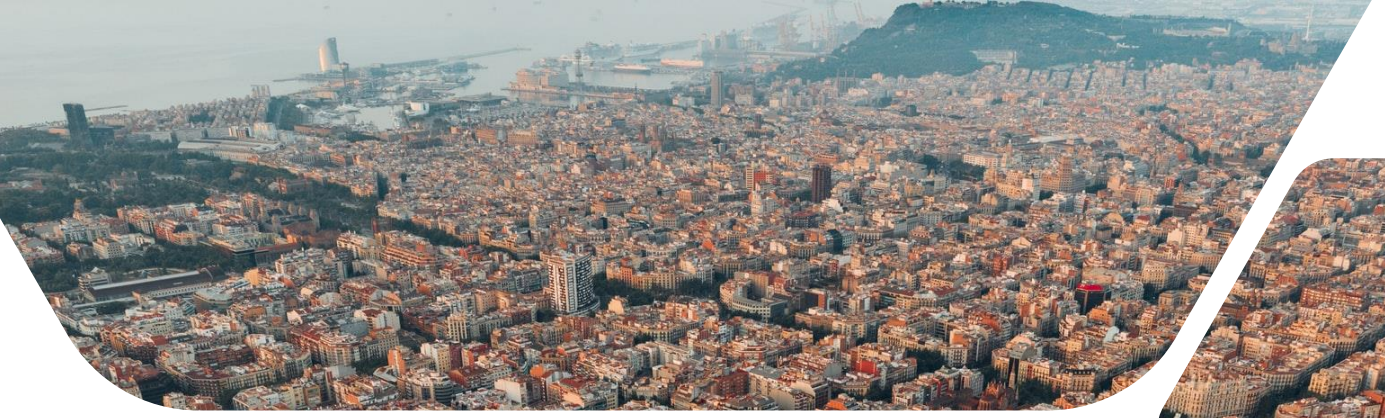
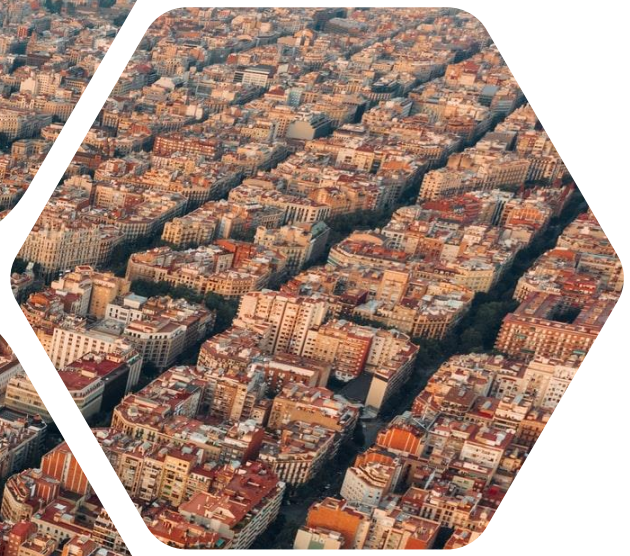
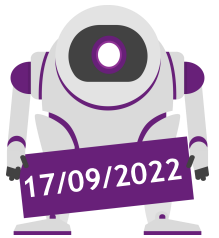




2022

Boost your power with Azure Container Apps

Manuel Sánchez
Txema González



Sponsors

Microsoft

NTT DATA

avanade

intelequia

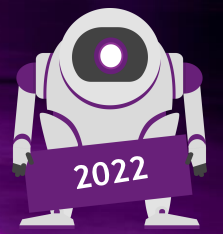
encamina

PIENSA EN COLORES

ERNI

TOKIOTA

#netcoreconf



Agenda

01 What is Azure Container Apps?

02 Container options in Azure

03 Azure Container Apps Benefits

04 What can you build with Azure Container Apps?

05 Demo

05 Q&A

Manuel Sánchez Rodríguez

Technical Manager & Azure Evangeslist at NTT DATA
Microsoft MVP Azure
@Manuss20 | Manuel Sánchez Rodríguez | LinkedIn

Txema González Balseiro

Azure Evangeslist at NTT DATA
Microsoft AI MVP
@tx_bals | Txema González Balseiro | LinkedIn

Azure Container Apps

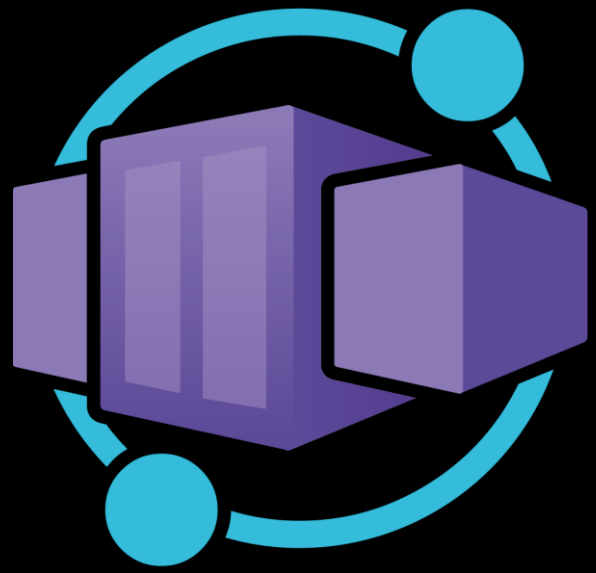
Serverless containers for microservices

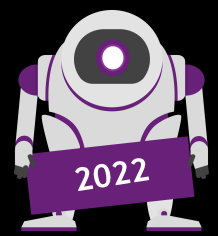
Build modern apps on open source

Focus on apps, not infrastructure

Seamlessly port to Kubernetes

kubernetes KEDA dapr envoy





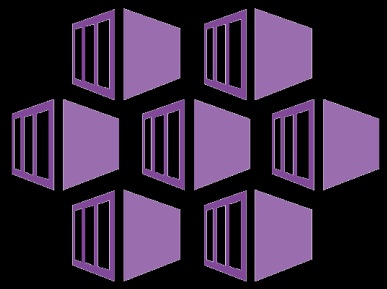
Container options in Azure

Azure Container Instance

- Simple solutions for deploying containers with a fixed scaling need.
- Does not support auto scaling or changing scale without recreating the service (downtime).
- Assumes ingress (must expose a port), so it is clunky to run headless jobs.

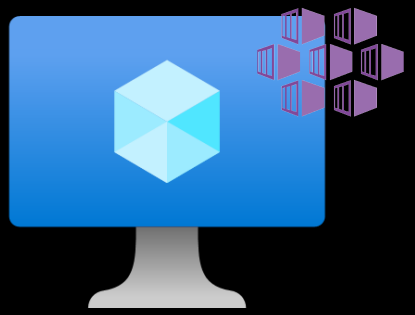
Azure App Services

- Easily launch and deploy HTTP-oriented services in containers.
- Built for HTTP base solutions.
- Allows Slot deployments for Green/Blue.
- Replication control (1 app instance per node, per-site scaling options).
- Supports auto-scale based on Azure Monitor metrics across the whole.



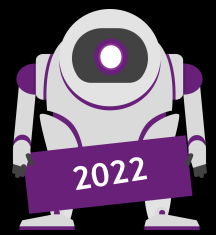
Azure Kubernetes Services

- Cluster-as-a-service
- Microsoft manages the master nodes, worker nodes size/scale defined by you.
- Tight-integration with Azure services such as application gateway, monitor, identity, etc.
- Auto scale options across nodes, access to K8s based scaling options for apps.



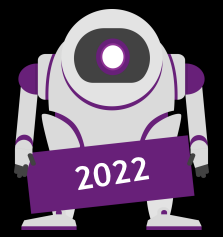
Kubernetes on Azure VM

- Bring-your-own-solution, run on Azure IaaS



Differences between Container Apps and other Azure Products

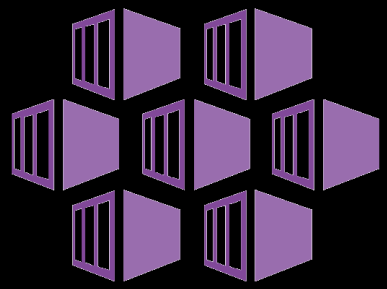
- **Azure Container Instances** – Single Container Instance Only
- **Azure Kubernetes Service** – You own and operate Kubernetes yourself.
- **Azure Functions** – Must use Functions SDK and limited to a set of languages.
- **Azure Web Apps** – No event-driven workloads.



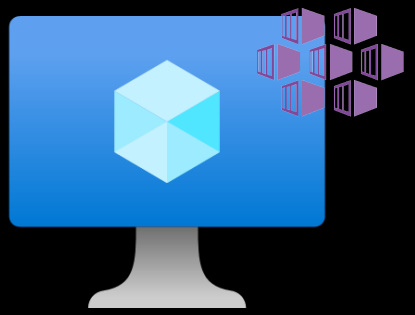
Container options in Azure

Azure Container Instance

Azure App Services



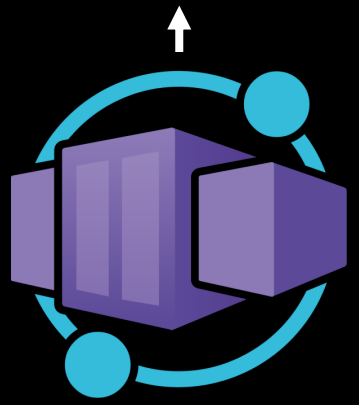
Azure Kubernetes Services



Kubernetes on Azure VM

Simple

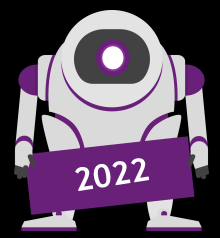
Less Flexible
Fewer Options



Azure Container Apps

Complex

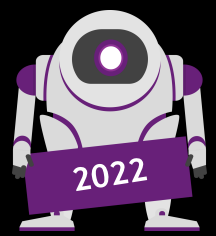
More Flexible
Most Options



Azure Container Apps Benefits

Don’t worry about a K8s cluster	Like Azure Container Instance, you just deploy the containers you want. There´s no cluster to worry about.
Auto-scaling with greater depth	More options for scaling your instances out than on Azure App Service with KEDA
Blue-Green by default	No downtime required for any deployment or scaling operation
Access to microservice management toolchain	Like AKS, Optional access to microservice management toolchain Dapr. gRPC or HTTP for internal service communication is available.

Azure Container Apps

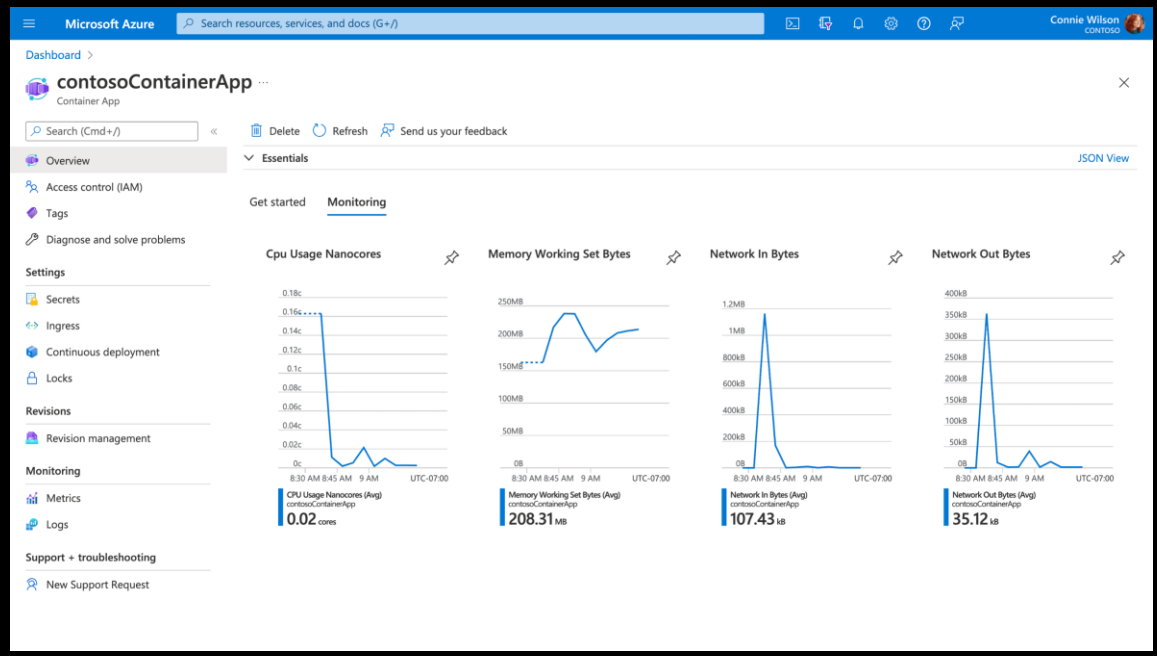


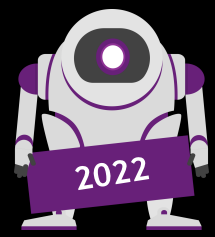
Scale with flexible serverless containers

Run containers and scale in response to HTTP traffic or a growing list of KEDA-supported scale trigger including Azure Event Hub, Apache Kafka, RabbitMQ Queue, MongoDB, MySQL and PostgreSQL.

Get robust autoscaling capabilities without the overhead of managing complex infrastructure.

Scale to zero and pay for only what you use, by the second.



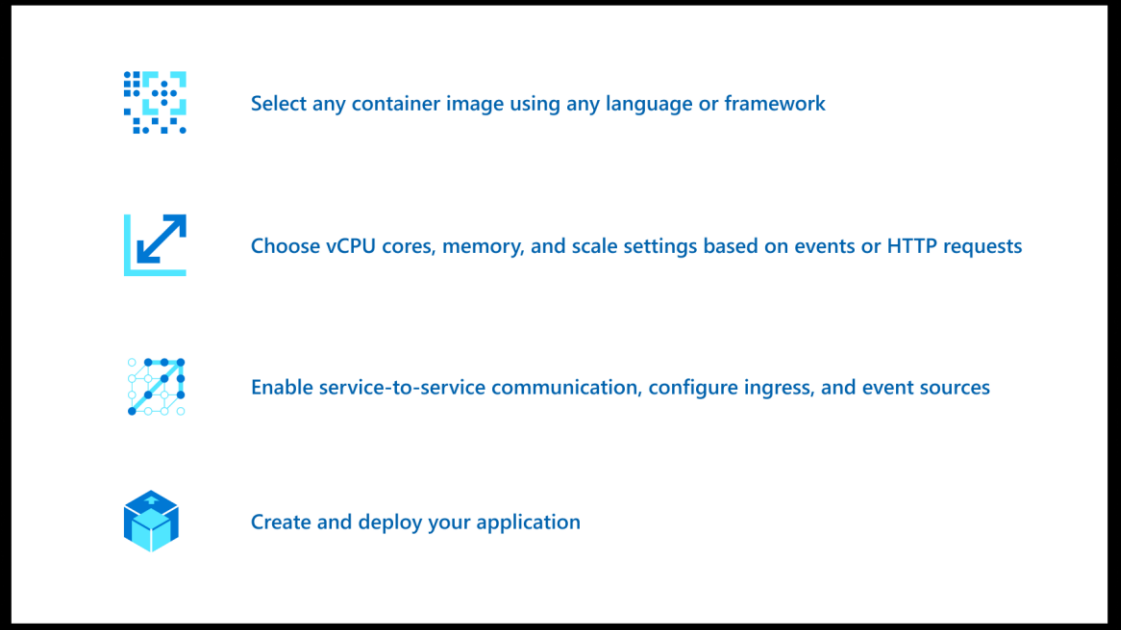


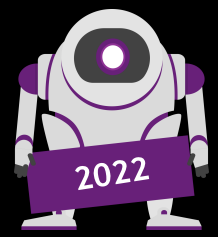
Accelerate developer productivity

Build microservices, APIs, event processing workers, and background jobs using containers.

Write code in you favorite programming language and accelerate development with built-in Distributed Application Runtime (Dapr) integration to simplify common tasks like event processing, pub/sub, and service invocation.

Set up code-to-cloud pipeline using GitHub Actions.



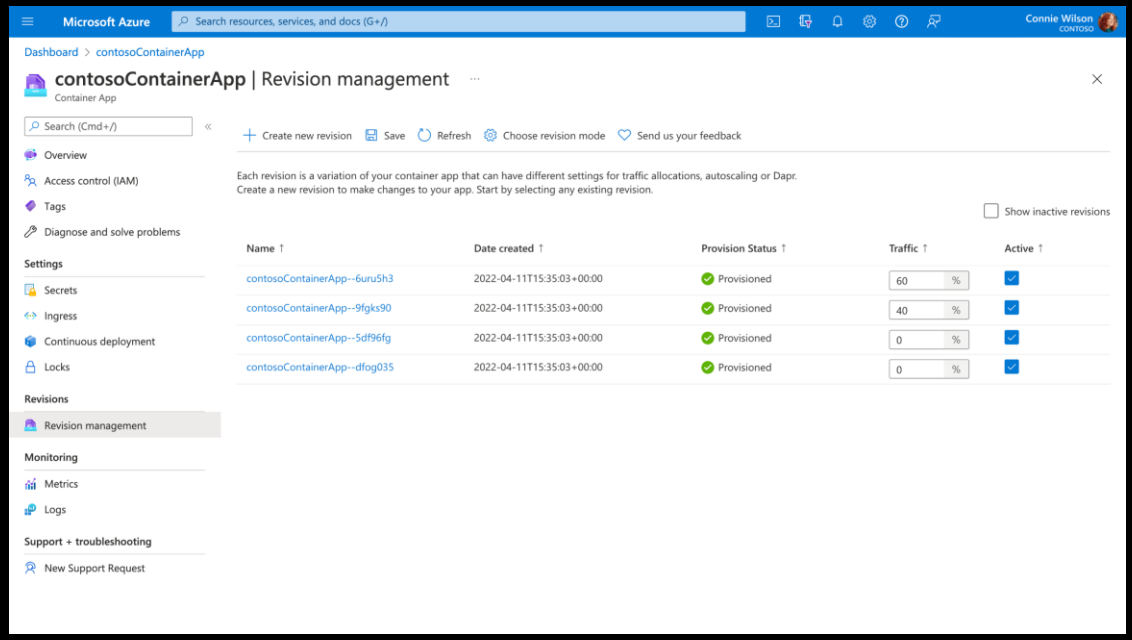


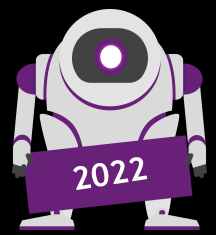
Build modern apps on open-source

Create modern apps with open standards on a Kubernetes foundation and portability in mind.

Contribute directly to OSS projects to influence product capabilities.

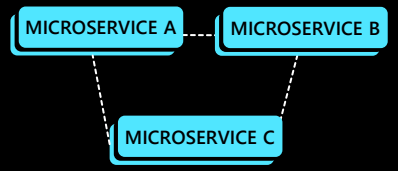
Rely on streamlined application lifecycle task such as application upgrades and versioning, traffic shifting, service discovery, and monitoring.





What can you build with Azure Container Apps?

Microservices



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

AUTO-SCALE CRITERIA

Individual microservices can scale independely using any KEDA scale trigger.

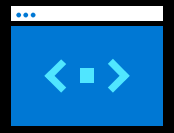
Event-driven processing

E.g., queue reader application that processes messages as the arrive in a queue.

AUTO-SCALE CRITERIA

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

Web Applications

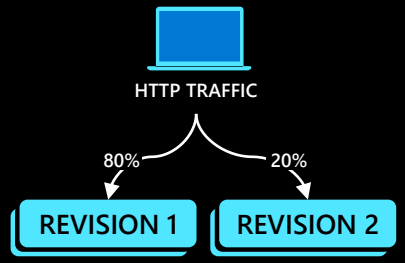


Deploy web apps with custom domains, TLS certificates, and integrated authentication.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP request.

Public API endpoints



HTTP requests are split between two revisions of the app – the first revision gets 80% of the traffic, while a new revision receives 20%

AUTO-SCALE CRITERIA

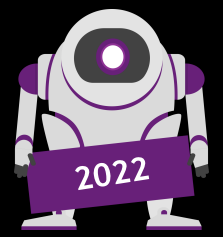
Scaling is determined by the number of concurrent HTTP request.

Background processing

E.g., continuously running background process that transforms data in a database.

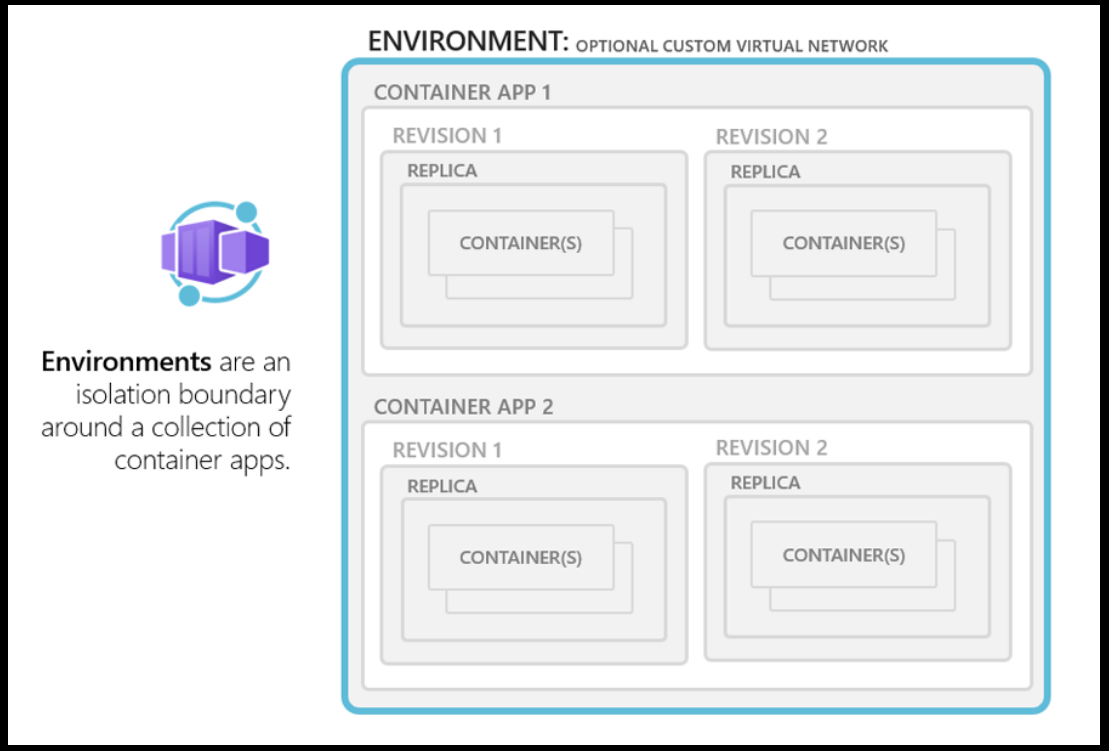
AUTO-SCALE CRITERIA

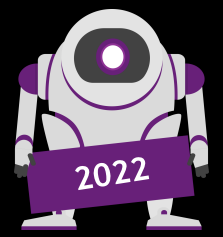
Scaling is determined by the level CPU or memory load.



Azure Container Apps enviroments

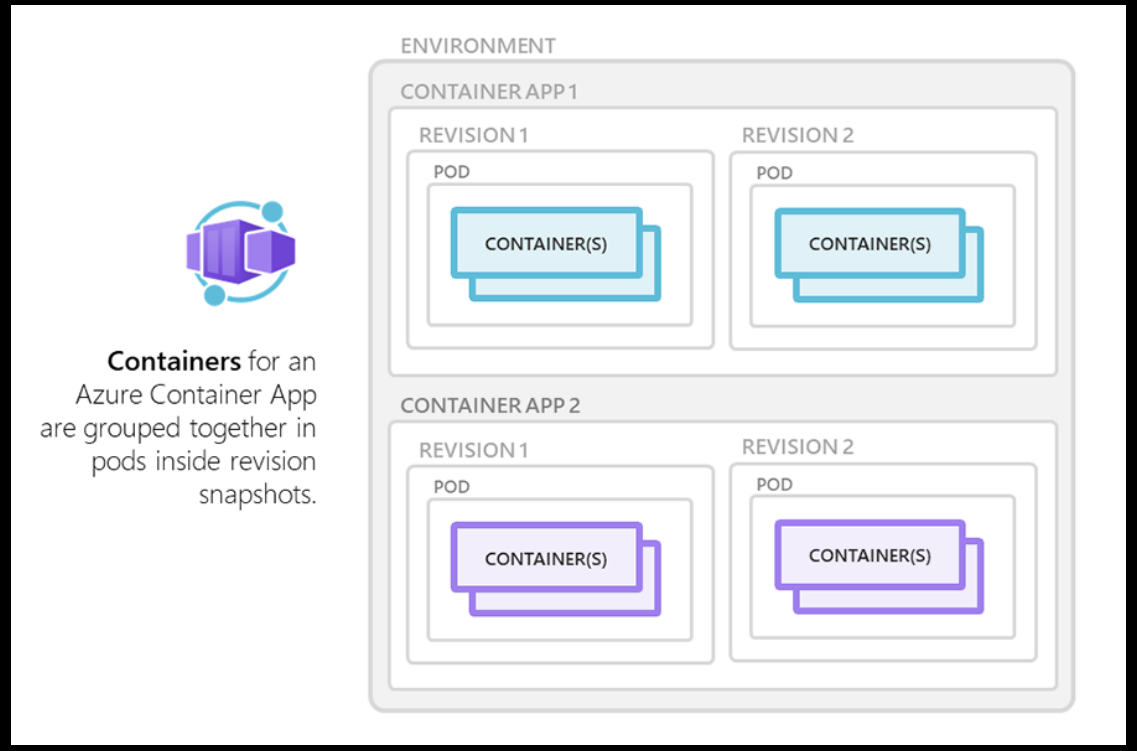
Environments define an isolation and observability boundary around a collection of container apps deployed in the same virtual network.

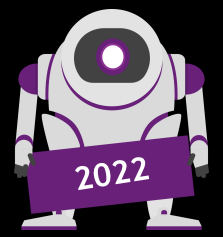




Containers

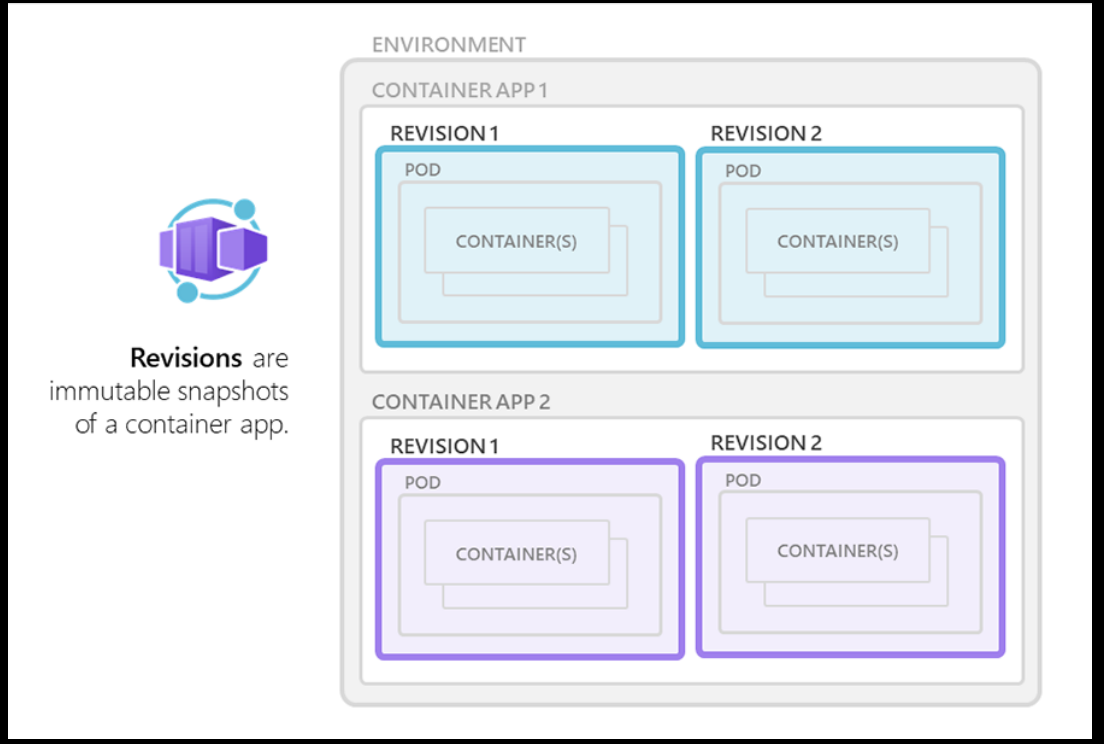
Containers for an Azure Container App are grouped together in pods inside revision snapshots.





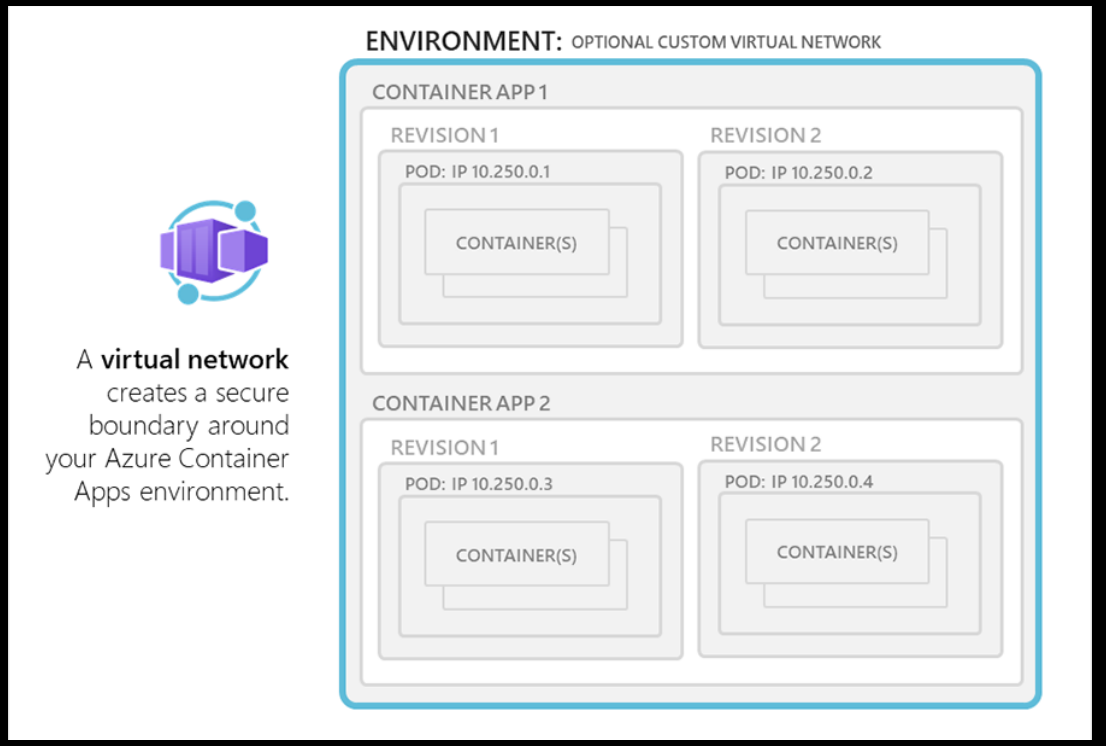
Revisions

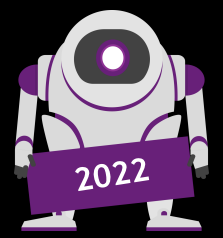
Revisions are immutable version snapshots of a container app.



Virtual Network

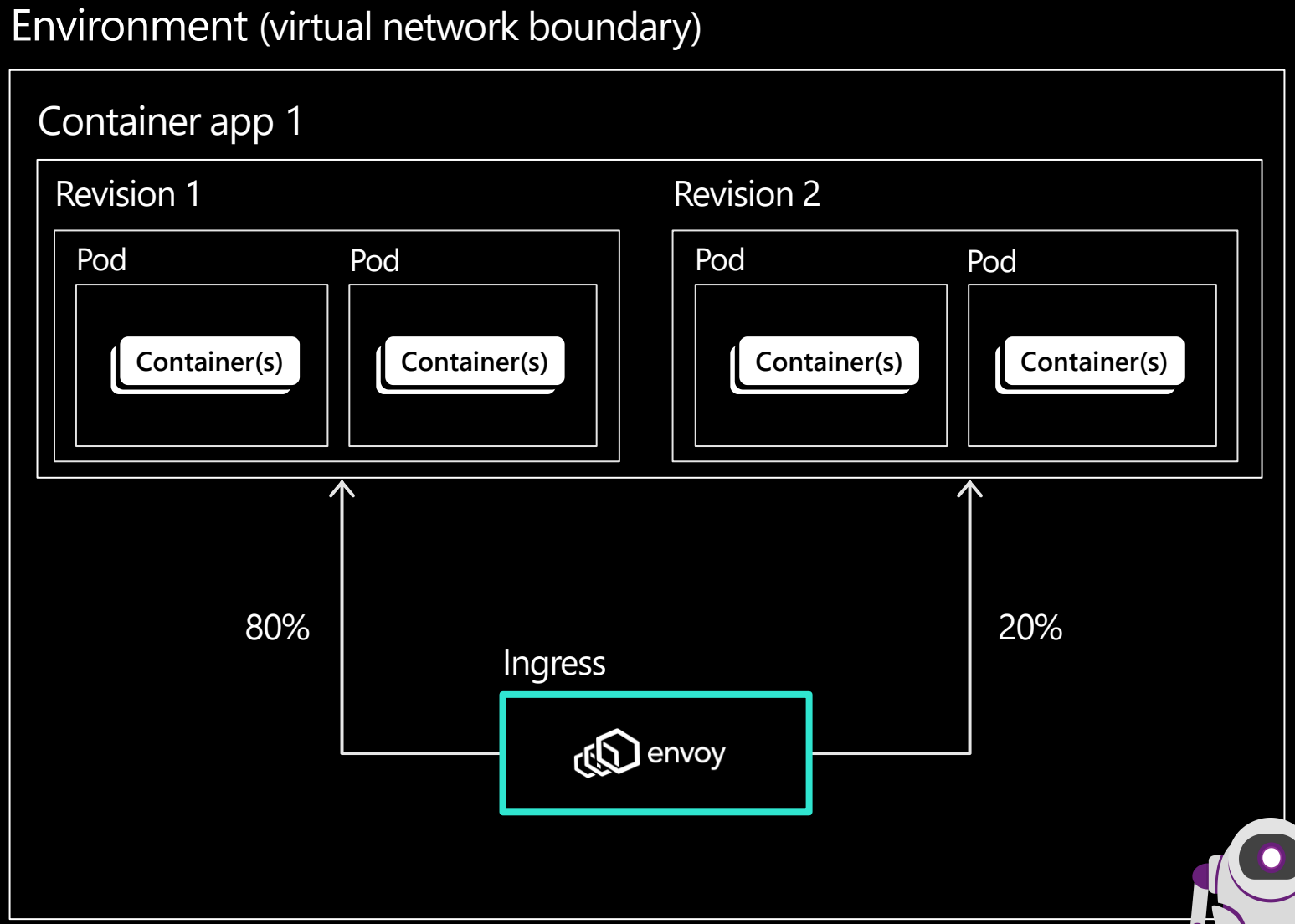
Internal or external visibility with TLS termination and support for HTPP/1.1 and HTTP/2 Ingress





Ingress

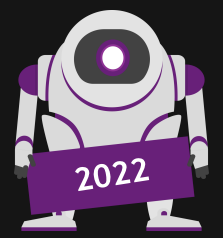
Internal or external visibility with TLS termination and support for HTTP/1.1 and HTTP/2



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"
                }
            ]
        }
    ],
}
```



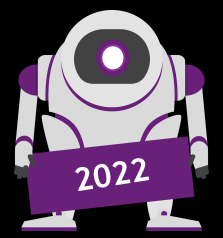
Managed Identity

A **system-assigned identity** is tied to your container app and is deleted when your container app is deleted. An app can only have one system-assigned identity.

A **user-assigned identity** is a standalone Azure resource that can be assigned to your container app and other resources. A container app can have multiple user-assigned identities. The identity exists until you delete them.

Limitations:

- Pull an image from Azure Container Registry
- Define scaling rules or Dapr configuration
- To access resources that require a connection string or key, such as storage resources, you'll still need to include the connection string or key in the secretRef of the scaling rule.

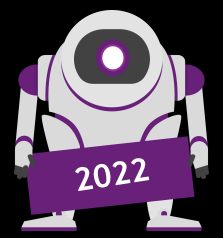


Observability

- Log Analytics – stderr/stdout, small 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:

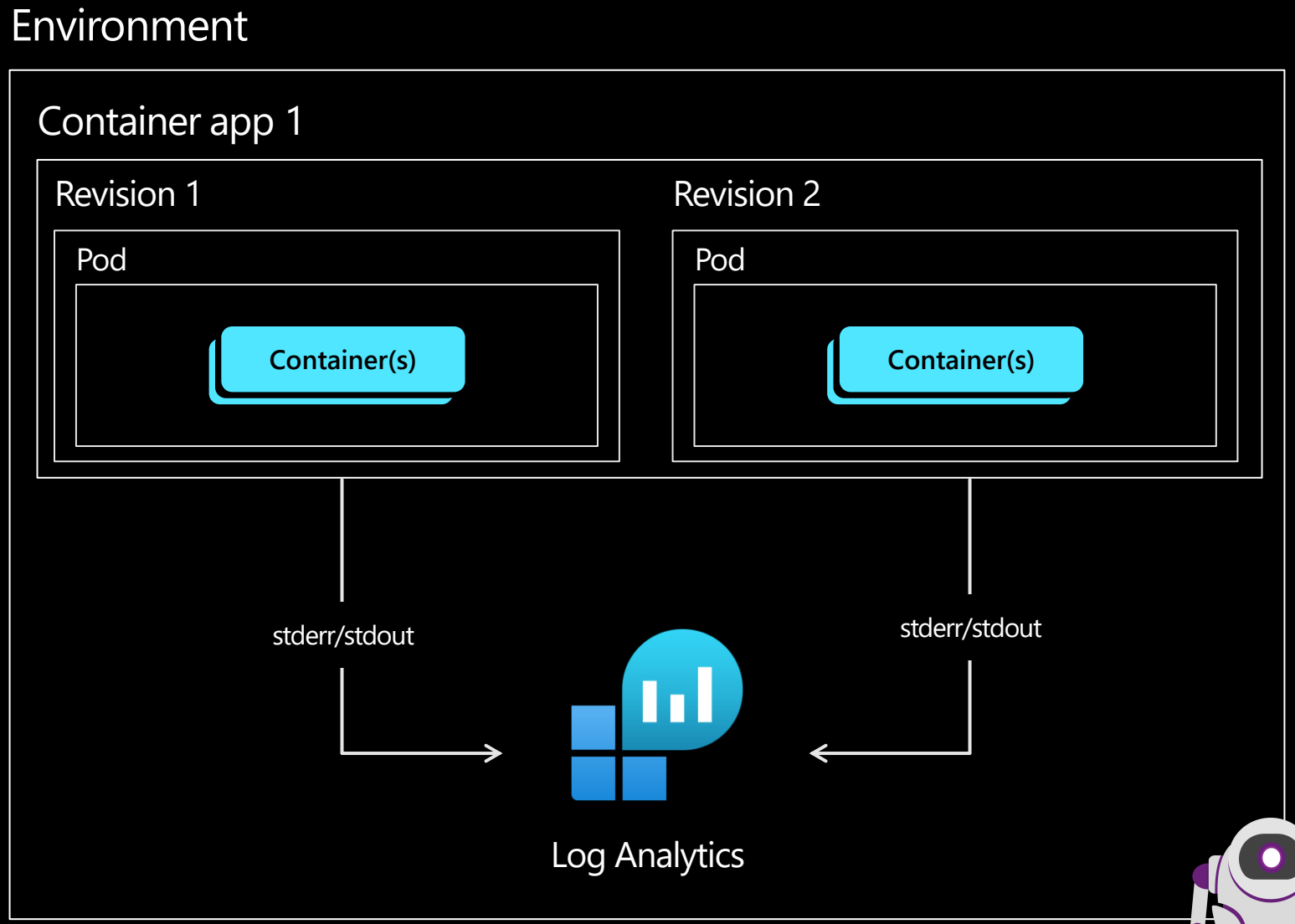
  az containerapp logs show -n MyContainerapp -g MyResourceGroup
- Connect to Console – connect to run shell commands

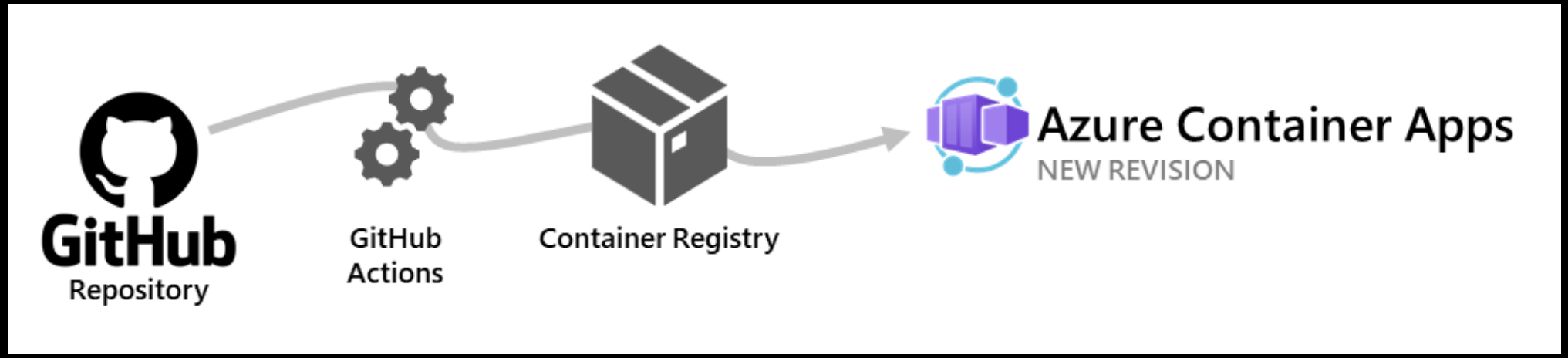
  az containerapp exec -n MyContainerapp -g MyResourceGroup --command bash



Logging

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




GitHub
Repository
GitHub
Actions
Container Registry
Azure Container Apps
NEW REVISION

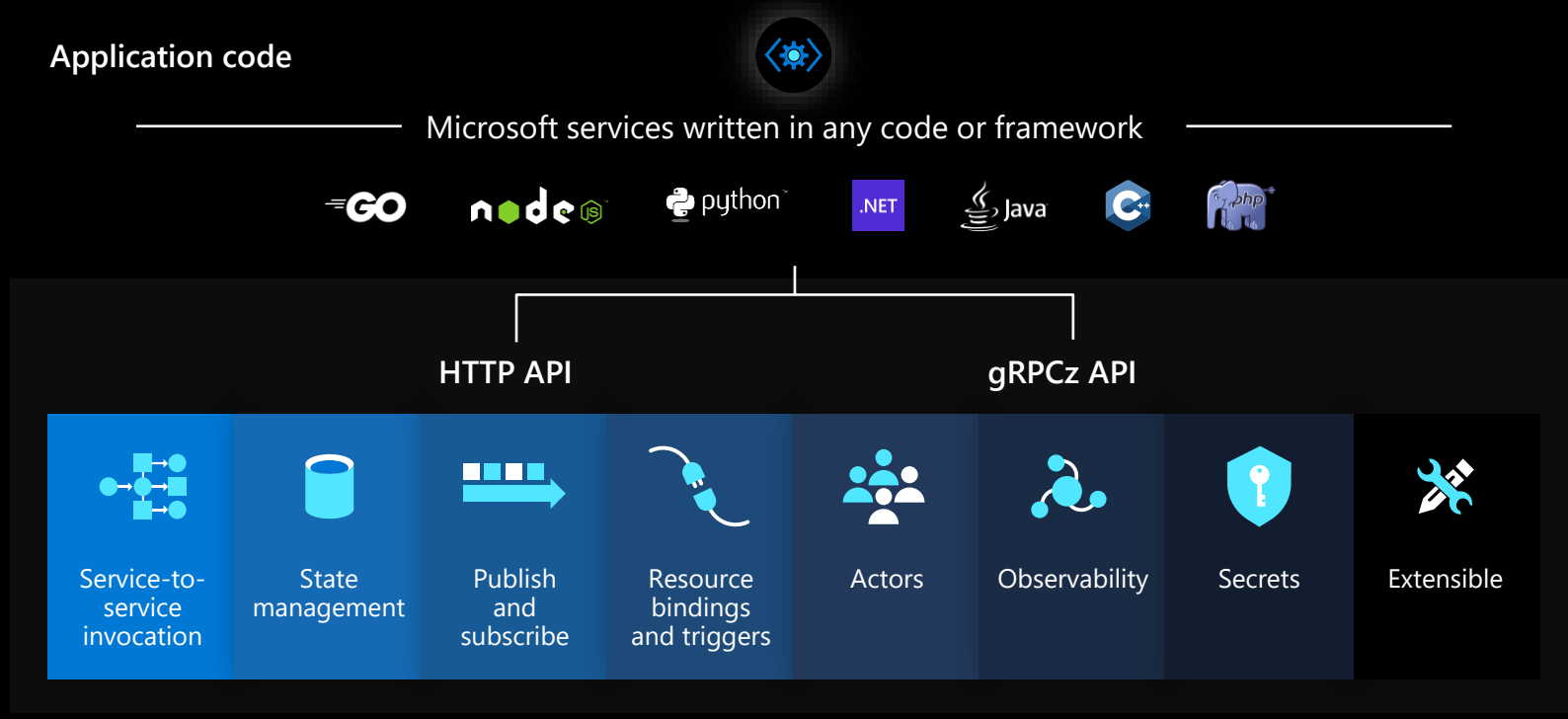
Microservices using any language or framework

Any cloud or edge infrastructure

dapr

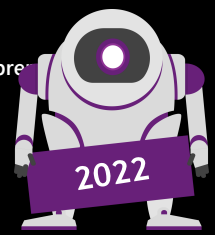
Distributed Application Runtime

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



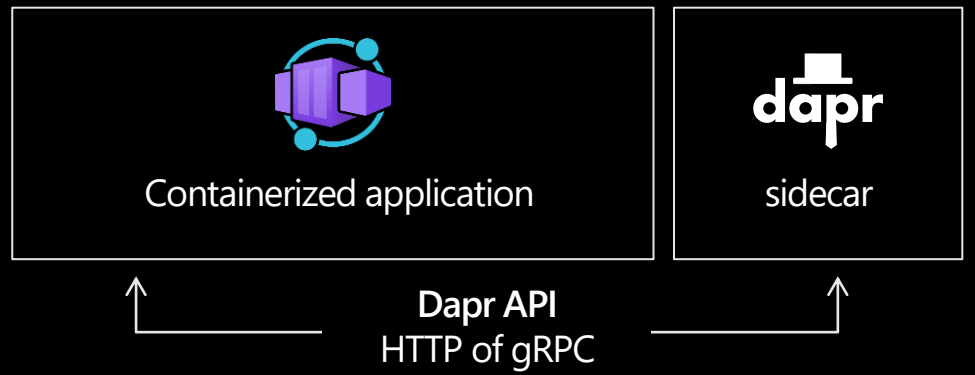
dapr.io

CLOUD NATIVE COMPUTING FOUNDATION



#netcoreconf

Fully managed Dapr using the sidecar model



Service-to-service invocation

```
POST http://localhost:3500/v1.0/invoke/cart/method/neworder
```

State management

```
GET http://localhost:3500/v1.0/state/inventory/item67
```

Publish and subscribe

```
POST http://localhost:3500/v1.0/publish/shipping/orders
```

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

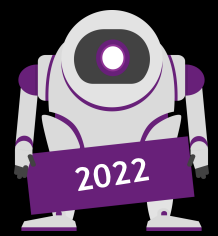
Support for triggers across variety of cloud providers & products

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

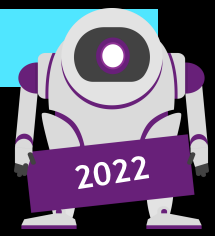
```
{
 "name": "cpu-rule",
  "custom": {
   "type": "cpu",
    "metadata": {
     "type": "Utilization",
      "value": "50"
   }
  }
}
```

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



DEMO

Sponsors

Microsoft

NTT DATA

avanade

intelequia

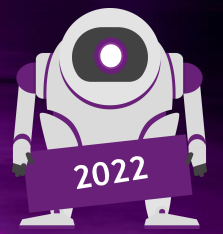
encamina

PIENSA EN COLORES

ERNI

TOKIOTA

#netcoreconf



More information:
info@netcoreconf.com
@Netcoreconf

Visit on:
netcoreconf.com