# Scaling to Success: Leveraging KEDA and Kubernetes for Optimal DevOps Pipeline Performance

Wolfgang Ofner

# Wolfgang Ofner

Freelance Cloud Architect, Toronto, Canada

Focus on Azure, Kubernetes, DevOps and .NET

https://programmingwithwolfgang.com

https://www.linkedin.com/in/wolfgangofner

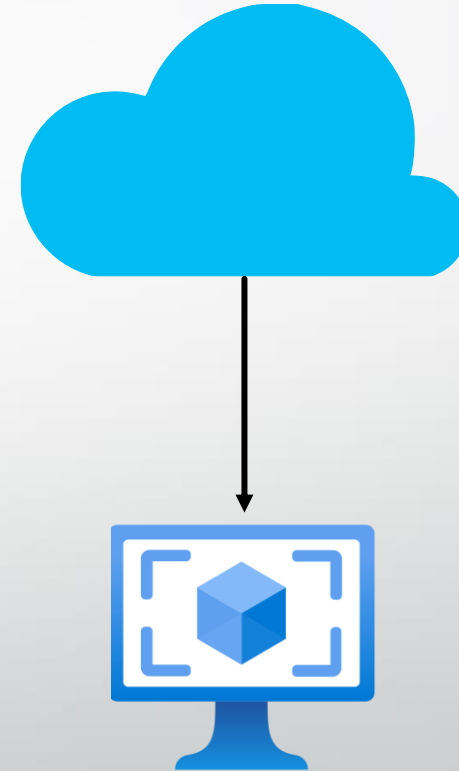https://twitter.com/wolfgang_ofner

# Agenda

- Architecture in SW projects

- Introduction to KEDA

- Scaling with messages in Azure Service Bus Queue

- Scaling Azure DevOps Agents in Kubernetes
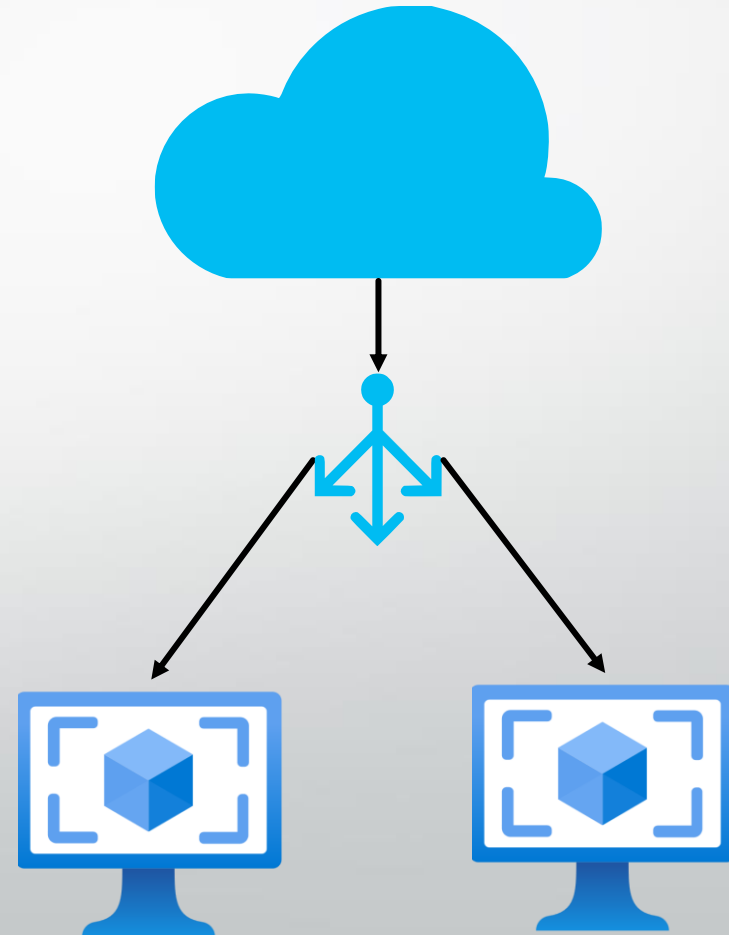
- KEDA Conclusion

- Q&A

# Slides

# Simplified Architecture History

- Server – Client Architecture

- Only few clients

- No redundancy
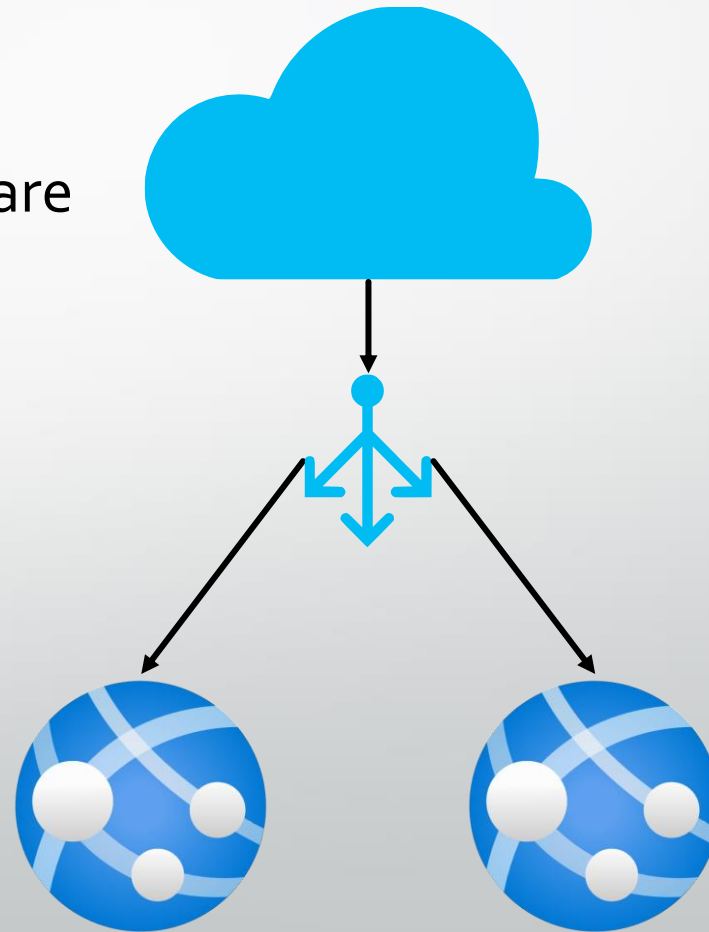
- No high availability

# Simplified Architecture History

- Static load balancing

- New VMs need to be added by hand

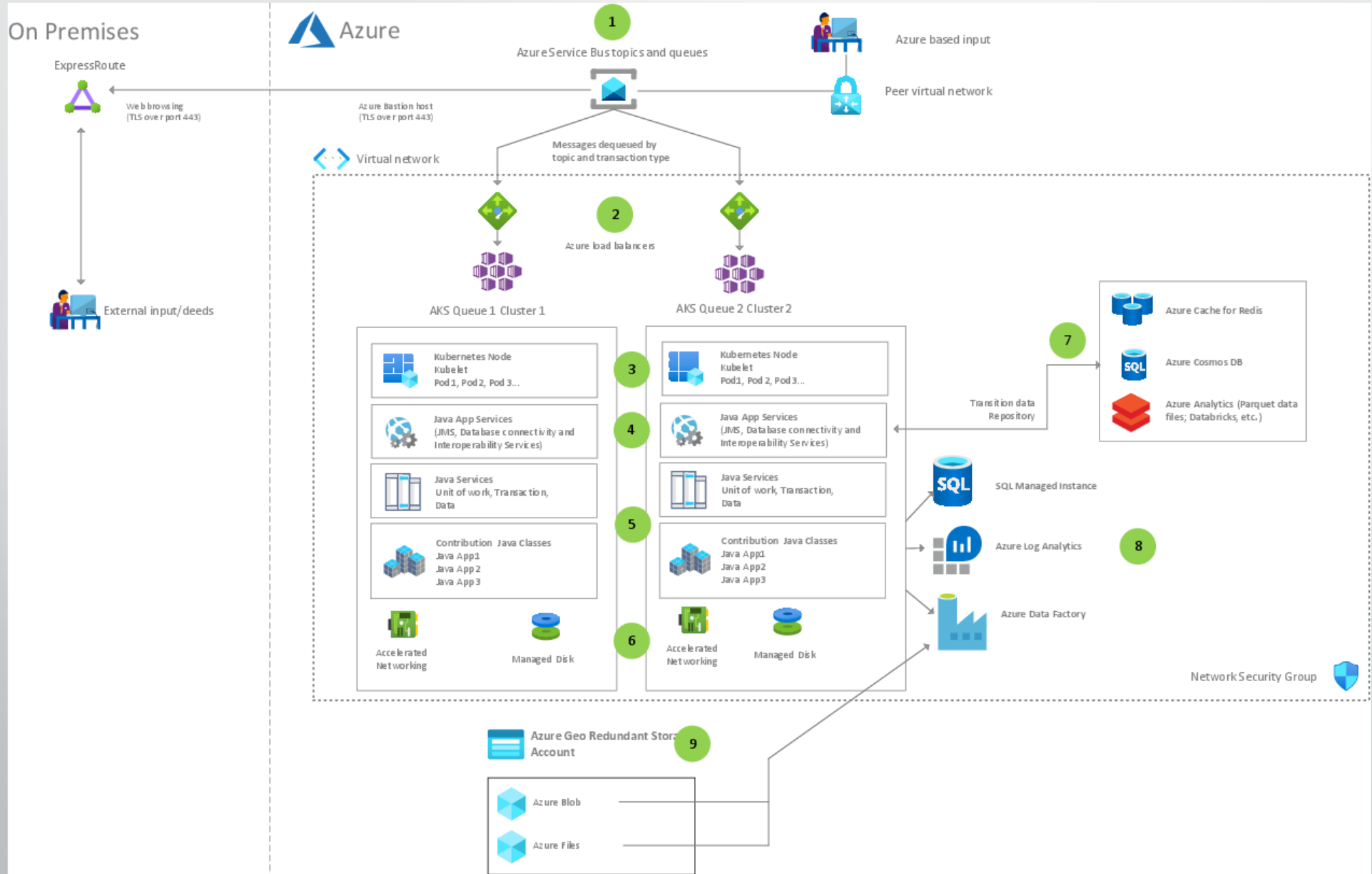- Expensive on-premises hardware

# Simplified Architecture History

- Automatically adding additional hardware

- Pay only what you need

- Mostly CPU or RAM based scaling

# Modern Architecture

# Kubernetes

- Horizontal Pod Autoscaler (HPA)

  - Scaling according to CPU and/or RAM

- Architectures get more and more complex

- Dependencies on external components

- Applications have to react to events

  - Database

  - Service Bus

  - Streams

# Horizontal Pod Autoscaler

- Scales Deployments or StatefulSets

- Adds or removes pods

- Scaling based on CPU or RAM usage

- Scaling on custom metrics

  - Query custom metrics from Kubernetes API

  - Prometheus

  - requests per second

# Horizontal Pod Autoscaler Configuration

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: customerapi
  namespace: customerapi-test
spec:
  maxReplicas: 10
  minReplicas: 1
  averageCpuUtilization: 50
  scaleTargetRef
    apiVersion: apps/v1
    kind: Deployment
    name: customerapi
```

```yaml
behavior:
  scaleDown:
    policies:
    - type: Pods
      value: 4
      periodSeconds: 60
    - type: Percent
      value: 10
      periodSeconds: 60
    selectPolicy: Min
```

```yaml
  scaleUp:
    policies:
    - type: Pods
      value: 5
      periodSeconds: 60
    - type: Percent
      value: 12
      periodSeconds: 60
  selectPolicy: Max
```

# Limitation of the HPA

- Black Friday

- Thousands of orders are stored in a queue

- Scaling using CPU or RAM is not sufficient

- No option for scaling in this scenario

# KEDA – Kubernetes Event-driven Autoscaling

- Kubernetes Event-driven Autoscaling

- Open source

- CNCF Project

- Maintained by
  - Docplanner Tech
  - Microsoft
  - Red Hat

# KEDA

- 62 built-in Scaler
  - Apache Kafka
  - Azure Blob Storage
  - Azure Monitor
  - Azure Service Bus
  - Elastic Search
  - MongoDB
  - Prometheus
  - Redis Streams

# KEDA Use Cases

- Scale according to external events

- Scale to Zero

  - Bring serverless to your datacenter

  - Recreate Azure Functions architecture

  - Better resource usage

# KEDA Installation

- Installation via Helm charts
- Namespace: keda

# KEDA Installation

kubectl create namespace keda

helm repo add kedacore https://kedacore.github.io/charts

helm repo update

helm install keda kedacore/keda --namespace keda

# KEDA Resources

```
PS C:\Users\Wolfgang> kubectl get all -n keda
NAME                                              READY   STATUS    RESTARTS   AGE
pod/keda-operator-5748df494c-mxz9p                1/1     Running   0          124m
pod/keda-operator-metrics-apiserver-cb649dd48-jjhpc  1/1  Running   0          124m

NAME                                     TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)         AGE
service/keda-operator-metrics-apiserver  ClusterIP   10.0.241.182   <none>        443/TCP,80/TCP  124m

NAME                                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/keda-operator                    1/1     1            1           124m
deployment.apps/keda-operator-metrics-apiserver  1/1     1            1           124m

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/keda-operator-5748df494c                   1         1         1       124m
replicaset.apps/keda-operator-metrics-apiserver-cb649dd48  1         1         1       124m
```
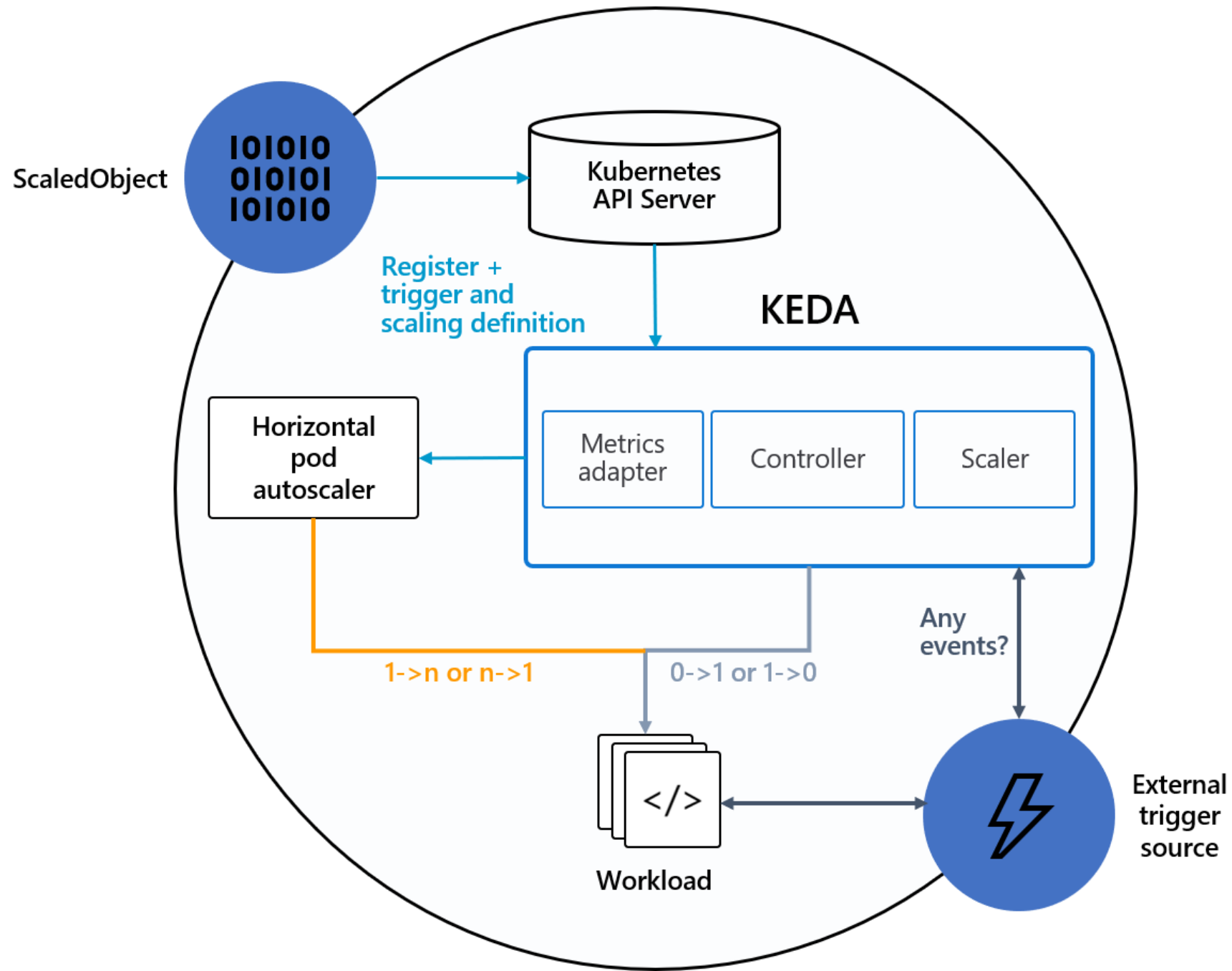
# KEDA Architecture

- 2 components for KEDA

  - Agent or Operator

  - Metrics Server

- Uses HPA for scaling

- Seamless integration into existing architecture

# KEDA Architecture

- 2 components for KEDA
  - Agent
  - Metrics Server
- Uses HPA for scaling
- Seamless integration into existing architecture
- 2 custom K8s resources for scaler
  - ScaledObject
  - TriggerAuthentication

# ScaledObject

```yaml
apiVersion:
keda.sh/v1alpha1

kind: ScaledObject

metadata:

  name: kedademoapi-scaler
```

```yaml
spec:

  scaleTargetRef:

    name: kedademoapi

  minReplicaCount: 0

  maxReplicaCount: 10

  pollingInterval: 30

  cooldownPeriod: 30
```

```yaml
triggers:

  - type: azure-servicebus

    metadata:

      queueName: KedaDemo

      queueLength: '5'

    authenticationRef:

      name: trigger-
authentication-kedademoapi
```

# TriggerAuthentication

```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: trigger-authentication-kedademoapi
spec:
  secretTargetRef:
  - parameter: connection
    name  kedademoapi-connectionstrings
    key: AzureServiceBus__ConnectionString
```

# Kubernetes Secret

```
PS C:\Users\Wolfgang> kubectl get secrets
NAME                                           TYPE                                 DATA   AGE
default-token-88lzb                            kubernetes.io/service-account-token  3      26h
kedademoapi-connectionstrings                  Opaque                               1      26h
kedademoapi-tls                                kubernetes.io/tls                    2      26h
sh.helm.release.v1.kedademoapi-kedademoapi-test.v1   helm.sh/release.v1            1      26h
sh.helm.release.v1.kedademoapi-kedademoapi-test.v2   helm.sh/release.v1            1      22h
PS C:\Users\Wolfgang> kubectl describe secret kedademoapi-connectionstrings
Name:        kedademoapi-connectionstrings
Namespace:   kedademoapi-test
Labels:      app.kubernetes.io/managed-by=Helm
Annotations: meta.helm.sh/release-name: kedademoapi-kedademoapi-test
             meta.helm.sh/release-namespace: kedademoapi-test


Type:  Opaque


Data
====
AzureServiceBus__ConnectionString:   165 bytes
```

# Kubernetes Secret

Namespace Overview ∨ ❯ Config and Storage ∨ ❯ Secrets ∨ ❯ kedademoapi-connectionstrings

## kedademoapi-connectionstrings

Summary    Metadata    Resource Viewer    YAML

```
1   ---
2   apiVersion: v1
3   data:
4     AzureServiceBus__ConnectionString: RW5kcG9pbnQ9c2I6Ly93b2xmZ2FuZ2tlZGFkZW1vLnNlcl...
5   kind: Secret
```

# Scaling based on Azure Service Queue Messages

# Demo

- Scale with messages in an Azure Service Bus Queue
- Scale to 0
- Scale to 1

```
PS C:\Users\Wolfgang> kubectl get pods
NAME                              READY   STATUS    RESTARTS    AGE
kedademoapi-6f986c4b76-hvbrq      1/1     Running   0           13m
```

# kedademo (wolfgangkedademo/kedademo) | Service Bus Explorer

Service Bus Queue

🔍 Search (Ctrl+/)  «

**Overview**

**Access control (IAM)**

**Diagnose and solve problems**

**Settings**

🔑 **Shared access policies**

▦ **Service Bus Explorer (preview)**

**Properties**

🔒 **Locks**

---

↻ Refresh

Authentication type ⓘ

[ **Access key** ] [ Active Directory ]

Send    **Receive**    Peek

Receive performs a destructive read (ReceiveAndDelete) from Queue *kedademo*.
from the Queue. Messages shown here are no longer stored.

| **Active** | **Dead-Lettered** | **Scheduled** |
| 1 MESSAGES | 0 MESSAGES | 0 MESSAGES |

Please Select Queue or DeadLetter

⦿ Queue    ◯ DeadLetter

**POST** `/v1/ServiceBusProcessing` Action to add new messages to the queue.

## Parameters

| Name | Description |
| --- | --- |
| numberOfQueueItems<br>integer($int32)<br>(query) | 270 |

Execute

## Responses

Curl

```
curl -X 'POST' \
  'https://test.kedademo.programmingwithwolfgang.com/v1/ServiceBusProcessing?numberOfQueueItems=270' \
  -H 'accept: */*' \
  -d ''
```

Request URL

```
https://test.kedademo.programmingwithwolfgang.com/v1/ServiceBusProcessing?numberOfQueueItems=270
```

Server response

| Code | Details |
| --- | --- |
| 200 | Response headers<br><br>```content-length: 0```<br>```date: Fri,18 Feb 2022 15:45:21 GMT```<br>```strict-transport-security: max-age=15724800; includeSubDomains``` |

| GET | /v1/ServiceBusProcessing | Action to start processing the queue items. |

**Parameters**

No parameters

<div style="text-align:center">Execute</div>

**Responses**

**Curl**

```
curl -X 'GET' \
  'https://test.kedademo.programmingwithwolfgang.com/v1/ServiceBusProcessing' \
  -H 'accept: application/json'
```

**Request URL**

```
https://test.kedademo.programmingwithwolfgang.com/v1/ServiceBusProcessing
```

**Server response**

| Code | Details |
|------|---------|
| 200 | **Response body**<br><br>271<br><br>**Response headers**<br><br>content-type: application/json; charset=utf-8<br>date: Fri,18 Feb 2022 15:51:31 GMT<br>strict-transport-security: max-age=15724800; includeSubDomains |

# kedademo (wolfgangkedademo/kedademo) | Service Bus Explorer

Service Bus Queue

🔍 Search (Ctrl+/)          «

🔄 Refresh

**Overview**

**Access control (IAM)**

**Diagnose and solve problems**

**Settings**

**Shared access policies**

**Service Bus Explorer (preview)**

**Properties**

**Locks**

Authentication type ⓘ

[ **Access key**  Active Directory ]

**Send**   Receive   Peek

Send Message to Queue *kedademo*

Content Type *

Text/Plain

new message

# kedademo (wolfgangkedademo/kedademo) | Service Bus Explorer

Service Bus Queue

---

🔍 Search (Ctrl+/)                    «

📧 Overview

👥 Access control (IAM)

🔧 Diagnose and solve problems

## Settings

🔑 Shared access policies

▦ Service Bus Explorer (preview)

⚙ Properties

🔒 Locks

---

↻ Refresh

Authentication type ⓘ

| Access key | Active Directory |

Send     **Receive**     Peek

Receive performs a destructive read ([ReceiveAndDelete](#)) from Queue *kedademo*. from the Queue. Messages shown here are no longer stored.

**Active**
**1** MESSAGES

**Dead-Lettered**
**0** MESSAGES

**Scheduled**
**0** MESSAGES

Please Select Queue or DeadLetter

◉ Queue     ○ DeadLetter

# KEDA Scaling Logs

- keda-operator pod writes logs during scaling events

{"scaledobject.Name": "kedademoapi-scaler", "scaledObject.Namespace": "kedademoapi-test", "scaleTarget.Name": "kedademoapi", "Original Replicas Count": 6, "New Replicas Count": 0}
er kind": "ScaledObject", "name": "kedademoapi-scaler", "namespace": "kedademoapi-test"}
"scaledObject.Namespace": "kedademoapi-test", "scaleTarget.Name": "kedademoapi" 'Original Replicas Count": 0, "New Replicas Count": 1}

# Limitations

- Scaler not available for used technology
- Cluster runs out of resources

# Pods

| Name | Labels | Ready | Phase | Restarts | Node |
|---|---|---|---|---|---|
| kedademoapi-6f986c4b76-2zfxc | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-6w9tc | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-777r8 | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-9vs76 | app:kedademoapi  draft:draft-app  1+ | 1/1 | Running | 0 | aks-nodepool1-35436033-vmss000000 |
| kedademoapi-6f986c4b76-jdd8x | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-mdj62 | app:kedademoapi  draft:draft-app  1+ | 1/1 | Running | 0 | aks-nodepool1-35436033-vmss000000 |
| kedademoapi-6f986c4b76-qg298 | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-rzgfm | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-s56q6 | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |
| kedademoapi-6f986c4b76-wb7rr | app:kedademoapi  draft:draft-app  1+ | 0/1 | Pending | 0 | \<not scheduled\> |

# Pods

| Name | Labels | | | Ready | Phase | Restarts | Node |
|------|--------|---|---|-------|-------|----------|------|
| ⓘ kedademoapi-6f986c4b76-2zfxc | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ⓘ kedademoapi-6f986c4b76-6w9tc | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ⓘ kedademoapi-6f986c4b76-777r8 | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ✓ kedademoapi-6f986c4b76-9vs76 | app:kedademoapi | draft:draft-app | 1+ | 1/1 | Running | 0 | aks-nodepool1-35436033-vmss000000 |
| ⓘ kedademoapi-6f986c4b76-jdd8x | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ✓ kedademoapi-6f986c4b76-mdj62 | app:kedademoapi | draft:draft-app | 1+ | 1/1 | Running | 0 | aks-nodepool1-35436033-vmss000000 |
| ⓘ kedademoapi-6f986c4b76-qg298 | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ⓘ kedademoapi-6f986c4b76-rzgfm | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ⓘ kedademoapi-6f986c4b76-s56q6 | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |
| ⓘ kedademoapi-6f986c4b76-wb7rr | app:kedademoapi | draft:draft-app | 1+ | 0/1 | Pending | 0 | <not scheduled> |

## Events

| Message | | Reason | |
|---|---|---|---|
| 0/1 nodes are available: 1 Insufficient cpu. | | FailedScheduling | |

# Limitations

- Scaler not available for used technology
- Cluster runs out of resources
  - Azure Cluster Autoscaler
  - Define replica limit
  - Monitor cluster usage

# Azure DevOps Agent with KEDA

# Scaling ADO Agent with KEDA

- Azure DevOps preparation

- Build Docker image

- Test locally

- Deploy to Kubernetes

- Apply KEDA scaling

# Copy the PAT

# Building the ADO Docker Image

- Dockerfile
- start.sh (with LF EOF)

```
FROM ubuntu:20.04
RUN DEBIAN_FRONTEND=noninteractive apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get upgrade -y

RUN DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends \
    apt-transport-https \
    apt-utils \
    ca-certificates \
    curl \
    git \
    iputils-ping \
    jq \
    lsb-release \
    software-properties-common \
    wget

RUN curl -sL https://aka.ms/InstallAzureCLIDeb | bash

RUN wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
RUN dpkg -i packages-microsoft-prod.deb
RUN rm packages-microsoft-prod.deb
RUN apt-get update && apt-get install -y dotnet-sdk-6.0
RUN apt-get update && apt-get install -y dotnet-sdk-7.0

# Can be 'linux-x64', 'linux-arm64', 'linux-arm', 'rhel.6-x64'.
ENV TARGETARCH=linux-x64

WORKDIR /azp

COPY ./start.sh .
RUN chmod +x start.sh

ENTRYPOINT [ "./start.sh" ]
```

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> docker build . -t adoagentkeda
[+] Building 376.5s (18/18) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 954B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:20.04
 => [ 1/13] FROM docker.io/library/ubuntu:20.04@sha256:9fa30fcef427e5e88c76bc41ad37b7cc573e1d79cecb23035e413c4be6e4
 => => resolve docker.io/library/ubuntu:20.04@sha256:9fa30fcef427e5e88c76bc41ad37b7cc573e1d79cecb23035e413c4be6e476
 => => sha256:9fa30fcef427e5e88c76bc41ad37b7cc573e1d79cecb23035e413c4be6e476ab 1.13kB / 1.13kB
 => => sha256:3626dff0d616e8ee7065a9ac8c7117e904a4178725385910eeecd7f1872fc12d 424B / 424B
 => => sha256:61c45d0e97988ff0cfa876e9ec145445974b9b384fe0a150b057ffc46039b3a0 2.30kB / 2.30kB
 => => sha256:47c7644723910b6dfc6ec8b3bd9fed3ac32778cf485ce3a6535ff6b6da06f743 27.50MB / 27.50MB
 => => extracting sha256:47c7644723910b6dfc6ec8b3bd9fed3ac32778cf485ce3a6535ff6b6da06f743
 => [internal] load build context
 => => transferring context: 2.52kB
 => [ 2/13] RUN DEBIAN_FRONTEND=noninteractive apt-get update
 => [ 3/13] RUN DEBIAN_FRONTEND=noninteractive apt-get upgrade -y
 => [ 4/13] RUN DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends      apt-transport-htt
 => [ 5/13] RUN curl -sL https://aka.ms/InstallAzureCLIDeb | bash
 => [ 6/13] RUN wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-mic
 => [ 7/13] RUN dpkg -i packages-microsoft-prod.deb
 => [ 8/13] RUN rm packages-microsoft-prod.deb
 => [ 9/13] RUN apt-get update && apt-get install -y dotnet-sdk-6.0
 => [10/13] RUN apt-get update && apt-get install -y dotnet-sdk-7.0
 => [11/13] WORKDIR /azp
 => [12/13] COPY ./start.sh .
 => [13/13] RUN chmod +x start.sh
 => exporting to image
 => => exporting layers
 => => writing image sha256:11167e8091f1222ebe05f84e4f7e711e20fcb1c9ec0bddc75276047477f39d03
```

# Building the ADO Docker Image

- Dockerfile

- start.sh (with LF EOF)

- Azure DevOps values:

  - PAT

  - Pool Name

  - URL

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> docker run -e AZP_URL=https://dev.azure.com/programmingwithwolfgang -e
AZP_TOKEN=qqjw2cvvhwtc4crqxpyye2gbrumtirgdff6wh6fvjlqir32qfzsa -e AZP_AGENT_NAME=agent -e AZP_POOL=Keda adoagentkeda
1. Determining matching Azure Pipelines agent...
2. Downloading and extracting Azure Pipelines agent...
3. Configuring Azure Pipelines agent...


   ___                    _____  _            ___
  /_  \                  |  ___|(_)          | (_)
 / /\ \ ___ _   _ _ __ __| |_    _ _ __   ___| |_ _ __   ___  ___
|  _  |/ _ \ | | | '__/ _ \  _|  | | '_ \ / _ \ | | '_ \ / _ \/ __|
| | | |  __/ |_| | | |  __/ |    | | |_) |  __/ | | | | |  __/\__ \
\_| |_/\___|\__,_|_|  \___\_|    |_| .__/ \___|_|_|_| |_|\___||___/
                                   | |
       agent v2.217.2              |_|         (commit ef0b5a5)


>> End User License Agreements:

Building sources from a TFVC repository requires accepting the Team Explorer Everywhere End User License Agreement. This
 step is not required for building sources from Git repositories.

A copy of the Team Explorer Everywhere license agreement can be found at:
   /azp/license.html


>> Connect:

Connecting to server ...


>> Register Agent:
```

main ⌄  ◈ K8sAdoAgent / **azure-pipelines.yml**

```yaml
1   trigger: none
2
3   pool: Keda
4
5   variables:
6       buildConfiguration: 'Release'
7
8   jobs:
9   - job: job1
10    steps:
          Settings
11      - task: Bash@3
12        inputs:
13          targetType: 'inline'
14          script: ''
15        displayName: Check if job is running
```

56

# Jobs in run #20230312.1

K8sAdoAgent

## Jobs

| | | |
|---|---|---|
| ✓ **job1** | | 6s |
| ✓ Initialize job | | <1s |
| ✓ Checkout K8sAdoAgent... | | 1s |
| ✓ Check if job is running | | <1s |
| ✓ Post-job: Checkout K8... | | <1s |
| ✓ Finalize Job | | <1s |
| ✓ Report build status | | <1s |

## ✓ job1

```
 1   Pool: Keda
 2   Queued: Just now [manage parallel jobs]
 3   Agent: agent
 4   Started: Just now
 5   Duration: 6s
 6
 7   The agent request is already running or has already completed.
 8 ▶ Job preparation parameters
 9   Job live console data:
10   Starting: job1
11   Finishing: job1
```

```
>> Connect:

Connecting to server ...

>> Register Agent:

Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
2023-03-12 11:12:31Z: Settings Saved.
4. Running Azure Pipelines agent...
Scanning for tool capabilities.
Connecting to the server.
2023-03-12 11:12:33Z: Listening for Jobs
2023-03-12 11:15:50Z: Running job: job1
2023-03-12 11:16:01Z: Job job1 completed with result: Succeeded
Cleanup. Removing Azure Pipelines agent...
Removing agent from the server
Connecting to server ...
Succeeded: Removing agent from the server
Removing .credentials
Succeeded: Removing .credentials
Removing .agent
Succeeded: Removing .agent
```

# Push the Docker Image

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> docker tag adoagentkeda wolfgangofner/adoagentkeda
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> docker push wolfgangofner/adoagentkeda
Using default tag: latest
The push refers to repository [docker.io/wolfgangofner/adoagentkeda]
9ab694b94e06: Pushed
95d099e671b0: Pushed
f5f935017b9d: Pushing [=============>                          ]  132.4MB/518.1MB
b5f309652342: Pushing [=======>                                ]  74.44MB/509.2MB
acd9adb1ef6d: Pushed
b79abcab2382: Pushing [================================================>]  726.5kB
ed3d14f25ed7: Pushed
e82e11b3ff01: Pushing [=>                                      ]  30.55MB/1.185GB
85d422d04b2e: Waiting
76674757e35e: Waiting
587658be1954: Waiting
6021993d84a2: Waiting
```

# Create a Secret with the PAT

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: ado-agent-secret
data:
  AZP_TOKEN: cXFqdzJjdnZod3RjNGNycXhweXllMmdicnVtdGlyZ2RmZjZ3aDZmdmpscWlyMzJxZnpzYQ== # replace with your value / (base64 encoded)
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ado-agent-deployment
  labels:
    app: ado-agent
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ado-agent
  template:
    metadata:
      labels:
        app: ado-agent
    spec:
      containers:
      - name: ado-agent
        image: wolfgangofner/adoagentkeda # replace with your value
        env:
          - name: AZP_URL
            value: https://dev.azure.com/programmingwithwolfgang # replace with your value
          - name: AZP_POOL
            value: Keda # replace with your value
          - name: AZP_TOKEN
            valueFrom:
              secretKeyRef:
                name: ado-agent-secret
                key: AZP_TOKEN
```

# Deploy the ADO Agent in K8s

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl create ns ado-agent
namespace/ado-agent created
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl config set-context --current --namespace=ado-agent
Context "microservice-aks" modified.
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl apply -f ./deployment.yaml
secret/ado-agent-secret created
deployment.apps/ado-agent-deployment created
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl get pod --watch
NAME                                    READY    STATUS             RESTARTS    AGE
ado-agent-deployment-5cdc9bc464-4sq65   0/1      ContainerCreating  0           4s
ado-agent-deployment-5cdc9bc464-4sq65   1/1      Running            0           88s
```

## General

-  Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Azure Active Directory

## Security

- Policies
- Permissions

## Boards

- Process

## Pipelines

- Agent pools
- Settings
- Deployment pools

---

 **Keda**

Jobs    **Agents**    Details    Security    Settings    Maintenance History    Analytics

Update all agents

| Name | Last run | Current status | Agent version | Enabled |
|---|---|---|---|---|
| **ado-agent-deployment-5cdc9bc464-4sq65**<br>● Online | | Idle | 2.217.2 | ⬤ On |

# Organization Settings

## General

- Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Azure Active Directory

## Security

- Policies
- Permissions

## Boards

- Process

## Pipelines

- Agent pools
- Settings
- Deployment pools

---

## Keda

Jobs   **Agents**   Details   Security   Settings   Maintenance History   Analytics

**Update all agents**

| Name | Last run | Current status | Agent version | Enabled |
|------|----------|----------------|---------------|---------|
| ado-agent-deployment-5cdc9bc464-4sq65<br>● Online | | Idle | 2.217.2 | Off |

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: ado-scaledjob
spec:
  jobTargetRef:
    template:
      spec:
        containers:
        - name: ado-agent-job
          image: wolfgangofner/adoagentkeda # replace with your value
          imagePullPolicy: Always
          env:
          - name: AZP_URL
            value: https://dev.azure.com/programmingwithwolfgang # replace with your value
          - name: AZP_TOKEN
            valueFrom:
              secretKeyRef:
                name: ado-agent-secret
                key: AZP_TOKEN
          - name: AZP_POOL
            value: Keda # replace with your value
  pollingInterval: 10
  successfulJobsHistoryLimit: 5
  failedJobsHistoryLimit: 5
  maxReplicaCount: 10
  scalingStrategy:
    strategy: "default"
  triggers:
  - type: azure-pipelines
    metadata:
      poolID:  "10" # <azure-devops-pool-id> (must be a string) (https://dev.azure.com/{Organization}/_apis/distributedtask/pools?api-version=7.0)
      organizationURLFromEnv: "AZP_URL"
      personalAccessTokenFromEnv: "AZP_TOKEN"
```

# Deploy the KEDA Scale Job

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl apply -f ./keda-scaled-jobs.yaml
scaledjob.keda.sh/ado-scaledjob created
```

main ∨    K8sAdoAgent / azure-pipelines.yml

```yaml
 1  trigger: none
 2
 3  pool: Keda
 4
 5  variables:
 6    buildConfiguration: 'Release'
 7
 8  jobs:
 9  - job: job1
10    steps:
      Settings
11    - task: Bash@3
12      inputs:
13        targetType: 'inline'
14        script: 'sleep 5m'
15        displayName: Wait for 5 minutes
16  - job: job2
17    steps:
      Settings
18    - task: Bash@3
19      inputs:
20        targetType: 'inline'
21        script: 'sleep 5m'
22        displayName: Wait for 5 minutes
23  - job: job3
24    steps:
      Settings
25    - task: Bash@3
26      inputs:
27        targetType: 'inline'
28        script: 'sleep 5m'
29        displayName: Wait for 5 minutes
```

```
PS C:\Users\Wolfgang\source\repos\Ado-Agent-Keda> kubectl get pod --watch
NAME                                      READY   STATUS             RESTARTS   AGE
ado-agent-deployment-5cdc9bc464-4sq65     1/1     Running            0          22m
ado-scaledjob-xlgcg-6ts6h                 0/1     Pending            0          0s
ado-scaledjob-6nvks-9x5wm                 0/1     Pending            0          0s
ado-scaledjob-45s68-st64h                 0/1     Pending            0          0s
ado-scaledjob-xlgcg-6ts6h                 0/1     Pending            0          0s
ado-scaledjob-vwbxw-dr5wf                 0/1     Pending            0          0s
ado-scaledjob-45s68-st64h                 0/1     Pending            0          0s
ado-scaledjob-6nvks-9x5wm                 0/1     Pending            0          0s
ado-scaledjob-vwbxw-dr5wf                 0/1     Pending            0          0s
ado-scaledjob-j6vcg-jvpn6                 0/1     Pending            0          0s
ado-scaledjob-xlgcg-6ts6h                 0/1     ContainerCreating  0          0s
ado-scaledjob-j6vcg-jvpn6                 0/1     Pending            0          0s
ado-scaledjob-45s68-st64h                 0/1     ContainerCreating  0          0s
ado-scaledjob-vwbxw-dr5wf                 0/1     ContainerCreating  0          0s
ado-scaledjob-6nvks-9x5wm                 0/1     ContainerCreating  0          0s
ado-scaledjob-j6vcg-jvpn6                 0/1     ContainerCreating  0          0s
ado-scaledjob-6nvks-9x5wm                 1/1     Running            0          2s
ado-scaledjob-j6vcg-jvpn6                 1/1     Running            0          3s
ado-scaledjob-vwbxw-dr5wf                 1/1     Running            0          13s
ado-scaledjob-45s68-st64h                 1/1     Running            0          20s
ado-scaledjob-xlgcg-6ts6h                 1/1     Running            0          20s
```

# #20230312.4 • Update azure-pipelines.yml for Azure Pipelines

🏭 K8sAdoAgent

Cancel ⋮

## Summary

Manually run by (WO) Wolfgang Ofner

View change

**Repository and version**

◆ K8sAdoAgent

ℙ main  ◇ b96efdac

**Time started and elapsed**

📅 Just now

🕑 32s

**Related**

📋 0 work items

🗄 0 artifacts

**Tests and coverage**

⚗ Get started

## Jobs

| Name | Status | Duration |
|------|--------|----------|
| 🕐 job1 | Queued | |
| 🕐 job2 | Queued | |
| 🛡 job3 | Running | 🕐 24s |
| 🛡 job4 | Running | 🕐 24s |
| 🕐 job5 | Queued | |

# Azure DevOps Limitations

- ADO Pipelines support scale to zero but need at least one agent registered

- ADO Pipelines can not queue a job with an empty agent pool

- Licensing limits parallel jobs
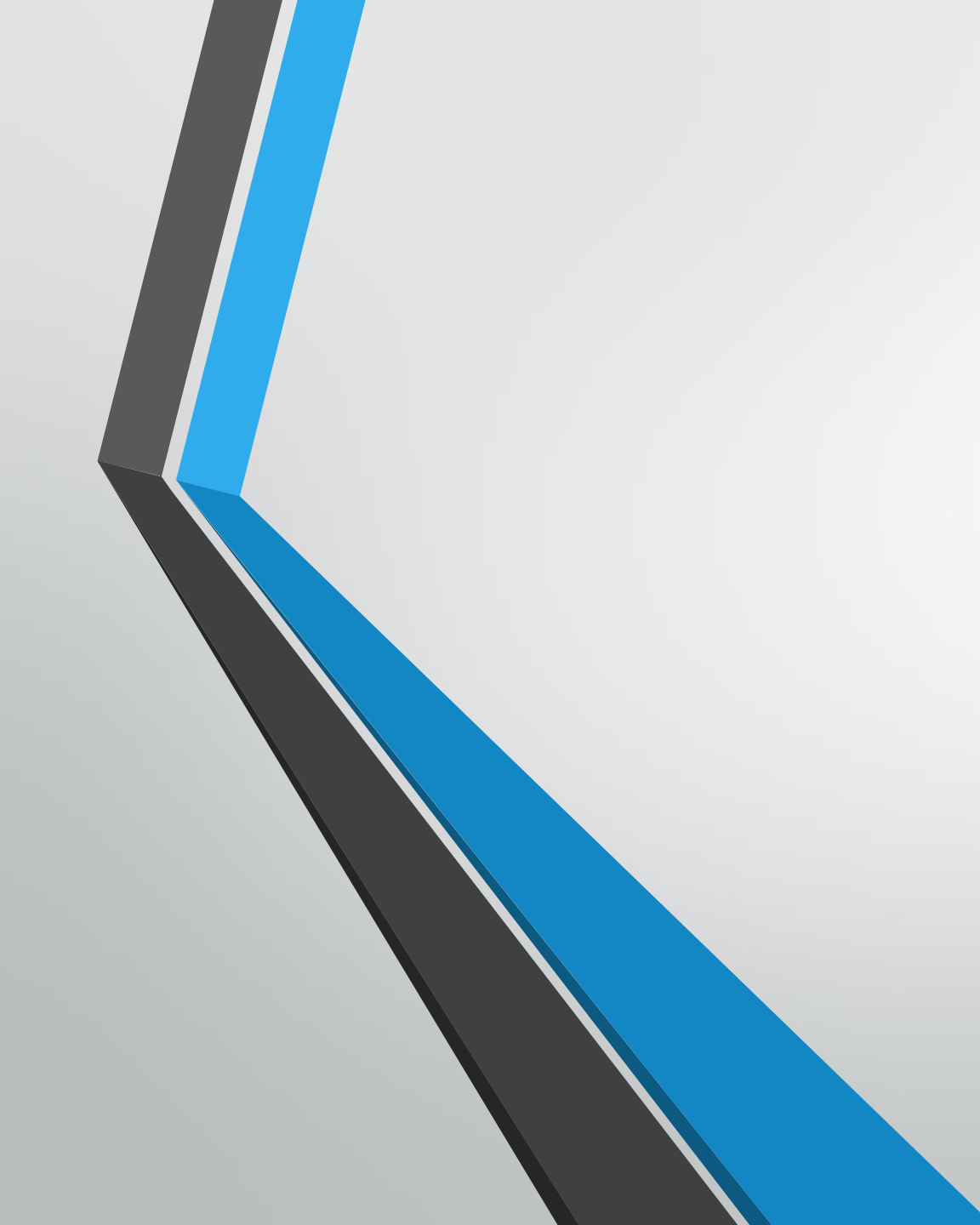
# KEDA ADO Scaling Limitations

- Cancelling a pipeline does not stop running pods

- KEDA does not remove completed pods

- Azure DevOps does not remove offline agents from the agent pool

# KEDA in Production

- Microsoft uses KEDA for Azure Services
  - Azure Container Apps
  - Azure App Services with Azure Arc


- KEDA 1.0.0 → 17. Nov 2019
- Currently 2.11.1
- Over 6k GitHub stars

# Resources

- Demo Application
  - https://github.com/WolfgangOfner/MicroserviceDemo/tree/master/KedaDemoApi
  - https://github.com/WolfgangOfner/Ado-Agent-Keda
- KEDA
  - https://keda.sh
- KEDA GitHub
  - https://github.com/kedacore/keda
- KEDA Architecture Screenshot
  - https://keda.sh/docs/2.6/concepts/#architecture

# Q&A

https://programmingwithwolfgang.com

https://www.linkedin.com/in/wolfgangofner

https://twitter.com/wolfgang_ofner