

Explore Kubernetes Event-driven Autoscaling (KEDA) and the KEDA HTTP Add-on



Who am I?

Aaron Wislang

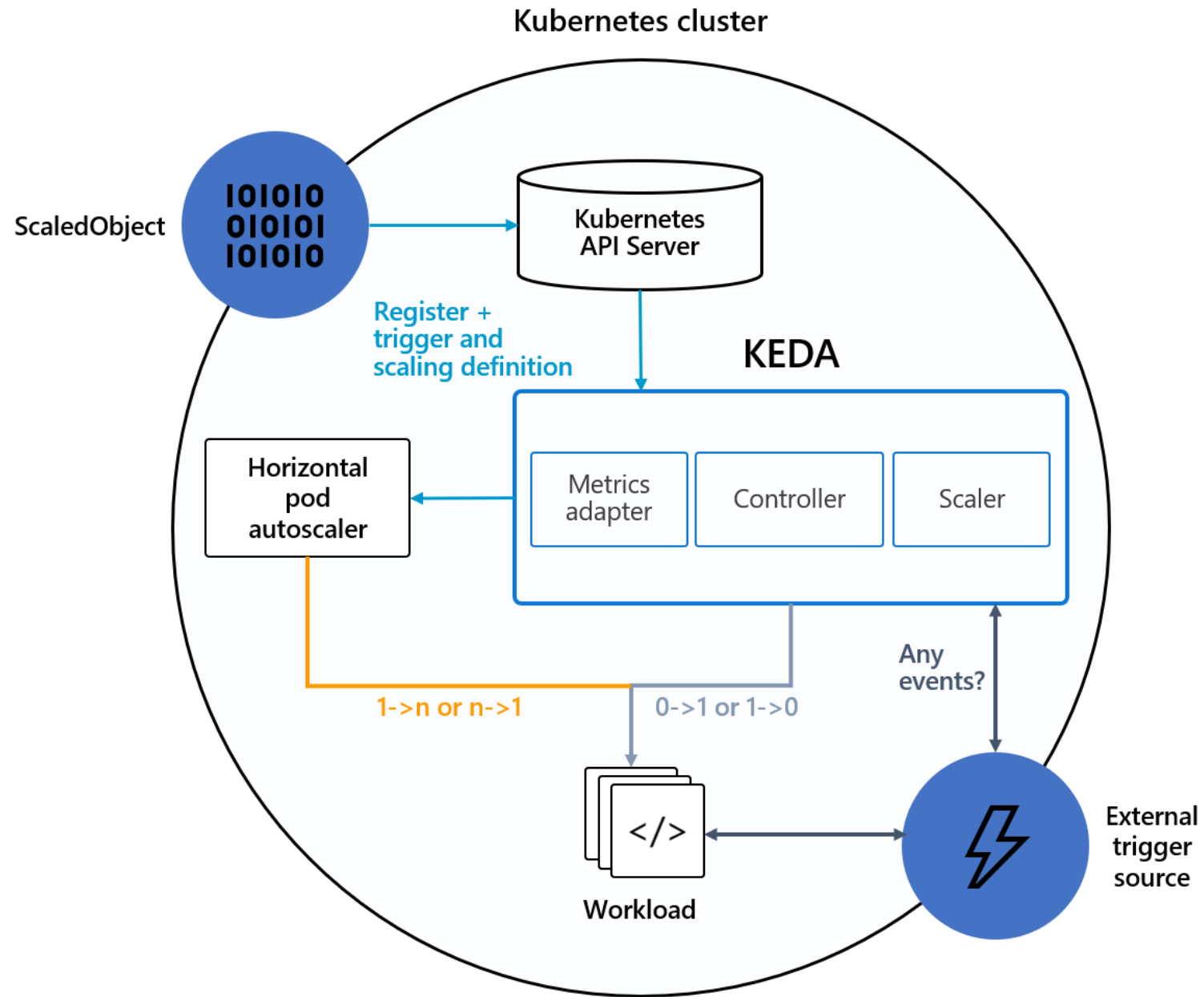
Cloud Advocate at Microsoft focusing on Open Source Engineering, Cloud Native & Go.

[@as_w](#) | aaronmsft.com/me/

What is KEDA?

[kedacore/keda](https://github.com/kedacore/keda) | keda.sh

- Announced in May 2019 by Microsoft and RedHat
- Became a CNCF Sandbox project in March 2020
- KEDA 2.0 released in November 2020
- Graduated to a CNCF Incubating project in August 2021
- KEDA now at 2.7.2 (May 2022)



KEDA & Kubernetes Resources

Deployment

StatefulSet

CustomResource (with `/scale` subresource)

Jobs

KEDA ScaledObject

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: {scaled-object-name}
spec:
  scaleTargetRef:
    apiVersion: {api-version-of-target-resource}
    kind: {kind-of-target-resource}
    name: {name-of-target-resource}
    envSourceContainerName: {container-name}
  pollingInterval: 30
  cooldownPeriod: 300
  idleReplicaCount: 0
  minReplicaCount: 1
  maxReplicaCount: 100
  ...
```

<https://keda.sh/docs/2.7/concepts/scaling-deployments/>

KEDA ScaledJob

```
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: {scaled-job-name}
spec:
  jobTargetRef:
    parallelism: 1 # max number of desired pods
    completions: 1 # desired number of successfully finished pods
    activeDeadlineSeconds: 600
    backoffLimit: 6
    template:
      # describes the job template
  # Optional
  pollingInterval: 30
  successfulJobsHistoryLimit: 100
  failedJobsHistoryLimit: 100
  envSourceContainerName: {container-name}
  maxReplicaCount: 100
  rolloutStrategy: default
```



Creating a Scaler

`desiredReplicas = ceil[currentReplicas * ( currentMetricValue / desiredMetricValue )]`

```
type Scaler interface {
    GetMetrics(
        ctx context.Context,
        metricName string,
        metricSelector labels.Selector
    )([]external_metrics.ExternalMetricValue, error)
    GetMetricSpecForScaling() []v2beta2.MetricSpec
    IsActive(ctx context.Context) (bool, error)
    Close() error
}

type PushScaler interface {
    Scaler
    Run(ctx context.Context, active chan<- bool)
}
```

<https://github.com/kedacore/keda/blob/main/CREATE-NEW-SCALER.md>

Scalers - Cron

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: cron-scaledobject
  namespace: default
spec:
  scaleTargetRef:
    name: my-deployment
  triggers:
    - type: cron
      metadata:
        timezone: Asia/Kolkata # From the IANA Time Zone Database.
        start: 30 * * * *      # Every hour on the 30th minute
        end: 45 * * * *        # Every hour on the 45th minute
        desiredReplicas: "10"
```

Scalers - PostgreSQL

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: airflow-worker
spec:
  scaleTargetRef:
    name: airflow-worker
  pollingInterval: 10
  cooldownPeriod: 300
  maxReplicaCount: 100
  triggers:
    - type: postgresql
      metadata:
        connectionFromEnv: AIRFLOW_CONN_AIRFLOW_DB
        query: "SELECT ceil(COUNT(*)::decimal / 16)
                FROM task_instance
                WHERE state='running' OR state='queued'"
        targetQueryValue: 1
```

See also: [Queues in PostgreSQL](#) by Thomas Munroe (@MengTangmu)

Scalers - External & ExternalPush

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: redis-scaledobject
  namespace: keda-redis-test
spec:
  scaleTargetRef:
    name: keda-redis-node
  triggers:
    - type: external
      metadata:
        scalerAddress: redis-external-scaler-service:8080
        address: REDIS_HOST
        password: REDIS_PASSWORD
        listName: mylist
        listLength: "5"
```

Implementing External Scalers

```
service ExternalScaler {  
    rpc IsActive(ScaledObjectRef) returns (IsActiveResponse) {}  
    rpc StreamIsActive(ScaledObjectRef) returns (stream IsActiveResponse) {}  
    rpc GetMetricSpec(ScaledObjectRef) returns (GetMetricSpecResponse) {}  
    rpc GetMetrics(GetMetricsRequest) returns (GetMetricsResponse) {}  
}
```

<https://keda.sh/docs/2.5/concepts/external-scalers/>

What is the KEDA HTTP Add-on

[kedacore/http-add-on](https://github.com/kedacore/http-add-on)

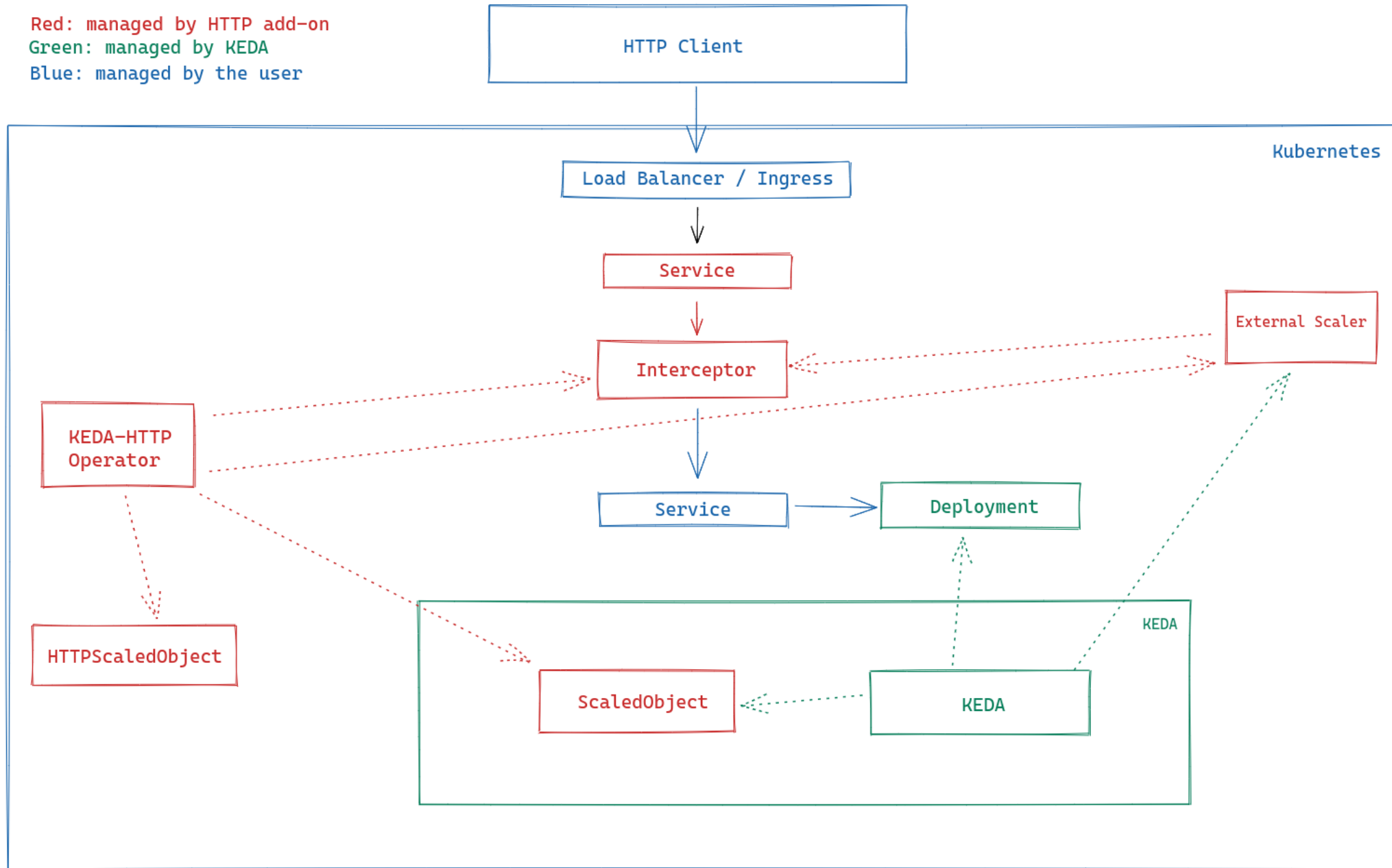
- [Announced June 2021](#)
- Led by Aaron Schlesinger ([@arschles](#)) at Microsoft
- Reached [version 0.3.0](#) in January 2022

See also: [The Launch Space - Scaling HTTP-based apps in Kubernetes using KEDA-HTTP](#)
(Aaron Schlesinger & Aaron Wislang)

KEDA HTTP Add-on - HTTPScaledObject

```
kind: HTTPScaledObject
apiVersion: http.keda.sh/v1alpha1
metadata:
  name: xkcd
spec:
  host: "myhost.com"
  targetPendingRequests: 100
  scaleTargetRef:
    deployment: xkcd
    service: xkcd
    port: 8080
```

Red: managed by HTTP add-on
Green: managed by KEDA
Blue: managed by the user



Thank you!

- Get involved @ [kedacore/keda](#) | [kedacore/http-add-on](#) | <https://keda.sh/community/>
- Try the KEDA + AKS Lab @ Open Source Labs (<https://aka.ms/oss-labs>)
- Drop by `#open-source` on Microsoft Open Source Discord (<https://aka.ms/open-source-discord>)

Aaron Wislang | [@as_w](#) | aaronmsft.com/me/