



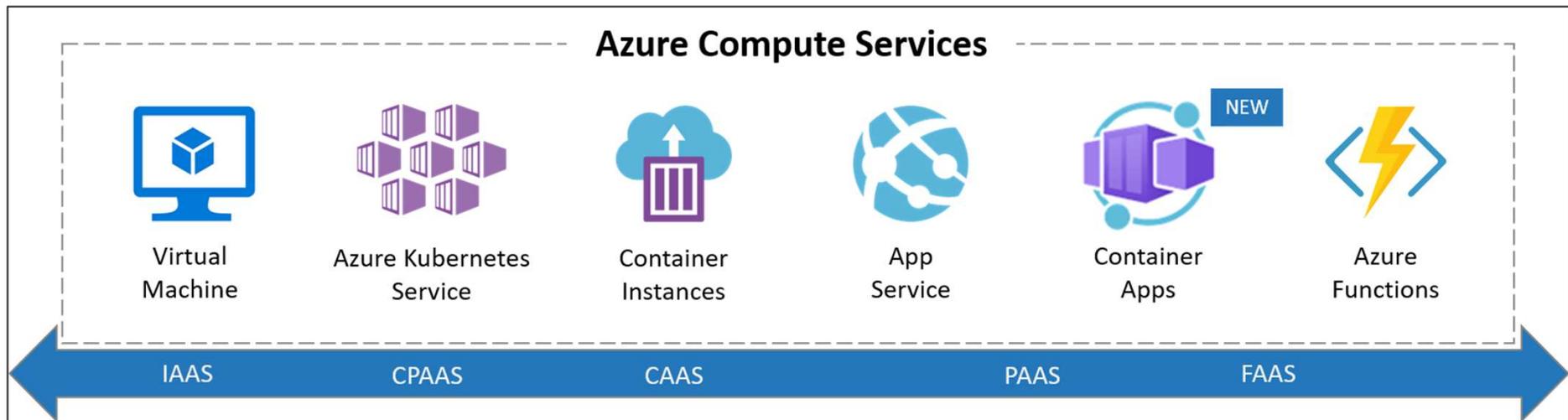
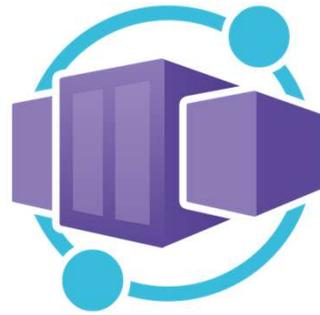
Azure Container Apps Jobs

Andrea Ceroni @ Elfo





Container in Azure



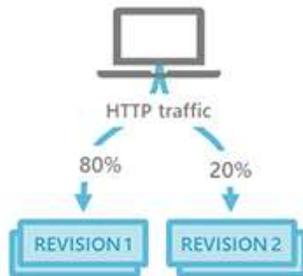


Azure Container Apps



Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS



HTTP requests are split between two versions of the container app where the first revision gets 80% of the traffic, while a new revision receives the remaining 20%.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests.

BACKGROUND PROCESSING



A continuously-running background process that transforms data in a database.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load.

EVENT-DRIVEN PROCESSING



A queue reader application that processes messages as they arrive in a queue.

AUTO-SCALE CRITERIA

Scaling is determined by the number of messages in the queue.

MICROSERVICES



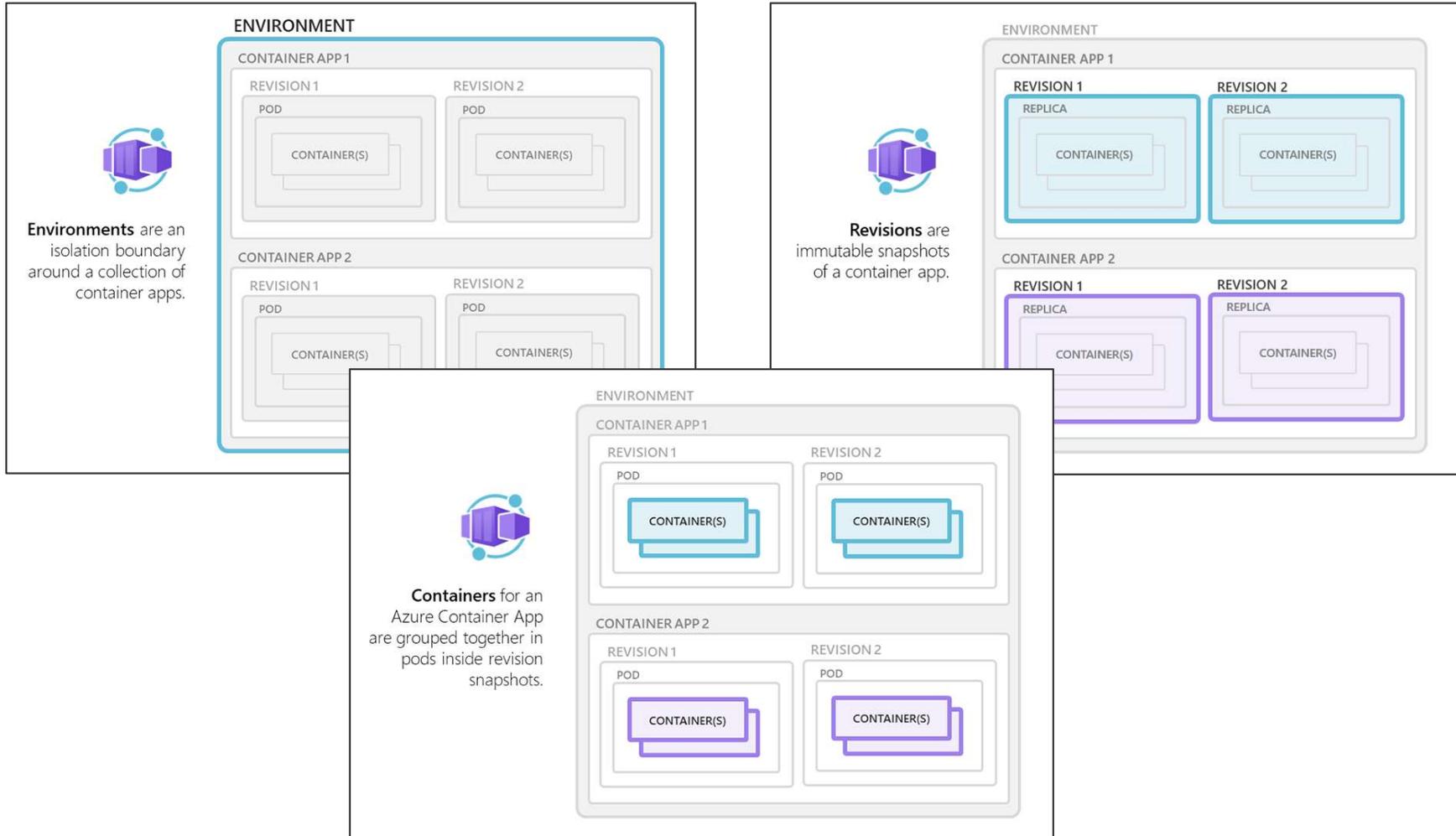
Deploy and manage a microservices architecture with the option to integrate with Dapr.

AUTO-SCALE CRITERIA

Individual microservices can scale according to any KEDA scale triggers.



Azure Container Apps





KEDA (K8S Event Driven Autoscaling)

KEDA

implementa l'autoscaling event-driven nel "linguaggio di Kubernetes"

Docker + KEDA

il runtime è fornito in un container, KEDA implementa lo scale controller

0...N scaling

fornendo metriche custom per Kubernetes Horizontal Pod Autoscaler
il numero di pod è zero se non ci sono eventi che triggerano un deployment

cresce dinamicamente in base agli eventi, per poi tornare di nuovo a zero

Si replica il modello serverless event-driven su un cluster K8s



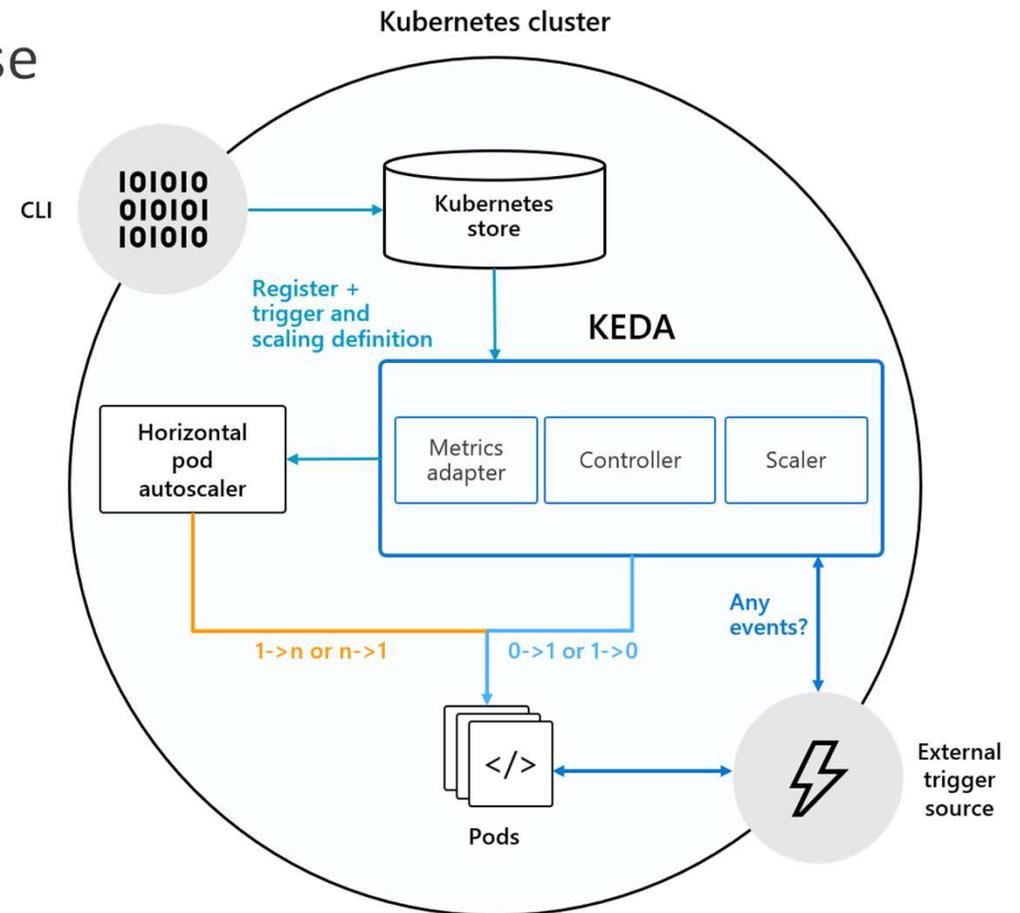
Come funziona

attiva o disattiva i deployment in base al carico di eventi

fa da **Metrics Server** esponendo metriche custom (lunghezza della coda...) che guidano l'HPA

l'evento può essere **consumato direttamente** e questo preserva l'integrazione, senza perdere nulla dell'evento stesso

diversi scaler (ASB RabbitMQ)





ScaledObject

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: {scaled-object-name}
  annotations:
    scaledobject.keda.sh/transfer-hpa-ownership: "true"
    validations.keda.sh/hpa-ownership: "true"
    autoscaling.keda.sh/paused: "true"
spec:
  scaleTargetRef:
    apiVersion: {api-version-of-target-resource}
    kind: {kind-of-target-resource}
    name: {name-of-target-resource}
    envSourceContainerName: {container-name}
  pollingInterval: 30
  initialCooldownPeriod: 0
  cooldownPeriod: 300
  idleReplicaCount: 0
  minReplicaCount: 1
  maxReplicaCount: 100
  fallback:
    failureThreshold: 3
    replicas: 6
  advanced:
    restoreToOriginalReplicaCount: true/false
    horizontalPodAutoscalerConfig:
      name: {name-of-hpa-resource}
      behavior:
        scaleDown:
          stabilizationWindowSeconds: 300
    policies:
      - type: Percent
        value: 100
        periodSeconds: 15
  triggers:
    # {list of triggers to activate scaling of the target}
```

Deployment (ma non solo)
definisce il target che verrà scalato

Polling
ogni quanto tempo devo riverificare la condizione del trigger? e quanto voglio essere prudente sullo scale-down?

Trigger
in base al tipo e ad una condizione scatenata lo scaling
posso avere più di un trigger
ne basta uno per lo scale-up, devono essere soddisfatti tutti per lo scale-down



Azure Container Jobs



A **job** defines its configuration, such as trigger type and container.

A **job execution** is a running instance of a job.

ENVIRONMENT

JOB 1

JOB EXECUTION 1

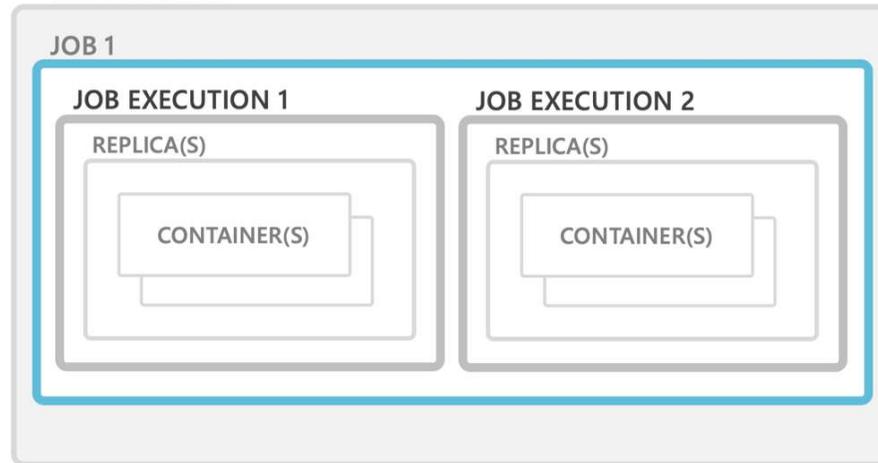
REPLICA(S)

CONTAINER(S)

JOB EXECUTION 2

REPLICA(S)

CONTAINER(S)





ScaledJob

```
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: {scaled-job-name}
  labels:
    my-label: {my-label-value}
  annotations:
    autoscaling.keda.sh/paused: true
    my-annotation: {my-annotation-value}
spec:
  jobTargetRef:
    parallelism: 1
    completions: 1
    activeDeadlineSeconds: 600
    backoffLimit: 6
    template:
      # describes the [job template](https://
  pollingInterval: 30
  successfulJobsHistoryLimit: 5
  failedJobsHistoryLimit: 5
  envSourceContainerName: {container-name}
  minReplicaCount: 10
  maxReplicaCount: 100
  rolloutStrategy: gradual
  rollout:
    strategy: gradual
    propagationPolicy: foreground
  scalingStrategy:
    strategy: "custom"
    customScalingQueueLengthDeduction: 1
    customScalingRunningJobPercentage: "0.5"
    pendingPodConditions:
      - "Ready"
      - "PodScheduled"
      - "AnyOtherCustomPodCondition"
    multipleScalersCalculation: "max" # Opti
  triggers:
    # {list of triggers to create jobs}
```

Long running task

i deployment non funzionano in caso di processi lunghi, voglio evitare che HPA uccida il pod quando magari gli manca poco per finire (HPA non sa nulla del lavoro da fare)

Graceful shutdown

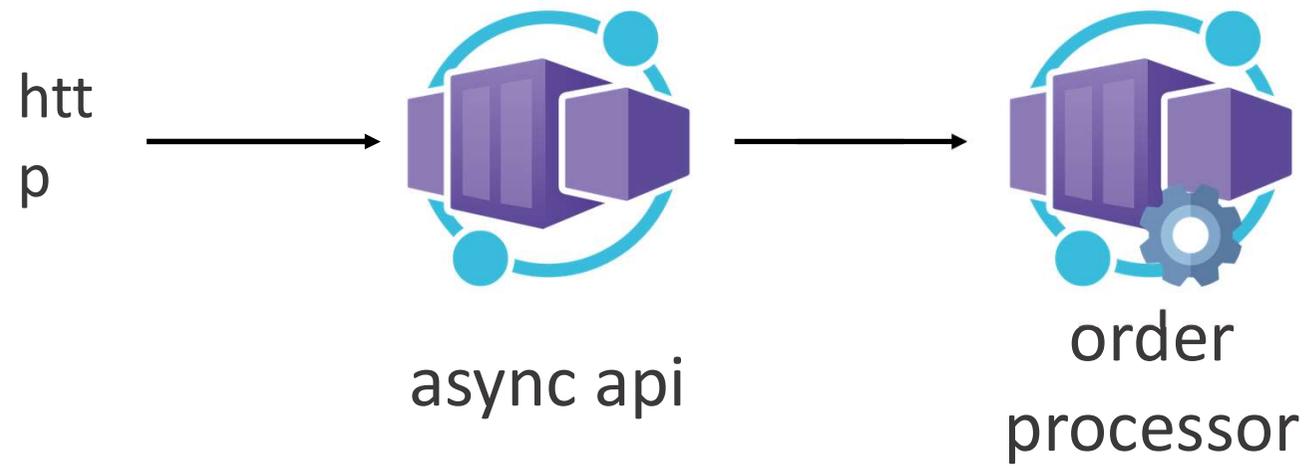
posso contare sugli eventi di lifecycle di K8s (pre-hook, SIGTERM...) per posticipare

KEDA + Job

oppure utilizzo ScaledJob per creare 1 job per ogni evento, processarlo, e solo alla fine essere sicuro di dismettere il pod

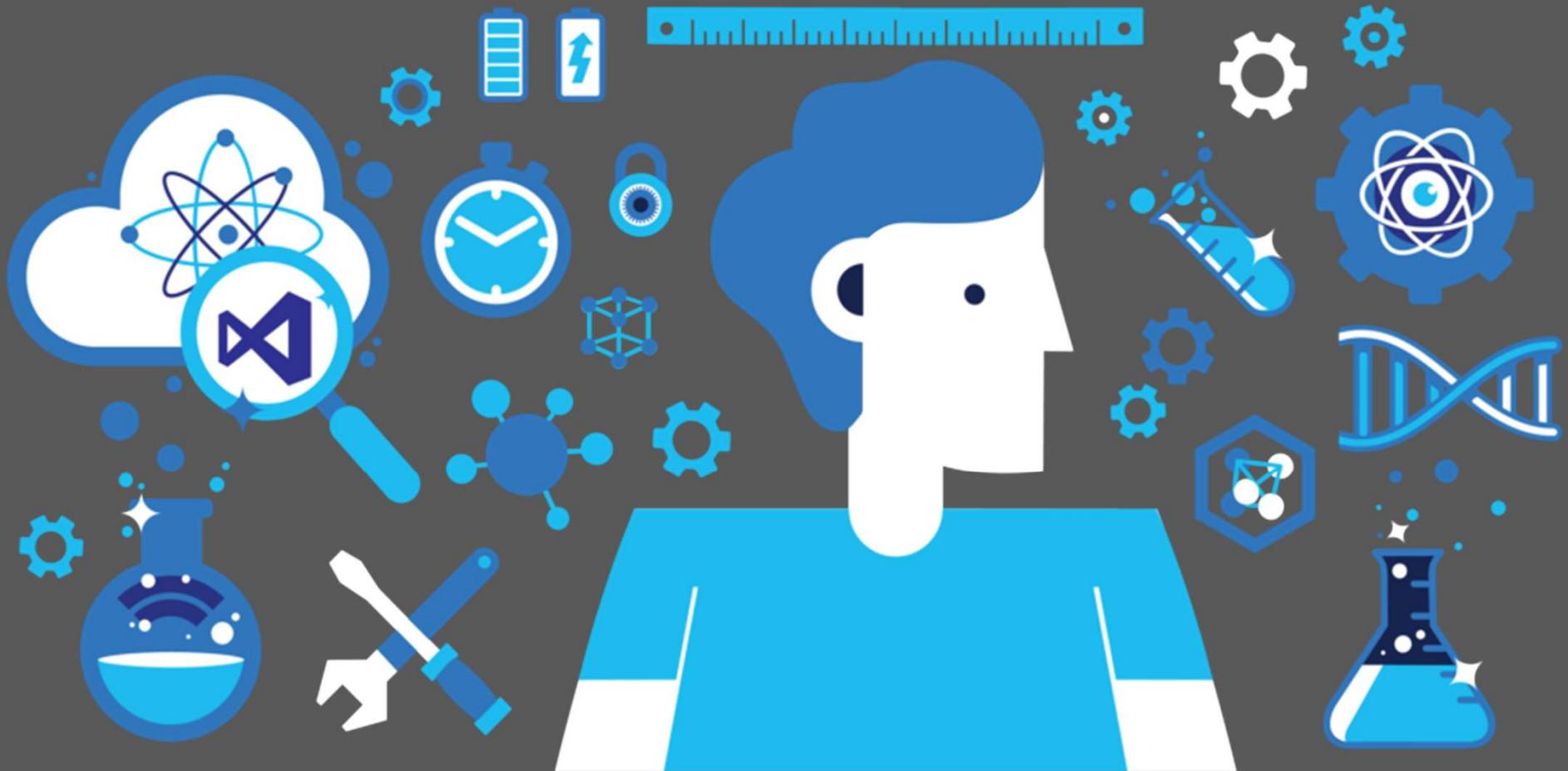


Crazy Bike Shop





Demo





az cli

```
az extension add --name containerapp --upgrade
```

```
az provider register --namespace Microsoft.App
```

```
az provider register --namespace Microsoft.OperationalInsights
```

```
az containerapp -h
```

<https://docs.microsoft.com/it-it/cli/azure/containerapp?view=azure-cli-latest>



az container app job start Q&A



Links

<https://github.com/andrekiba/CrazyBikeShop>

<https://docs.microsoft.com/en-us/azure/container-apps/overview>

<https://learn.microsoft.com/en-us/azure/container-apps/jobs?tabs=azure-cli>

<https://keda.sh>

<https://keda.sh/docs/2.15/concepts/scaling-deployments/#scaling-deployments-and-statefulsets>

<https://keda.sh/docs/2.15/concepts/scaling-jobs/>

<https://docs.microsoft.com/en-us/cli/azure/containerapp?view=azure-cli-latest>

<https://github.com/Azure/azure-sdk-for-net/tree/main/sdk/containerapps/Azure.ResourceManager.AppContainers>



@andrekiba6



andrea.ceroni@gmail.com



<https://github.com/andr>

ekiba



<https://www.linkedin.com/in/and>

reaceroni



elfo.net

ELFO



klabcommunity.org

