Serverless Microservices with Azure Container Apps

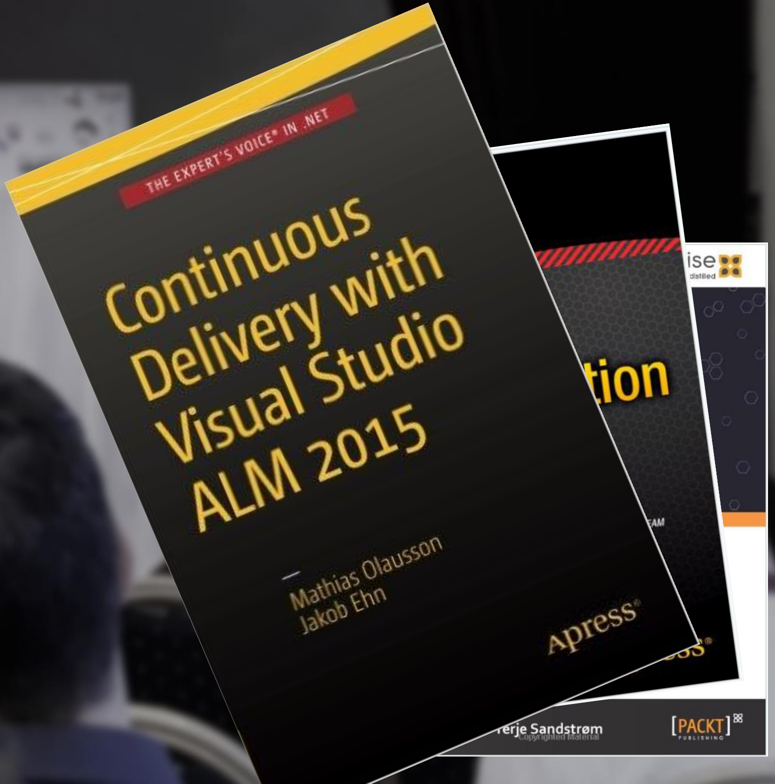
Cloudburst 2022

Jakob Ehn
@jakobehn
https://blog.ehn.nu

Jakob Ehn
Microsoft Azure MVP

https://blog.ehn.nu

@jakobehn



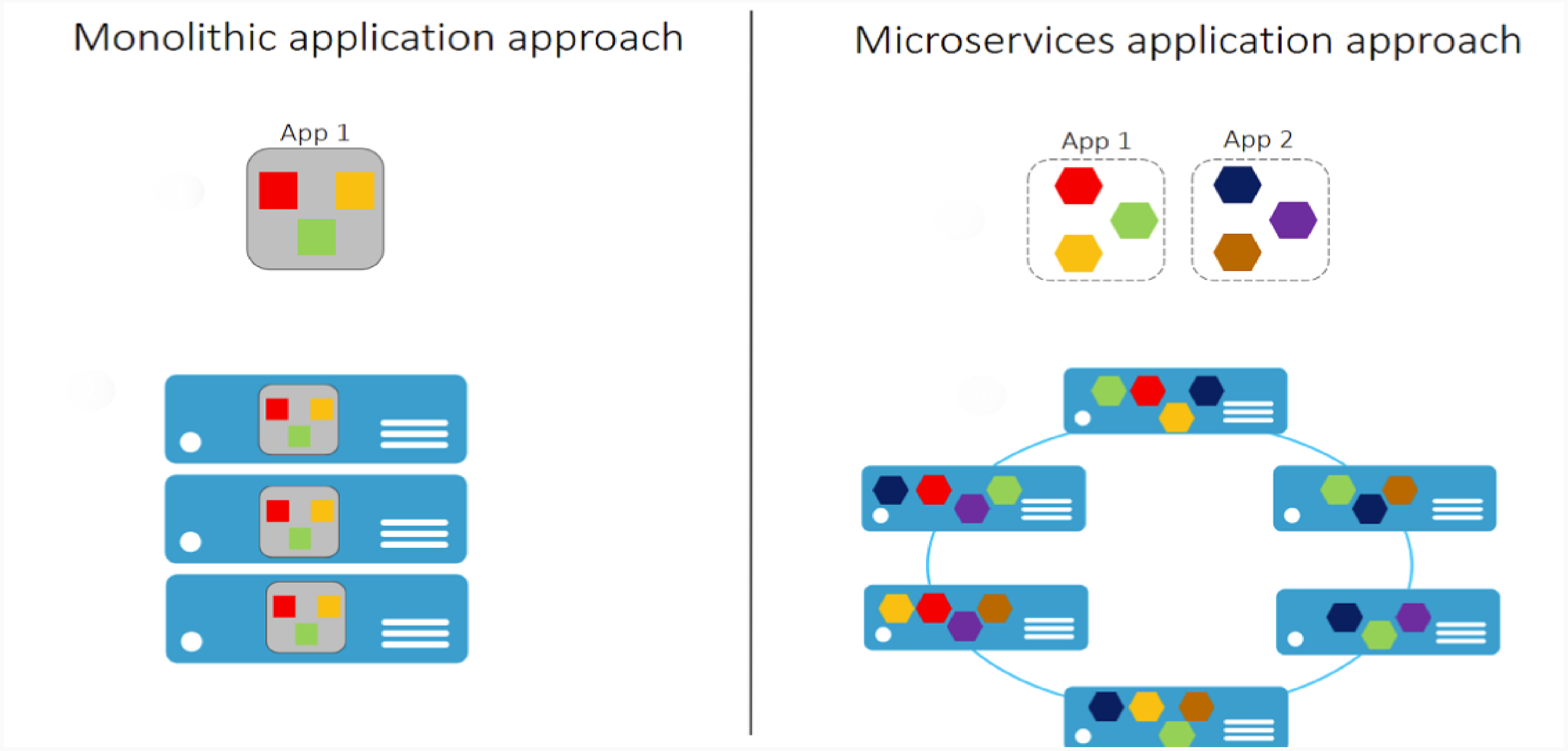
THE EXPERT'S VOICE® IN .NET

Continuous Delivery with Visual Studio ALM 2015

Mathias Olausson
Jakob Ehn

Apress®

Microservices and Containers


Monolithic application approach
App 1
Microservices application approach
App 1
App 2

✓ With Microservices every part of the application is deployed as a fully self-contained component

Running Multi-Container Apps

What do we need?

- Service discovery and service-to-service communication
- Zero downtime deployments
- Autoscale apps on metrics and events
- Monitoring and distributed tracing
- Don't want to care about infrastructure
- Pay for what we use

Options for running Containers in Azure

Azure Container Instances

Azure App Service

Azure Kubernetes Service

Azure Container Apps

Azure Container Apps

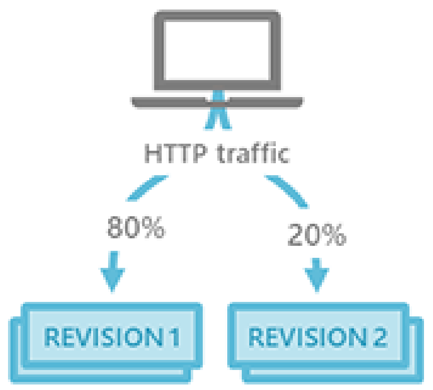
"Serverless containers for microservices"

- Build modern apps on open source
- Focus on apps, not infrastructure
- Language/Framework agnostic
- Pay for what you use

kubernetes KEDA dapr envoy

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.

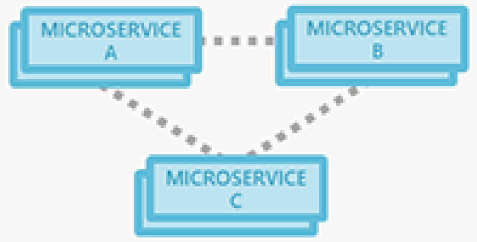
EVENT-DRIVEN PROCESSING

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.

Deploying a container app - CLI

Create environment

```
az containerapp env create  --name my-env
                             --resource-group myGroup
                             --logs-workspace-id myWorkspaceId
                             --logs-workspace-key myWorkspaceKey
                             --location myLocation
```

Create container app

```
az containerapp create --name my-app
                       --resource-group myGroup
                       --environment my-env
                       --image mcr.microsoft.com/azuredoc/azurecontainerapps-hellworld:latest
                       --target-port 80
                       --ingress 'external'
                       --query configuration.ingress.fqdn
```

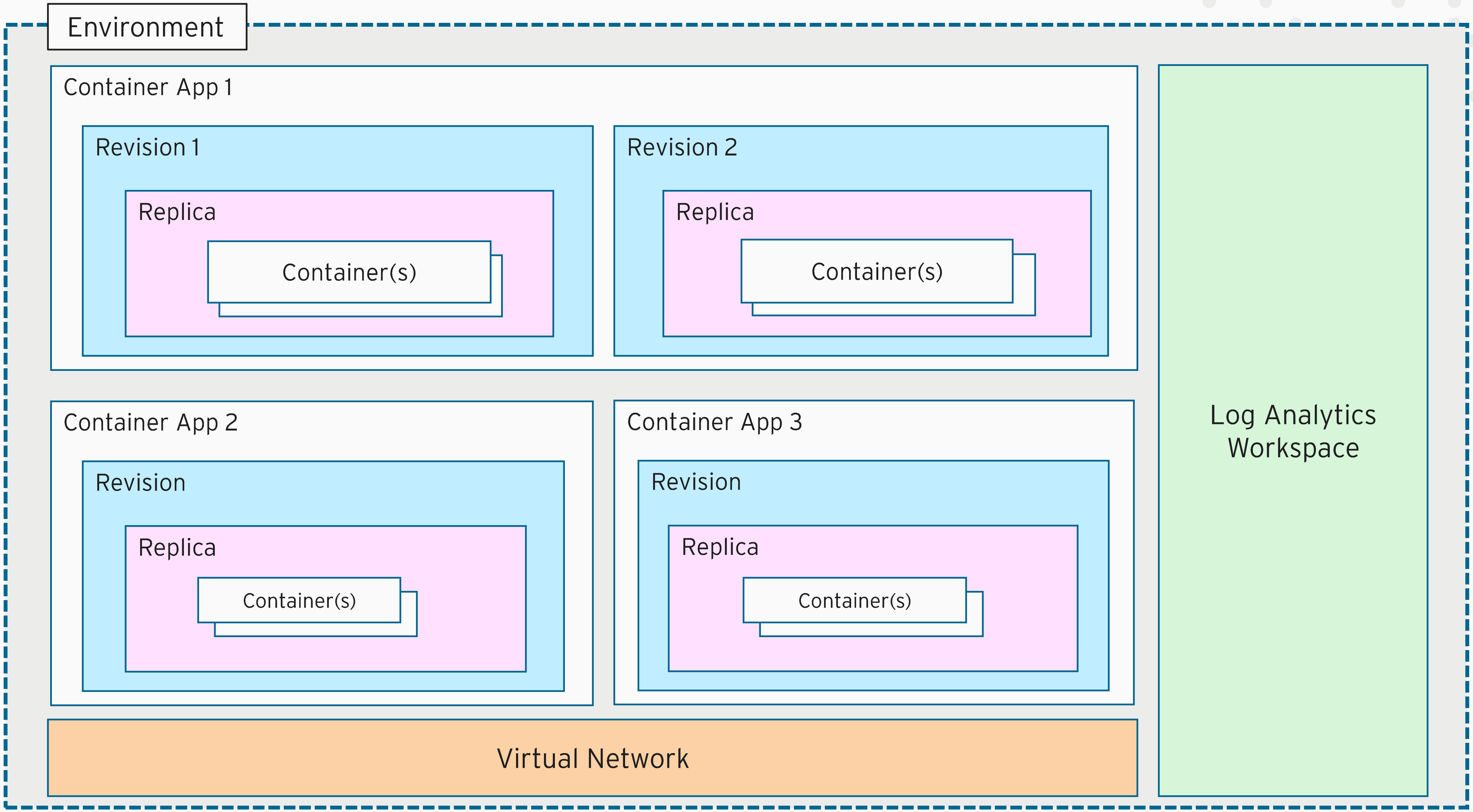
Deploying a container app - ARM/Bicep

```
resource app 'Microsoft.App/containerApps@2022-03-01' = {
  name: 'mycontainerapp'
  location: location
  properties: {
    managedEnvironmentId: resourceId('Microsoft.App/managedEnvironments', environment_name)
    configuration: {
      ingress: {
        external: true
        targetPort: 5000
      }
    }
    template: {
      containers: [
        {
          image: 'mcr.microsoft.com/azuredocs/containerapps-helloworld:latest'
          name: 'hello-k8s-node’
        }
      ]
      scale: {
        minReplicas: 1
        maxReplicas: 10
      }
   …
```

DEMO

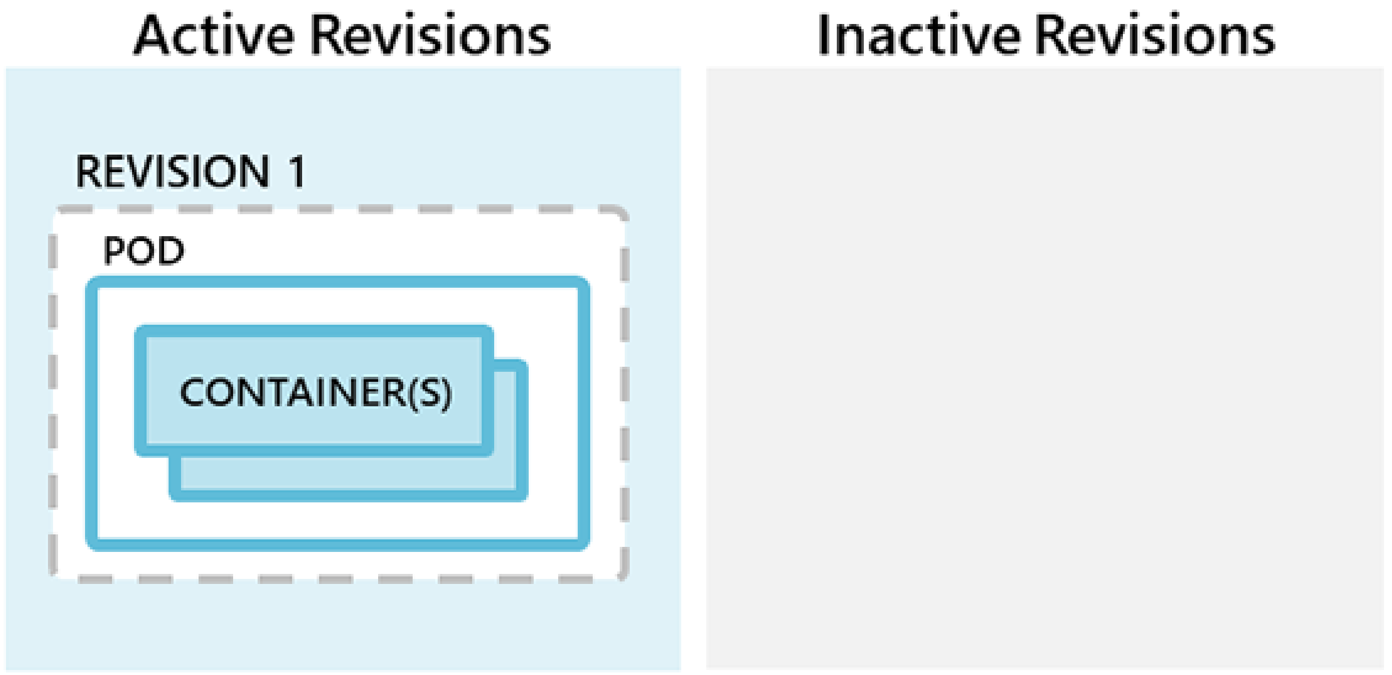
Deploying an Azure Container App

Azure Container Apps - Concepts



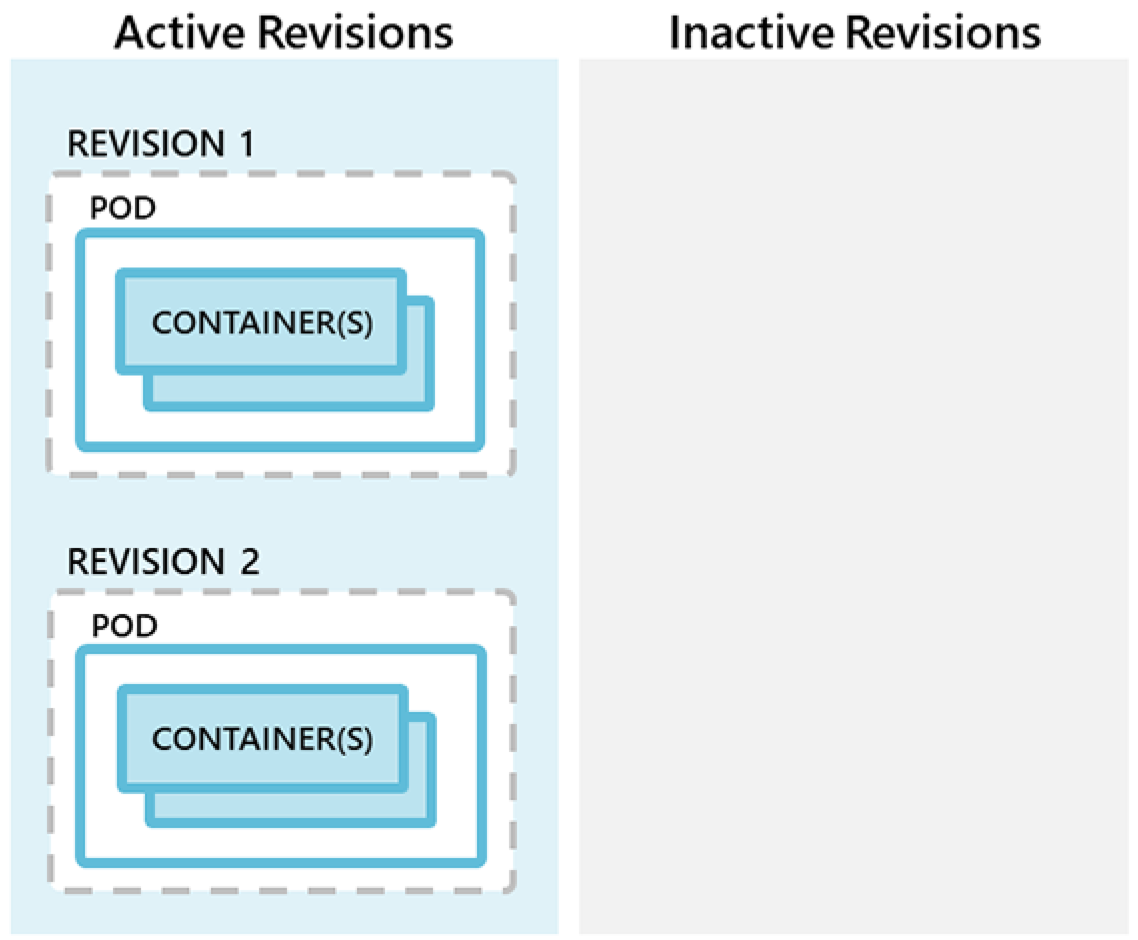
Revisions

Upon **deployment**, the first revision is automatically created.



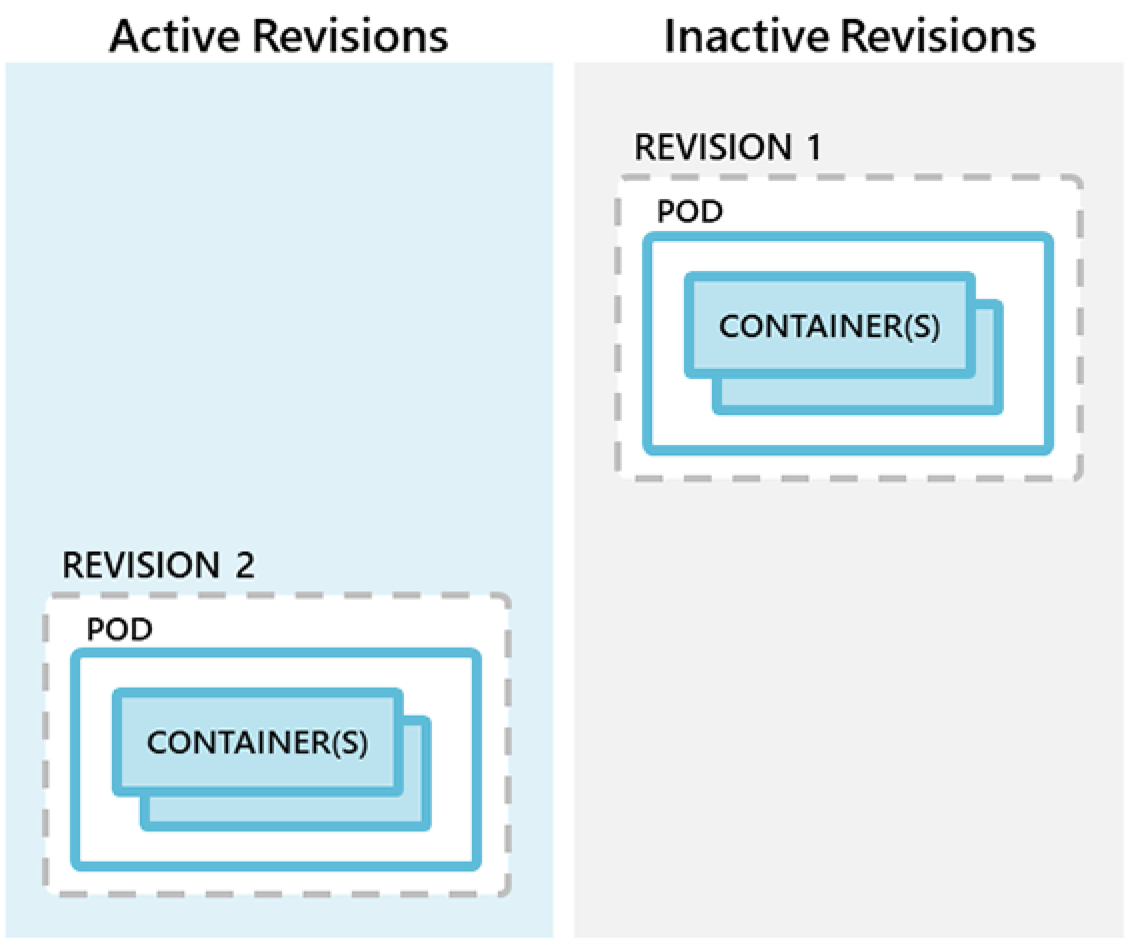
Revisions

As the container app is **updated**, a new revision is created.



Revisions

Once a revision is no longer needed, you can **deactivate** individual revisions, or choose to automatically deactivate old revisions.



Revisions

<container app name>--<revision suffix>

Revision-scope changes

- Revision suffix
- Container configuration and images
- Scale rules
- …

Application-scope changes

- Secret valus
- Ingress configuration
- Credentials for private container registries
- …

Revisions - Traffic splitting

NTAINER(S)
NTAINER(S)

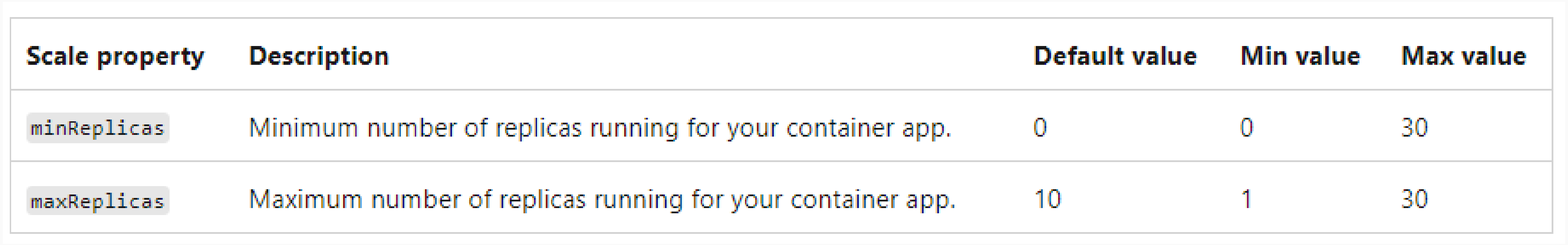
```
{
  ...
  "configuration": {
    "ingress": {
      "traffic": [
        {
          "revisionName": <REVISION1_NAME>,
          "weight": 50
        },
        {
          "revisionName": <REVISION2_NAME>,
          "weight": 30
        },
        {
          "latestRevision": true,
          "weight": 20
        }
      ]
    }
  }
}
```

DEMO

Azure Container App Revisions

Scaling Azure Container Apps

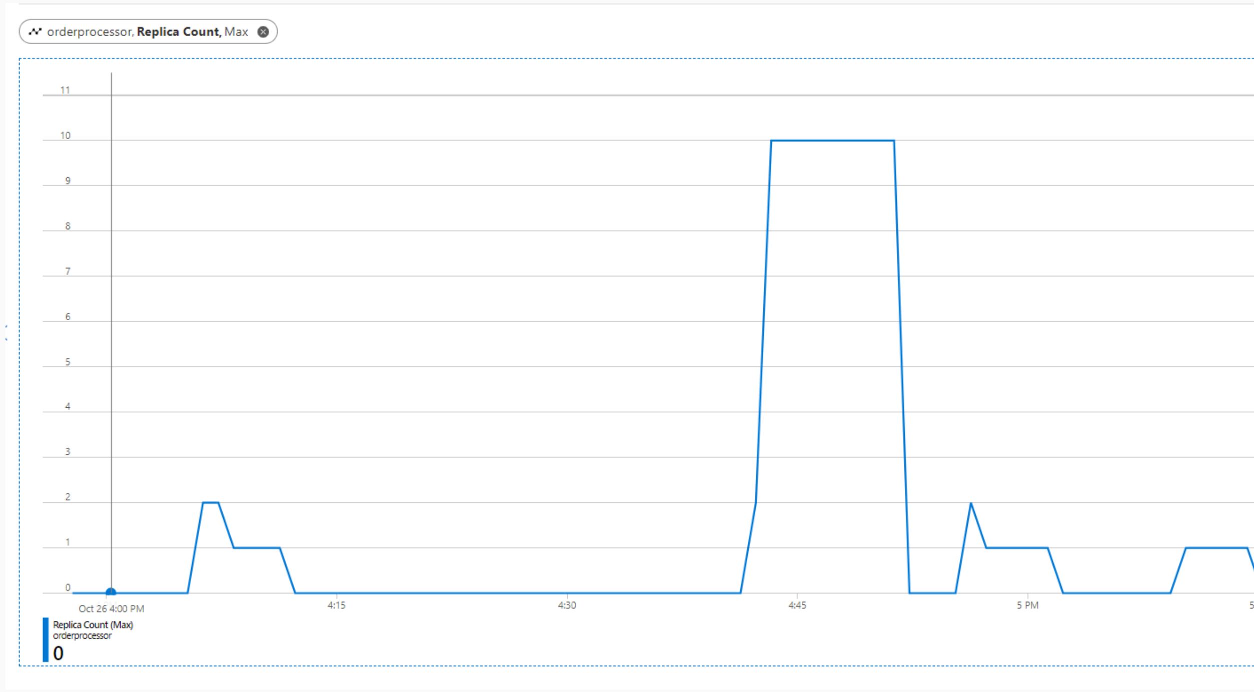
- Automatic horizontal scaling through scaling rules
- Scale triggers
  - HTTP/TCP
  - Event-driven (queues, storage, event hubs, redis...)
  - CPU/Memory
- Uses KEDA under the hood



Scale property	Description	Default value	Min value	Max value
`minReplicas`	Minimum number of replicas running for your container app.	0	0	30
`maxReplicas`	Maximum number of replicas running for your container app.	10	1	30

HTTP traffic scaling

```
scale: {
       minReplicas: 0
       maxReplicas: 10
       rules: [
         {
           name: 'http-autoscale'
           http: {
             metadata: {
               concurrentRequests: '50'
             }
           }
         }
       ]
     }
```



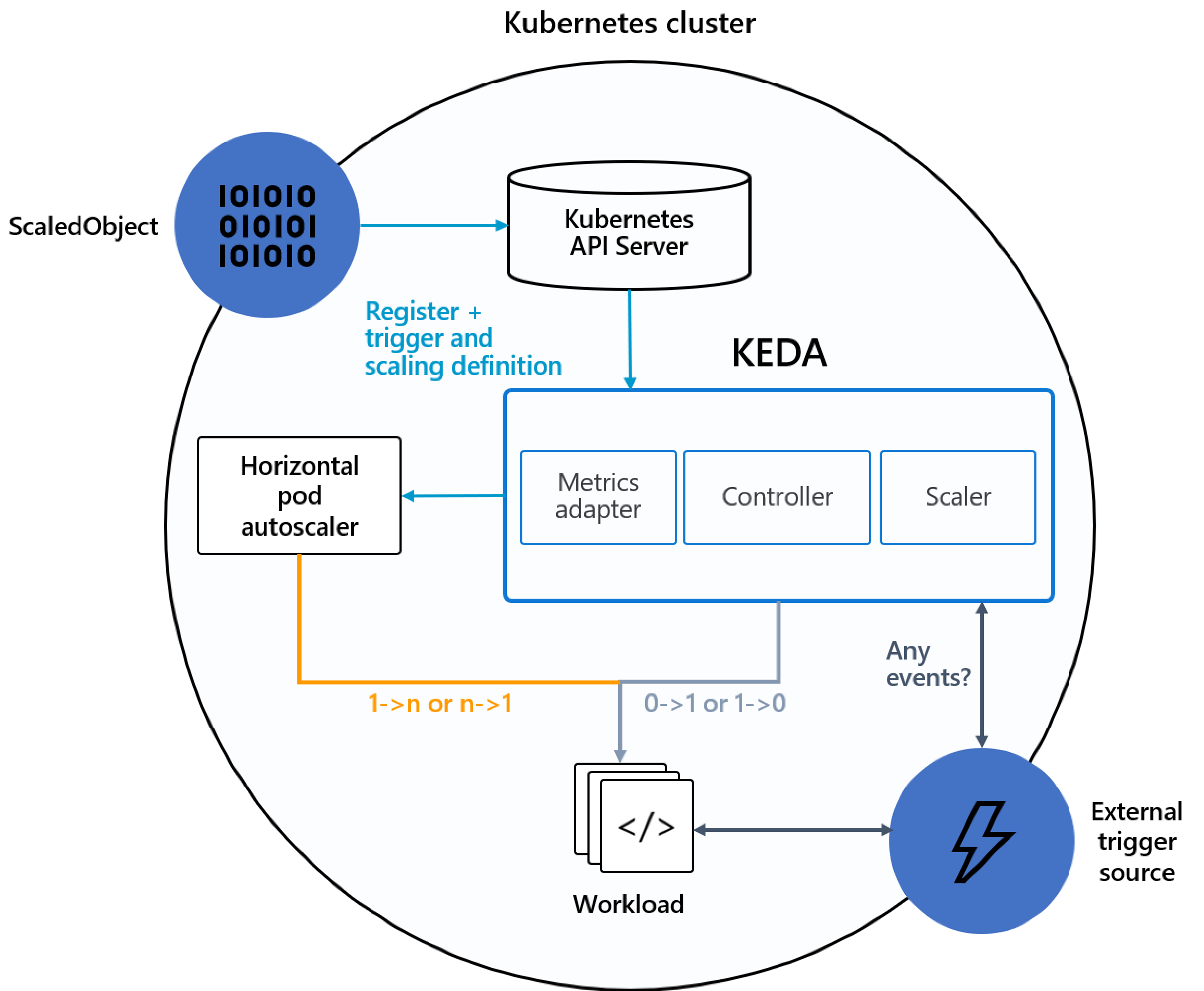
Event-driven scaling

```
scale: {
        minReplicas: 0
        maxReplicas: 10
        rules: [
          {
            name: 'queue-based-autoscaling'
            custom: {
              type: 'azure-servicebus'
              metadata: {
                topicName: 'ordercreated'
                subscriptionName: 'orderprocessor'
                queueLength: ’20'
              }
              auth: [
                {
                  secretRef: 'servicebus-connectionstring'
                  triggerParameter: 'connection'
                }
              ]
            }
          }
```

KEDA

- Open source component for event-driven scaling in Kubernetes
- Provides 30+ built-in scalers
- Scale to zero or to thousands
- Run and scale Azure Functions in Kubernetes

https://keda.sh/



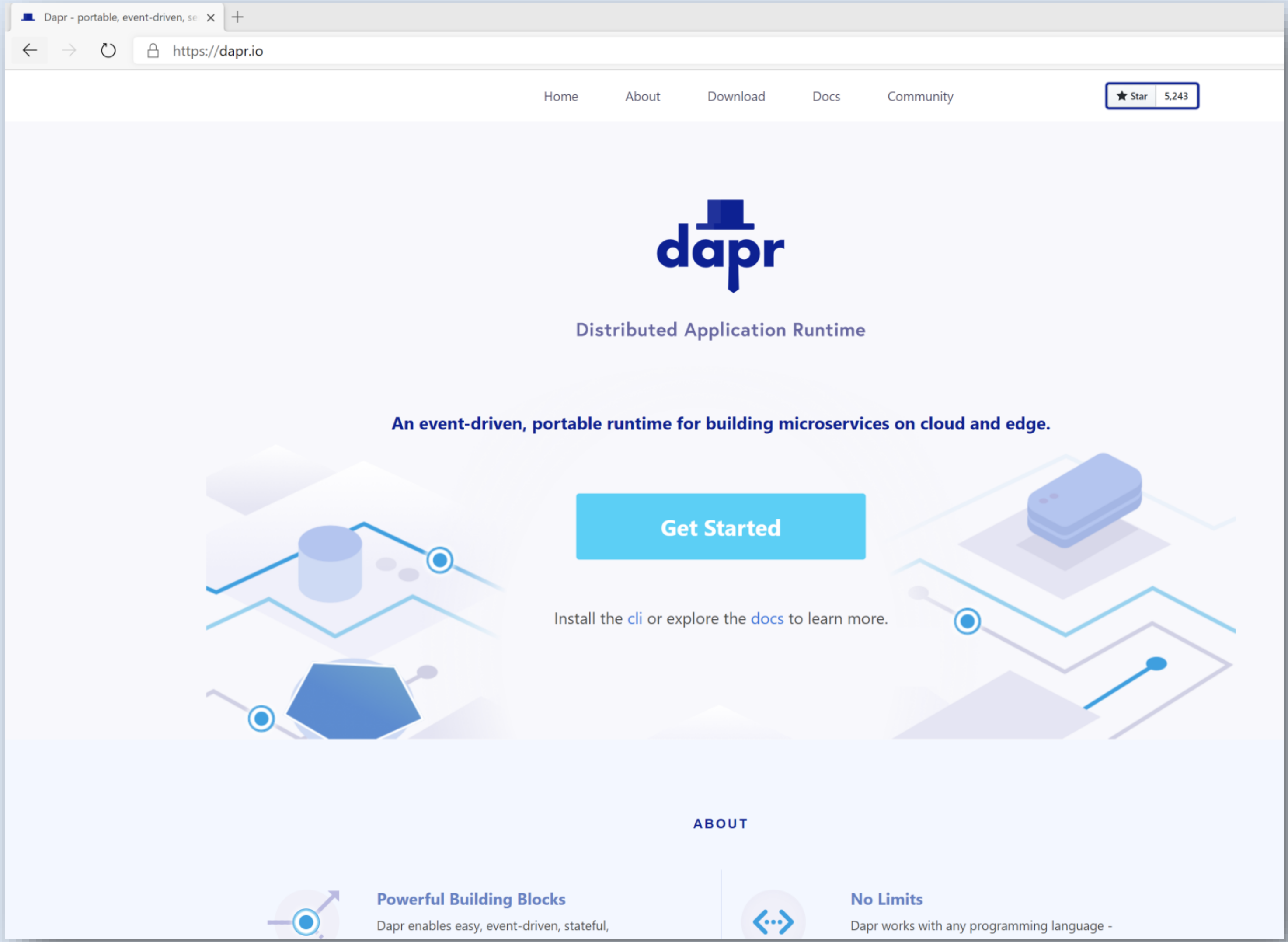
KEDA Scalers

- Apache Kafka
- AWS CloudWatch
- AWS Simple Queue Service
- Azure Event Hub
- Azure Monitor
- Azure Service Bus Queues & Top
- Azure Storage Queues
- GCP PubSub
- IBM MQ
- Influx DB
- Kafka
- Liiklus
- MongoDB
- MySQL
- Nats Streaming
- Prometheus
- RabbitMQ
- Redis Lists, Streams
- …

Distributed Application Runtime

Portable, event-driven, runtime for building distributed applications across cloud and edge

https://dapr.io



Dapr - portable, event-driven, se

https://dapr.io

Home About Download Docs Community ★ Star 5,243

Distributed Application Runtime

An event-driven, portable runtime for building microservices on cloud and edge.

Get Started

Install the cli or explore the docs to learn more.

ABOUT

Powerful Building Blocks

Dapr enables easy, event-driven, stateful,

No Limits

Dapr works with any programming language -

Dapr - Microservice building blocks

Service-to-service invocation

Perform direct, secure, service-to-service method calls

State management

Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

Resource bindings and triggers

Trigger code through events from a large array of inputs

Output bindings to external resources including databases and queues

Secrets

Securely access secrets from your application

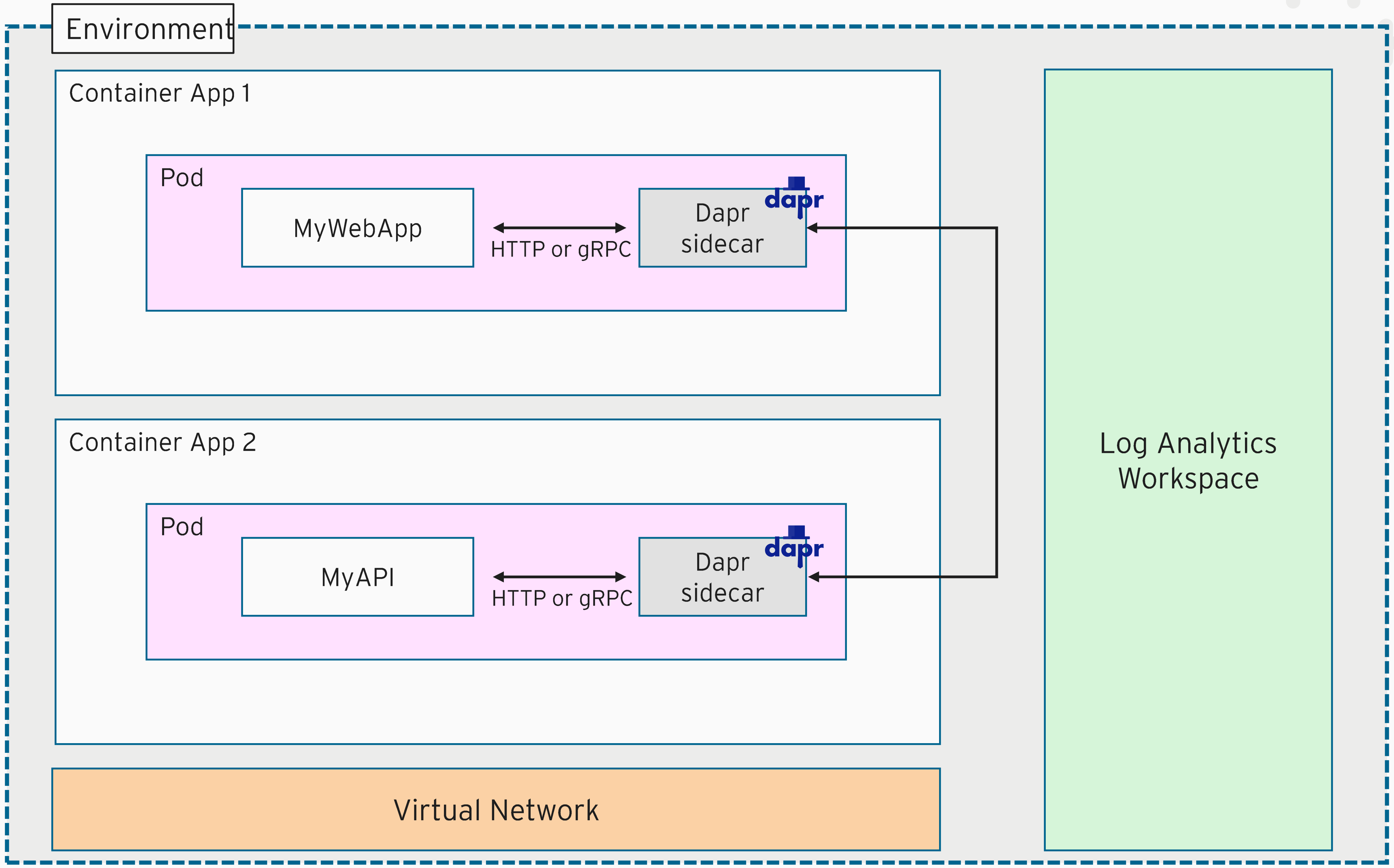
Distributed tracing

See and measure the message calls across components and networked services

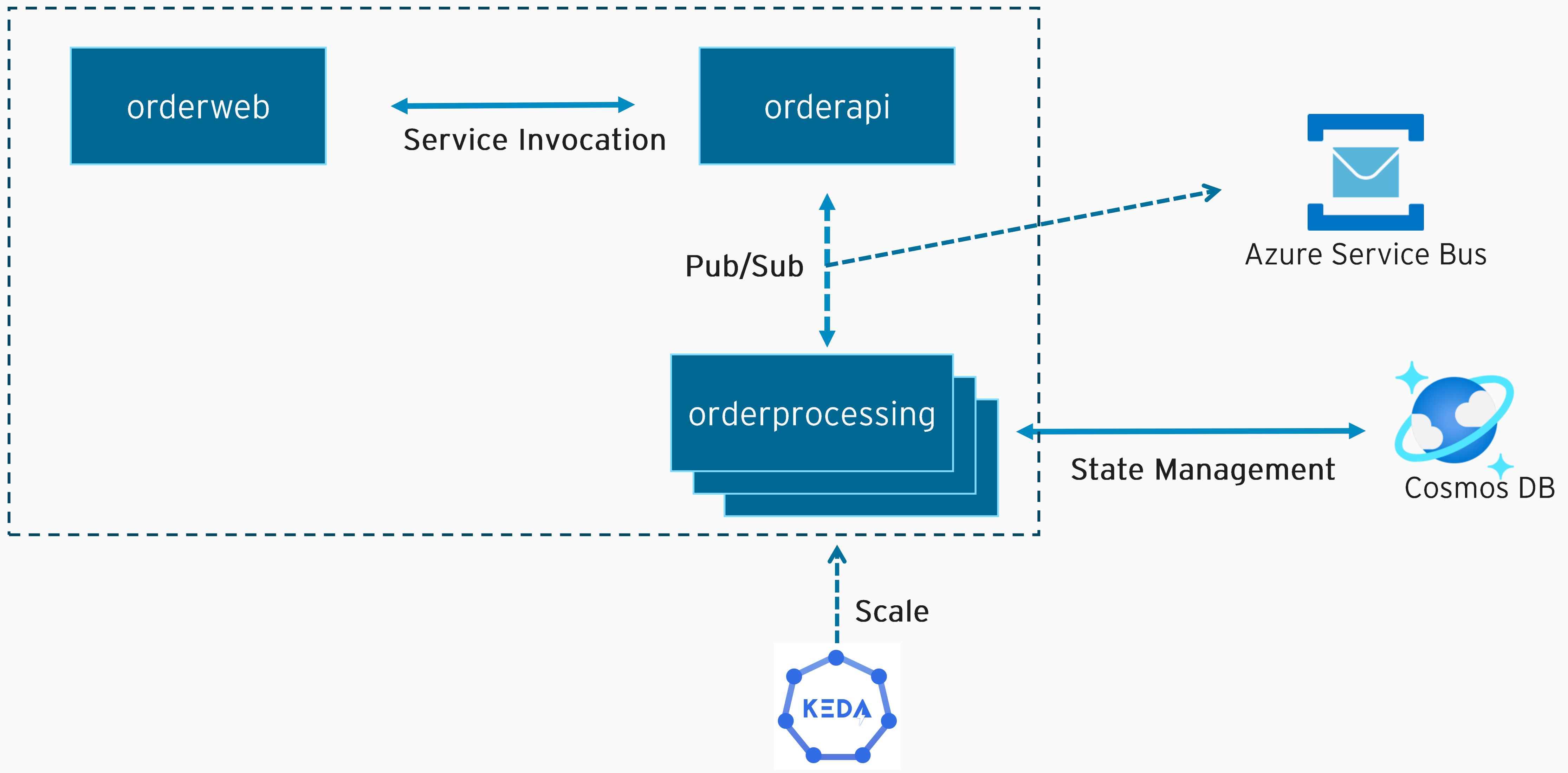
Actors

Encapsulate code and data in reusable actor objects as a common microservices design pattern

Azure Container Apps – With Dapr



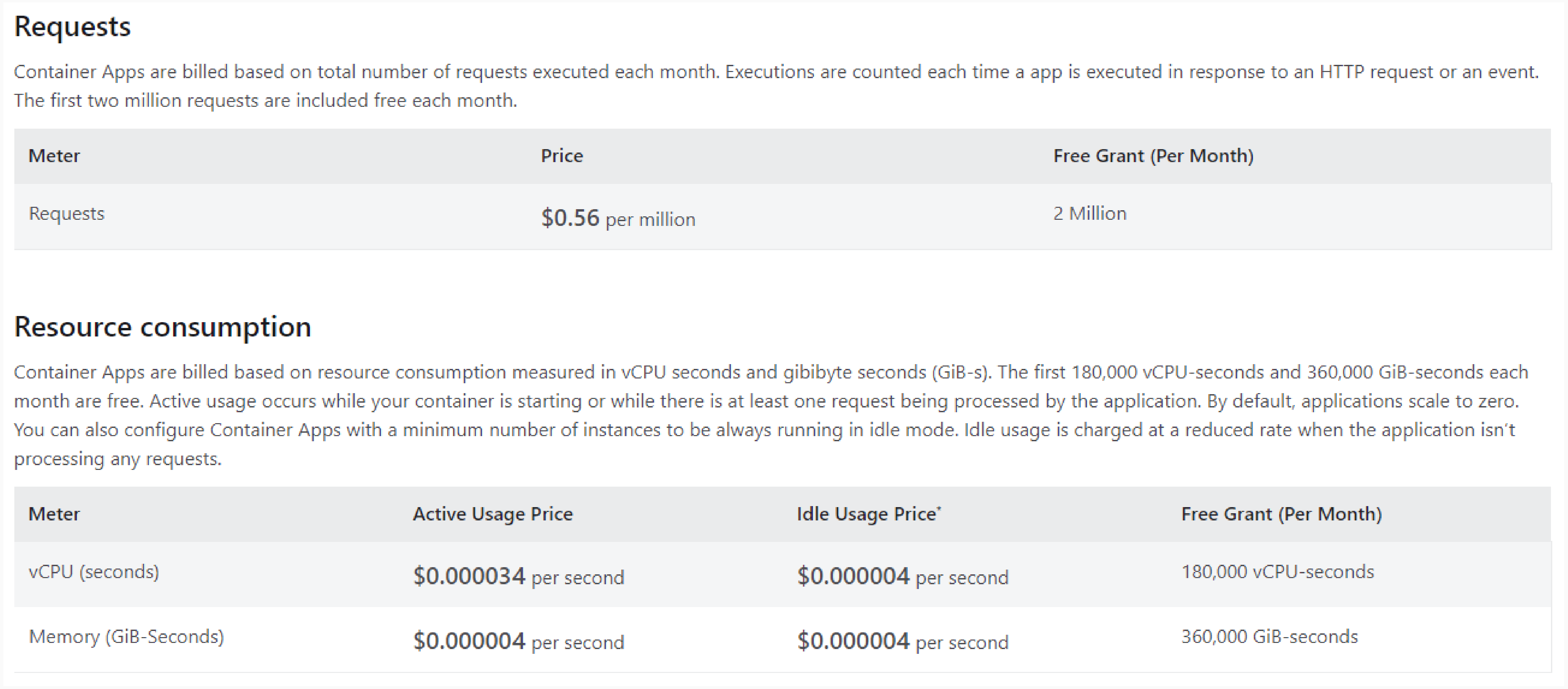
DEMO


orderweb
Service Invocation
orderapi
Pub/Sub
Azure Service Bus
orderprocessing
State Management
Cosmos DB
Scale
KEDA

Azure Container Apps

Dapr + scaling

Azure Container Apps - Pricing



Requests

Container Apps are billed based on total number of requests executed each month. Executions are counted each time a app is executed in response to an HTTP request or an event. The first two million requests are included free each month.

Meter	Price	Free Grant (Per Month)
Requests	**$0.56** per million	2 Million

Resource consumption

Container Apps are billed based on resource consumption measured in vCPU seconds and gibibyte seconds (GiB-s). The first 180,000 vCPU-seconds and 360,000 GiB-seconds each month are free. Active usage occurs while your container is starting or while there is at least one request being processed by the application. By default, applications scale to zero. You can also configure Container Apps with a minimum number of instances to be always running in idle mode. Idle usage is charged at a reduced rate when the application isn't processing any requests.

Meter	Active Usage Price	Idle Usage Price*	Free Grant (Per Month)
vCPU (seconds)	**$0.000034** per second	**$0.000004** per second	180,000 vCPU-seconds
Memory (GiB-Seconds)	**$0.000004** per second	**$0.000004** per second	360,000 GiB-seconds

Thank you!

Sample code available at:
https://github.com/jakobehn/azurecontainerapp-demo

Jakob Ehn
@jakobehn
https://blog.ehn.nu