

# Autoscaling containers



Kubernetes Event Driven Autoscaler

# What is the problem solved by KEDA ?

In Kubernetes, you can **autoscale** pods based (only) on CPU and Memory utilization available within the cluster.

Some applications needs to autoscaler based on **metrics or events** from outside the cluster.  
An example is **number of messages** within a **Queue** or **number of HTTP requests**.

KEDA will be the enabler for that use case.

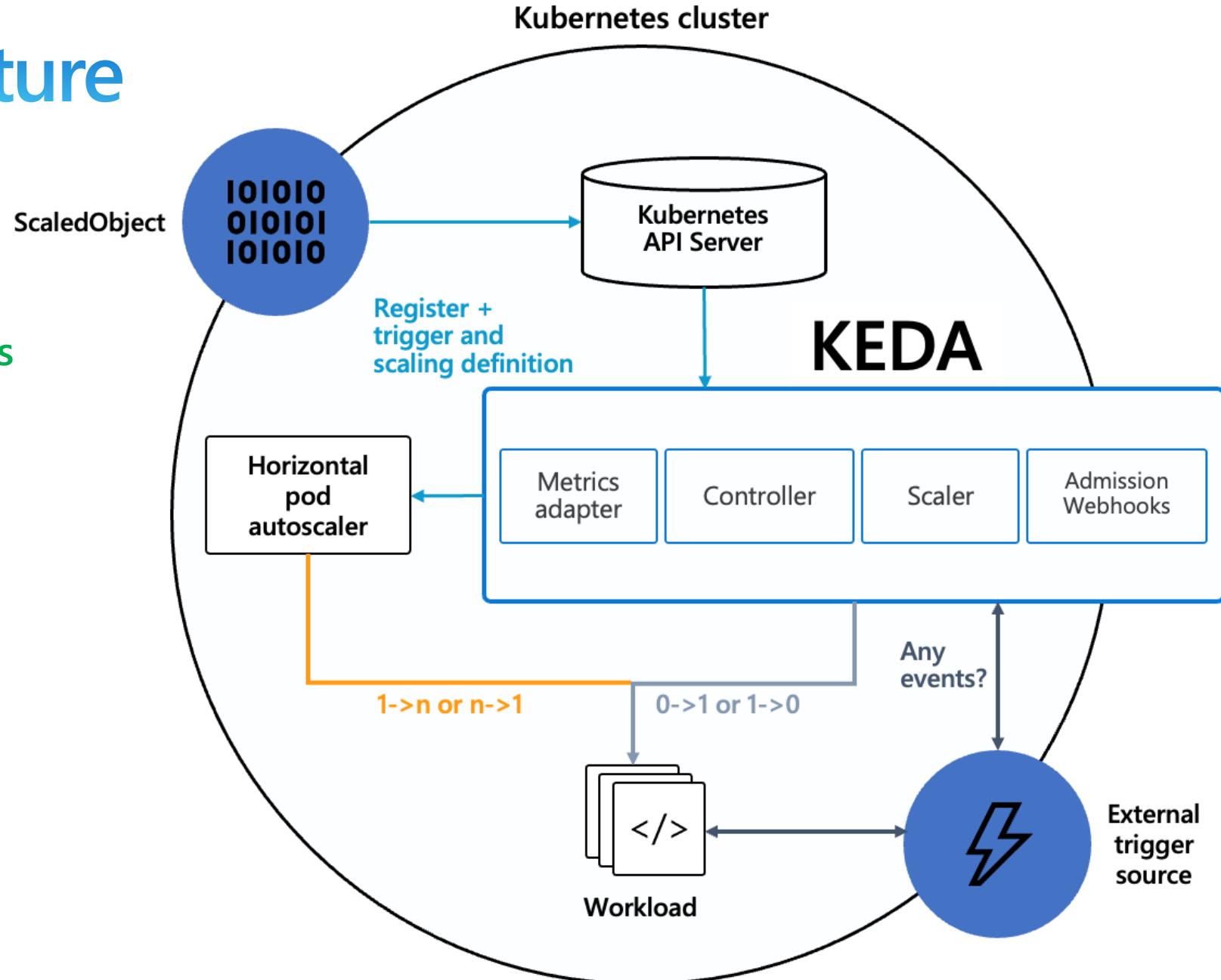


# KEDA Architecture

KEDA watches for the events or metrics.

Then KEDA triggers application autoscaling.

KEDA acts on the HPA and deployment activation.



# KEDA the project

Supports building event-driven applications in Kubernetes ([AKS](#), [Openshift](#), [EKS](#), [GKE...](#))

Now it supports also [Azure Container Apps](#)

Native integration with Horizontal Pod Autoscaler (HPA)

Supports [autoscaling Deployments and Jobs](#) (1 event -> 1 job)

Built in conjunction with [Red Hat](#)

[CNCF](#) graduated project



kedacore / keda

 Type ⌘ to search

Code

Issues 181

Pull requests 39

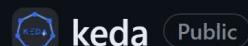
Discussions

Actions

Projects 1

Security 1

Insights



keda

Public



Watch

93



Fork

863



Starred 6.8k



&lt;&gt; Code ▾



jeevanragula

AWS SDK V2 Upgrade and update scalers (#4953)



791c895 · 2 days ago

2,042 Commits

## About

KEDA is a Kubernetes-based Event Driven Autoscaling component. It provides event driven scale for any container running in Kubernetes

## keda.sh

## Used by 22



## Contributors 302



+ 291 contributors

## Languages

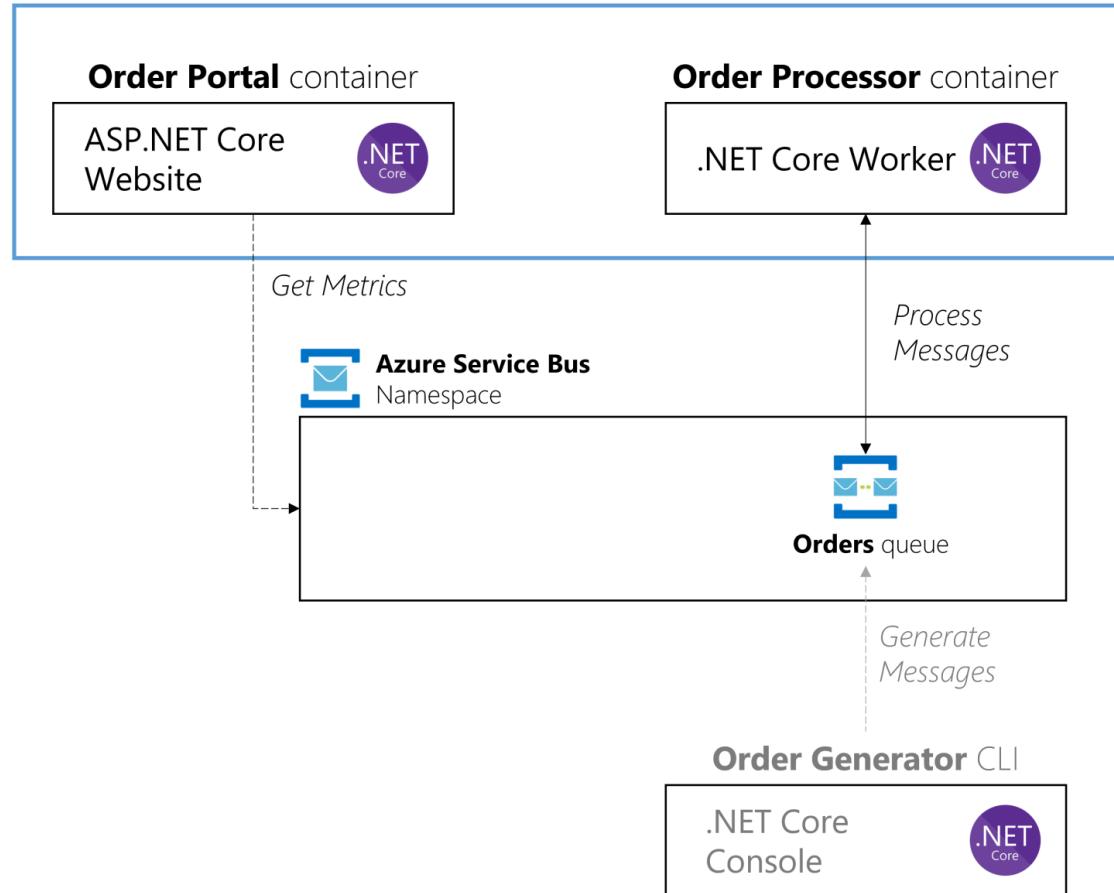


● Go 98.8% ● Other 1.2%

# KEDA event sources and scalers

ActiveMQ	ActiveMQ Artemis	Apache Kafka	Apache Pulsar	ArangoDB	AWS CloudWatch
AWS DynamoDB	AWS DynamoDB Streams		AWS Kinesis Stream	AWS SQS Queue	Azure Application Insights
Azure Blob Storage	Azure Data Explorer	Azure Event Hubs		Azure Log Analytics	Azure Monitor
Azure Pipelines	Azure Service Bus	Azure Storage Queue	Cassandra	CouchDB	CPU
Cron	Datadog	Elasticsearch	Etcfd	External	External Push
Github Runner Scaler	Google Cloud Platform Stackdriver	Google Cloud Platform Storage		Google Cloud Platform Pub/Sub	Graphite
Huawei Cloudeye	IBM MQ	InfluxDB	Kubernetes Workload	Liiklus Topic	Loki
Memory	Metrics API	MongoDB	MSSQL	MySQL	NATS JetStream
New Relic	NATS Streaming	OpenStack Metric	OpenStack Swift	PostgreSQL	Predictkube
RabbitMQ Queue	Prometheus	Redis Lists	Redis Lists (supports Redis Cluster)		Redis Streams
Redis Streams (supports Redis Cluster)	Redis Streams (supports Redis Sentinel)			Selenium Grid Scaler	
Solace PubSub+ Event Broker	Solr				

# KEDA sample for Kubernetes



[github.com/kedacore/sample-dotnet-worker-servicebus-queue](https://github.com/kedacore/sample-dotnet-worker-servicebus-queue)

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: order-processor-scaler
  labels:
    app: order-processor
    name: order-processor
spec:
  scaleTargetRef:
    name: order-processor
  # minReplicaCount: 0 Change to define how many minimum replicas you want
  maxReplicaCount: 10
  triggers:
  - type: azure-servicebus
    metadata:
      queueName: orders
      queueLength: '5'
    authenticationRef:
      name: trigger-auth-service-bus-orders

apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: trigger-auth-service-bus-orders
spec:
  secretTargetRef:
  - parameter: connection
    name: secrets-order-management
    key: servicebus-order-management-connectionstring
```

# ScaledObject CRD – Deployment, StatefulSets, Custom Resources

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: {scaled-object-name}
spec:
  scaleTargetRef:
    apiVersion: {api-version-of-target-resource} # Optional. Default: apps/v1
    kind: {kind-of-target-resource} # Optional. Default: Deployment
    name: {name-of-target-resource} # Mandatory. Must be in the same namespace
    envSourceContainerName: {container-name} # Optional. Default: .spec.template.env
  pollingInterval: 30 # Optional. Default: 30 seconds
  cooldownPeriod: 300 # Optional. Default: 300 seconds
  idleReplicaCount: 0 # Optional. Must be less than minReplicaCount
  minReplicaCount: 1 # Optional. Default: 0
  maxReplicaCount: 100 # Optional. Default: 100
  fallback:
    failureThreshold: 3 # Optional. Section to specify failure threshold
    replicas: 6 # Mandatory if fallback section is present
  advanced:
    restoreToOriginalReplicaCount: true/false # Optional. Default: false
    horizontalPodAutoscalerConfig:
      behavior:
        scaleDown:
          stabilizationWindowSeconds: 300 # Optional. Section to specify HPA's scale down behavior
          policies:
            - type: Percent
              value: 100
              periodSeconds: 15
  triggers:
    # {list of triggers to activate scaling of the target resource}
```

# ScaledObject CRD – Job

```
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: {scaled-job-name}
spec:
  jobTargetRef:
    parallelism: 1                                # [max number of desired pods](https://kubernetes.io/docs/concepts/workloads/controllers/pod-controller/#parallelism)
    completions: 1                                 # [desired number of successfully finished pods](https://kubernetes.io/docs/concepts/workloads/controllers/pod-controller/#completions)
    activeDeadlineSeconds: 600                      # Specifies the duration in seconds relative to the startTime
    backoffLimit: 6                                # Specifies the number of retries before requeueing
    template:
      # describes the [job template](https://kubernetes.io/docs/concepts/workloads/controllers/jobs-pods/#template)
      pollingInterval: 30                            # Optional. Default: 30 seconds
      successfulJobsHistoryLimit: 5                  # Optional. Default: 100. How many completed jobs to keep in history
      failedJobsHistoryLimit: 5                      # Optional. Default: 100. How many failed jobs to keep in history
      envSourceContainerName: {container-name}       # Optional. Default: .spec.JobTargetRef.template.spec.containers[0].name
      maxReplicaCount: 100                           # Optional. Default: 100
      scalingStrategy:
        strategy: "custom"                         # Optional. Default: default. Which Scaling Strategy to use
        customScalingQueueLengthDeduction: 1          # Optional. A parameter to optimize custom scaling
        customScalingRunningJobPercentage: "0.5"      # Optional. A parameter to optimize custom scaling
        pendingPodConditions:                       # Optional. A parameter to calculate pending pods
          - "Ready"
          - "PodScheduled"
          - "AnyOtherCustomPodCondition"
  triggers:
    # {list of triggers to create jobs}
```

# Triggers

## Service Bus Trigger

```
triggers:
- type: azure-servicebus
  metadata:
    # Required: queueName OR topicName and subscriptionName
    queueName: functions-sbqueue
    # or
    topicName: functions-sbtopic
    subscriptionName: sbtopic-sub1
    # Optional, required when pod identity is used
    namespace: service-bus-namespace
    # Optional, can use TriggerAuthentication as well
    connectionFromEnv: SERVICEBUS_CONNECTIONSTRING_ENV_NAME # This must be a connection
    # Optional
    messageCount: "5" # Optional. Count of messages to trigger scaling on. Default: 5 messages
    cloud: Private # Optional. Default: AzurePublicCloud
    endpointSuffix: servicebus.airgap.example # Required when cloud=Private
```

## Kafka Trigger

```
triggers:
- type: kafka
  metadata:
    bootstrapServers: kafka.svc:9092
    consumerGroup: my-group
    topic: test-topic
    lagThreshold: '5'
    offsetResetPolicy: latest
    allowIdleConsumers: false
    version: 1.0.0
```

## Prometheus Trigger

```
triggers:
- type: prometheus
  metadata:
    # Required
    serverAddress: http://<prometheus-host>:9090
    metricName: http_requests_total
    query: sum(rate(http_requests_total{deployment="my-deployment"}[2m])) # Note: query
    threshold: '100'
```

# Trigger Authentication (Env Var, Secret, Pod Identity, Vault)

## Pod Identity Auth

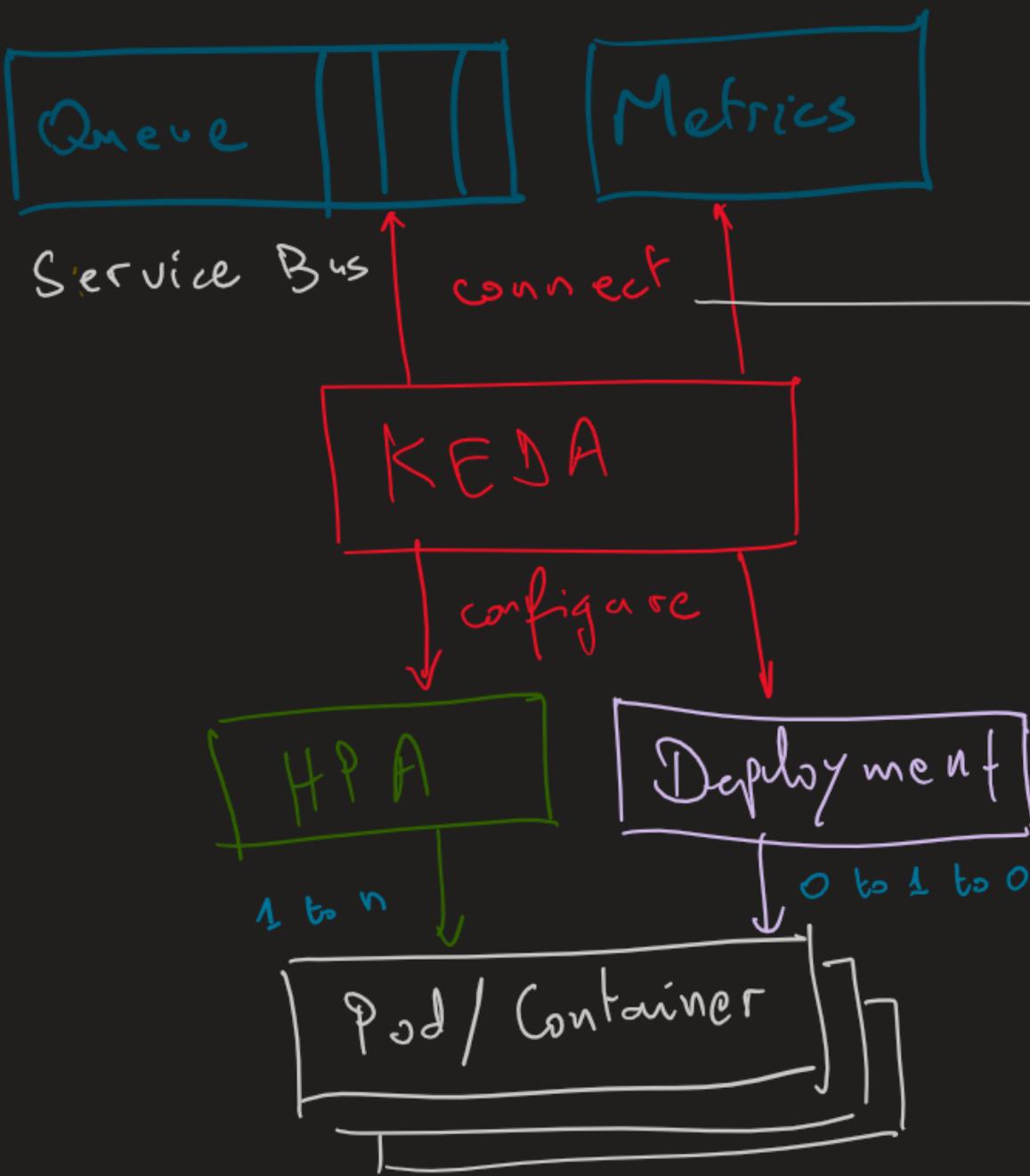
```
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: azure-servicebus-auth
spec:
  podIdentity:
    provider: azure
```

## Secret Auth (connection string)

```
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: mongodb-trigger
spec:
  secretTargetRef:
    - parameter: connectionString
      name: mongodb-secret
      key: connect
```

## Secret Auth (bearer token)

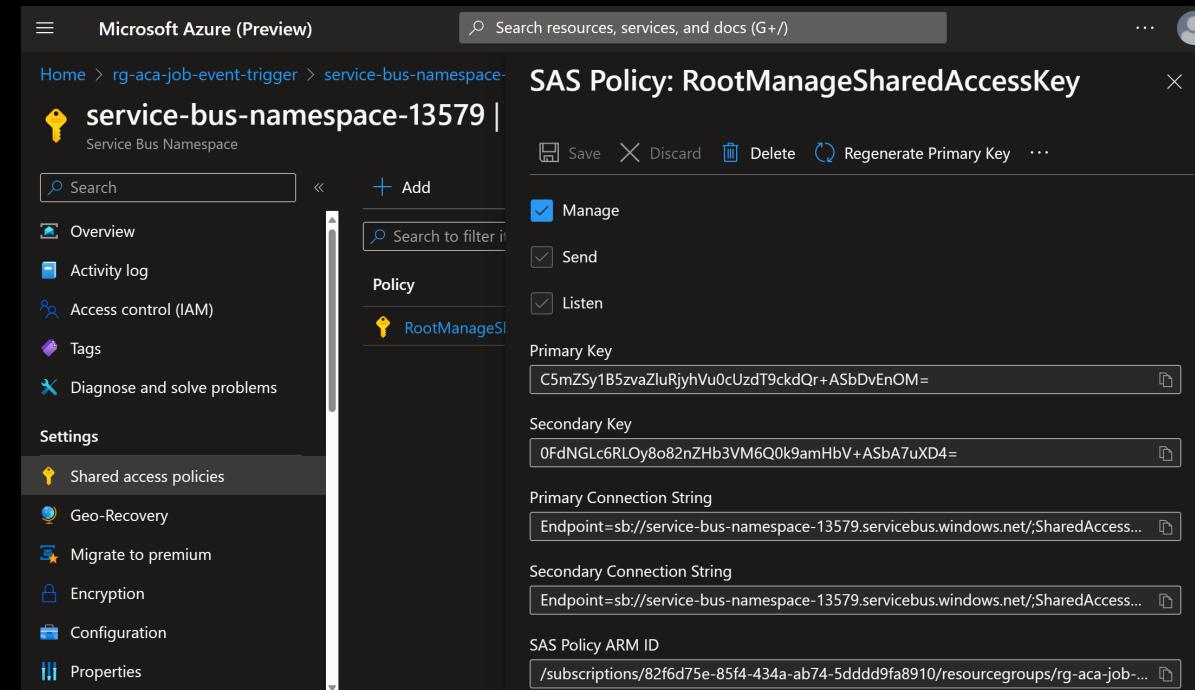
```
apiVersion: v1
kind: Secret
metadata:
  name: keda-prom-secret
  namespace: default
data:
  bearerToken: "BEARER_TOKEN"
  ca: "CUSTOM_CA_CERT"
---
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: keda-prom-creds
  namespace: default
spec:
  secretTargetRef:
    - parameter: bearerToken
      name: keda-prom-secret
      key: bearerToken
      # might be required if you're using a custom CA
    - parameter: ca
      name: keda-prom-secret
      key: ca
```



AuthN → SB Connection String  
→ Scale Rule  
→ Service Principal  
→ Managed Identity ?

# Scaling Container App or Job using KEDA

```
az containerapp job create \
--name "aca-job-demo" \
--resource-group "rg-aca-job-event-trigger" \
--environment "aca-environment" \
--replica-timeout 600 \
--replica-retry-limit 1 \
--replica-completion-count 1 \
--parallelism 1 \
--image "quickstart-jobs:latest" \
--cpu "0.25" \
--memory "0.5Gi" \
--min-executions 0 \
--max-executions 1 \
--trigger-type "Event" \
--secrets service-bus-connection-string='Endpoint=sb://servicebus...'" \
--scale-rule-name "azure-servicebus-queue-rule" \
--scale-rule-type "azure-servicebus" \
--scale-rule-auth "connection=service-bus-connection-string" \
--scale-rule-metadata "namespace=servicebus-ns-job" \
"queueName=queue-messages" \
"messageCount=1"
```



# KEDA sample with Job connecting to Service Bus

```
az containerapp job create `  
    --trigger-type "Event" `  
    --secrets service-bus-connection-string="Endpoint=sb://servicebus...`  
    --scale-rule-name azure-servicebus-queue-rule `  
    --scale-rule-type azure-servicebus `  
    --scale-rule-auth "connection=service-bus-connection-string" `  
    --scale-rule-metadata "queueName=queue-video-input" `  
        "namespace= my-servicebus-ns" `  
        "messageCount=1" `  
    --env-vars `  
        AZURE_CLIENT_ID="40a2eab6-6e69-48d4-b719-d06d2c58d0c9" `  
        MANAGED_IDENTITY_CLIENT_ID="40a2eab6-6e69-48d4-b719-d06d2c58d0c9" `  
        FULLY_QUALIFIED_NAMESPACE="my-servicebus-ns.servicebus.windows.net" `  
        QUEUE_NAME="queue-video-input"
```

