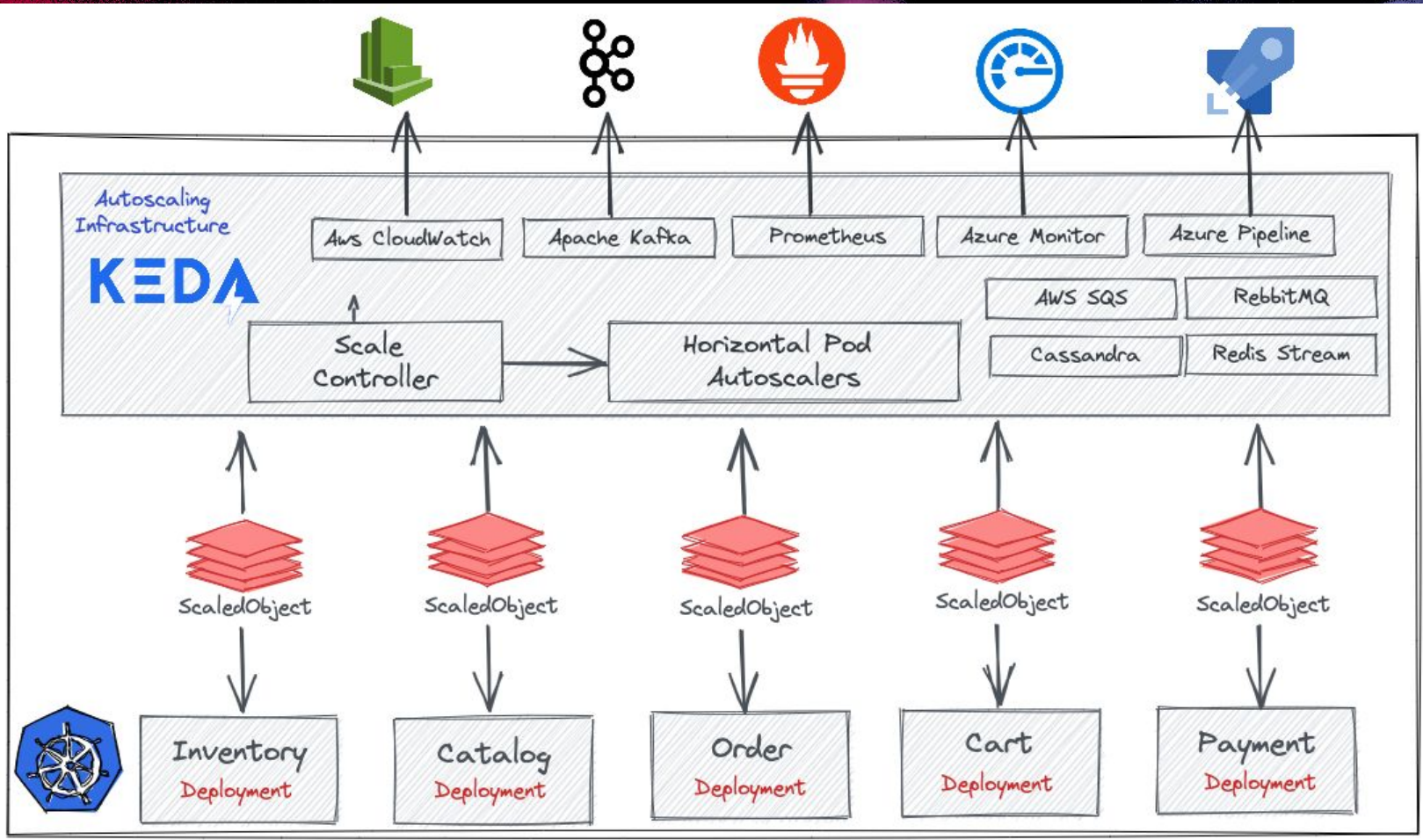
- Project aims to make Kubernetes Event Driven Autoscaling dead simple
- Started as a partnership between Red Hat and Microsoft (Feb 2019)
- Donated into CNCF as a Sandbox project (Mar 2020)
- KEDA 2.0 brought major redesign (Nov 2020)
- Promoting to CNCF Incubation project (Aug 2021)
- KEDA 2.8 has been released recently (Sep 2022)
- <https://keda.sh>



KEDA

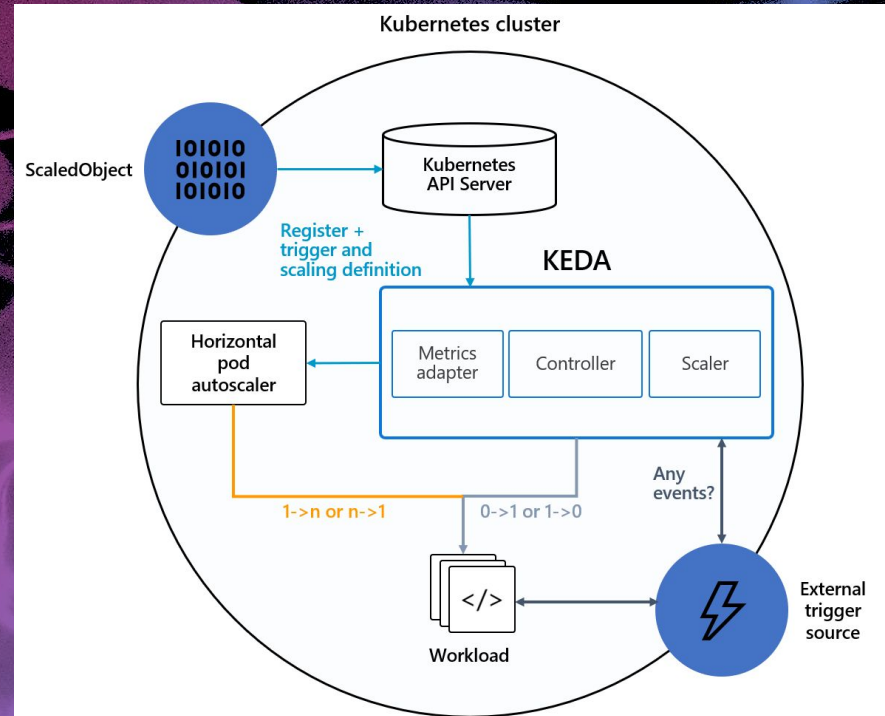
- *Automatically scale Kubernetes Deployments, Jobs & Custom Resources*
- *Provides 56+ built-in scalers, but users can build own external scalers*
- *Kafka, Prometheus, RabbitMQ, AWS services, Azure Services,...*
- *Scale resources based on events in the target scalers, eg. messages in Kafka topic*
- *KEDA does not manipulate the data, just scales the workload*
- *Installation through OLM Operator or Helm*





HOW DOES KEDA WORKS?

- *KEDA is built on top of Kubernetes*
- *Use ScaledObject/ScaledJob to define scaling metadata*
- *Manages workloads to scale to 0*
- *Registers itself as k8s Metric Adapter*
- *Provides metrics for Horizontal Pod Autoscaler (HPA) to scale on*



SCALED OBJECT

- *Can target Deployment, StatefulSet or Custom Resource with scale*
- *Multiple scalers can be defined as triggers for the target workload*
- *User can specify HPA related settings to tweak the scaling behavior*

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: example-so
spec:
  scaleTargetRef:
    name: example-deployment
  minReplicaCount: 0
  maxReplicaCount: 100
  triggers:
    - type: kafka
      metadata:
        bootstrapServers: kafka.svc:9092
        consumerGroup: my-group
        topic: test-topic
        lagThreshold: '5'
```



The background of the image is a grid of speakers, arranged in four vertical columns. The top two rows of each column consist of small, square speakers. The middle three rows consist of larger, circular speakers. The bottom row consists of seven large, circular speakers. The entire grid is illuminated with a color gradient that transitions from a deep red on the left side to a dark blue on the right side. The text "How about Serverless Autoscaling?" is centered over the middle section of the speaker array in a white, handwritten-style font.

*How about Serverless
Autoscaling?*



K

n



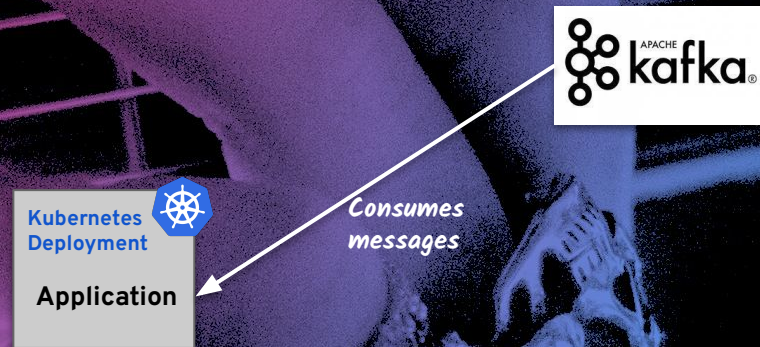
- Operates on standard k8s resources
- Can scale existing deployed apps
- Pull based approach
- Doesn't manage data delivery
- K8s Horizontal Pod Autoscaler (HPA)
- Focus is on event driven autoscaling
- Operates on Knative Service
- Existing apps must be converted
- Push based approach
- Manages data delivery (Eventing)
- Knative Autoscaler
- Demand-based autoscaling (HTTP)



USE CASE #1

APPLICATION CONSUMING MESSAGES FROM KAFKA TOPIC

- Application is deployed as standard Kubernetes Deployment
- Can be autoscaled only via standard k8s HPA: CPU & Memory
- No event-driven autoscaling



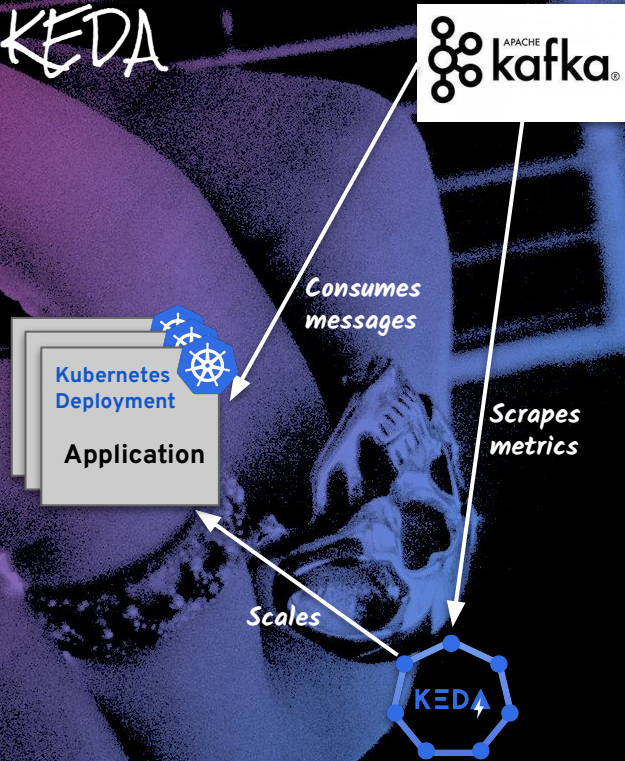
DEMO



USE CASE #2

REDESIGNED TO UTILIZE KEDA

- Application remains the same and is being deployed the same way
- Event-driven autoscaling enabled through KEDA



DEMO



USE CASE #3

REDESIGNED TO UTILIZE KNATIVE

- Application needs to be rewritten from Kafka consumer to CloudEvents consumer
- Application needs to be redeployed as Knative Service
- Needs Knative Eventing Kafka Source
- Event-driven autoscaling enabled through Knative Autoscaler



DEMO





Event-driven

*How about Serverless
Autoscaling?*

Integrate KEDA with Knative

KEDA



KEDA AND KNATIVE INTEGRATION

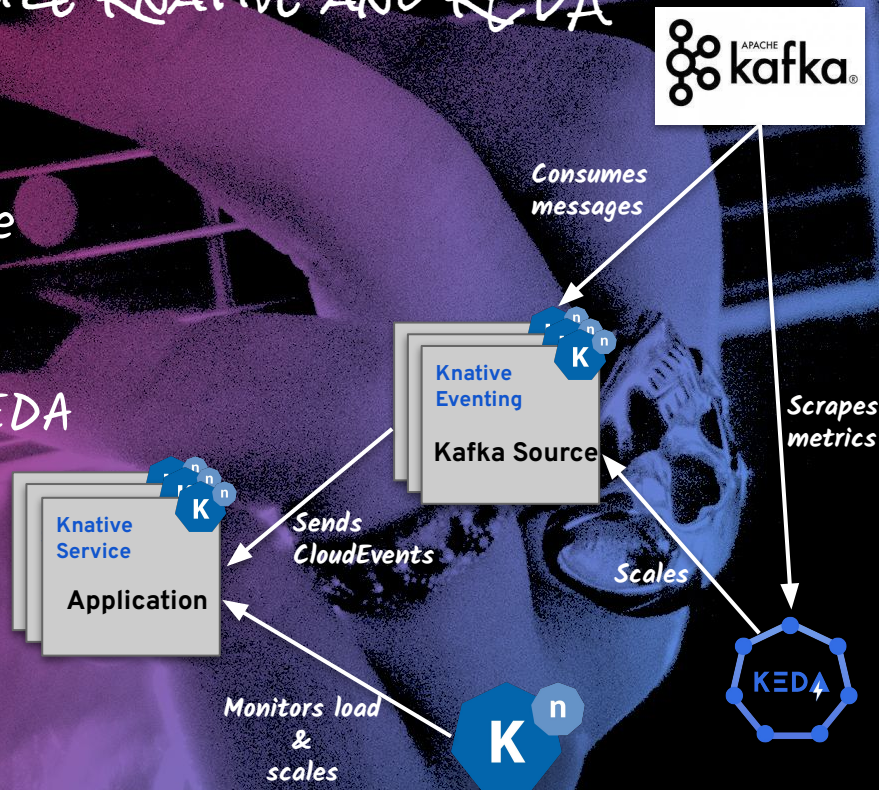
- *KEDA can be used to autoscale Knative Eventing Infrastructure*
 - *Knative Eventing Sources, Channels*
 - *Autoscaling allows infrastructure to handle higher loads or save resources (by scaling to 0) when idle*
- *KEDA could be potentially used to scale Knative Service, in case users don't want to utilize Knative Eventing for event driven workloads (currently not implemented)*



USE CASE #4

REDESIGNED TO UTILIZE KNative AND KEDA

- Application deployed as Knative Service and autoscaled by Knative
- Knative Eventing Infrastructure - Kafka Source is autoscaled by KEDA



DEMO



KNATIVE-SANDBOX/EVENTING-AUTOSCALER-KEDA

- *KafkaSource*
- *AWS SQS Source*
- *Redis Stream Source*
- *RabbitMQ Broker*

```
metadata:  
  annotations:  
    autoscaling.knative.dev/class: keda.autoscaling.knative.dev  
    autoscaling.knative.dev/minScale: "0"  
    autoscaling.knative.dev/maxScale: "5"  
    keda.autoscaling.knative.dev/pollingInterval: "30"  
    keda.autoscaling.knative.dev/cooldownPeriod: "30"  
  
  # Kafka Source  
    keda.autoscaling.knative.dev/kafkaLagThreshold: "10"  
  
  # AWS SQS Source  
    keda.autoscaling.knative.dev/awsSqsQueueLength: "5"  
  
  # Redis Stream Source  
    keda.autoscaling.knative.dev/redisStreamPendingEntriesCount: "5"
```



TAKEAWAYS







- *KEDA - Kubernetes event driven autoscaling dead simple*
- *KEDA - pull model vs. Knative - push model*
- *KEDA - Standard Kubernetes resources vs. Knative Service*
- *KEDA can autoscale Knative Eventing Infrastructure*
 - *knative-sandbox/eventing-autoscaler-keda*



bit.ly/danielohtv



KUBERNETES LEARN BY EXAMPLE ▶ PLAY ALL

 KUBERNETES LEARN BY EXAMPLE #9 Persistent Volumes 10:31	 KUBERNETES LEARN BY EXAMPLE #8 ConfigMaps 6:45	 KUBERNETES LEARN BY EXAMPLE #7 Managing Secrets 7:12	 KUBERNETES LEARN BY EXAMPLE #6 DaemonSet 7:00	 KUBERNETES LEARN BY EXAMPLE #5 StatefulSets 8:54	 KUBERNETES LEARN BY EXAMPLE #4 Deployment and ReplicaSet 11:44
--	--	---	--	---	---

Persistent Volumes - Learn by Example [9]
Daniel Oh
11 views • 1 day ago

ConfigMaps - Learn by Example [8]
Daniel Oh
124 views • 2 weeks ago







Managing Secrets - Learn by Example [7]
Daniel Oh
115 views • 1 month ago

DaemonSet - Learn by Example [6]
Daniel Oh
110 views • 1 month ago

StatefulSets - Learn by Example [5]
Daniel Oh
137 views • 2 months ago

Deployment and ReplicaSet - Learn by Example [4]
Daniel Oh
125 views • 2 months ago

QUARKUS ▶ PLAY ALL

 Microsweeper Quarkus on Red Hat OpenShift Service on AWS 19:21	 Build your first Java Serverless Function using Quarkus Quick start 9:55	 Microsweeper Quarkus on Azure Red Hat OpenShift 18:57	 Getting Started with Reactive Programming with Kotlin on... 12:55	 Cloud Native Buildpacks with Quarkus 5:09	 Extend Service Discovery with Quarkus and Stork 11:16
--	---	---	---	---	---

Microsweeper Demo with Quarkus on Red Hat...
Daniel Oh
5 views • 6 minutes ago

Build your first Java Serverless Function using...
Daniel Oh
74 views • 5 days ago



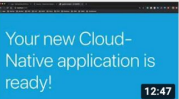

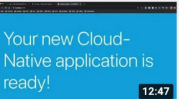
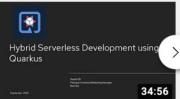
Microsweeper Demo with Quarkus on Azure Red Hat...
Daniel Oh
119 views • 11 days ago

Getting Started with Reactive Programming with Kotlin on...
Daniel Oh
125 views • 3 weeks ago

Cloud Native Buildpacks with Quarkus
Daniel Oh
141 views • 1 month ago

Extend Service discovery with Quarkus and Stork
Daniel Oh
157 views • 1 month ago

SERVERLESS & FUNCTION ▶ PLAY ALL

 Build your first Java Serverless Function using Quarkus Quick start 9:55	 Drag and Drop your Quarkus App on the Developer Sandbox 6:18	 Your new Cloud-Native application is ready! 12:47	 DEMO 15:41	 Your new Cloud-Native application is ready! 12:47	 Hybrid Serverless Development using Quarkus 34:56
--	---	---	--	---	---

Build your first Java Serverless Function using...
Daniel Oh
74 views • 5 days ago

Drag and Drop your Quarkus Serverless App on the...
Daniel Oh
6 views • 1 week ago

Deploying Multiple CloudNative Apps with...
Daniel Oh
1 view • 1 week ago

Quarkus builds your AWS Lambdas
Daniel Oh
1 view • 1 week ago

Deploying Multiple Cloud Native Apps with OpenShift...
Daniel Oh
1 view • 1 week ago

Hybrid Serverless Development using Quarkus
Daniel Oh
1 view • 1 week ago



Thank you!
Questions?

