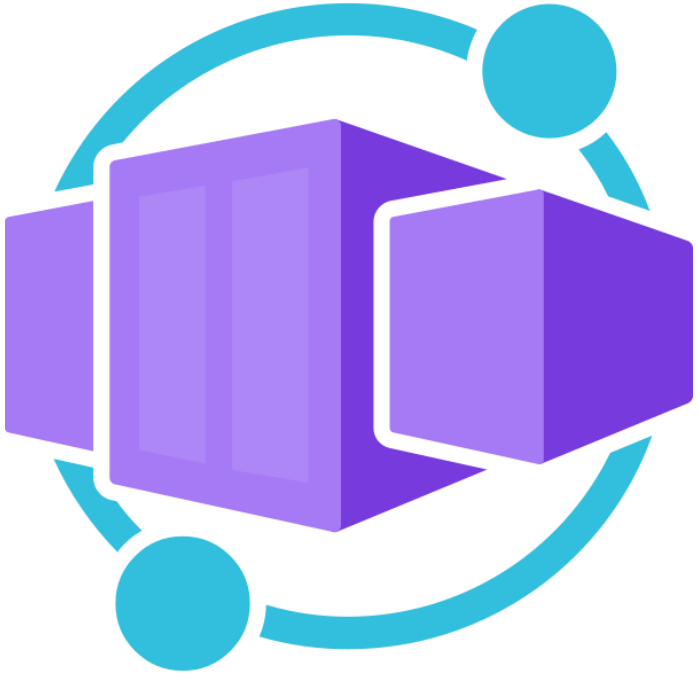
Azure Container Apps

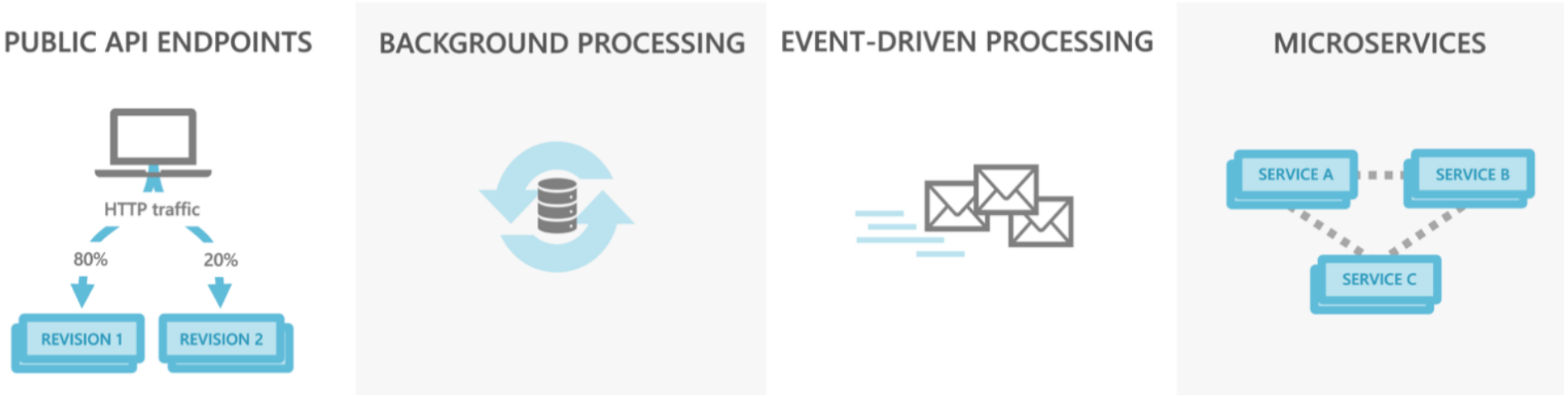
Memi Lavi
www.memilavi.com



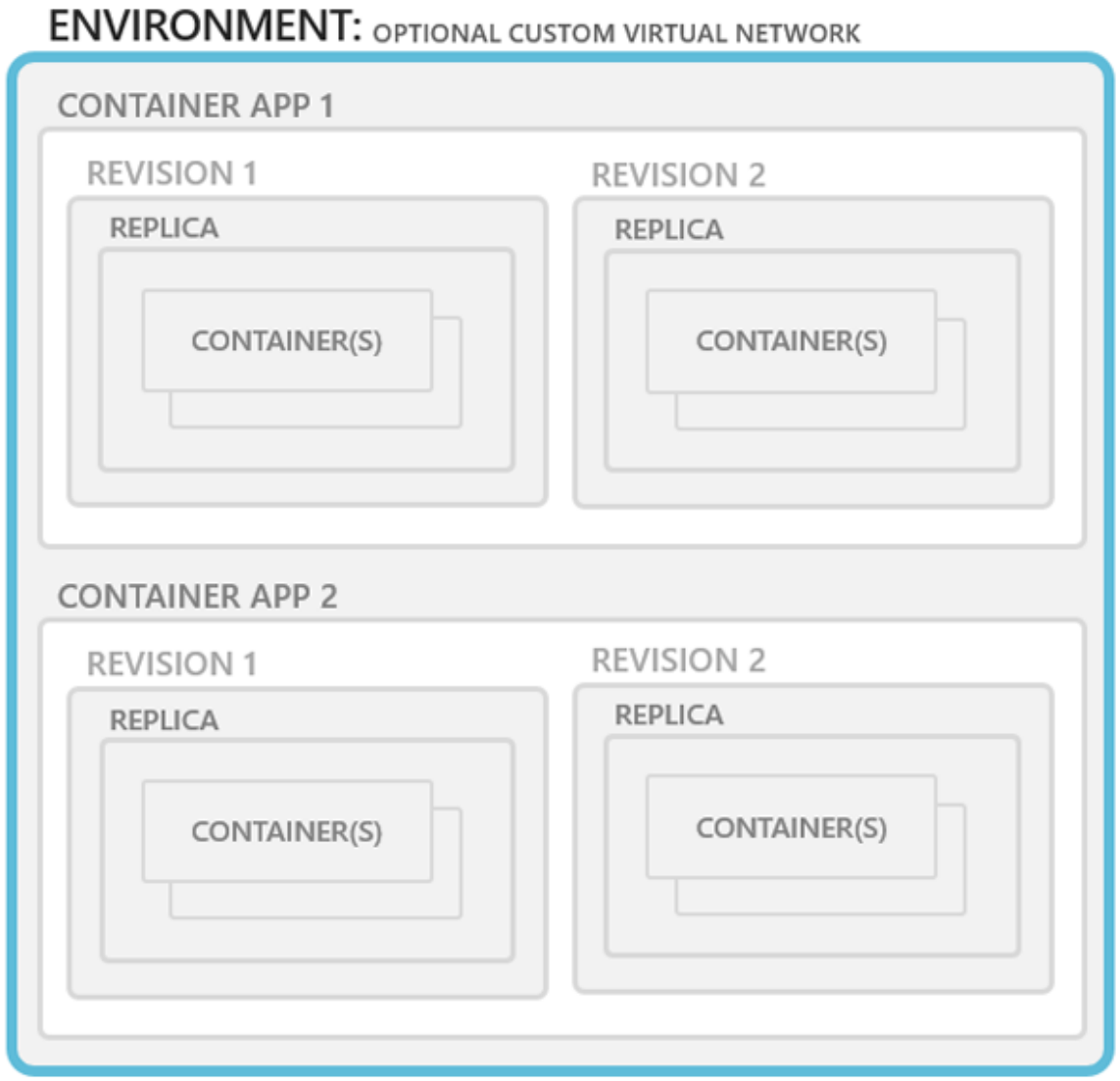
- Build and deploy fully managed apps and microservices using serverless containers
- Sophisticated autoscaling
- Ingress configuration for traffic security
- Inter-service communication with Dapr
- Rich logging and monitoring
- SLA of 99.95%

Scenarios Supported

- With Azure Container Apps you can deploy various types of apps



Architecture of Container Apps



Environment

- Boundary around one or more container app
- Powers the underlying infrastructure
- Provides the VNet used by the container apps
  - You can use an existing VNet
- Provides unified logging and monitoring
- Handles OS upgrades, scale, failover, balancing and more

Environment VNet

Generated VNet

- Automatically created
- Inaccessible to you
- Can reach only internet accessible endpoints

Existing VNet

- Finer control
- Supports peering, NSG, private endpoints

Environment

Single Environment

- Related apps
- Apps using the same VNet
- Apps using the same log destination
- Apps communicating with the same Dapr service

vs

Multiple Environments

- Don't want to share compute resources
- Apps not using the same VNet, log or Dapr
- Separate teams
- Separate environment usage (test, prod)

Environment Types

Consumption Only

- Autoscaling
- Pay only for what you use
- Scale-to-zero (no payment)

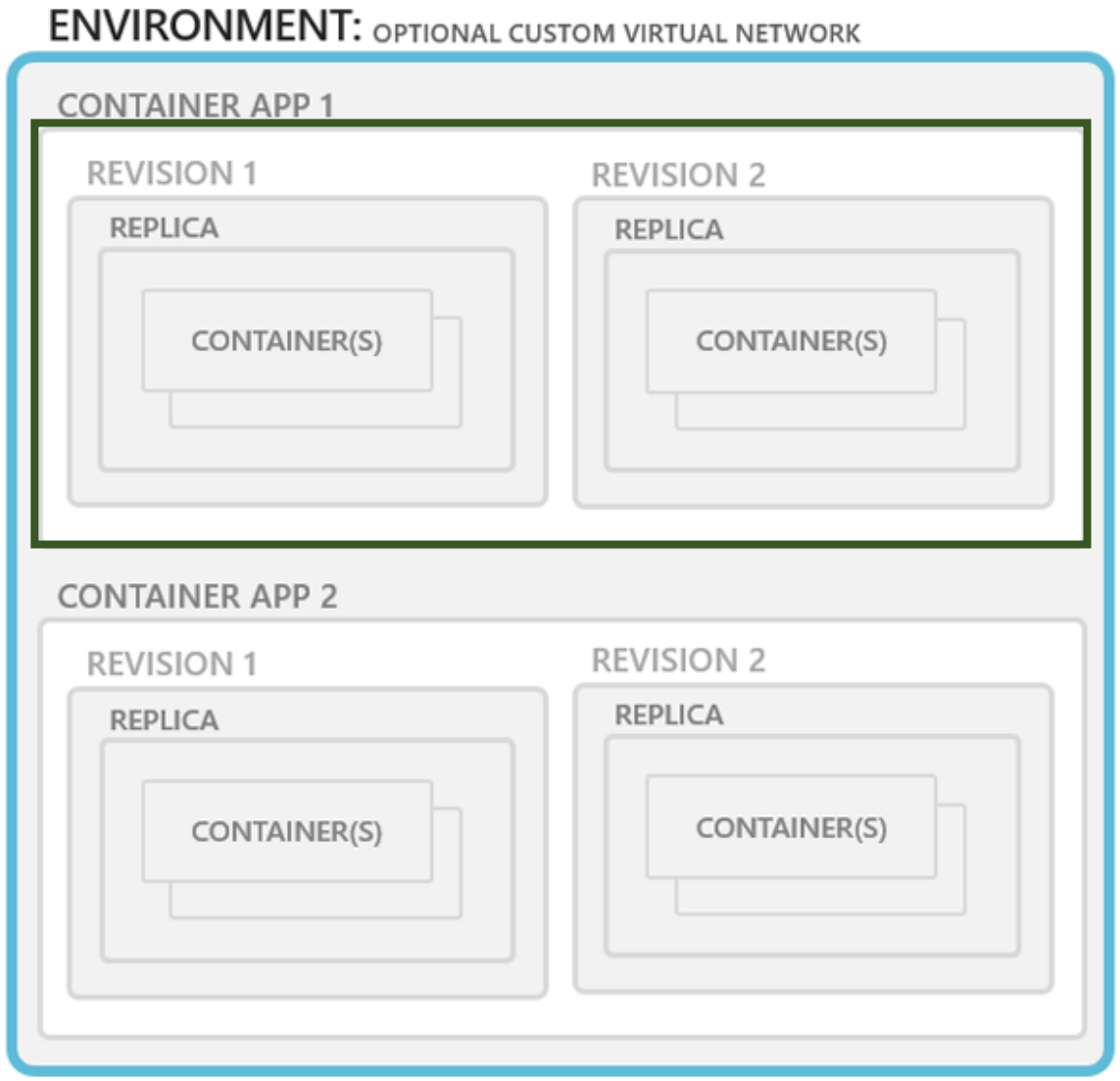
Workload Profile

- Choose between Consumption Only and dedicated hardware
- Allows hardware customization and isolation
- Predictable cost

Creating Environments

- Environments are created as part of Container Apps deployment
- Can also be created separately

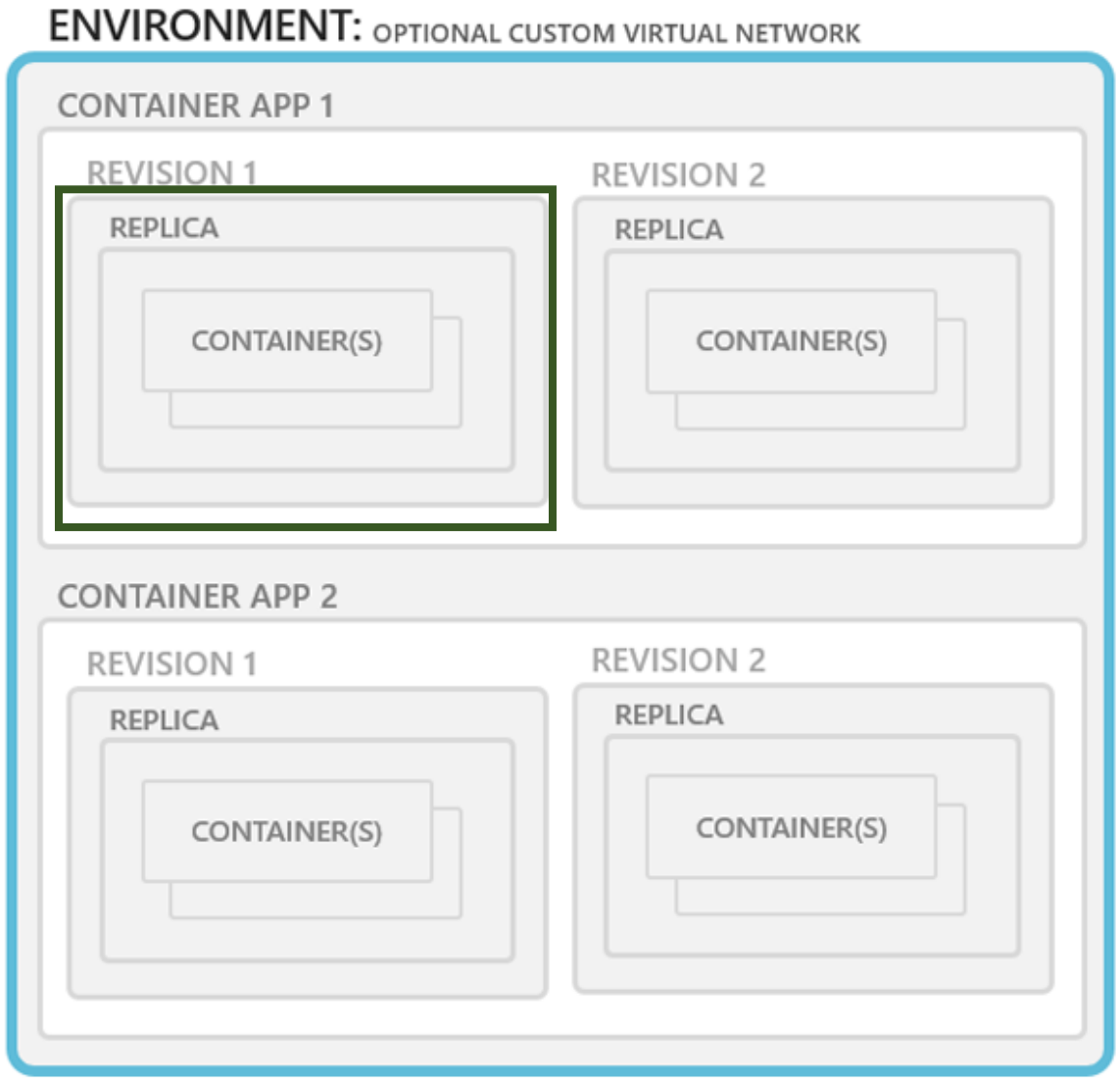
Architecture of Container Apps



Container App

- Where the container runs
- Uses the environment's resources
  - Networking, logging, security etc.
- The main entry point in the portal
- Has its own revisions, replicas etc.

Architecture of Container Apps



Revision

- Manages versions of the container app
- A container app runs a revision
- Each revision is a snapshot of the version that was deployed
- Once deployed – cannot be changed
- Every deployment creates a new revision
- Up to 100 revisions per container app

Revision Characteristics

Immutable	Once deployed – cannot be changed
Automatically provisioned	A new revision is automatically created when a new container app is deployed
Historical record	Up to 100 revisions are retained and can be used as historical records for past versions
Multiple revisions	Multiple revisions can be active, allowing traffic splitting between revisions
Zero downtime deployment	When deploying a new revision the app will not experience any downtime

Traffic Splitting

- When multiple revisions are active traffic can be split between them
- Very useful for various deployment scenarios
  - ie. Rolling deployment

Rolling Deployment

- Instances are updated gradually in batches
- Only if no errors are found the deployment resumes

v1

v2

Rolling Deployment

- Instances are updated gradually in batches
- Only if no errors are found the deployment resumes

v1

v2

Rolling Deployment

- Instances are updated gradually in batches
- Only if no errors are found the deployment resumes

v1

v2

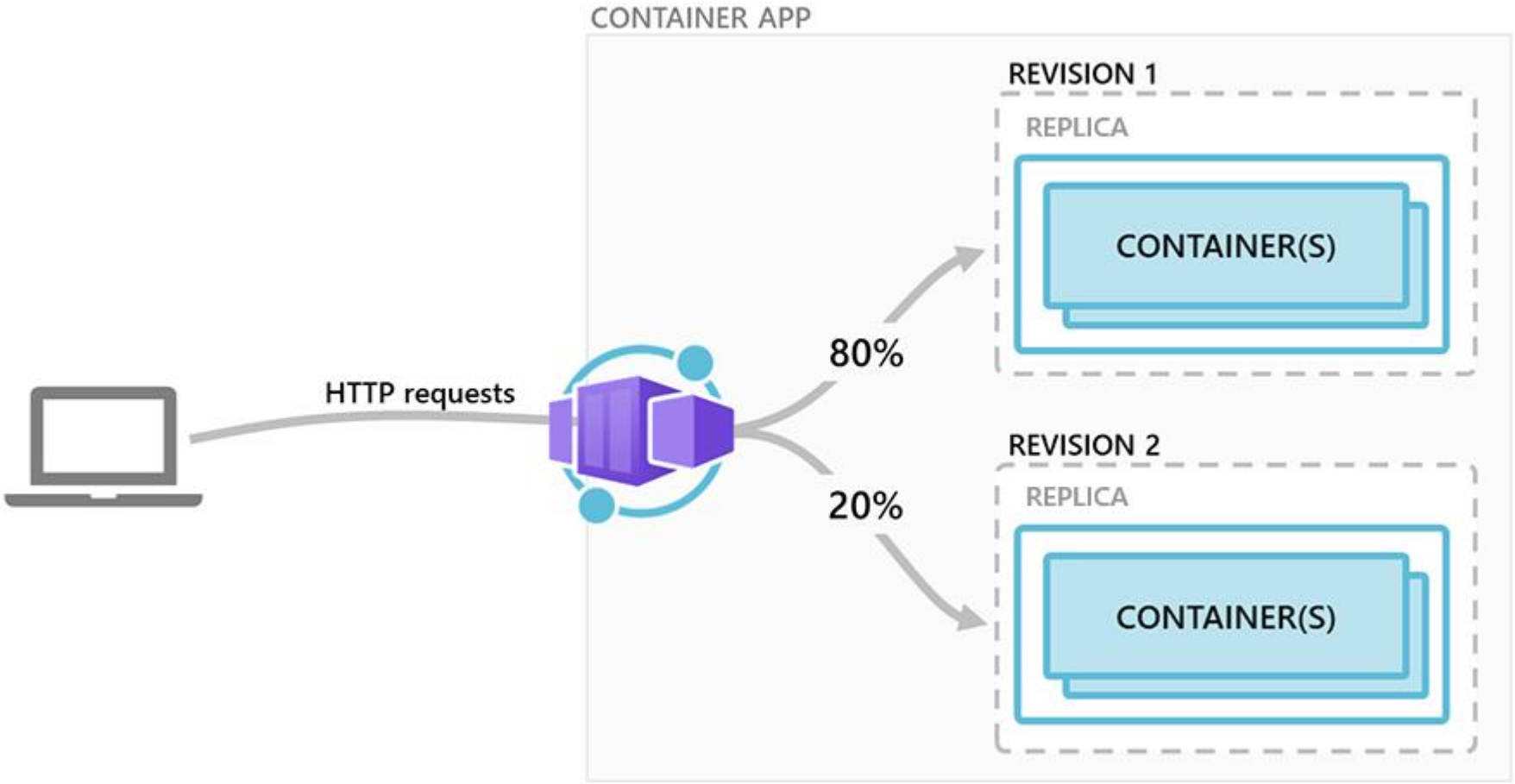
Rolling Deployment

- Instances are updated gradually in batches
- Only if no errors are found the deployment resumes

v1

v2

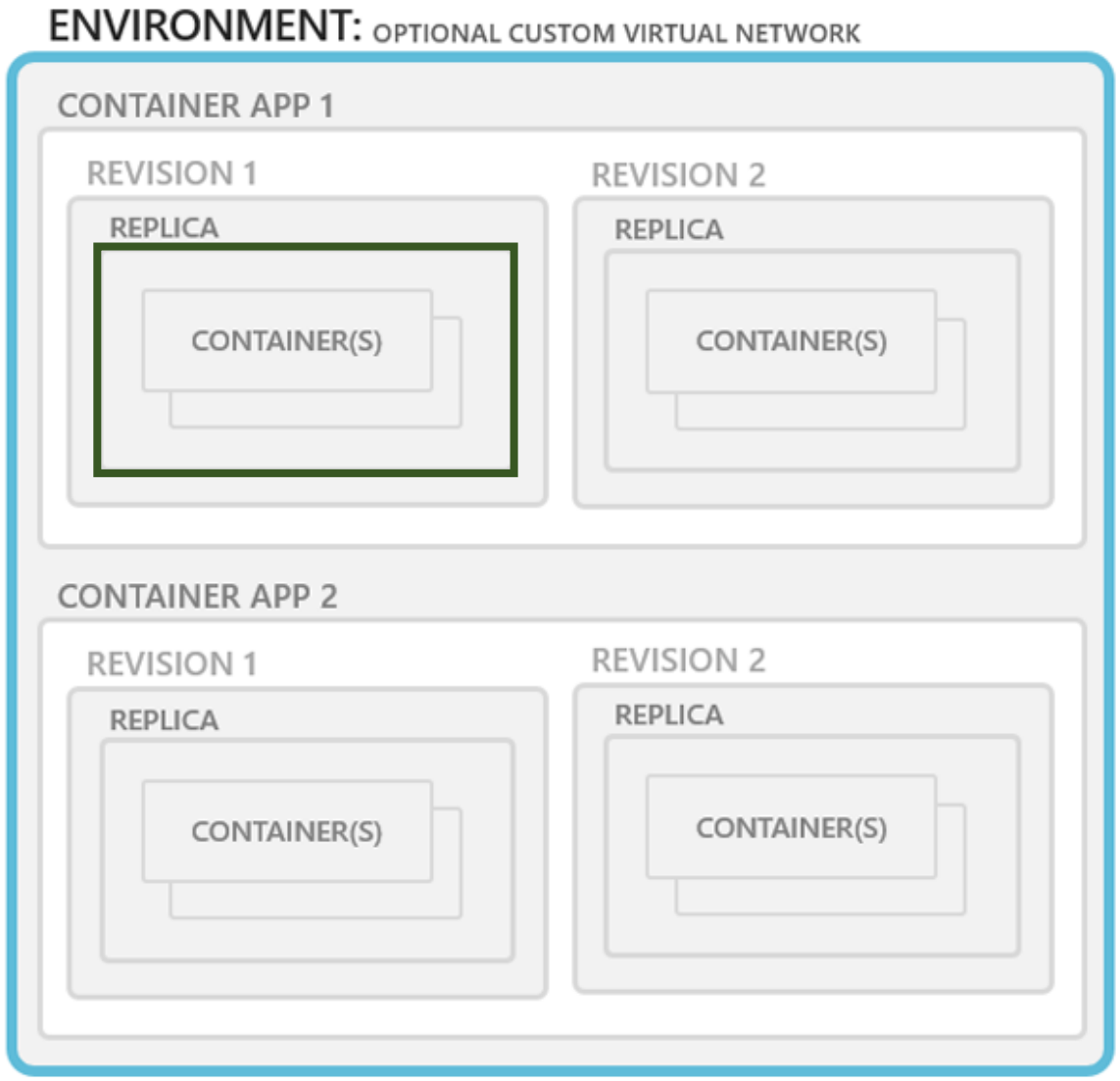
Traffic Splitting



Direct Revision Access

- Specific revisions can be directly accessed
- Done using a specific URL

Architecture of Container Apps



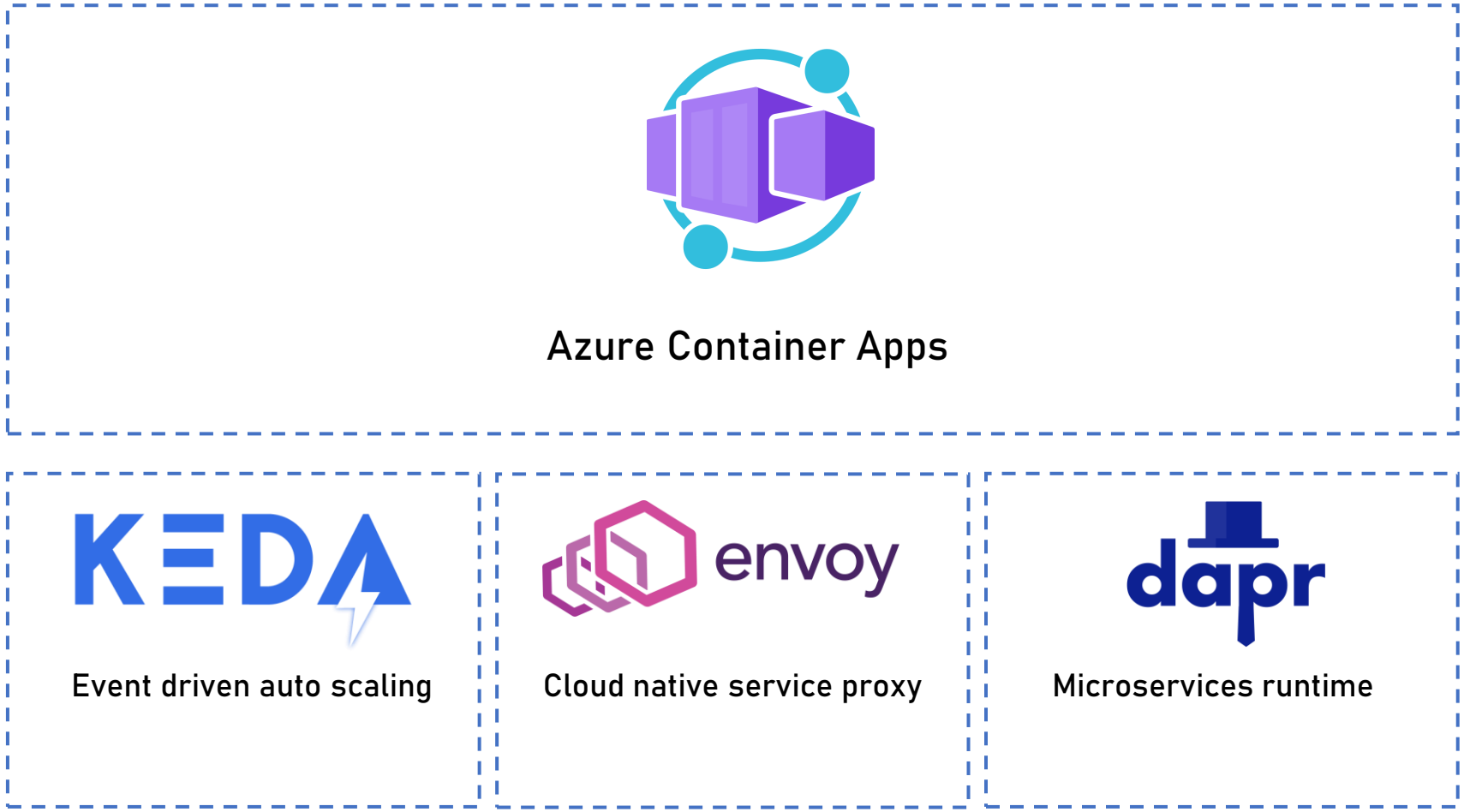
Replica

- Azure Container Apps can scale out automatically as needed
- Done by adding or removing replicas
- Each replica is an instance of a revision
- Provides the compute and memory for running the revision
- Up to 300 replicas per revision
- We'll cover scaling later in this course

Underlying Components

- Azure Container Apps uses open source components for providing multiple services
- These components are optional and don't have to be used
- Add real value to the Container App deployed

Underlying Components



Pricing

- Depends on the plan used

Consumption

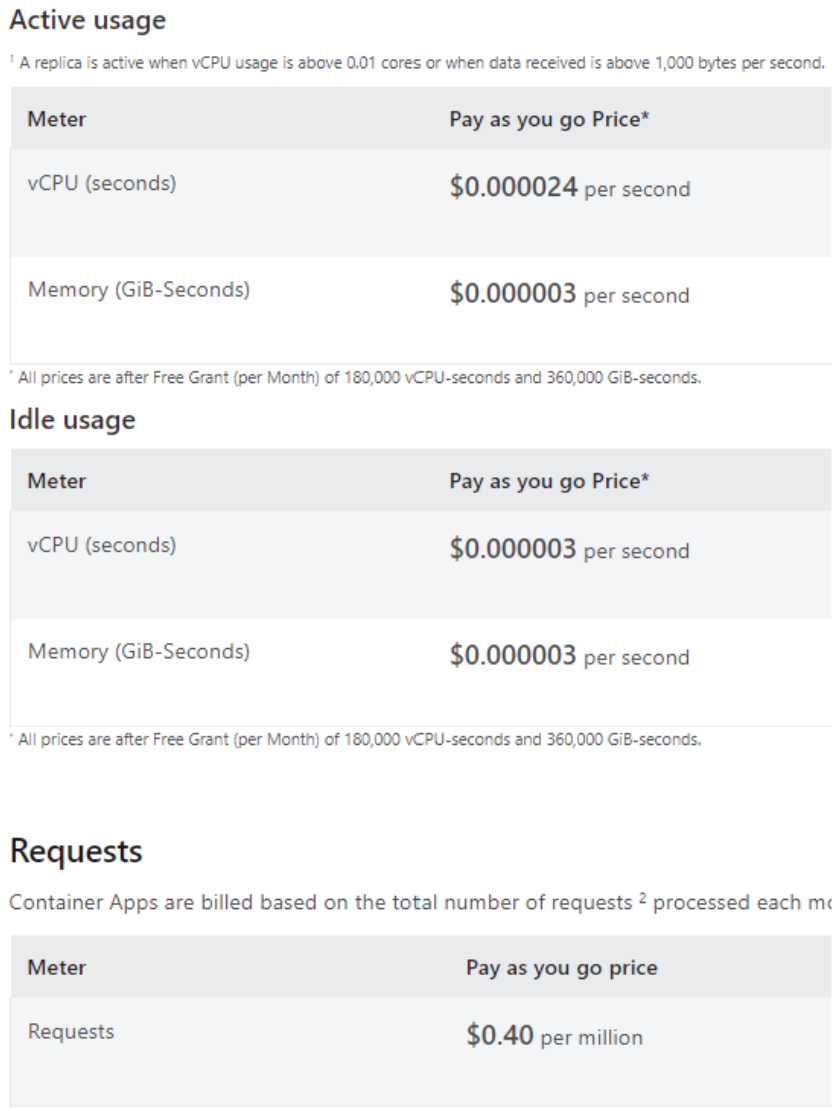
- Serverless architecture
- Scale-to-zero
- Pay for what you use
  - Based on CPU, memory and requests
- Includes free tier

Dedicated

- Use a dedicated hardware
- Can be customized to your needs
- ie. Powerful CPU, GPU
- Pay per instance

Consumption Plan Pricing

- Free tier: 180,000 vCPU-seconds, 360,000 GB-seconds, 2 million requests per month



Active usage

[1] A replica is active when vCPU usage is above 0.01 cores or when data received is above 1,000 bytes per second.

Meter	Pay as you go Price*
vCPU (seconds)	$0.000024 per second
Memory (GiB-Seconds)	$0.000003 per second

* All prices are after Free Grant (per Month) of 180,000 vCPU-seconds and 360,000 GiB-seconds.

Idle usage

Meter	Pay as you go Price*
vCPU (seconds)	$0.000003 per second
Memory (GiB-Seconds)	$0.000003 per second

* All prices are after Free Grant (per Month) of 180,000 vCPU-seconds and 360,000 GiB-seconds.

Requests

Container Apps are billed based on the total number of requests [2] processed each mo

Meter	Pay as you go price
Requests	$0.40 per million

Dedicated Plan Pricing