Azure Container Apps

Championing Azure – Cloud Native

Jakob Ehn

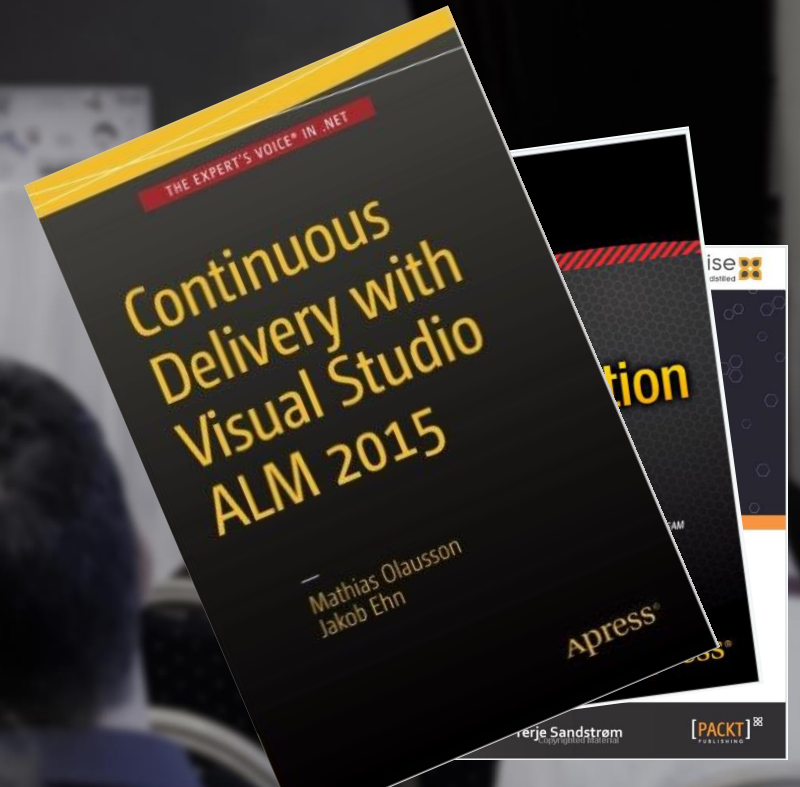@jakobehn

https://blog.ehn.nu

active
SOLUTION

Jakob Ehn
Microsoft Azure MVP

https://blog.ehn.nu

@jakobehn

Microservices and Containers

Monolithic application approach | Microservices application approach
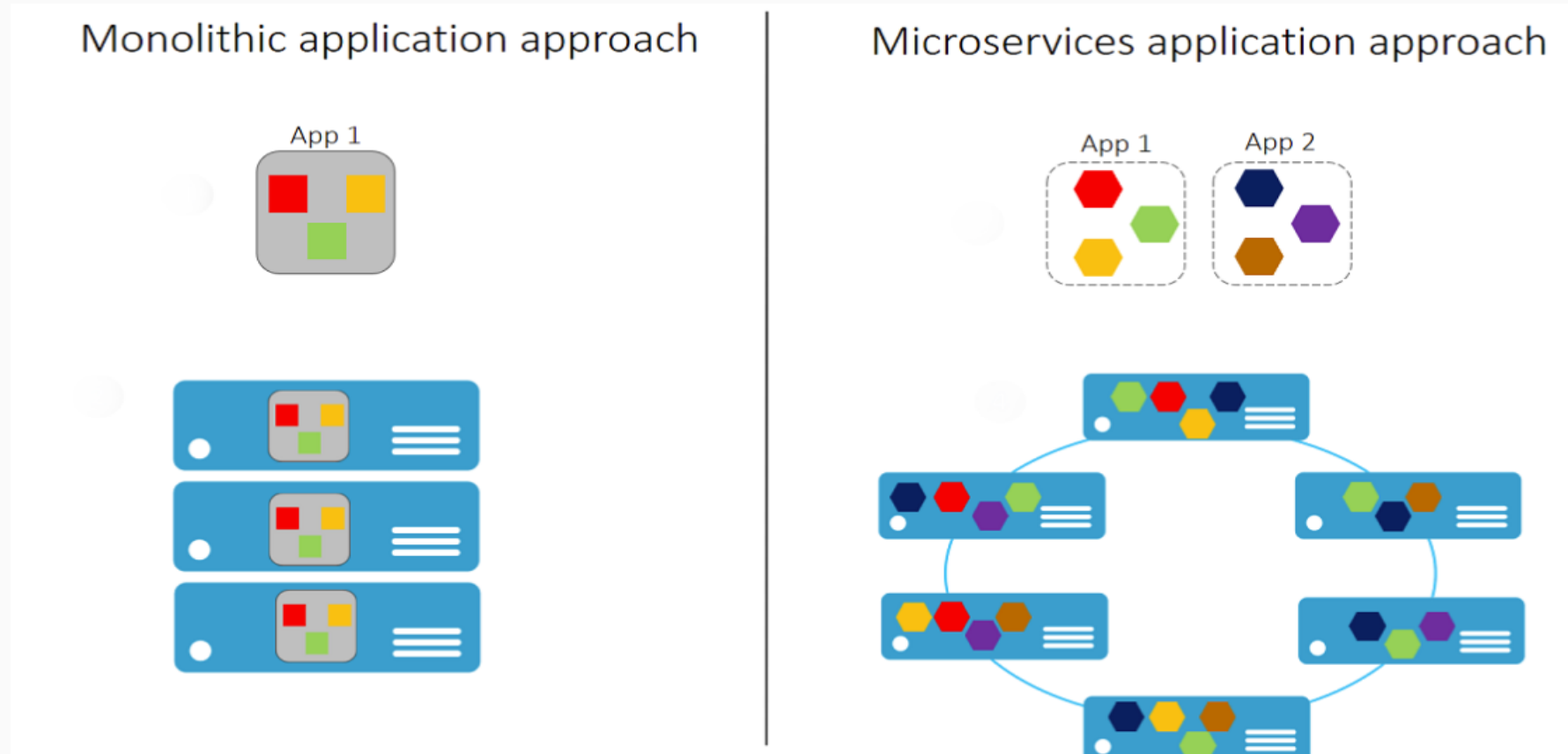
✔ With Microservices every part of the application is deployed as a fully self-contained component

Running Multi-Container Apps

What do we need?

- Service discovery and service-to-service communication
- Deploy new versions without downtime
- 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
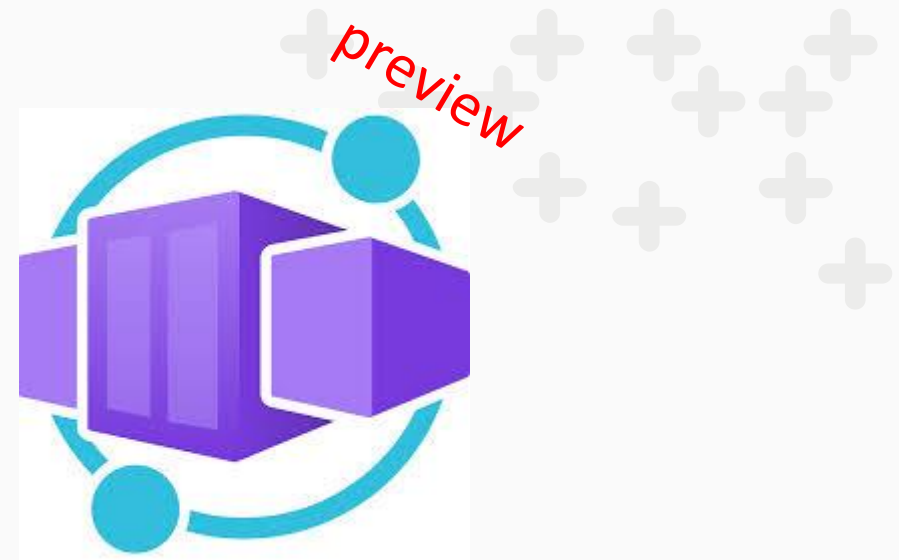

preview
Azure Container Apps

Azure Container Apps

preview

Serverless containers for microservices

- Build modern apps on open source
- Focus on apps, not infrastructure
- Language/Framework agnostic
- Seamlessly upgrade to Kubernetes

Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS

HTTP traffic

80%   20%

REVISION 1   REVISION 2

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

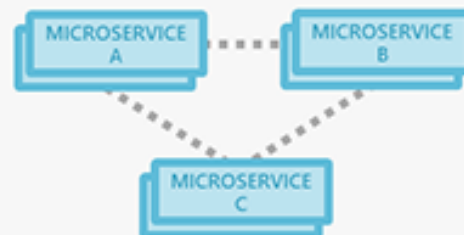A queue reader application that processes messages as they arrive in a queue.

AUTO-SCALE CRITERIA

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

MICROSERVICES

MICROSERVICE A   MICROSERVICE B
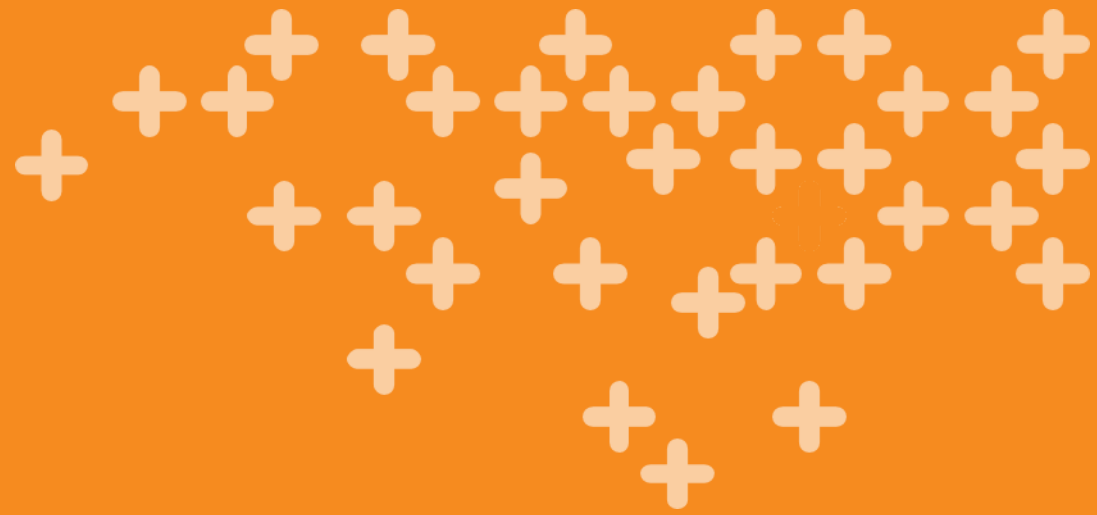
MICROSERVICE C

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 my-container-image:1.0
--target-port 80
--ingress 'external'
--query configuration.ingress.fqdn

Deploying a container app – ARM/Bicep
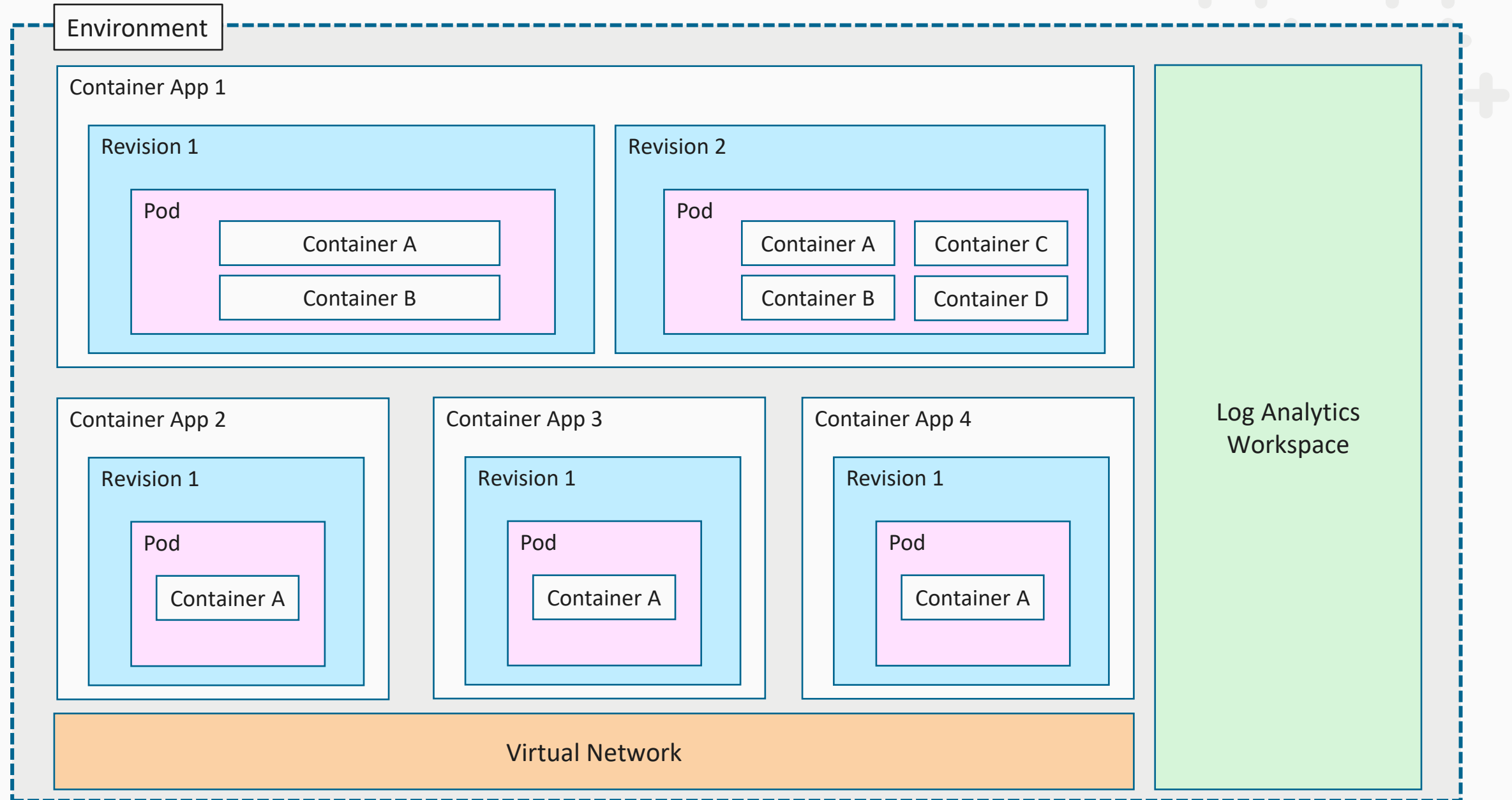
```
resource nodeapp 'Microsoft.Web/containerapps@2021-03-01' = {
  name: 'nodeapp'
  kind: 'containerapp'
  location: location
  properties: {
    kubeEnvironmentId: resourceId('Microsoft.Web/kubeEnvironments', environment_name)
    configuration: {
      ingress: {
        external: true
        targetPort: 3000
      }
    }
    template: {
      containers: [
        {
          image: 'dapriosamples/hello-k8s-node:latest'
          name: 'hello-k8s-node'
          resources: {
            cpu: '0.5'
            memory: '1Gi'
          }
        }
      ]
      scale: {
        minReplicas: 1
        maxReplicas: 1
```
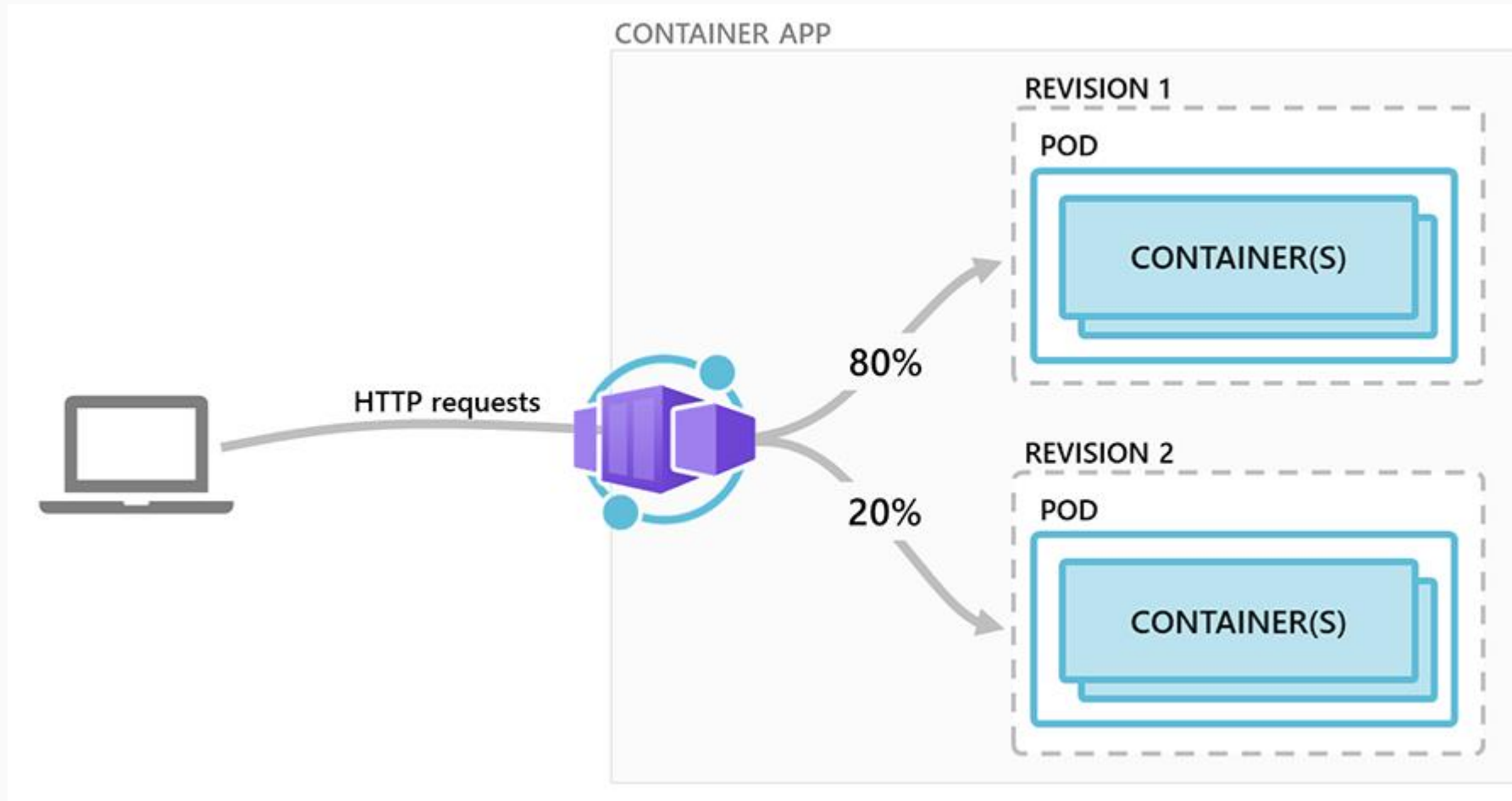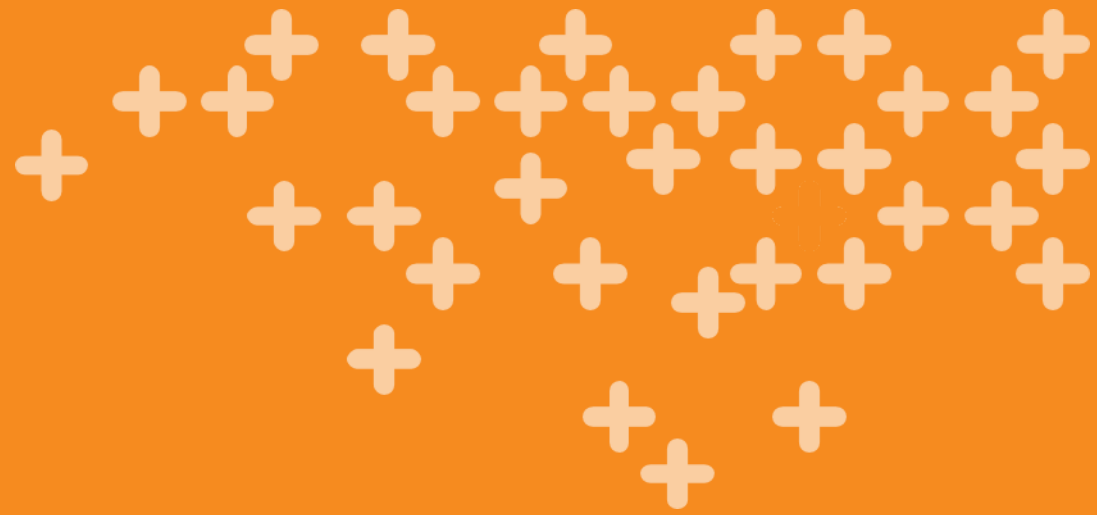
DEMO

Deploying an Azure Container App

Azure Container Apps - Concepts

Environment

Container App 1

Revision 1

Pod

| Container A |
| Container B |

Revision 2

Pod

| Container A | Container C |
| Container B | Container D |

Container App 2

Revision 1

Pod

| Container A |

Container App 3

Revision 1

Pod

| Container A |

Container App 4

Revision 1

Pod

| Container A |

Log Analytics Workspace

Virtual Network

Revisions – Traffic splitting

DEMO

Azure Container App Revisions

Scaling Azure Container Apps

- Automatic horizontal scaling through scaling rules
- Large number of scale triggers
  - HTTP
  - 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	1	25
`maxReplicas`	Maximum number of replicas running for your container app.	n/a	1	25

KEDA

- Open source component for event-driven scaling in Kubernetes

- Provides 30+ built-in scalers

- Scale to zero or to thousands

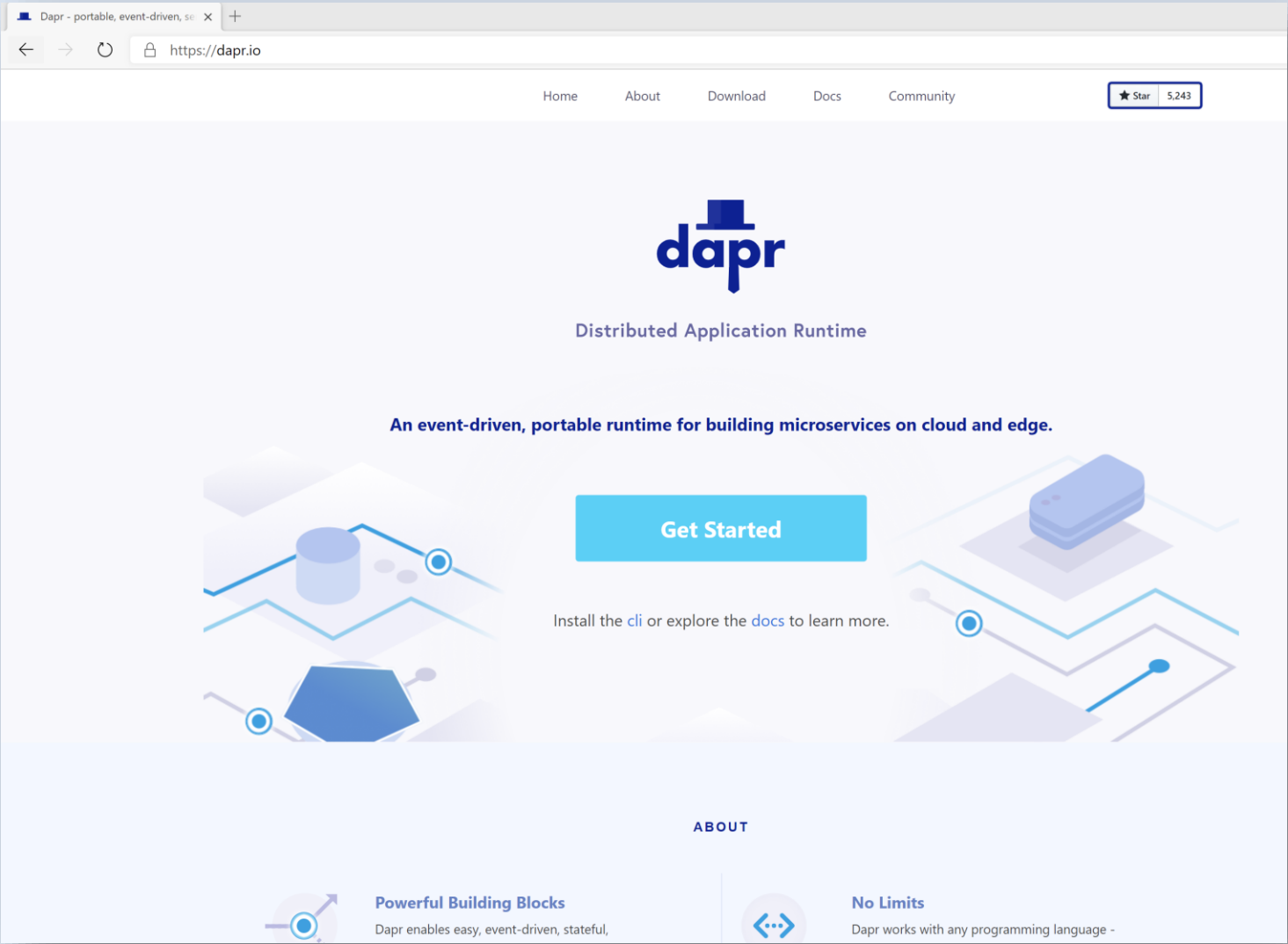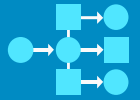- Run and scale Azure Functions in Kubernetes

https://keda.sh/

dapr

Distributed Application Runtime

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

https://dapr.io

Dapr - Microservice building blocks

| Service-to-service invocation	State management	Publish and subscribe	Resource bindings and triggers	Secrets	Distributed tracing	Actors
Perform direct, secure, service-to-service method calls	Create long running, stateless and stateful services	Secure, scalable messaging between services	Trigger code through events from a large array of inputs Output bindings to external resources including databases and queues	Securely access secrets from your application	See and measure the message calls across components and networked services	Encapsulate code and data in reusable actor objects as a common microservices design pattern

Sidecar and component architecture
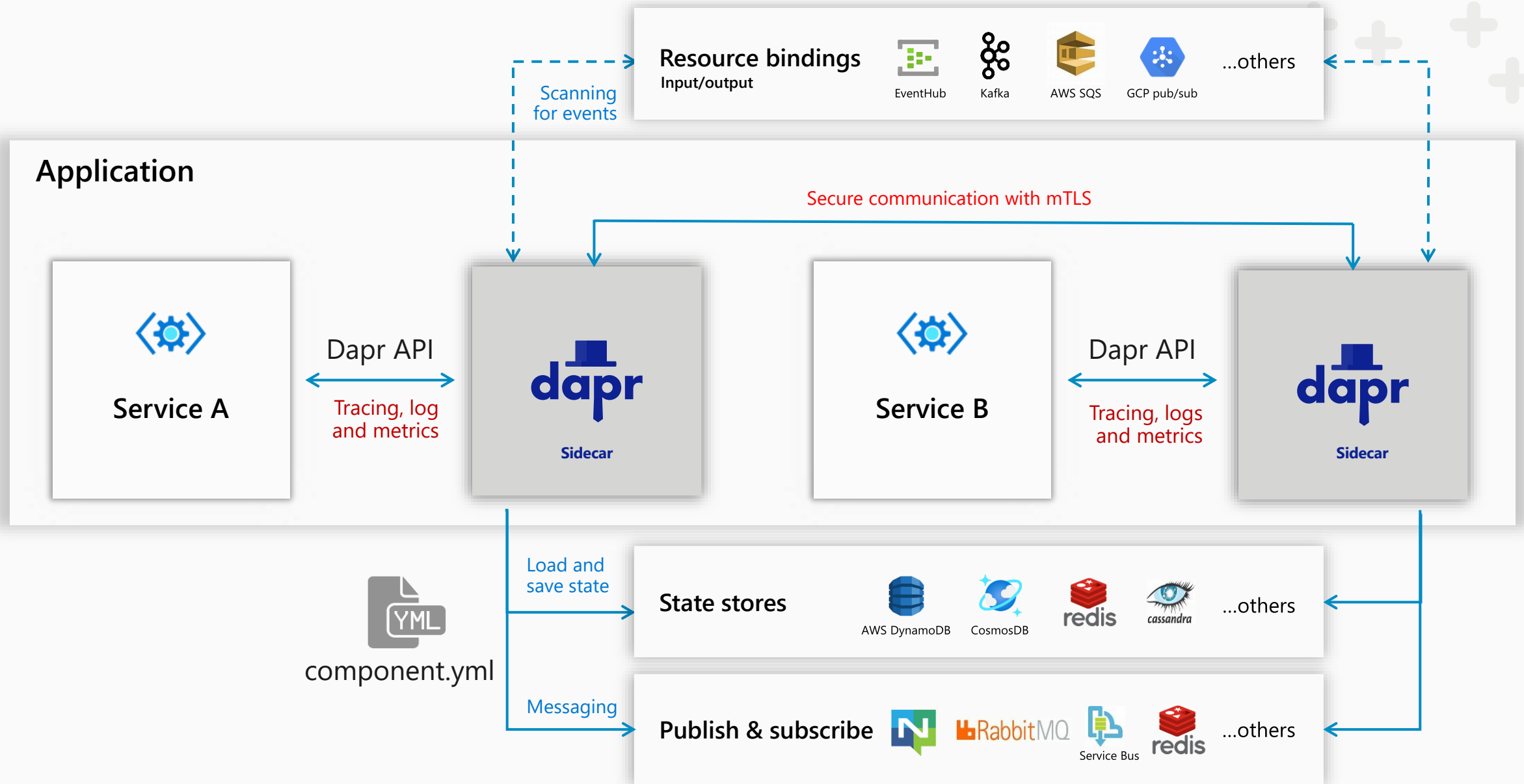
Azure Container Apps – With Dapr
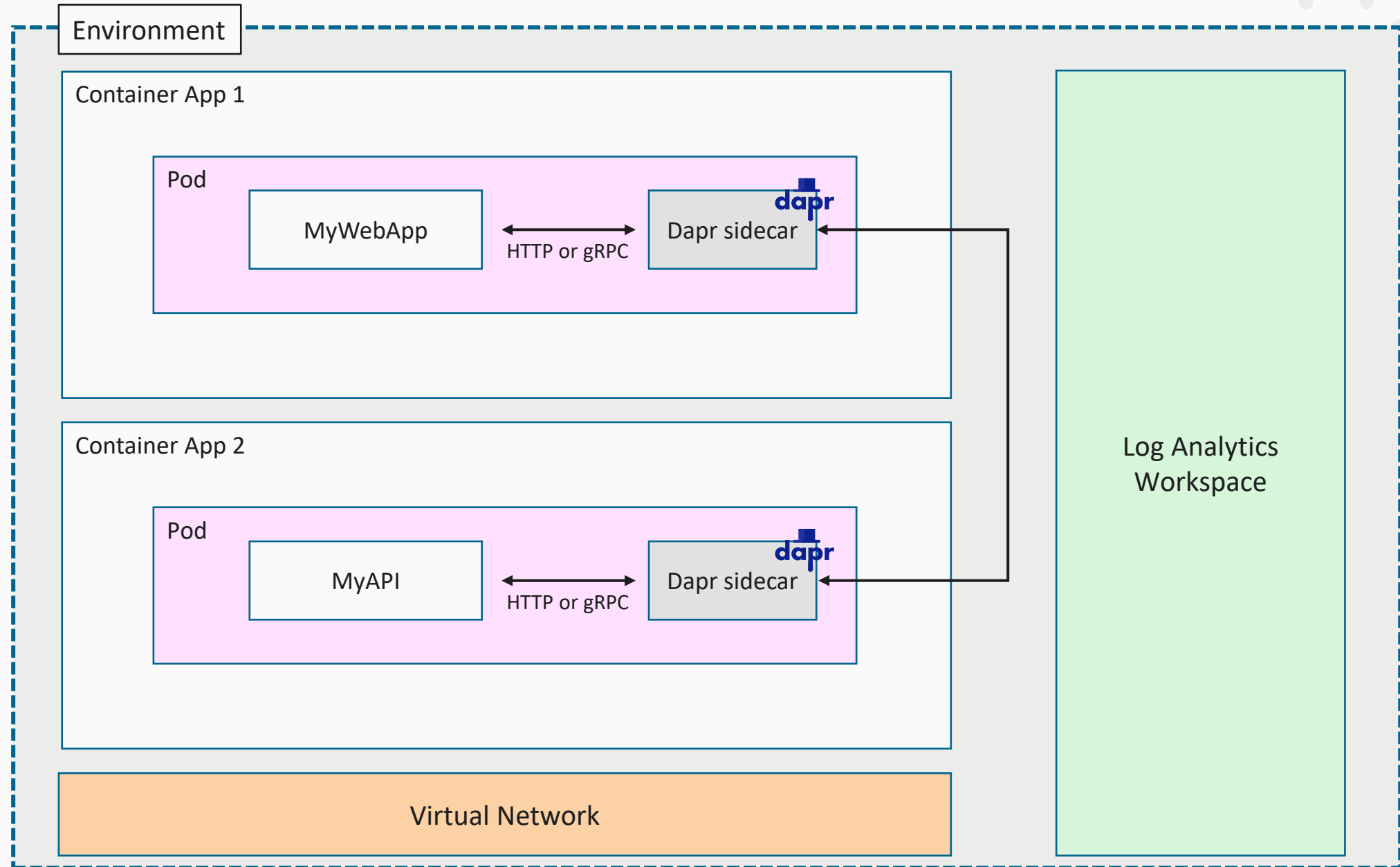
DEMO
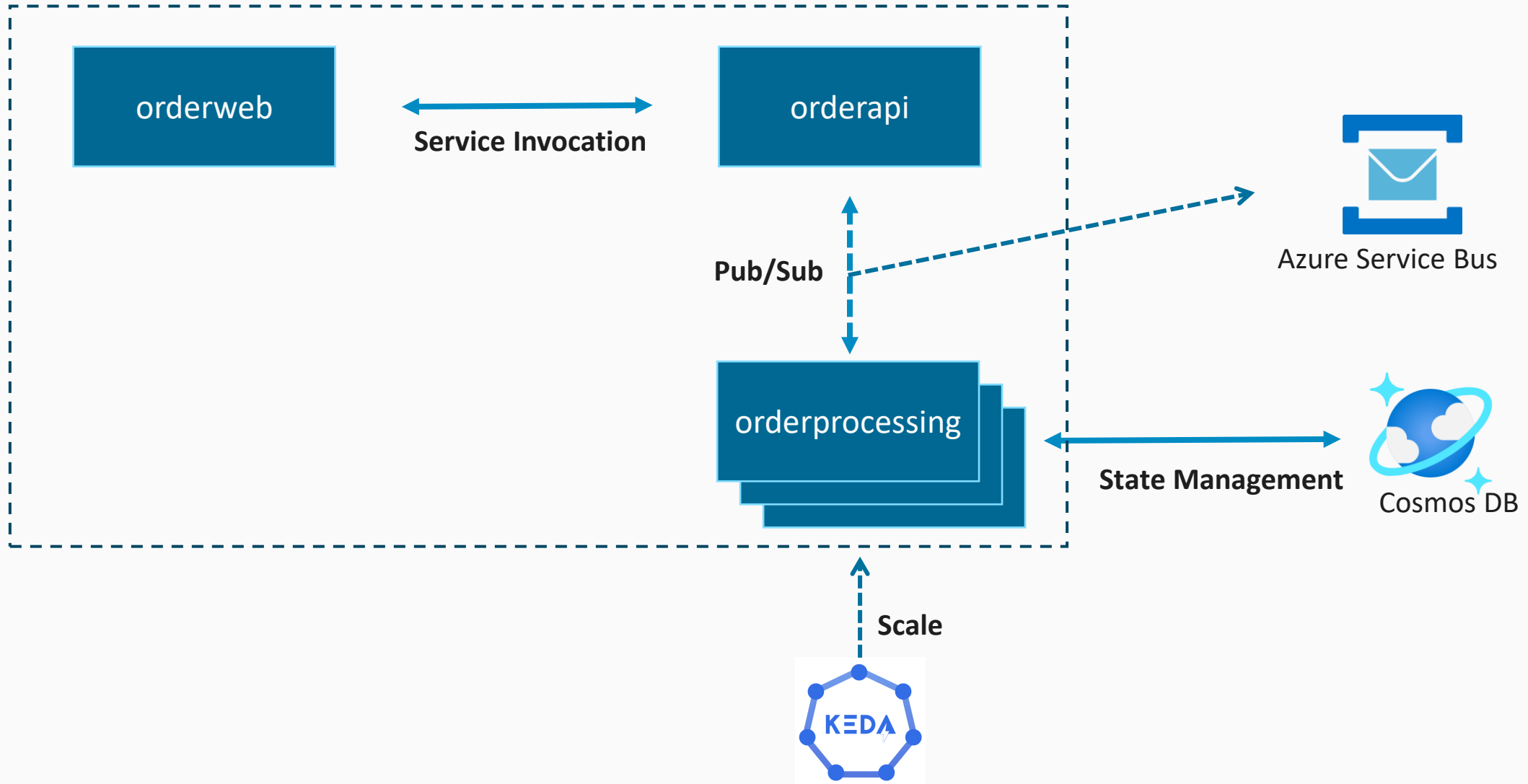
Azure Container Apps

Enabling Dapr

Scaling with KEDA
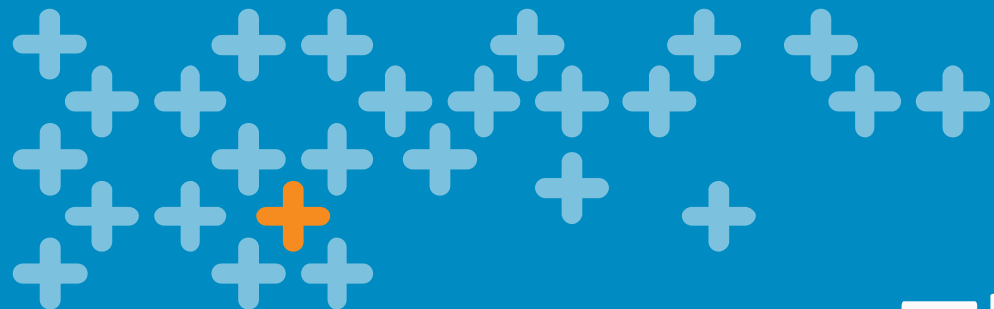
Azure Container Apps - Pricing

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

Living in Sweden?
Sign up for some free stickers!

https://activesolution.se/stickers

Championing Azure: AI & Machine Learning

Jan 18th — AI & ML Intro – Alan Smith & Peter Örneholm

Jan 25th — Democratizing AI – Peter Örneholm

Feb 1st — Machine Learning Theroy – Alan Smith

Feb 8th — Reinforcement Learning in Gaming – Alan Smith & Eve Pardi

Feb 15th — Azure Custom Vision – Alan Smith

Feb 22nd — Azure Machine Learning – Robert Folkesson

Championing Cloud Native

Jan 19th — Introduction to Cloud Native Options - Chris Klug

Jan 26th — Event Driven Architechture - Alan Smith

Feb 2nd — Serverless Architechtures in Microsoft Azure - Alan Smith

Feb 9th — Containers and Dockers - Chris Klug

Feb 16th — Azure Container Apps - Jakob Ehn

Feb 23rd — Azure Kubernetes Service - Jakob Ehn

active
SOLUTION