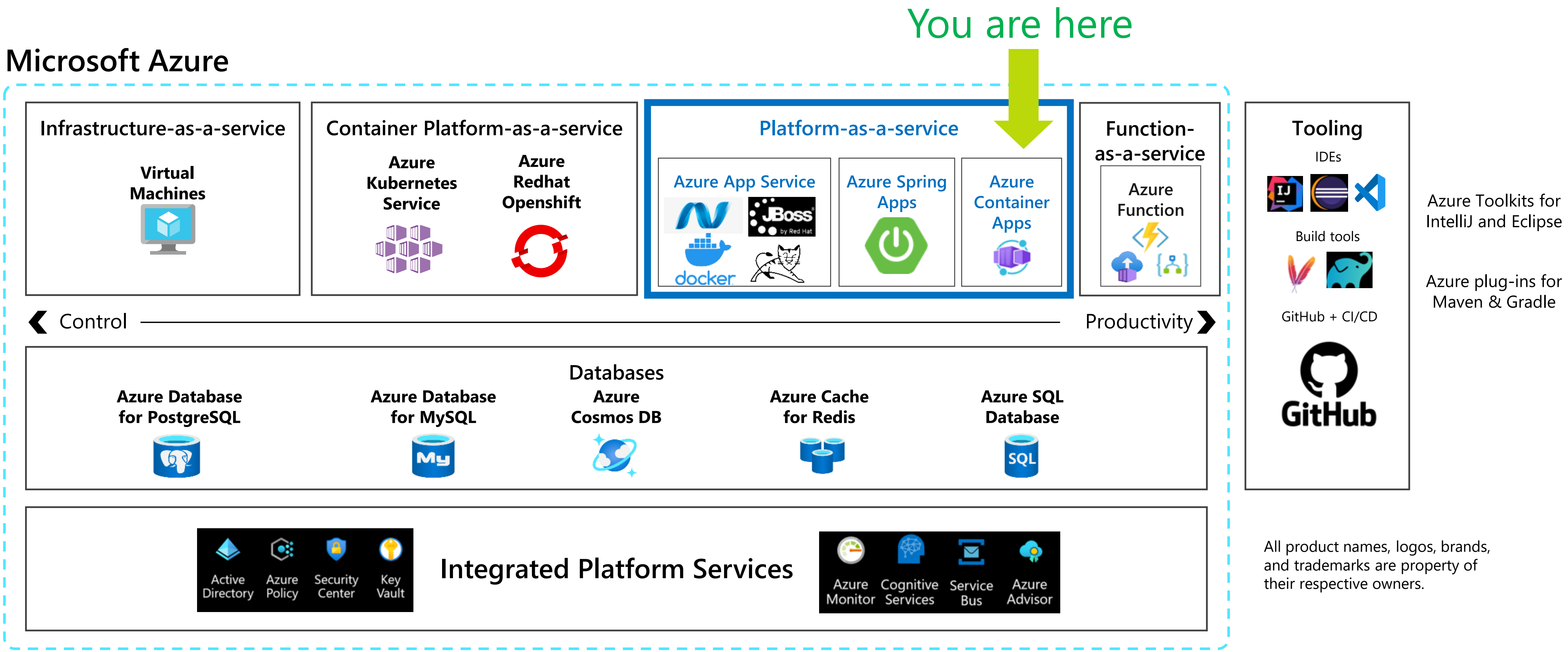
Azure Container Apps

Overview

maheshk@microsoft.com

Azure Application Platform



You are here

Microsoft Azure

Infrastructure-as-a-service

Virtual Machines

Container Platform-as-a-service

Azure Kubernetes Service

Azure Redhat Openshift

Platform-as-a-service

Azure App Service

Azure Spring Apps

Azure Container Apps

Function-as-a-service

Azure Function

Tooling

IDEs

Build tools

GitHub + CI/CD

Azure Toolkits for IntelliJ and Eclipse

Azure plug-ins for Maven & Gradle

Control

Productivity

Databases

Azure Database for PostgreSQL

Azure Database for MySQL

Azure Cosmos DB

Azure Cache for Redis

Azure SQL Database

Active Directory

Azure Policy

Security Center

Key Vault

Integrated Platform Services

Azure Monitor

Cognitive Services

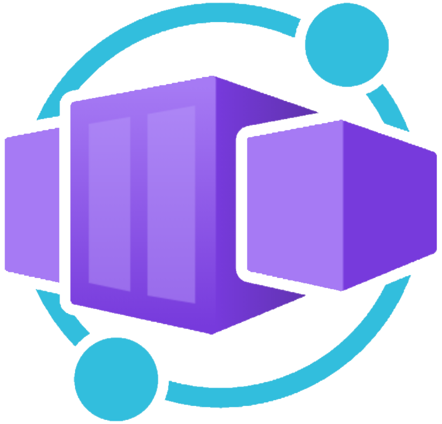
Service Bus

Azure Advisor

All product names, logos, brands, and trademarks are property of their respective owners.

Azure Container Apps

A new serverless container platform for building modern apps and microservices



Built on a foundation of AKS, KEDA, Dapr, and Envoy

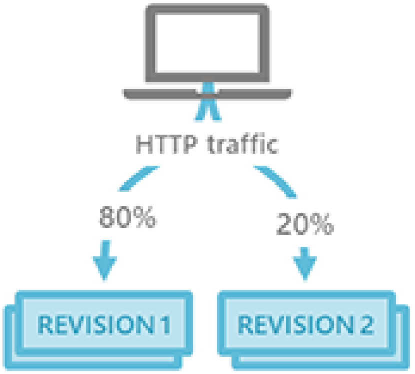
Azure Container Apps (public preview)

"Azure Container Apps enables executing application code packaged in any container and is unopinionated about runtime or programming model."

- Enjoy the **benefits of running containers** while leaving behind the concerns of **managing cloud infrastructure** and **complex container orchestrators**.
- **Serverless** (scale to zero support)
- **Scale** on HTTP requests, events, or run always-on background jobs
- **Automatic encryption** for ingress and service-to-service communications
- Built on a foundation of AKS, KEDA, Dapr, and 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.

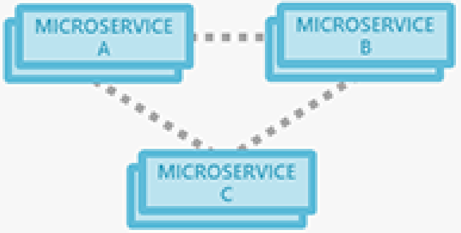
EVENT-DRIVEN PROCESSING

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

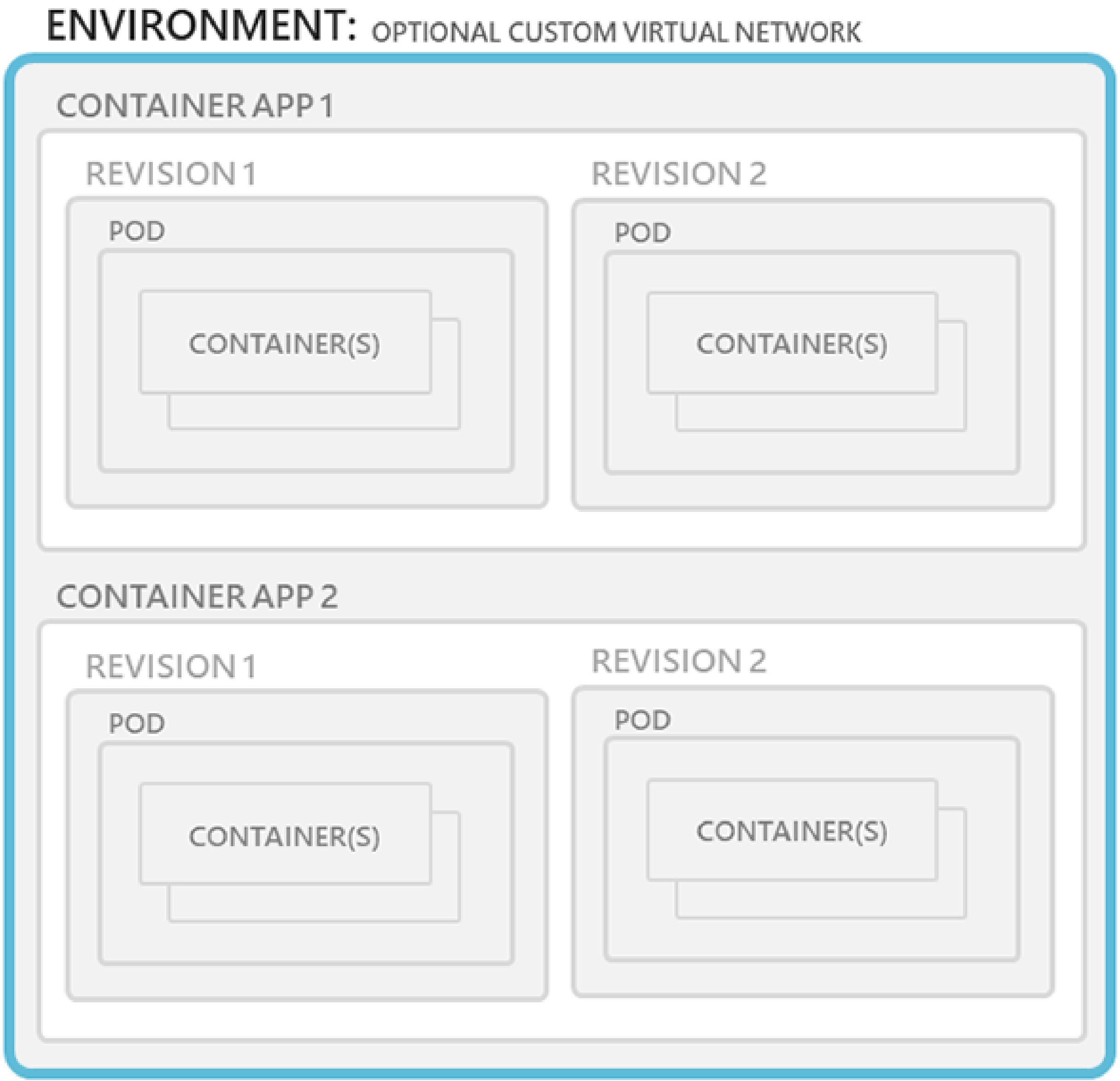


Deploy and manage a microservices architecture with the option to integrate with Dapr.

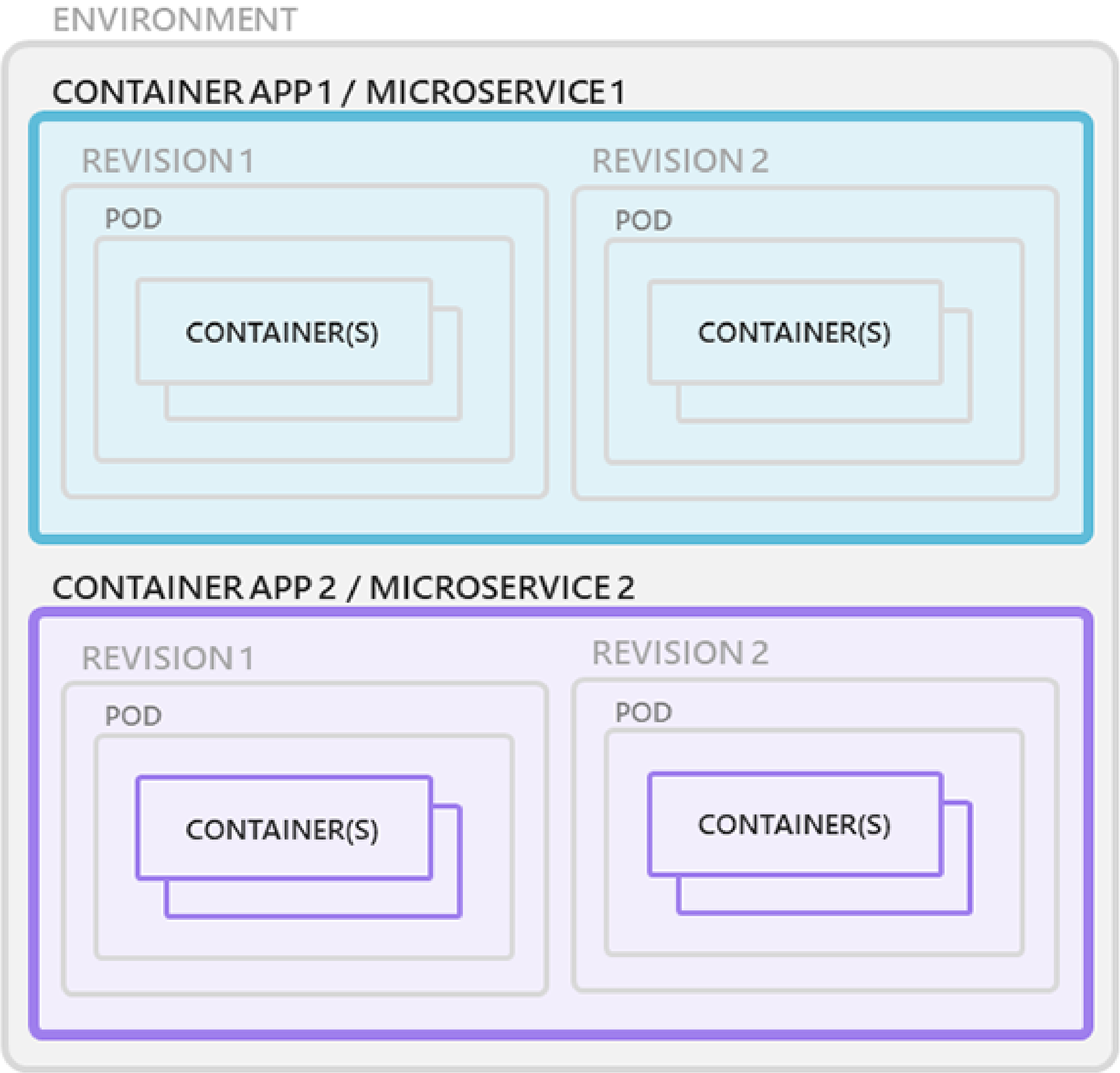
AUTO-SCALE CRITERIA

Individual microservices can scale according to any KEDA scale triggers.

Environments are an isolation boundary around a collection of container apps.



Container apps are deployed as **microservices**.



ENVIRONMENT

CONTAINER APP 1 / MICROSERVICE 1

REVISION 1

POD

CONTAINER(S)

REVISION 2

POD

CONTAINER(S)

CONTAINER APP 2 / MICROSERVICE 2

REVISION 1

POD

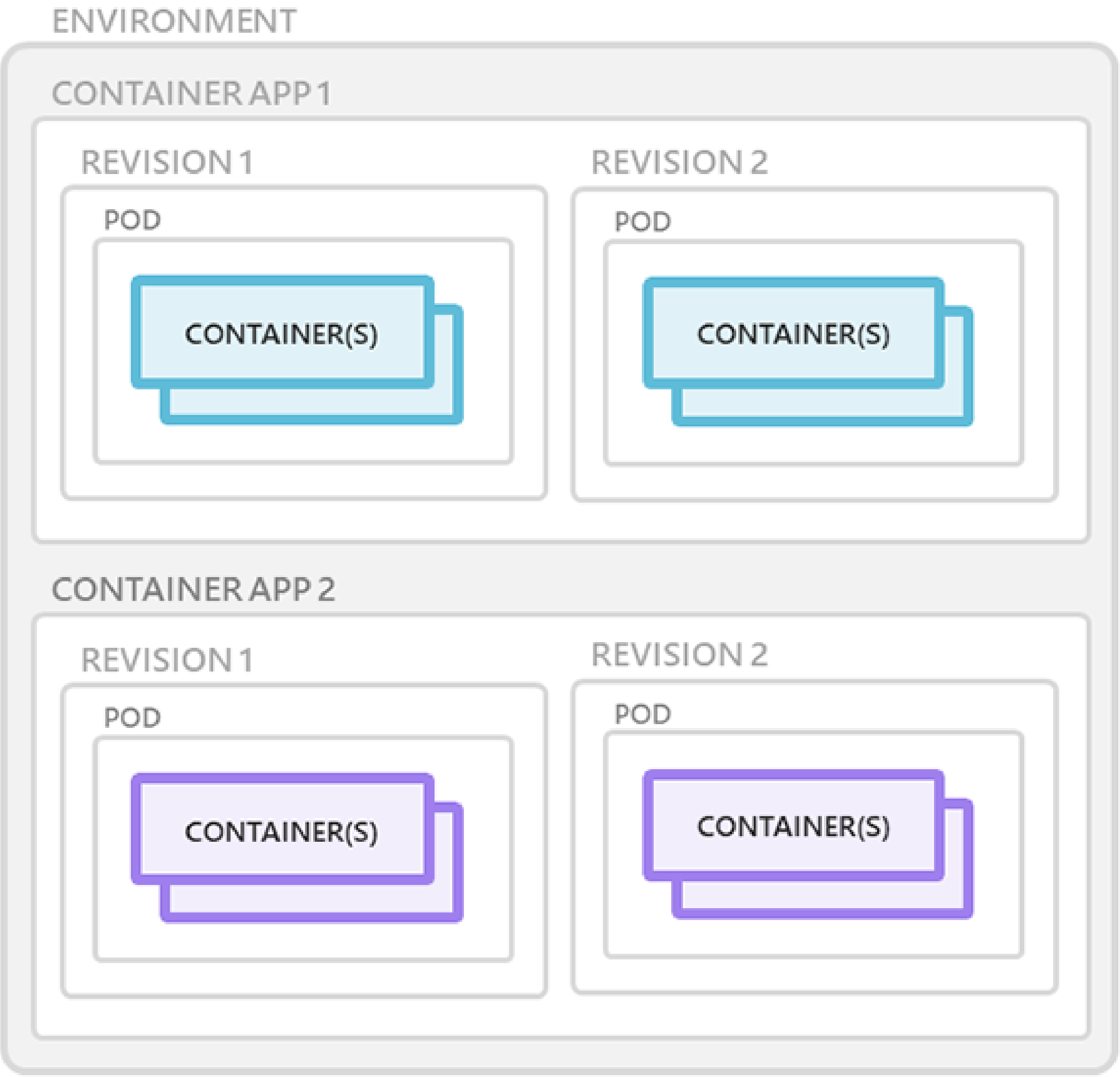
CONTAINER(S)

REVISION 2

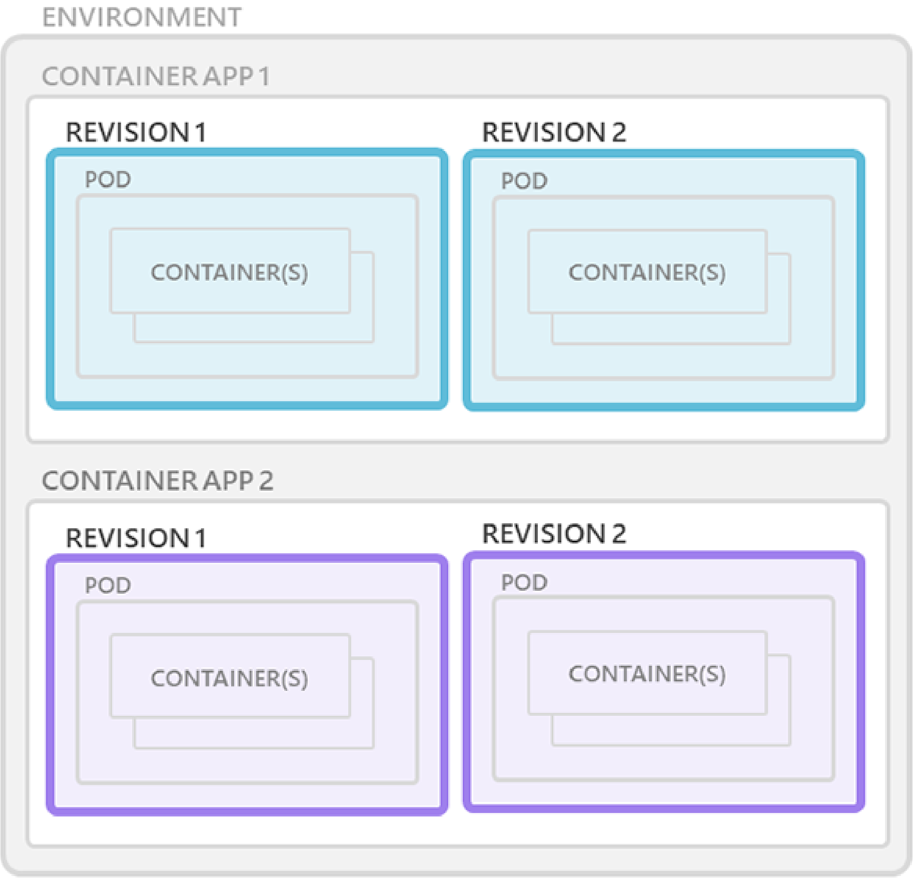
POD

CONTAINER(S)

Containers for an Azure Container App are grouped together in pods inside revision snapshots.



Revisions are immutable snapshots of a container app.



ENVIRONMENT

CONTAINER APP 1

REVISION 1

POD

CONTAINER(S)

REVISION 2

POD

CONTAINER(S)

CONTAINER APP 2

REVISION 1

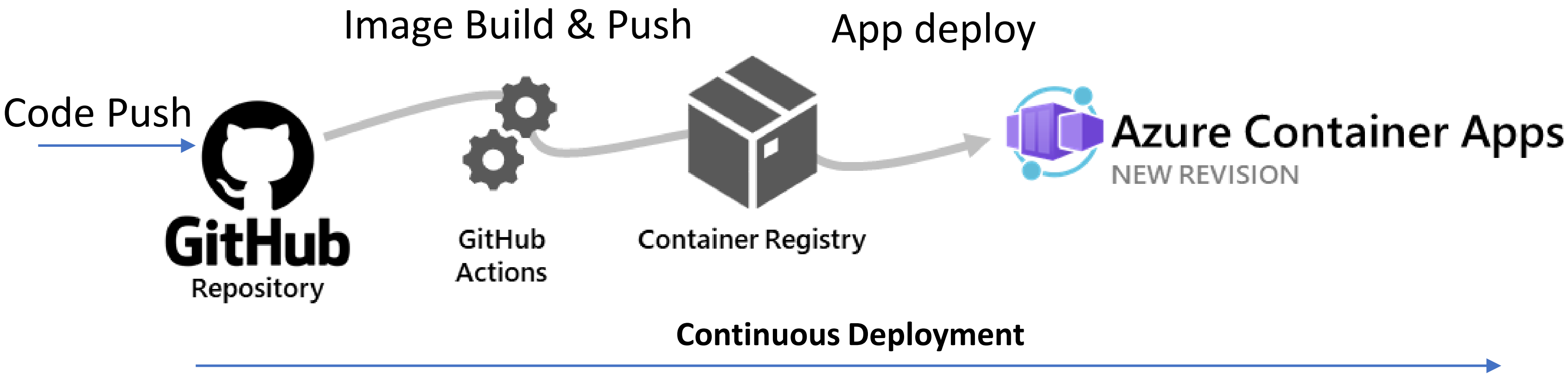
POD

CONTAINER(S)

REVISION 2

POD

CONTAINER(S)



Code Push

GitHub Repository

Image Build & Push

GitHub Actions

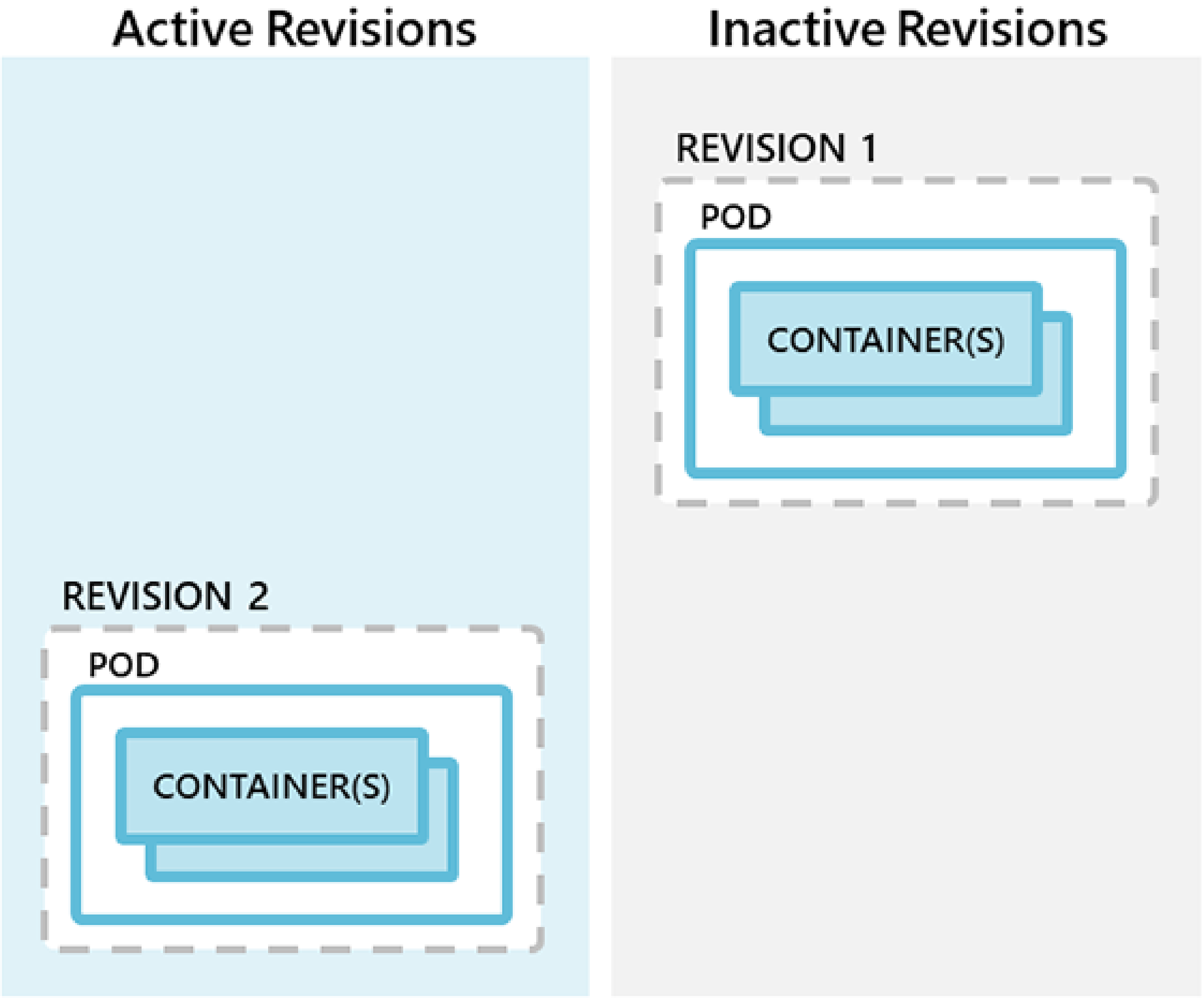
Container Registry

App deploy

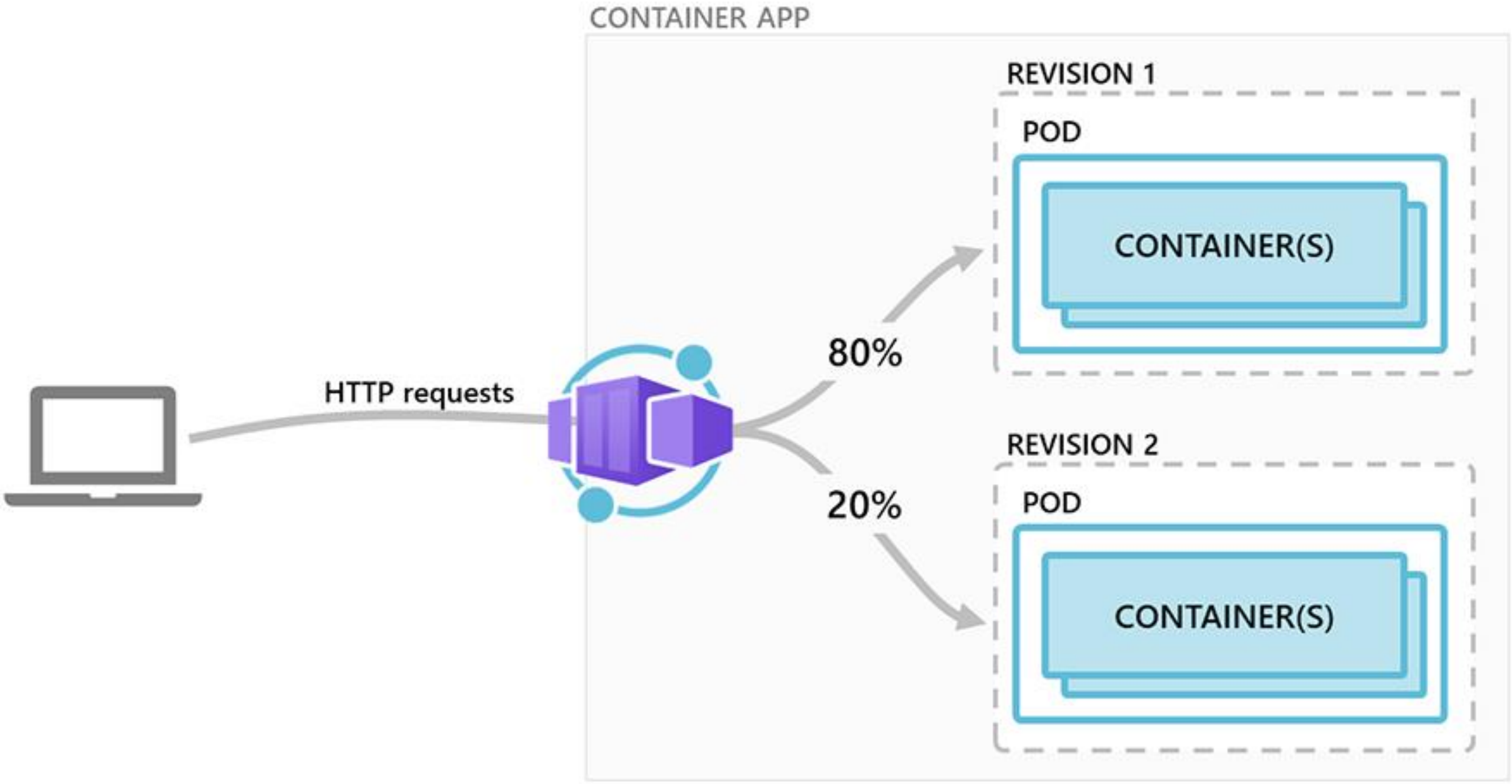
Azure Container Apps NEW REVISION

Continuous Deployment

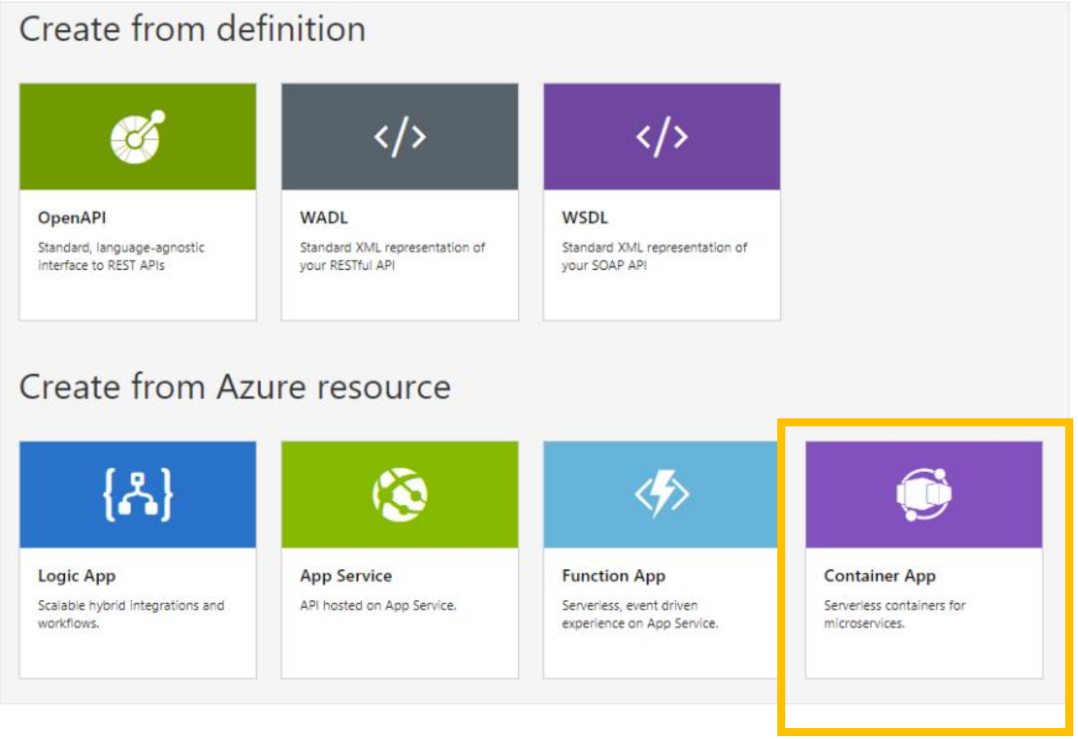
Once a revision is no longer needed, you can **deactivate** individual revisions, or choose to automatically deactivate old revisions.



Ingress traffic splitting



API Management Import



Create from definition

OpenAPI
Standard, language-agnostic interface to REST APIs
WADL
Standard XML representation of your RESTful API
WSDL
Standard XML representation of your SOAP API

Create from Azure resource

Logic App
Scalable hybrid integrations and workflows.
App Service
API hosted on App Service.
Function App
Serverless, event driven experience on App Service.
Container App
Serverless containers for microservices.

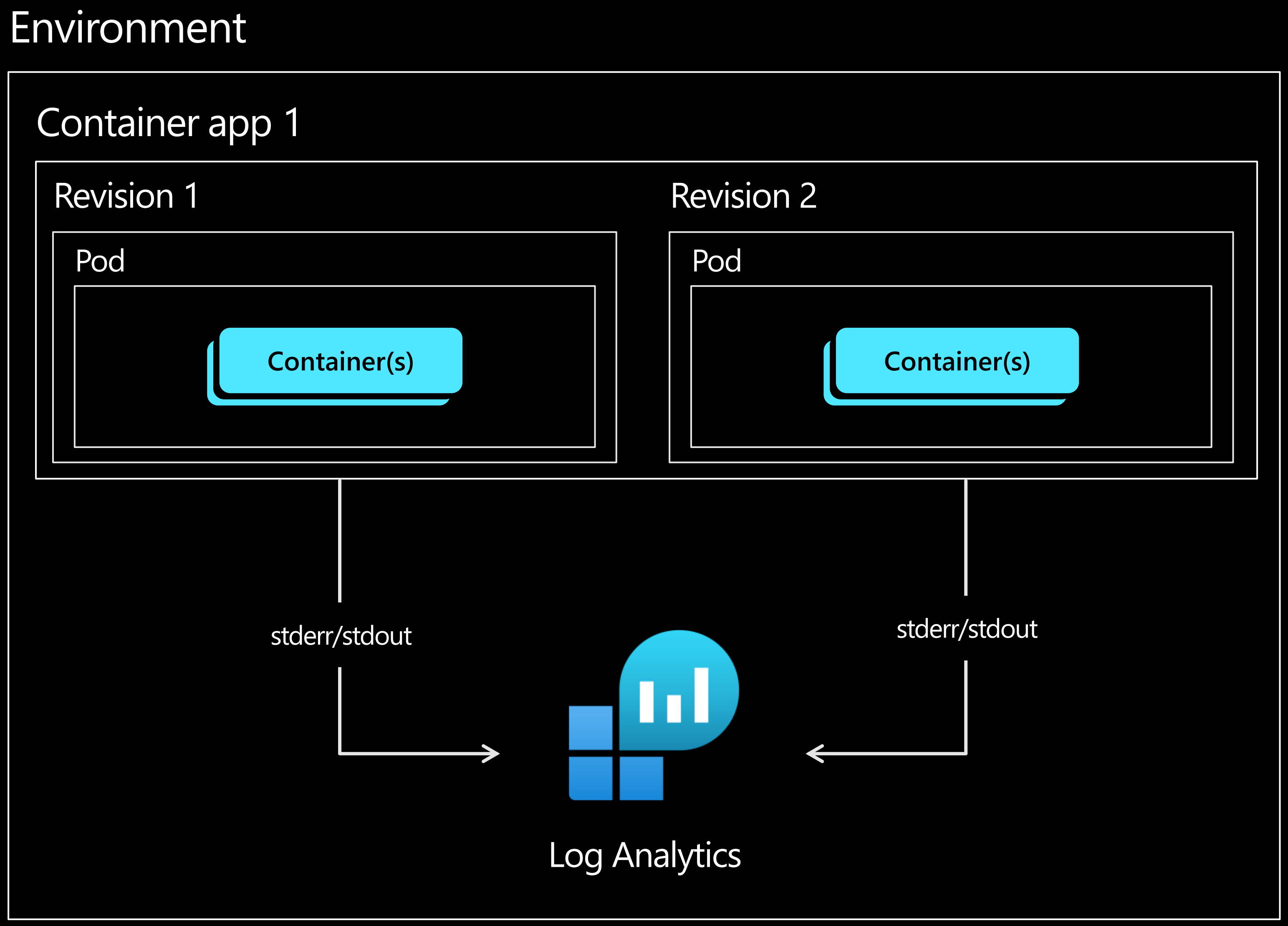
API Management will look in several locations for an OpenAPI Specification:

- The Container App configuration
- /openapi.json
- /openapi.yml
- /swagger/v1/swagger.json

https://docs.microsoft.com/en-us/azure/api-management/import-container-app-with-oas

Logging

Containers write logs to standard output or standard error streams surfaced via Log Analytics



Observability

- **Log Analytics** – stderr/stdout, small ingestion delay
- **Metrics** – CPU, Memory, Bytes in/out, Requests
- **Alerts** – based on metrics, log search, admin signals (e.g., create, update, delete container app)
- **Streaming Logs** – stderr/stdout, real-time
- **Connect to Console** – connect to run shell commands
- **Events** – emitted from underlying orchestrator (e.g., container start failure, scale up/down)

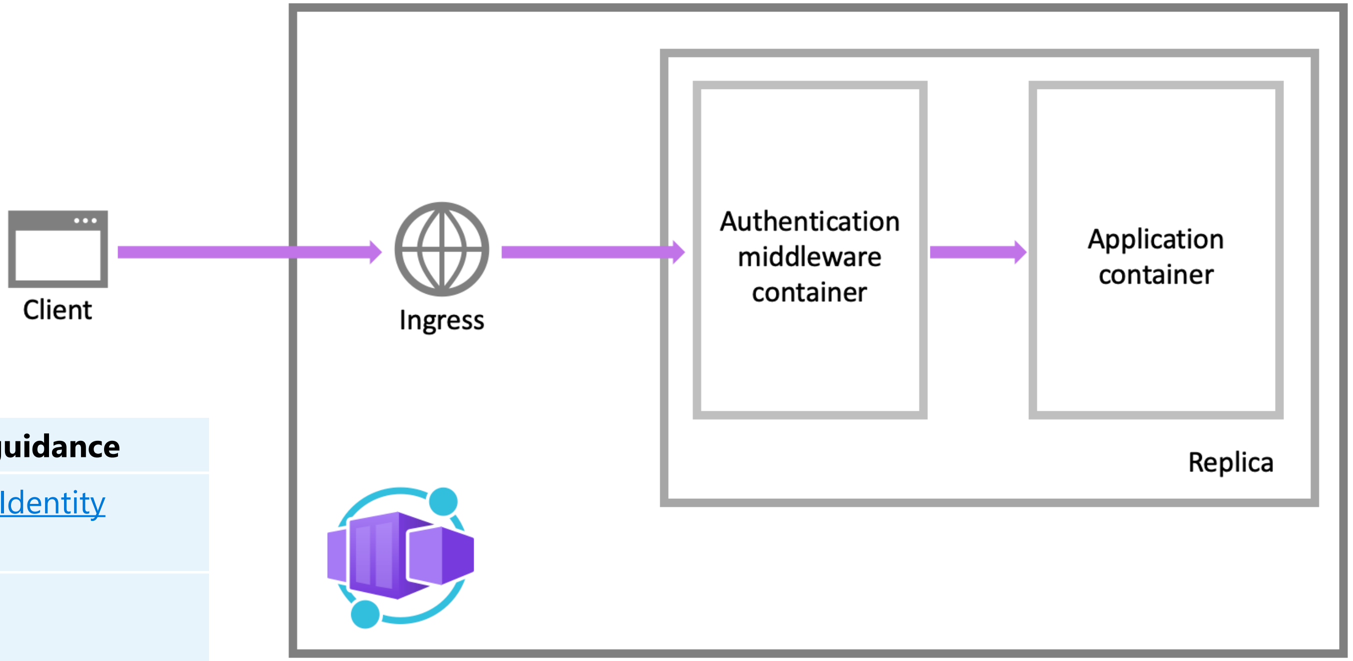
Secrets management

Securely store sensitive configuration elements that are then available to containers through environment variables, scale rules, and Dapr

```
"template": {
    "containers": [
        {
            "image": "myregistry/myQueueApp:v1",
            "name": "myQueueApp",
            "env": [
                {
                    "name": "QueueName",
                    "value": "myqueue"
                },
                {
                    "name": "ConnectionString",
                    "secretref": "queue-connection-string"
                }
            ]
        }
    ],
}
```

Authentication and Authorization with Federated Identity

Built-in authentication and authorization features (sometimes referred to as "Easy Auth"), to secure your external ingress-enabled container app with minimal or no code.



Provider	Sign-in endpoint	How-To guidance
Microsoft Identity Platform	/.auth/login/aad	Microsoft Identity Platform
Facebook	/.auth/login/facebook	Facebook
GitHub	/.auth/login/github	GitHub
Google	/.auth/login/google	Google
Twitter	/.auth/login/twitter	Twitter
Any OpenID Connect provider	/.auth/login/<providerName>	OpenID Connect

Application autoscaling made simple

Open-source, extensible, and vendor agnostic

KEDA

Kubernetes-based Event Driven Autoscaler

Drive the scaling of any container based on a growing list of 35+ event sources, known as: scalers

Metrics Adapter | Controller | Scaler

Event-driven

Intelligently scale your event-driven applications

Built-in scalers

Out-of-the-box scalers for various vendors, databases, messaging systems, telemetry systems, CI/CD, and more

Vendor-agnostic

Support for triggers across variety of cloud providers & products

Rich capabilities

Bring rich scaling to every workload

keda.sh

CLOUD NATIVE
COMPUTING FOUNDATION

Scaling

KEDA

HTTP

```
{
 "name": "http-rule",
  "http": {
   "metadata": {
    "concurrentRequests": 50
   }
  }
}
```

Event-driven

artemis-queue, kafka, aws-cloudwatch, aws-kinesis-stream, aws-sqs-queue, azure-blob, azure-eventhub, azure-servicebus, azure-queue, cron, external, gcp-pubsub, huawei-cloudeye, ibmmq, influxdb, mongodb, mssql, mysql, postgresql, rabbitmq, redis, redis-streams, selenium-grid, solace-event-queue, ..

CPU

```
{
 "name": "cpu-rule",
  "custom": {
   "type": "cpu",
    "metadata": {
     "type": "Utilization",
      "value": "50"
   }
  }
}
```

Memory

```
{
 "name": "mem-rule",
  "custom": {
   "type": "memory",
    "metadata": {
     "type": "AverageValue",
      "value": "512"
   }
  }
}
```

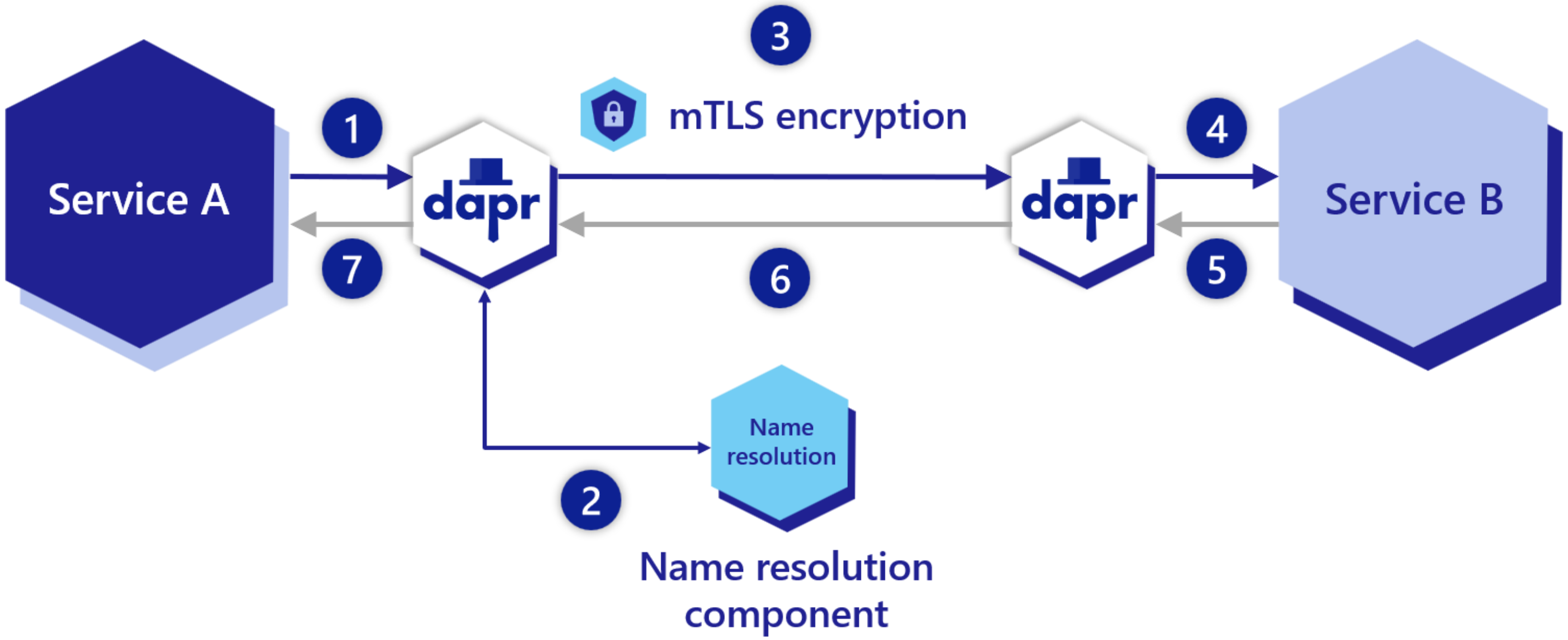
Support for scale to zero and specifying minimum/maximum replicas

Support for specifying minimum/maximum replicas

KEDA – Event Sources and Scalers

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

Dapr integration (mTLS, service discovery, tracing, etc.)



Service- to- service invocation

State management

Publish and subscribe

Resource bindings and triggers

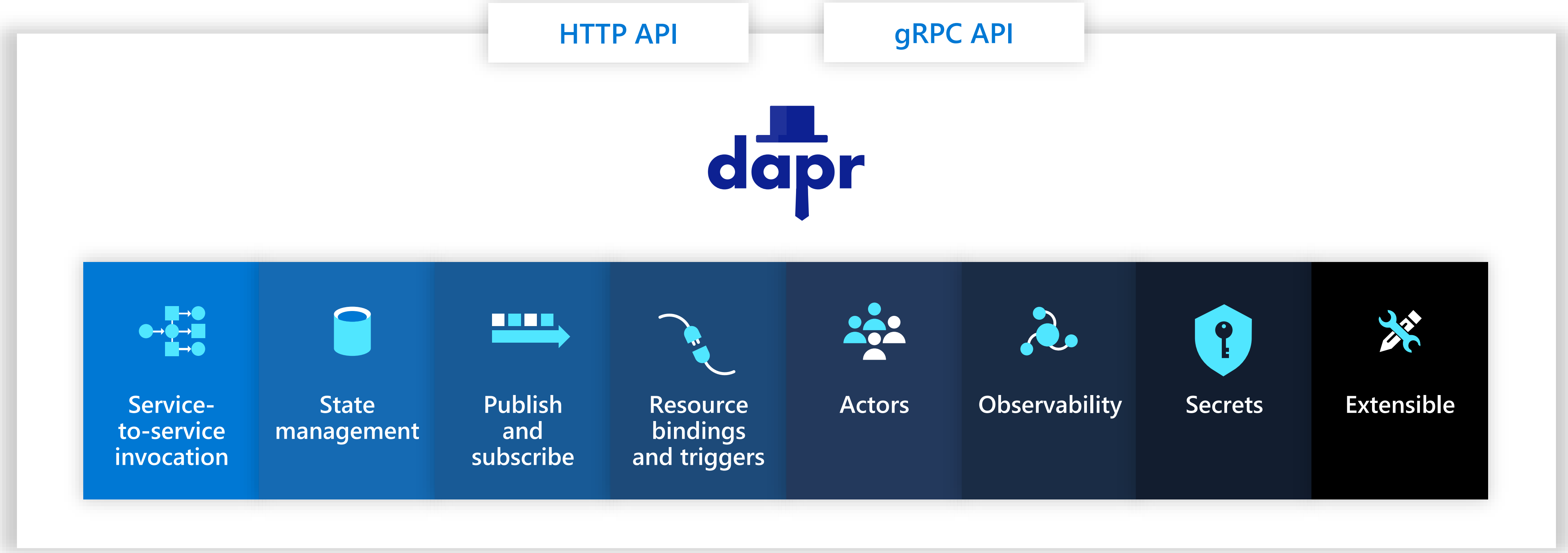
Actors

Observability

Secrets

Extensible

Microservice building blocks



HTTP API

gRPC API

dapr

Service-to-service invocation

State management

Publish and subscribe

Resource bindings and triggers

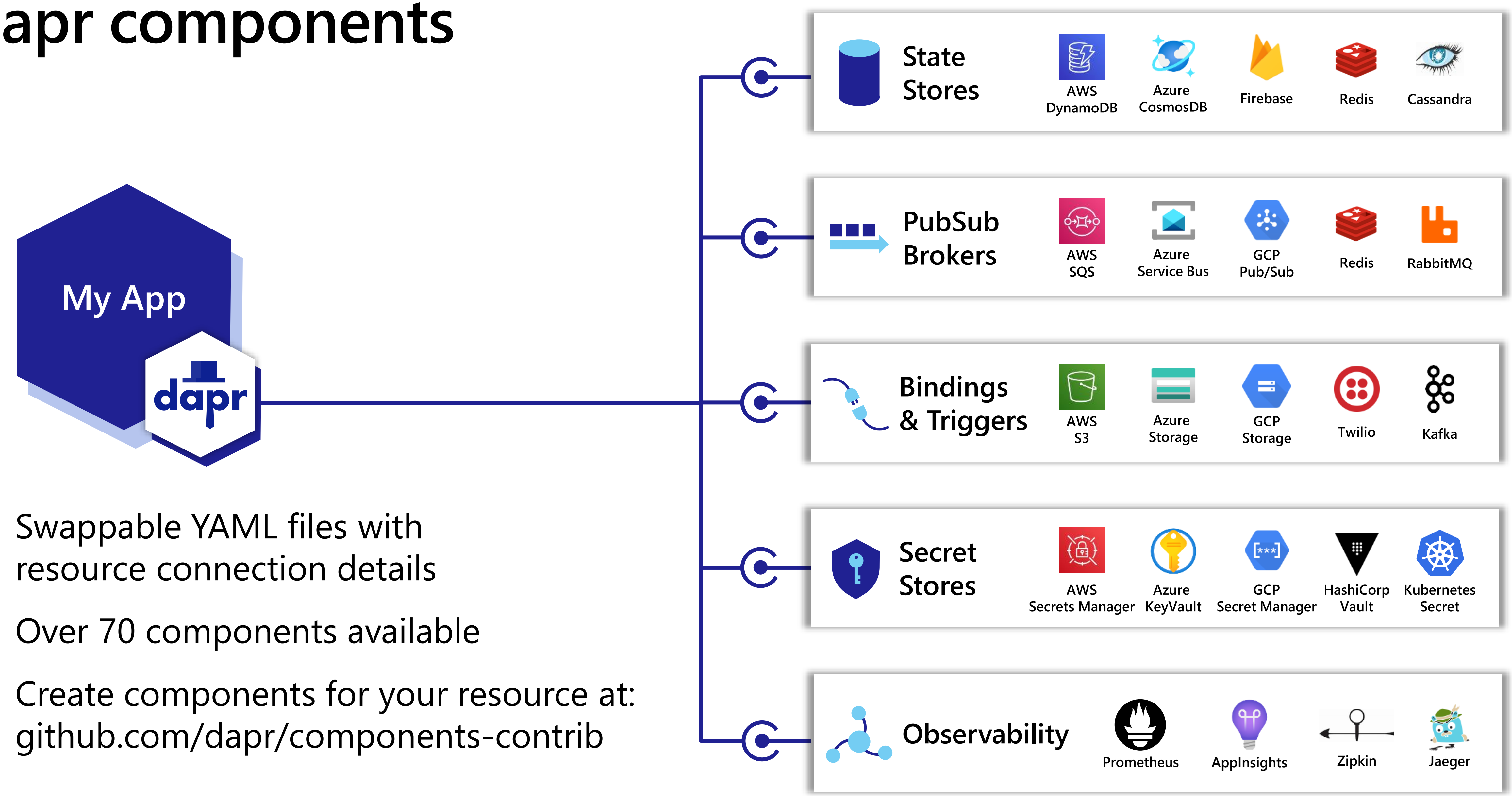
Actors

Observability

Secrets

Extensible

Dapr components



My App

dapr

State Stores

AWS DynamoDB

Azure CosmosDB

Firebase

Redis

Cassandra

PubSub Brokers

AWS SQS

Azure Service Bus

GCP Pub/Sub

Redis

RabbitMQ

Bindings & Triggers

AWS S3

Azure Storage

GCP Storage

Twilio

Kafka

Secret Stores

AWS Secrets Manager

Azure KeyVault

GCP Secret Manager

HashiCorp Vault

Kubernetes Secret

Observability

Prometheus

AppInsights

Zipkin

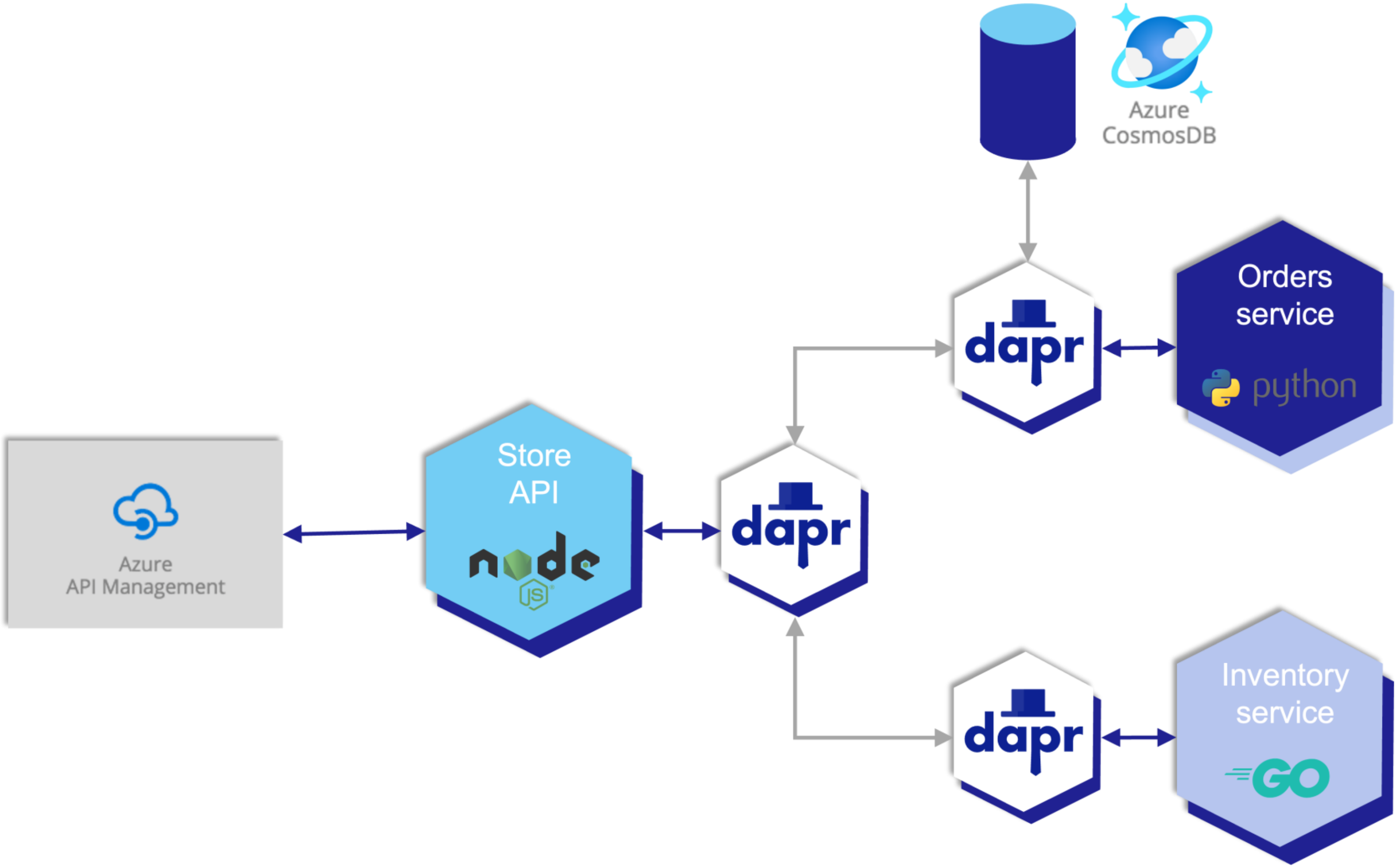
Jaeger

Swappable YAML files with resource connection details

Over 70 components available

Create components for your resource at: github.com/dapr/components-contrib

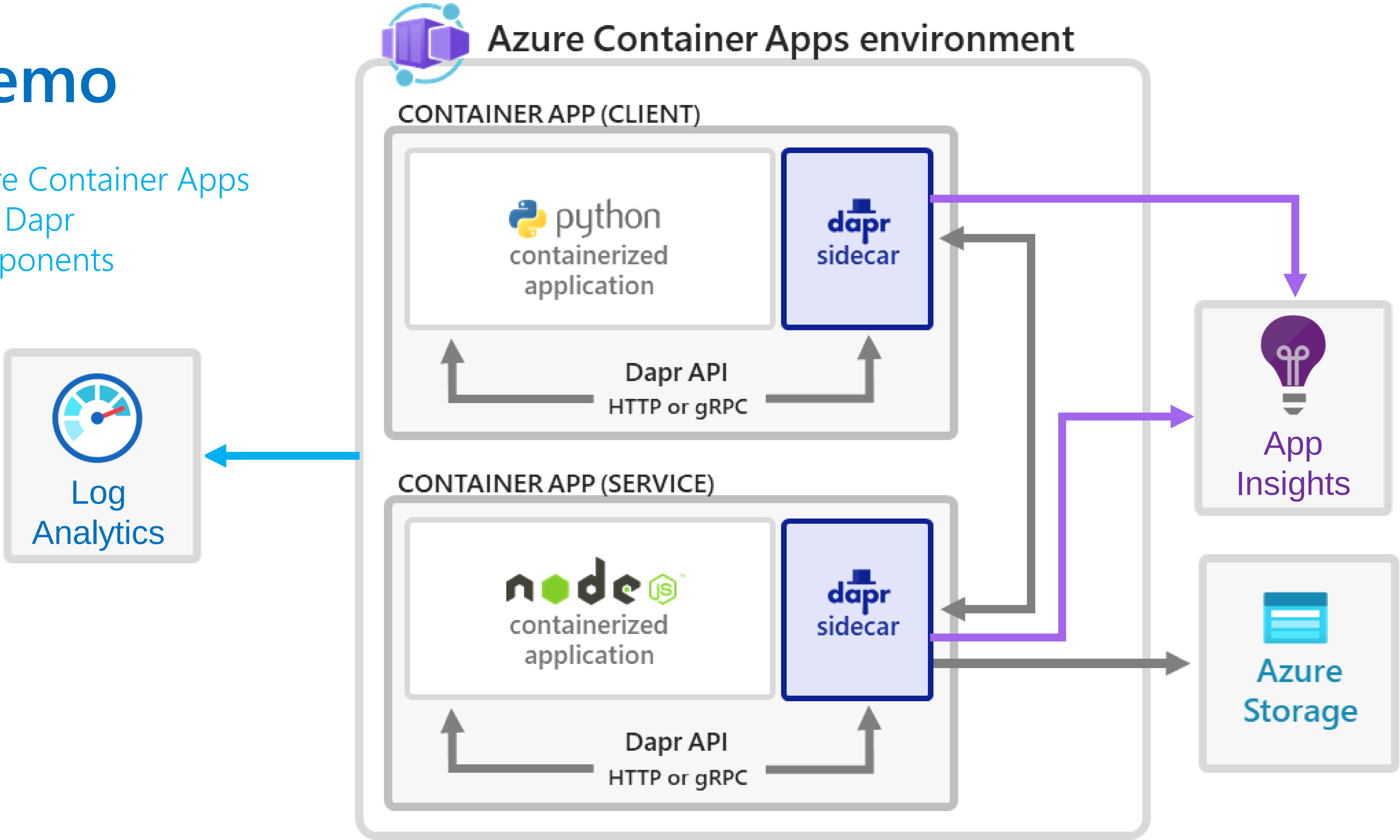
Container Apps Sample App



Container App Store Microservice Sample

Demo

Azure Container Apps with Dapr components



https://github.com/clarenceb/tutorial-dapr-cli

Learn more about Container Apps

- Introducing Azure Container Apps: a serverless container service for running modern apps at scale (Microsoft Tech Community)
- Azure Container Apps Preview documentation
- Azure Container Apps product page
- Container App Store Microservice Sample (GitHub)

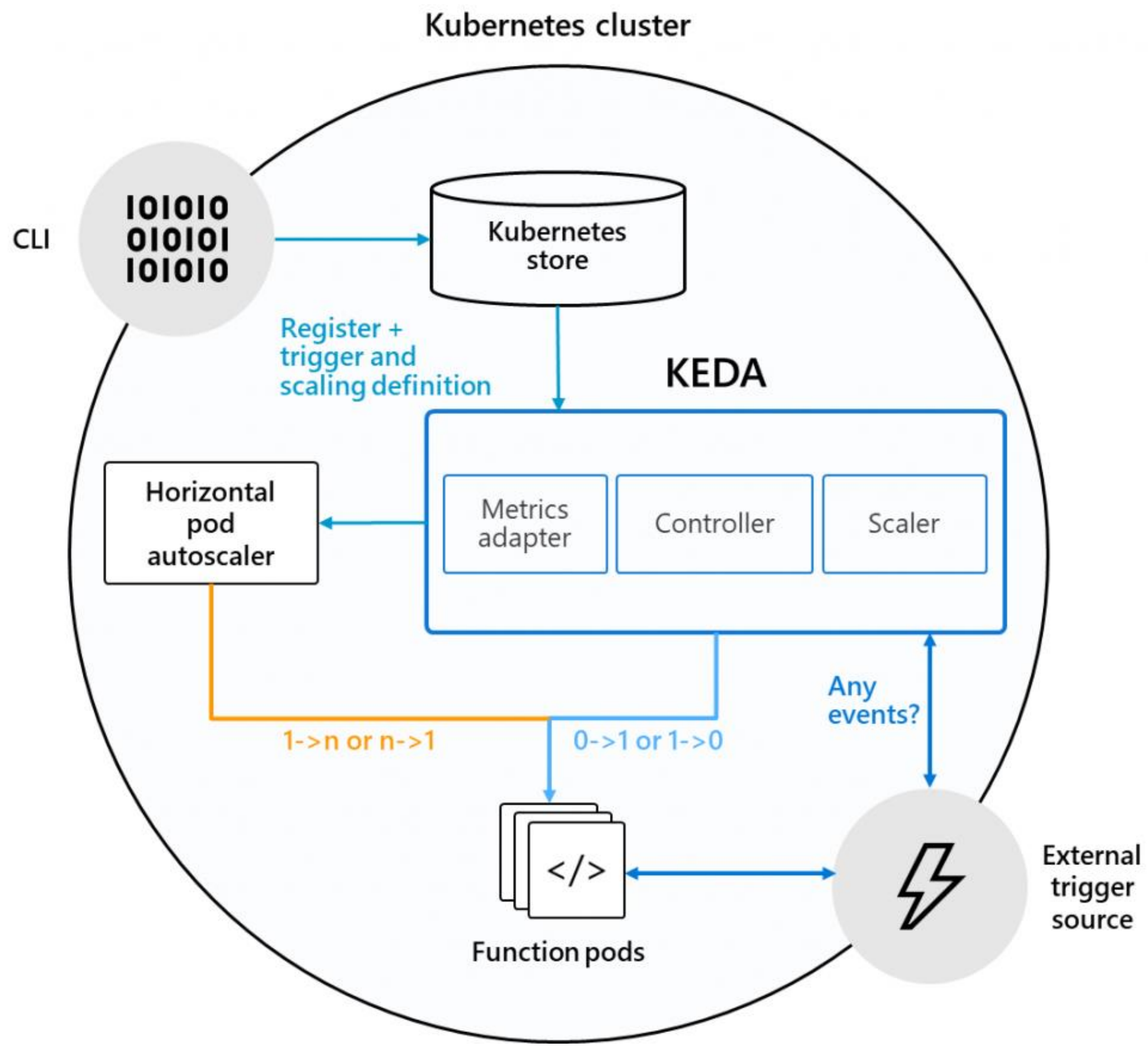
Appendix - KEDA primer

KEDA - Kubernetes Event-driven Autoscaling

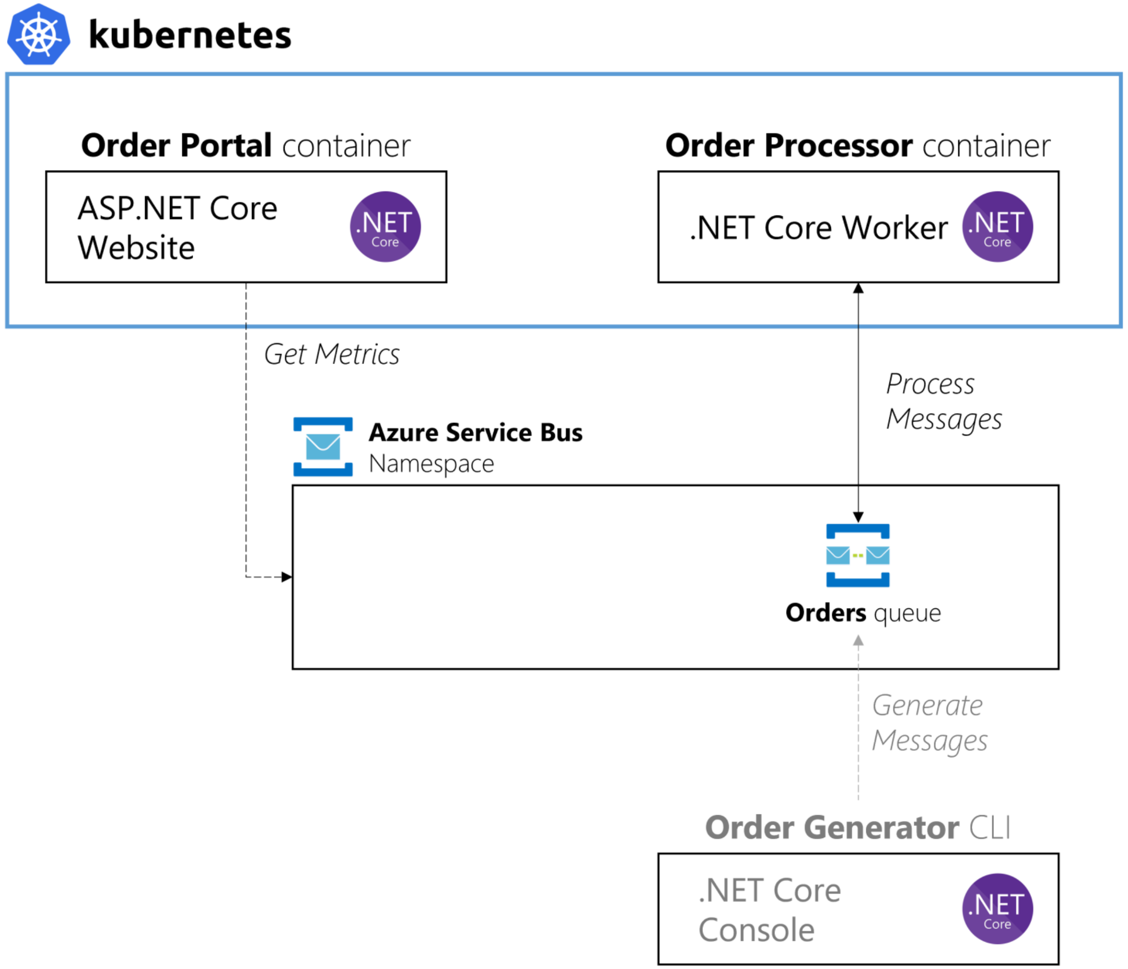
https://keda.sh/
https://www.cncf.io/projects/keda/

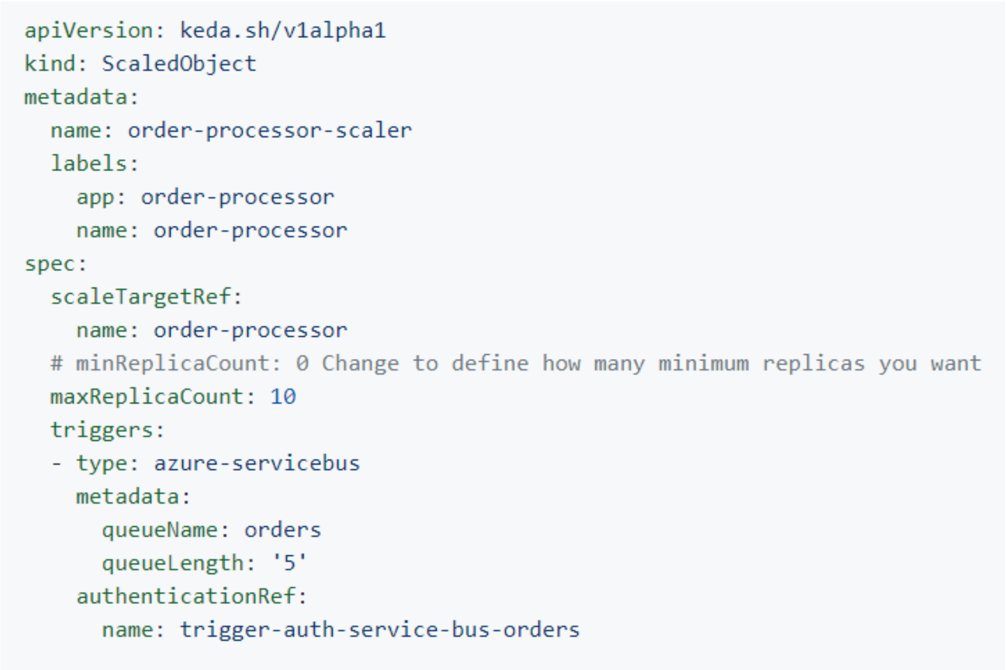
- Supports building event-driven applications in Kubernetes
- Fine grained autoscaling off of event sources for *any* container in Kubernetes
- Runs *anywhere* Kubernetes/OpenShift runs
- Native integration with Horizontal Pod Autoscaler (HPA)
- Supports scaling via Jobs (1 event -> 1 job)
- Pods get direct access to event sources
- New hosting option for Azure Functions via containers in Kubernetes
- Built in conjunction with Red Hat
- CNCF incubating project

Architecture

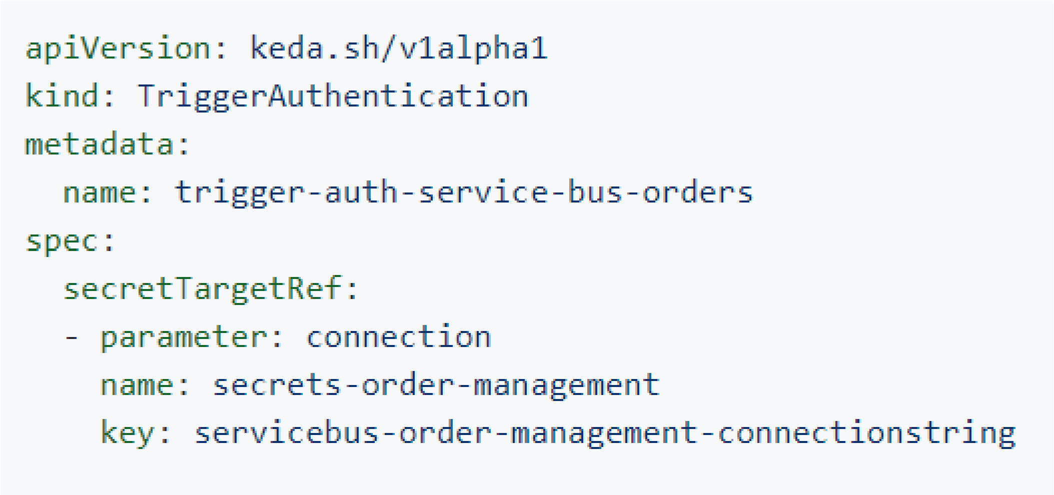


Example with queue scaler





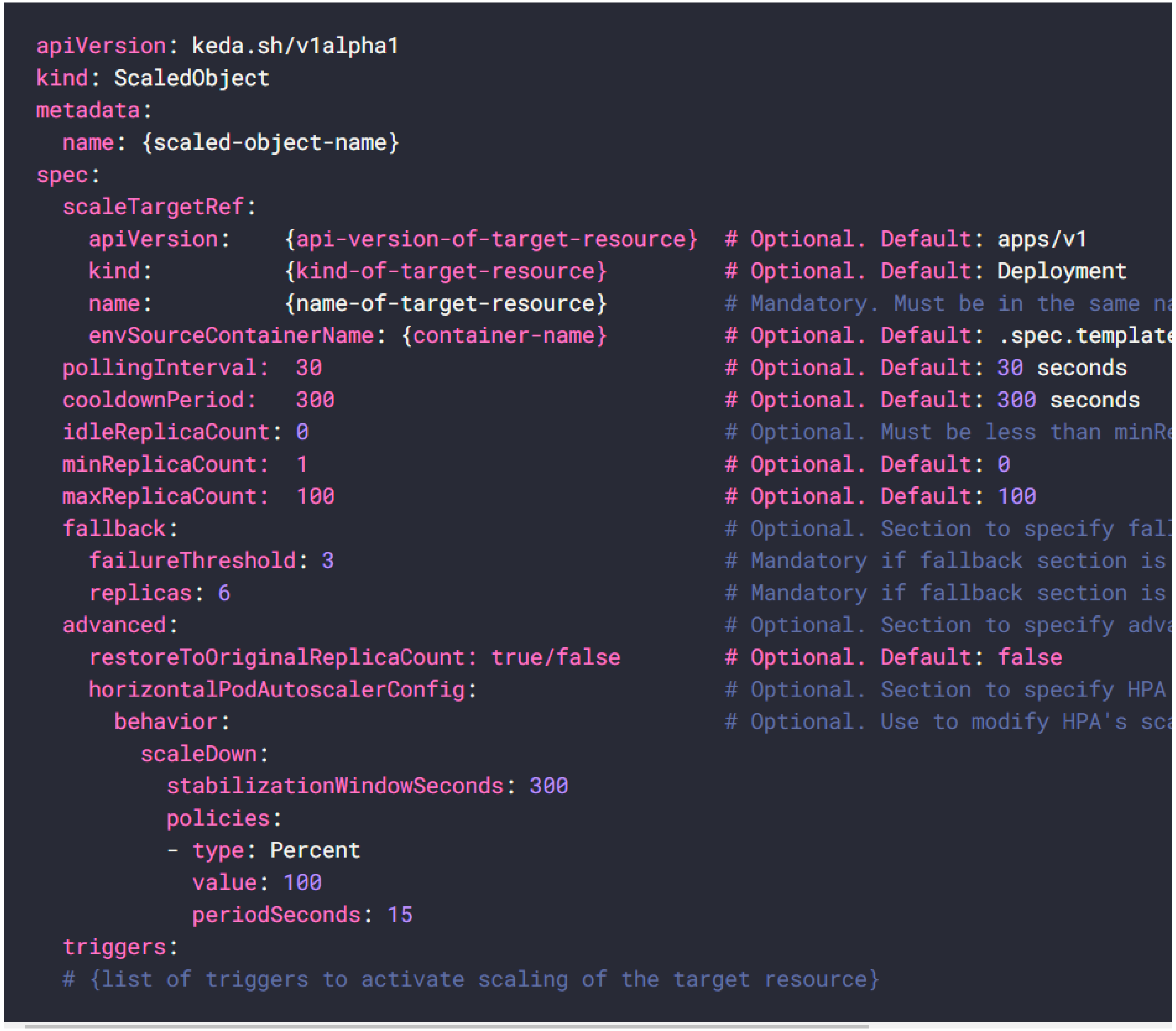
```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: order-processor-scaler
  labels:
    app: order-processor
    name: order-processor
spec:
  scaleTargetRef:
    name: order-processor
  # minReplicaCount: 0 Change to define how many minimum replicas you want
  maxReplicaCount: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



```yaml
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
```

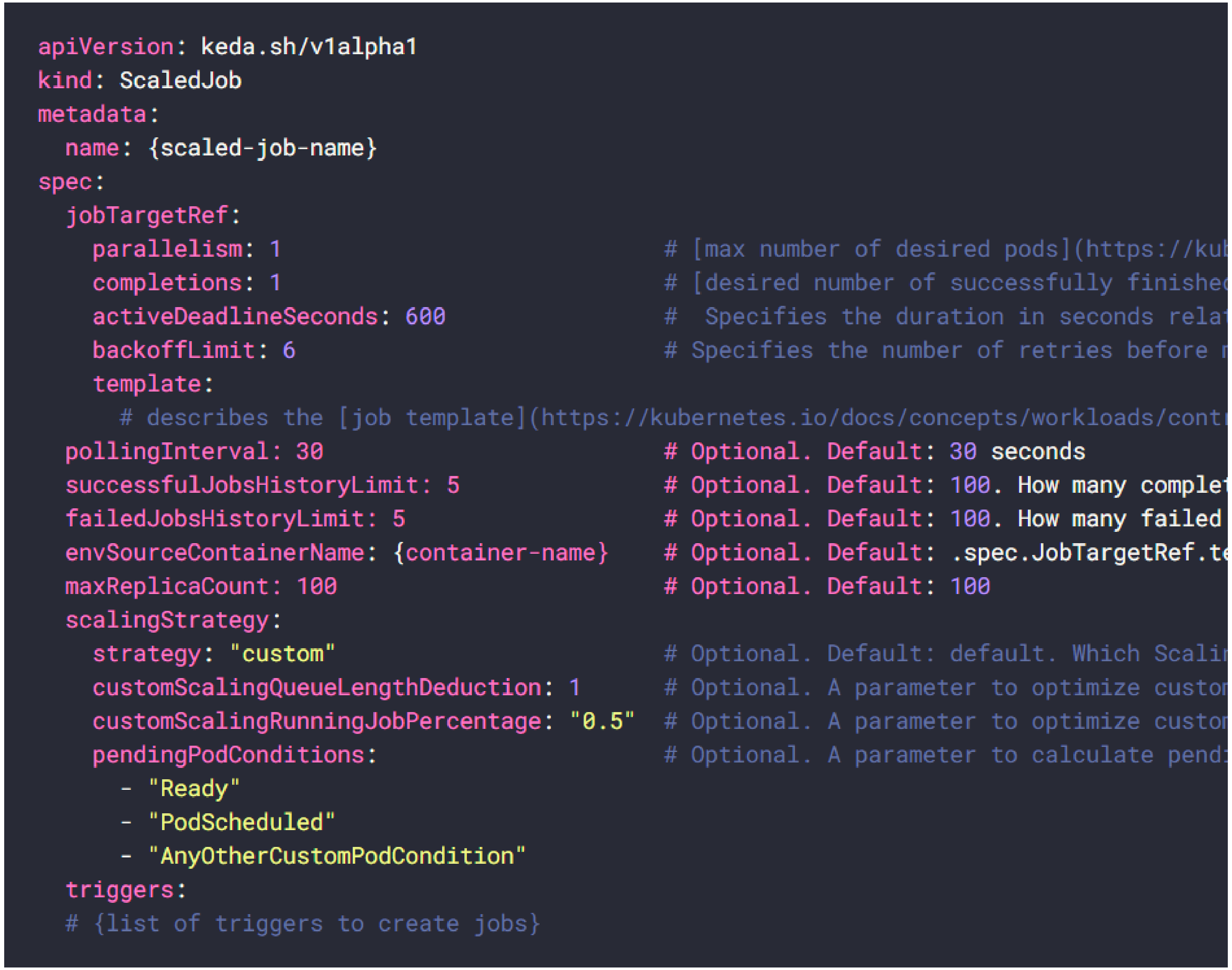
https://github.com/kedacore/sample-dotnet-worker-servicebus-queue

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 na
    envSourceContainerName: {container-name}         # Optional. Default: .spec.template
  pollingInterval:  30                               # Optional. Default: 30 seconds
  cooldownPeriod:   300                              # Optional. Default: 300 seconds
  idleReplicaCount: 0                                # Optional. Must be less than minRe
  minReplicaCount:  1                                # Optional. Default: 0
  maxReplicaCount:  100                              # Optional. Default: 100
  fallback:                                          # Optional. Section to specify fal
    failureThreshold: 3                              # Mandatory if fallback section is
    replicas: 6                                      # Mandatory if fallback section is
  advanced:                                          # Optional. Section to specify adva
    restoreToOriginalReplicaCount: true/false        # Optional. Default: false
    horizontalPodAutoscalerConfig:                   # Optional. Section to specify HPA
      behavior:                                      # Optional. Use to modify HPA's sca
        scaleDown:
          stabilizationWindowSeconds: 300
          policies:
          - type: Percent
            value: 100
            periodSeconds: 15
  triggers:
  # {list of triggers to activate scaling of the target resource}
```

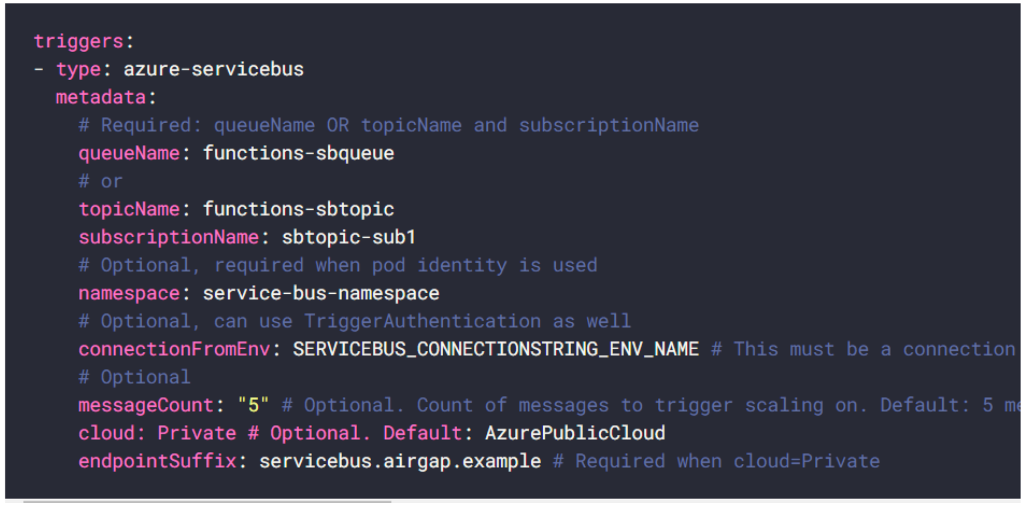
ScaledObject CRD – Job



```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: {scaled-job-name}
spec:
  jobTargetRef:
    parallelism: 1                            # [max number of desired pods](https://kub
    completions: 1                            # [desired number of successfully finishe
    activeDeadlineSeconds: 600                #  Specifies the duration in seconds relat
    backoffLimit: 6                           # Specifies the number of retries before r
    template:
      # describes the [job template](https://kubernetes.io/docs/concepts/workloads/contr
  pollingInterval: 30                         # Optional. Default: 30 seconds
  successfulJobsHistoryLimit: 5               # Optional. Default: 100. How many complet
  failedJobsHistoryLimit: 5                   # Optional. Default: 100. How many failed
  envSourceContainerName: {container-name}    # Optional. Default: .spec.JobTargetRef.te
  maxReplicaCount: 100                        # Optional. Default: 100
  scalingStrategy:
    strategy: "custom"                        # Optional. Default: default. Which Scalin
    customScalingQueueLengthDeduction: 1      # Optional. A parameter to optimize custom
    customScalingRunningJobPercentage: "0.5"  # Optional. A parameter to optimize custom
    pendingPodConditions:                     # Optional. A parameter to calculate pendi
      - "Ready"
      - "PodScheduled"
      - "AnyOtherCustomPodCondition"
  triggers:
  # {list of triggers to create jobs}
```

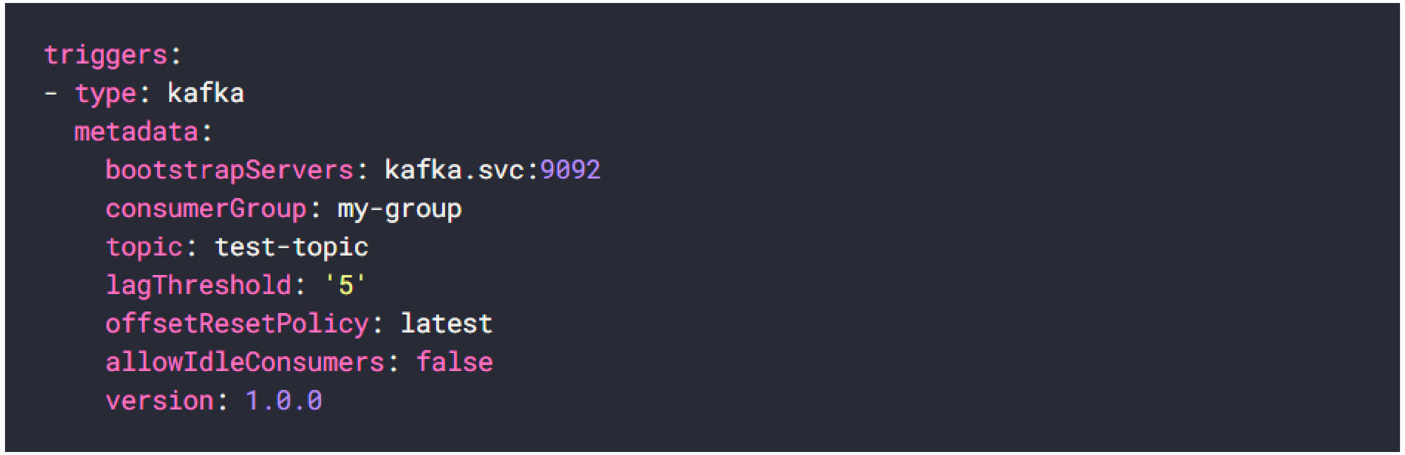
Triggers

Service Bus Trigger



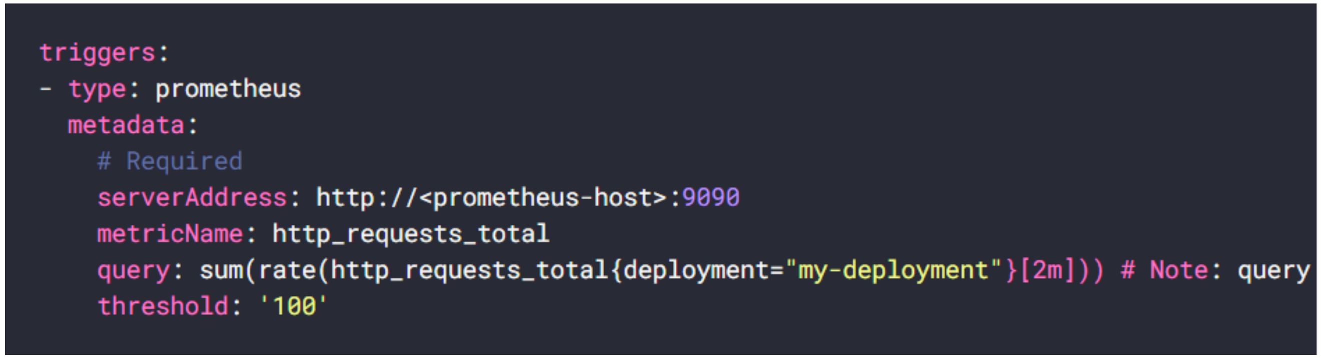
```
triggers:
- type: azure-servicebus
  metadata:
    # Required: queueName OR topicName and subscriptionName
    queueName: functions-sbqueue
    # or
    topicName: functions-sbtopic
    subscriptionName: sbtopic-sub1
    # Optional, required when pod identity is used
    namespace: service-bus-namespace
    # 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 me
    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
```

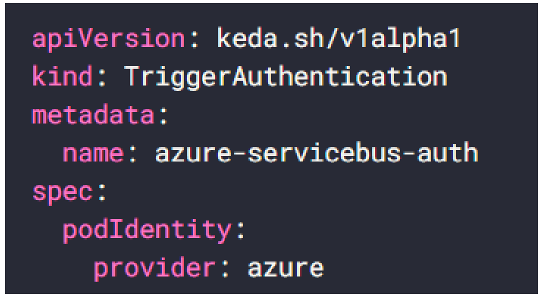
Prometheus Trigger



```
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'
```

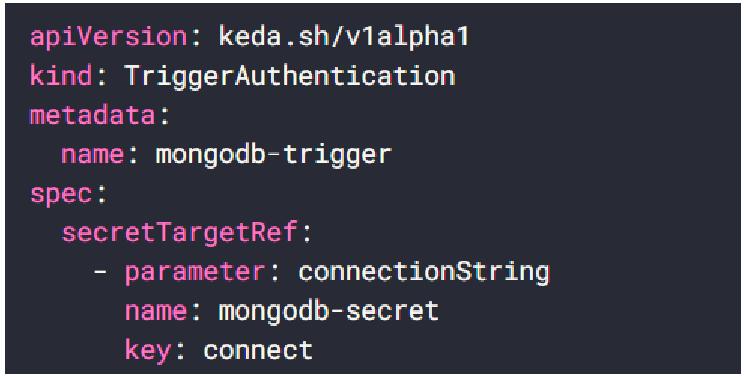
Trigger Authentication (Env Var, Secret, Pod Identity, Vault)

Pod Identity Auth



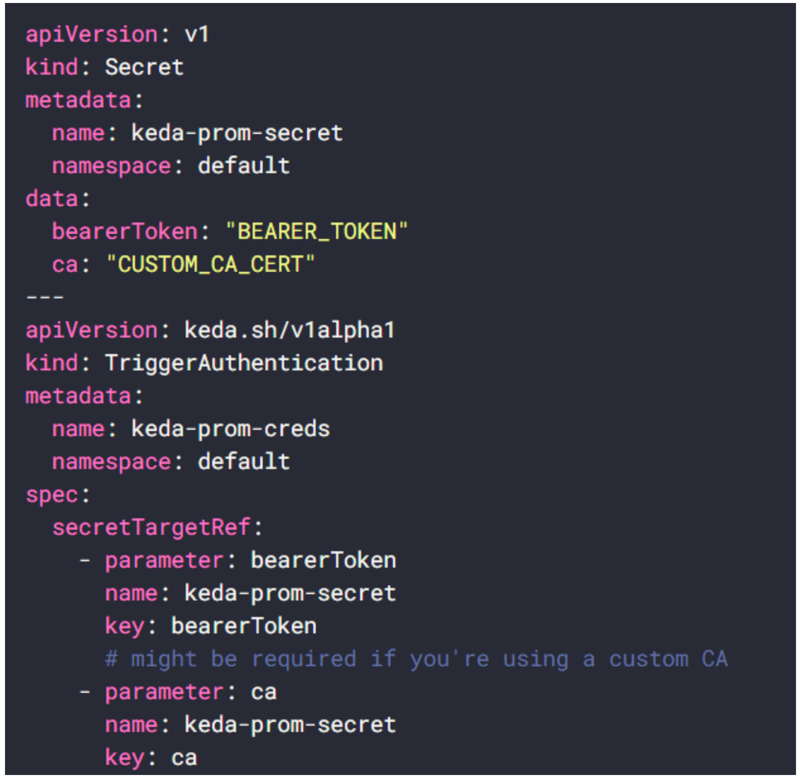
```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: azure-servicebus-auth
spec:
  podIdentity:
    provider: azure
```

Secret Auth (connection string)



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

Secret Auth (bearer token)



```yaml
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
```

KEDA – Event Sources and Scalers

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

Appendix - Dapr primer

State of enterprise developers

Deploying scale-out apps for flexibility, cost, and efficiency

Developing resilient, scalable, microservice-based apps that interact with services

Focusing on building applications, not infrastructure

Trending toward serverless platforms with simple code to cloud pipelines

Using multiple languages and frameworks during development

What is holding back microservice development?

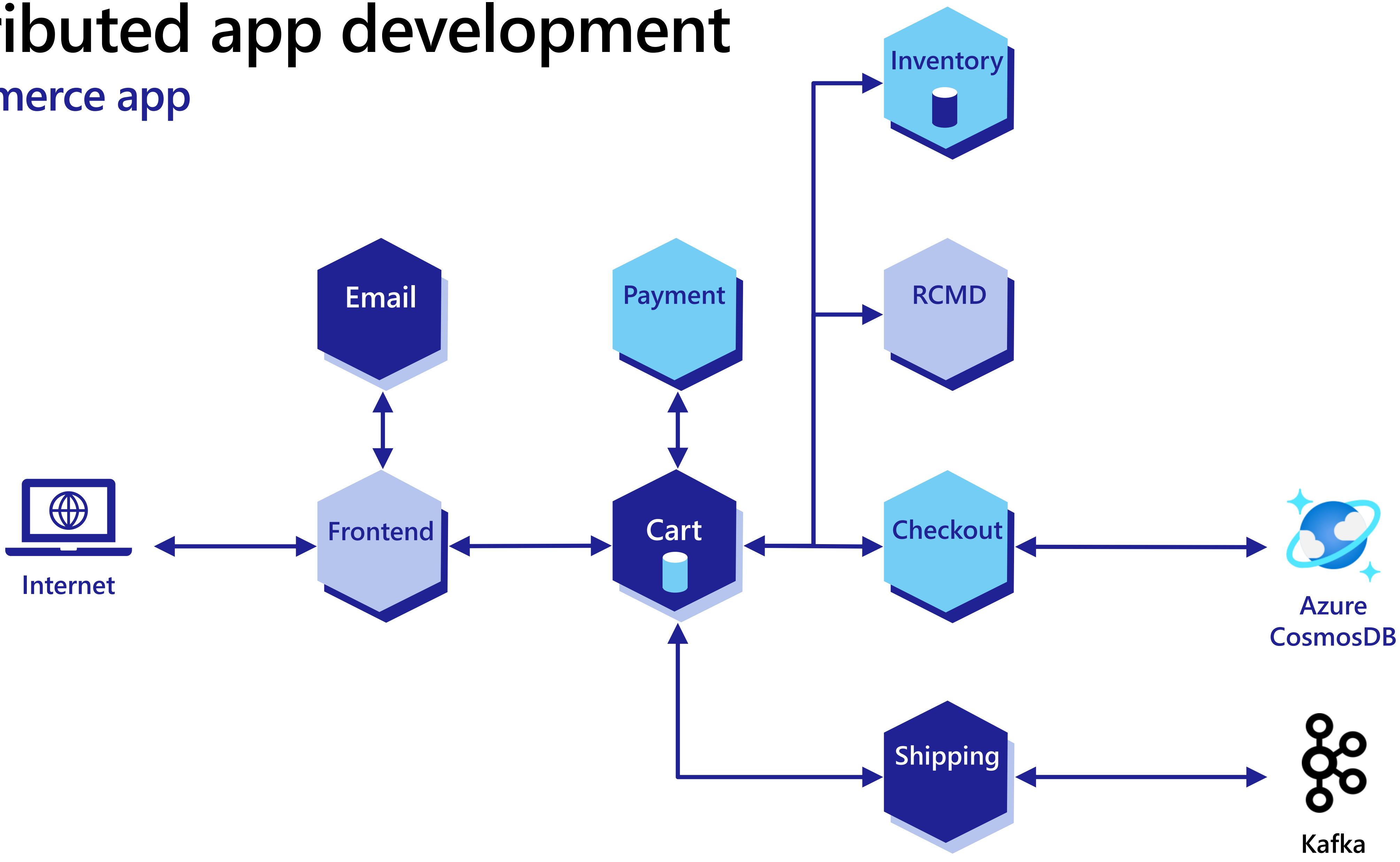
Limited tools and runtimes to build distributed applications

Runtimes have limited language support and tightly controlled feature sets

Runtimes only target specific infrastructure platforms with limited portability

Distributed app development

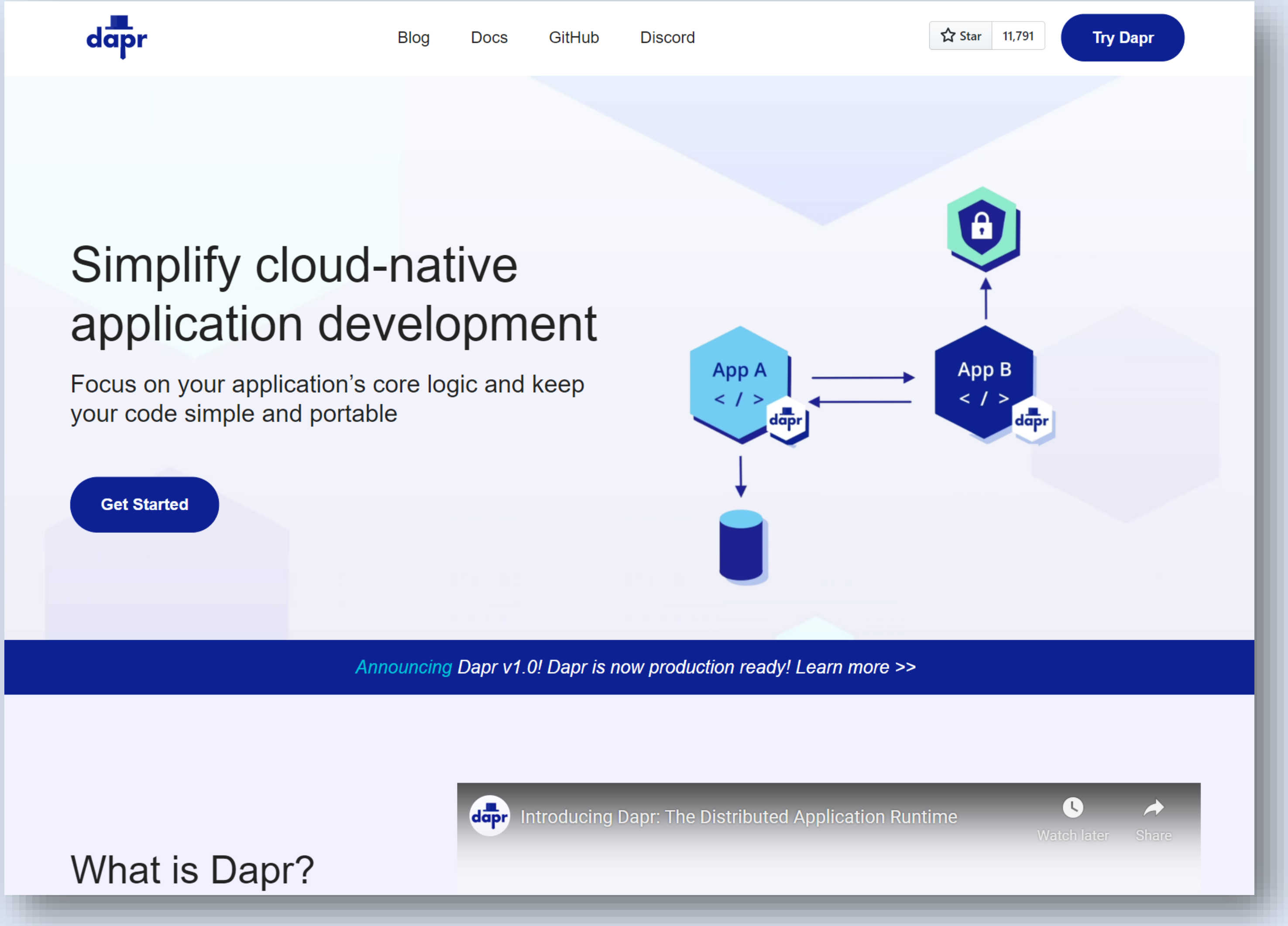
E-commerce app



Distributed Application Runtime

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

dapr.io



Simplify cloud-native application development

Focus on your application's core logic and keep your code simple and portable

Announcing Dapr v1.0! Dapr is now production ready! Learn more >>

Introducing Dapr: The Distributed Application Runtime

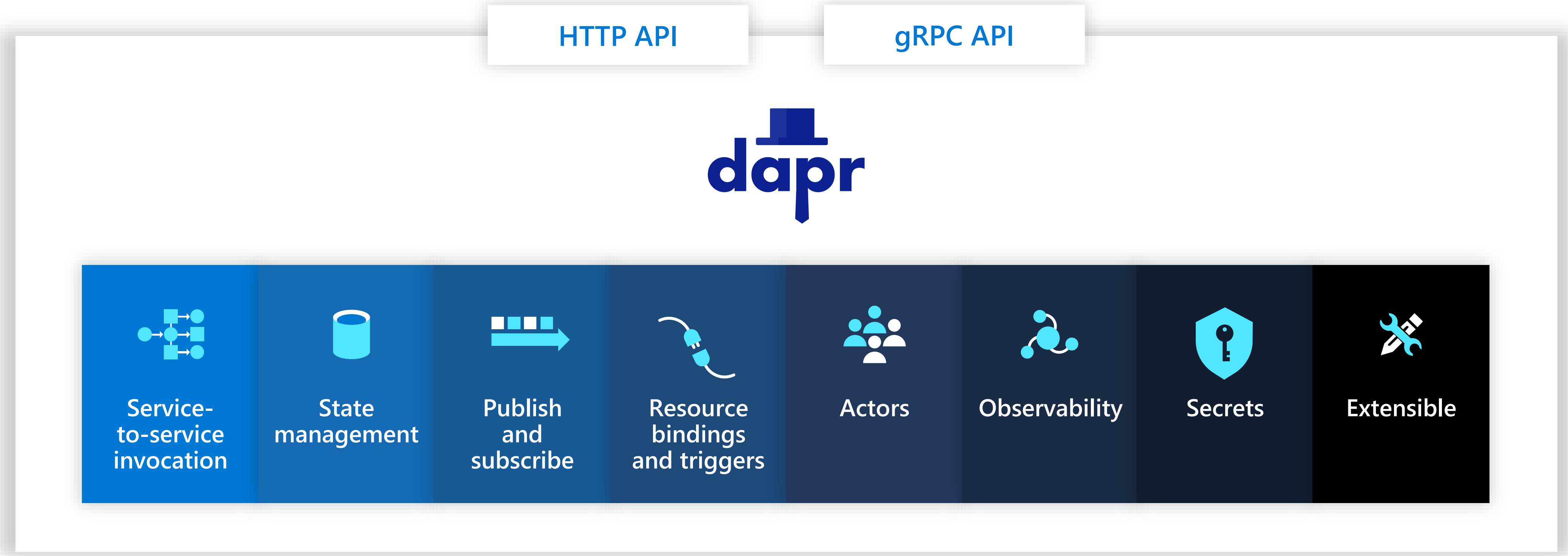
What is Dapr?

Dapr Goals

- Best-practices building blocks
- Any language or framework
- Consistent, portable, open APIs
- Adopt standards
- Extensible and pluggable components
- Platform agnostic cloud + edge

- Community driven, vendor neutral

Microservice building blocks



HTTP API

gRPC API

dapr

Service-to-service invocation

State management

Publish and subscribe

Resource bindings and triggers

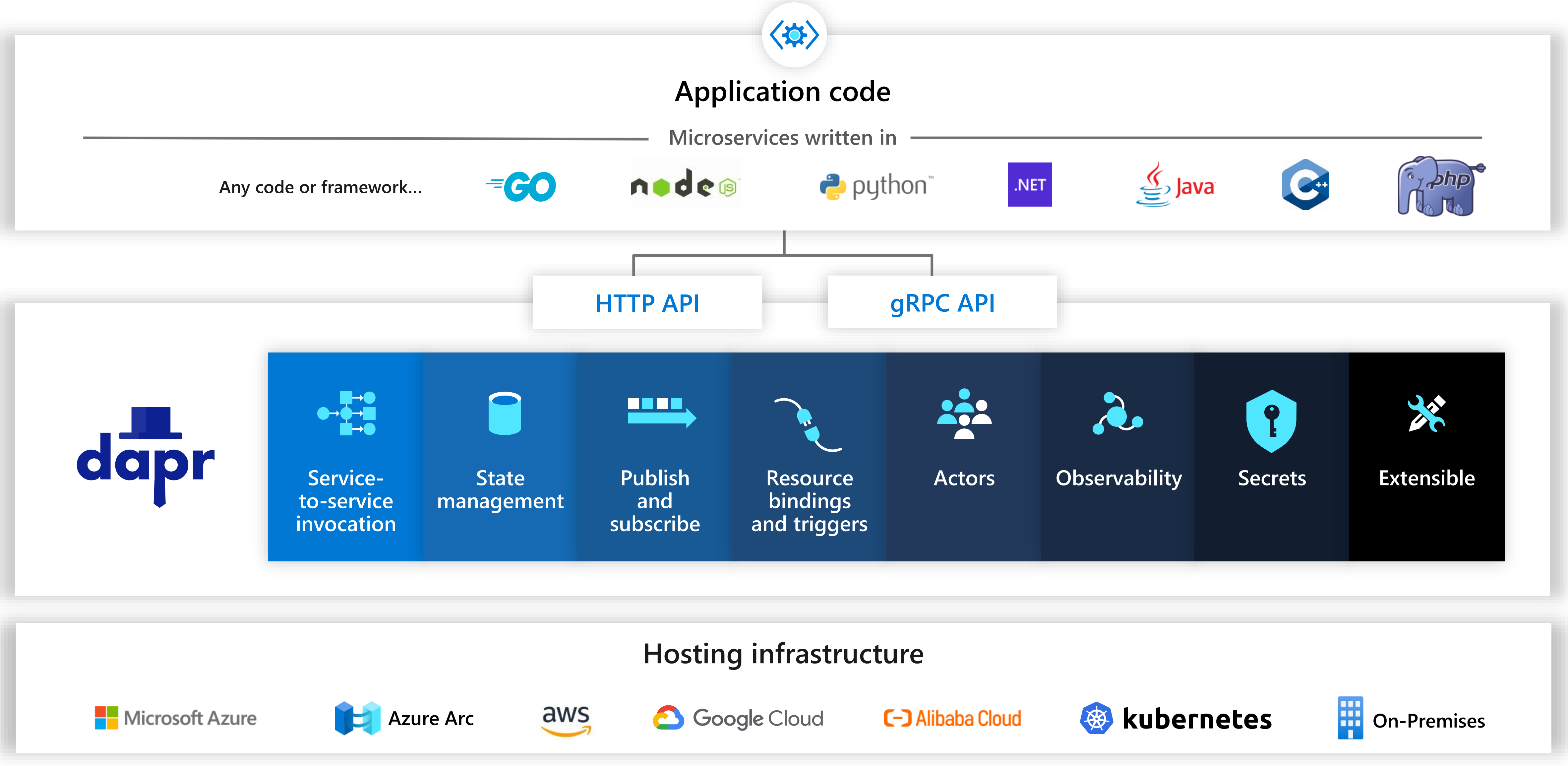
Actors

Observability

Secrets

Extensible

Any cloud or edge infrastructure



Application code

Microservices written in

Any code or framework...

HTTP API

gRPC API

- Service-to-service invocation
- State management
- Publish and subscribe
- Resource bindings and triggers
- Actors
- Observability
- Secrets
- Extensible

Hosting infrastructure

Microsoft Azure

Azure Arc

aws

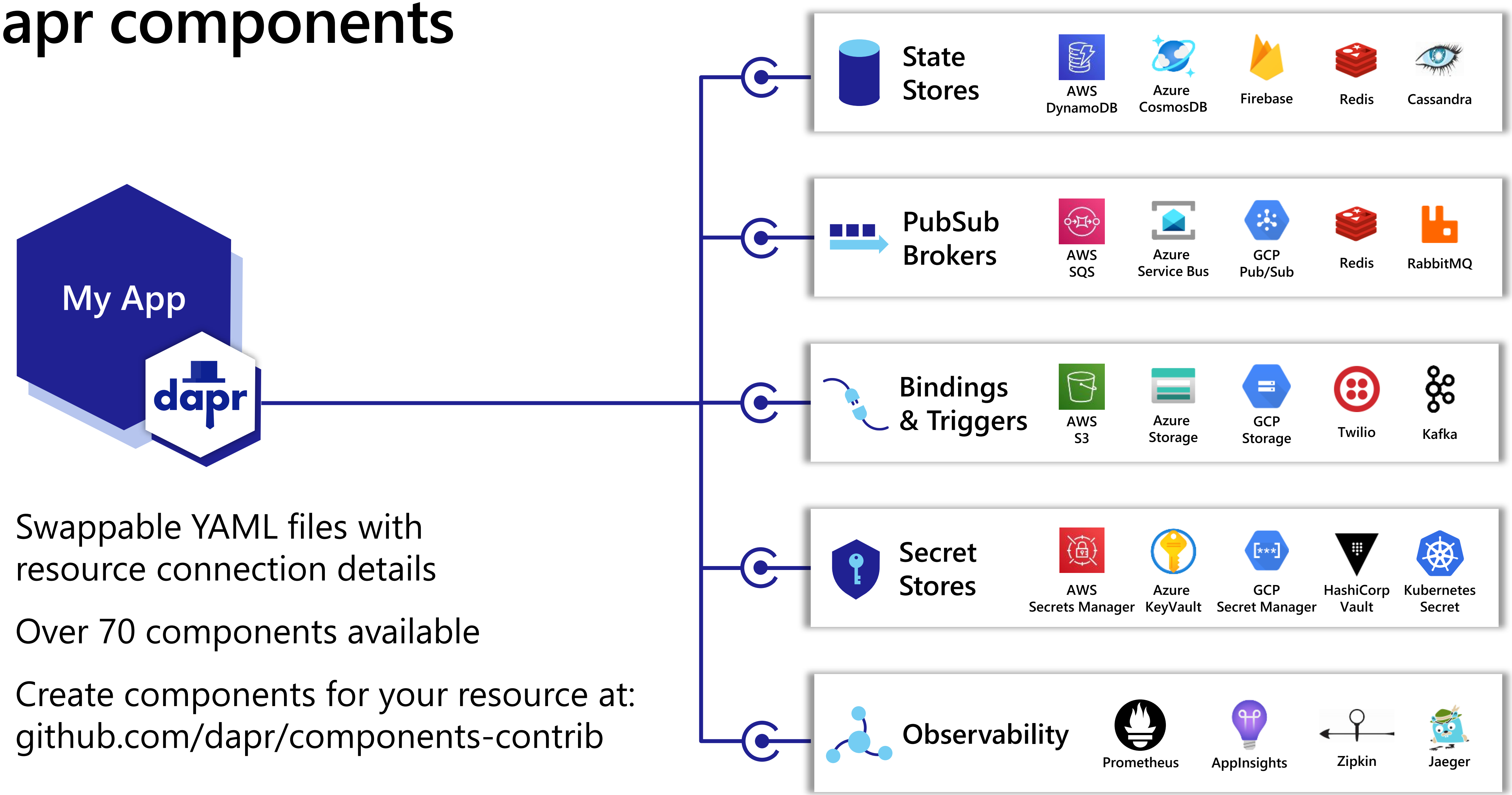
Google Cloud

Alibaba Cloud

kubernetes

On-Premises

Dapr components



My App

dapr

State Stores
- AWS DynamoDB
- Azure CosmosDB
- Firebase
- Redis
- Cassandra

PubSub Brokers
- AWS SQS
- Azure Service Bus
- GCP Pub/Sub
- Redis
- RabbitMQ

Bindings & Triggers
- AWS S3
- Azure Storage
- GCP Storage
- Twilio
- Kafka

Secret Stores
- AWS Secrets Manager
- Azure KeyVault
- GCP Secret Manager
- HashiCorp Vault
- Kubernetes Secret

Observability
- Prometheus
- AppInsights
- Zipkin
- Jaeger

Swappable YAML files with resource connection details

Over 70 components available

Create components for your resource at: github.com/dapr/components-contrib

Microsoft