KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KEDA -

Real Time And Serverless Scaling In Kubernetes

Zbynek Roubalik, Principal Software Engineer @ Red Hat
Jeff Hollan, Director of Product @ Snowflake

About us

- Jeff Hollan
  - Director of Product @ Snowflake
  - KEDA founding member / maintainer while at Microsoft (Director of product for serverless / containers)
  - https://twitter.com/jeffhollan
  - https://linkedin.com/in/jeffhollan
- Zbyněk Roubalík
  - Principal Software Engineer @ Red Hat
  - KEDA founding member / maintainer, Knative (TOC), Microsoft MVP
  - https://github.com/zroubalik
  - https://twitter.com/zroubalik
  - https://www.linkedin.com/in/zbynek-roubalik/

Agenda

1. What is KEDA?
2. KEDA project and community
3. Demo!
4. KEDA concepts and architecture
5. Future development
6. Q&A

A tale of two scalings….

Goal: Provide enough pizza for everyone at a KubeCon party 🍕

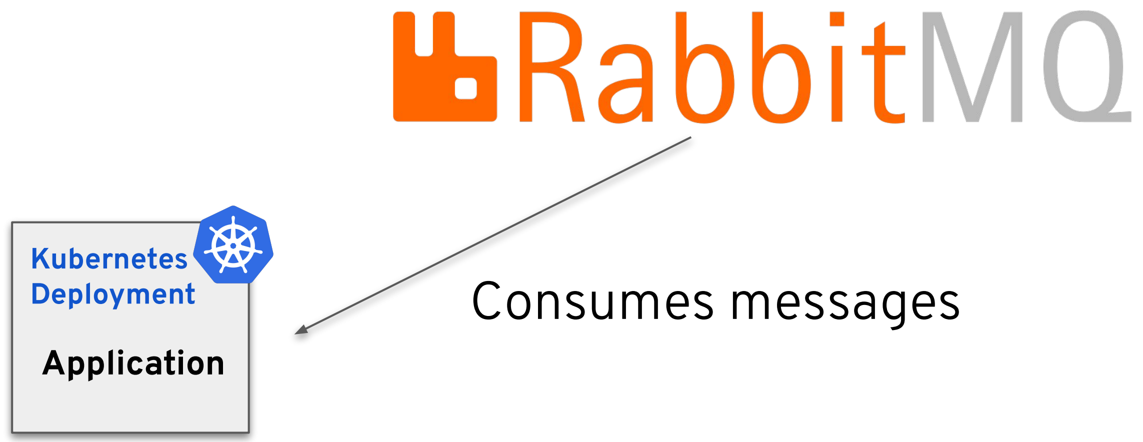
Strategy #1: Start with 1 pizza, see if it gets eaten, and add another

Strategy #2: Find out how many people accepted the invite. Bring the amount of pizzas estimated to feed the group

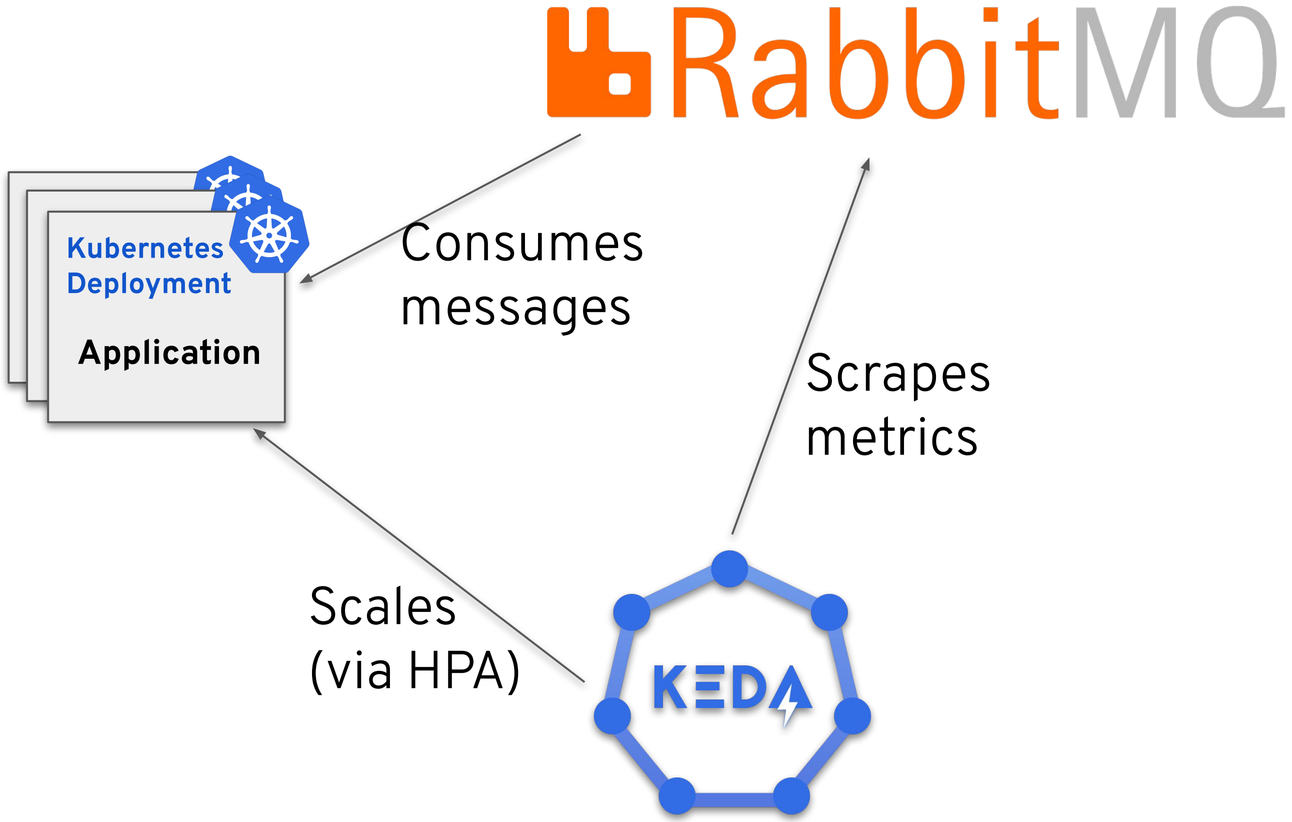
KEDA

- Project aims to make **K**ubernetes **E**vent **D**riven **A**utoscaling dead simple
- Allows you to scale any deployment resource or job based on **events**, not only on CPU / Memory
- 55+ integrated event sources (Prometheus, RabbitMQ, Kafka, SQS, PostgreSQL, …..)
- https://keda.sh

Hello KEDA Demo

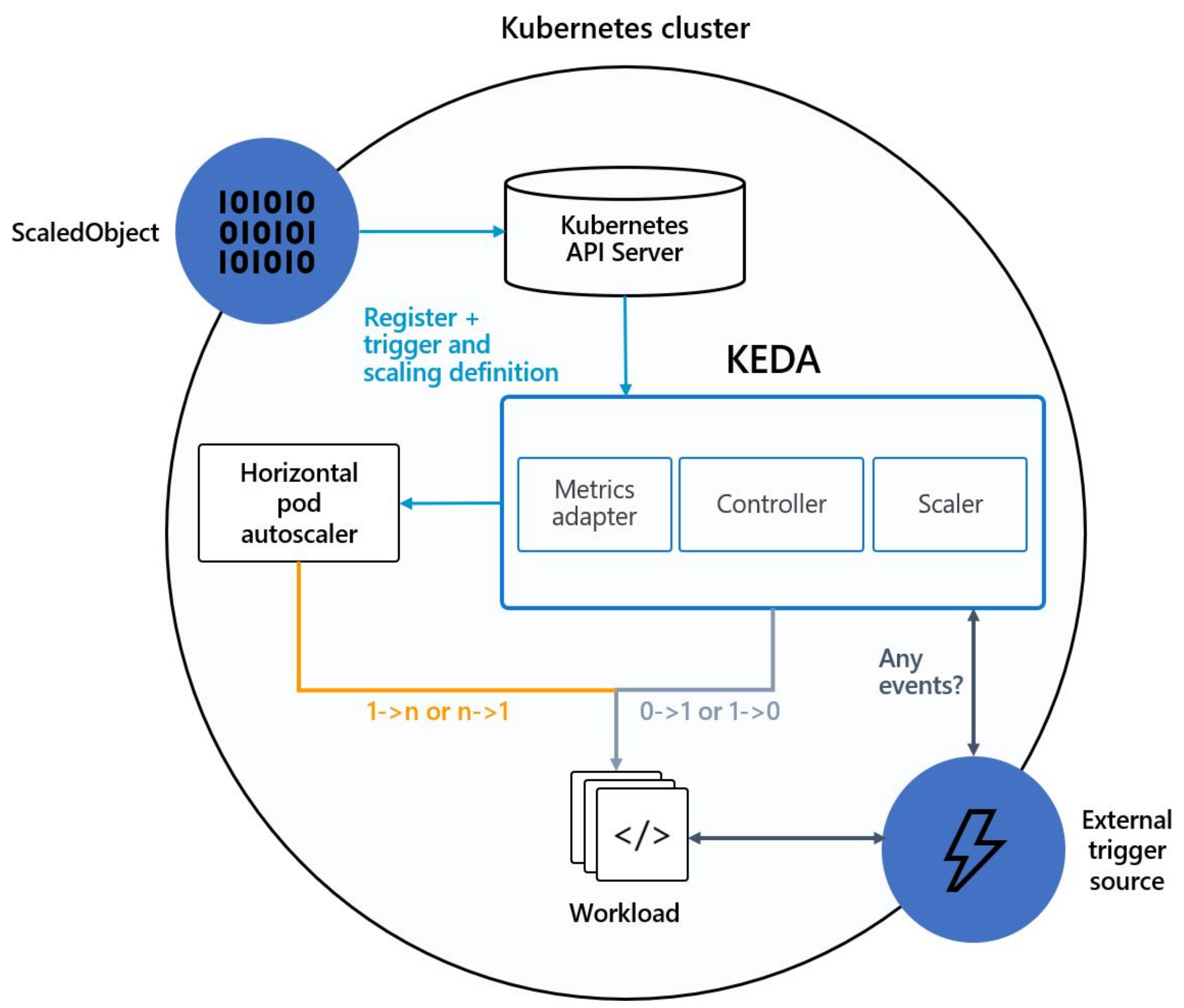


Hello KEDA Demo



Architecture

- KEDA is built on top of Kubernetes
- Use **ScaledObject/ScaledJob** to define scaling metadata
- Manages workloads to scale to 0
- Registers itself as Kubernetes Metric Adapter
- Publishes metrics to the Horizontal Pod Autoscaler (HPA) which makes most scale decisions



ScaledObject

- 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

```yaml
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'
```

ScaledJob

- Schedule **Kubernetes Job** based on events
- Useful option to handle **processing long running executions**

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
 name: example-sj
spec:
 jobTargetRef:
        ... # standard k8s Job definition

 maxReplicaCount: 100
 triggers:
 - type: kafka
   metadata:
        bootstrapServers: kafka.svc:9092
        consumerGroup: my-group
        topic: test-topic
        lagThreshold: '5'
```

KEDA vs Custom Metrics Adapter

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
 name: example-scaled-object
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'
```

```yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: web-prometheus-adapter
spec:
  scaleTargetRef:
     apiVersion: apps/v1
     kind: Deployment
     name: web
 minReplicas: 1
 maxReplicas: 50
 metrics:
 - type: Object
  object:
    metric:
     name: http_requests_per_second_per_pod
    describedObject:
     apiVersion: v1
     kind: Service
     name: web-prometheus-adapter-service
    target:
     type: Value
     value: 25
```

```yaml
rules:
 default: false
 custom:
 - seriesQuery: 'http_requests_received_total'
   resources:
     overrides:
       namespace: {resource: "namespace"}
       service: {resource: "service"}
   name:
     matches: ""
     as: "http_requests_per_second_per_pod"
   metricsQuery:
'max(irate(<<.Series>>{<<.LabelMatchers>>}[1
m])) by (<<.GroupBy>>)'
```

Advanced features

- Ability to specify **Fallback replicas count** - in case of problems
- Users can still tweak HPA settings if they want to (**scaling behavior**)
- Ability to **Pause autoscaling**
- KEDA exposes Prometheus metrics
- Users can extend KEDA implementing **External scalers** via gRPC interface or **Metrics API scalers** via Rest API.
- https://keda.sh/docs/2.8/concepts/scaling-deployments/#scaledobject-spec

Authentication

- Re-use trigger authentication across ScaledObject/ScaledJobs with **TriggerAuthentication** (namespaced) or **ClusterTriggerAuthentication**

- Out-of-the-box integration with sources such as:
  - Environment variables (on scale target)
  - Kubernetes secrets
  - Pod Identity (“No secret authentication” - Azure / AWS)
  - HashiCorp Vault
  - Azure Key Vault

What is next?

- Cache metrics values in KEDA Metrics Server
  - Reduced load on the source of metrics (eg. Prometheus server…)
  - Smoother autoscaling (apply AI/ML model to incoming metrics)
- Custom logic for evaluation when there are multiple triggers in ScaledObject
- CloudEvents integration
- OpenTelemetry integration
- Open interface for Predictive autoscaling

What is next?

- Environmental impact
  - CNCF TAG Environmental Sustainability
  - Integrate CO2 emission intensity, power consumption,... data into KEDA scaling decision
  - Carbon aware autoscaling
    - - POC & recording

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
    name: my-scaled-object
spec:
    scaleTargetRef:
        name:  my-resource
    minReplicaCount: 1
    maxReplicaCount: 100
    environmentalImpact:
        carbon:
          - carbonIntensity: 90
           allowedMaxReplicaCount: 81
          - carbonIntensity: 100
            allowedMaxReplicaCount: 53
          - carbonIntensity: 110
            allowedMaxReplicaCount: 44
    triggers:
         - type: kafka
          metadata:
            bootstrapServers: kafka.svc:9092
            consumerGroup: my-group
            topic: test-topic
            lagThreshold: '5'
```

Please scan the QR Code above to leave feedback on this session