Escenarios avanzados en AKS

Eduard Tomàs - @eiximenis

¿Quien soy yo?

Eduard Tomàs
Principal Tech Lead en Plain Concepts BCN
etomas@plainconcepts.com

Beer crafter & drinker
@eiximenis

¡Gracias!

Patrocinadores Locales

avanade

Bravent
IT consulting company

capside
architects of the digital society

clevertask

encamina
PIENSA EN COLORES

enimbos
the cloud enabler company

ERNI

everis
an NTT DATA Company

ilitia

Insight

intelequia
SOFTWARE SOLUTIONS

isolutions

KABEL

myCloudDoor

plain concepts

raona

東京 TOKIOTA

Colabora

Microsoft

Agenda

Https en 5 minutos ☺

Escalabilidad en Kubernetes

Nodos virtuales

Dev spaces

Https en 5 minutos

Https en Kubernetes

Si se usa ingress para tener HTTPS, solo se debe:

Crear el secreto en Kuebernetes con el certificado TLS

Añadir la sección TLS en los recursos ingress

Cert manager

Cert-manager es un “plugin” para Kubernetes que gestiona la creación (y renovación) de certificados TLS

Mediante CDOs se definen el emisor (p. ej. Let’s Encrypt) y la petición de certificado

Cert-manager usa ACME para obtener un certificado TLS a partir de la petición usando el emisor definido

Nosotros solo debemos modificar los ingress

Escalabilidad en Kubernetes

Horizontal pod scaling

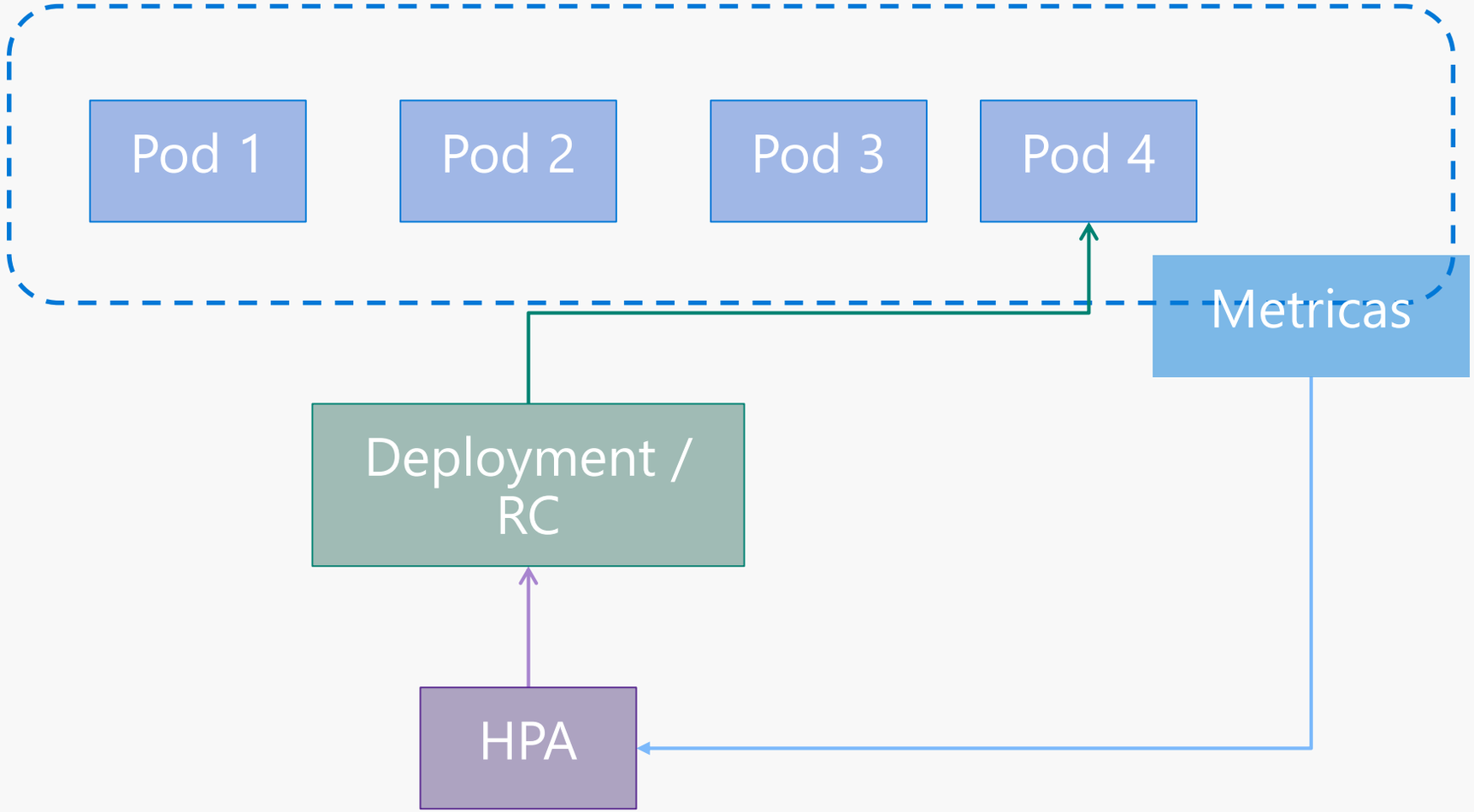
Si un pod alcanza determinados límites (CPU, memoria, …) otro pod es creado para repartir la carga

Funciona perfecto para pods *stateless*

*(Todas los vuestros lo son… ¿**verdad**?)*

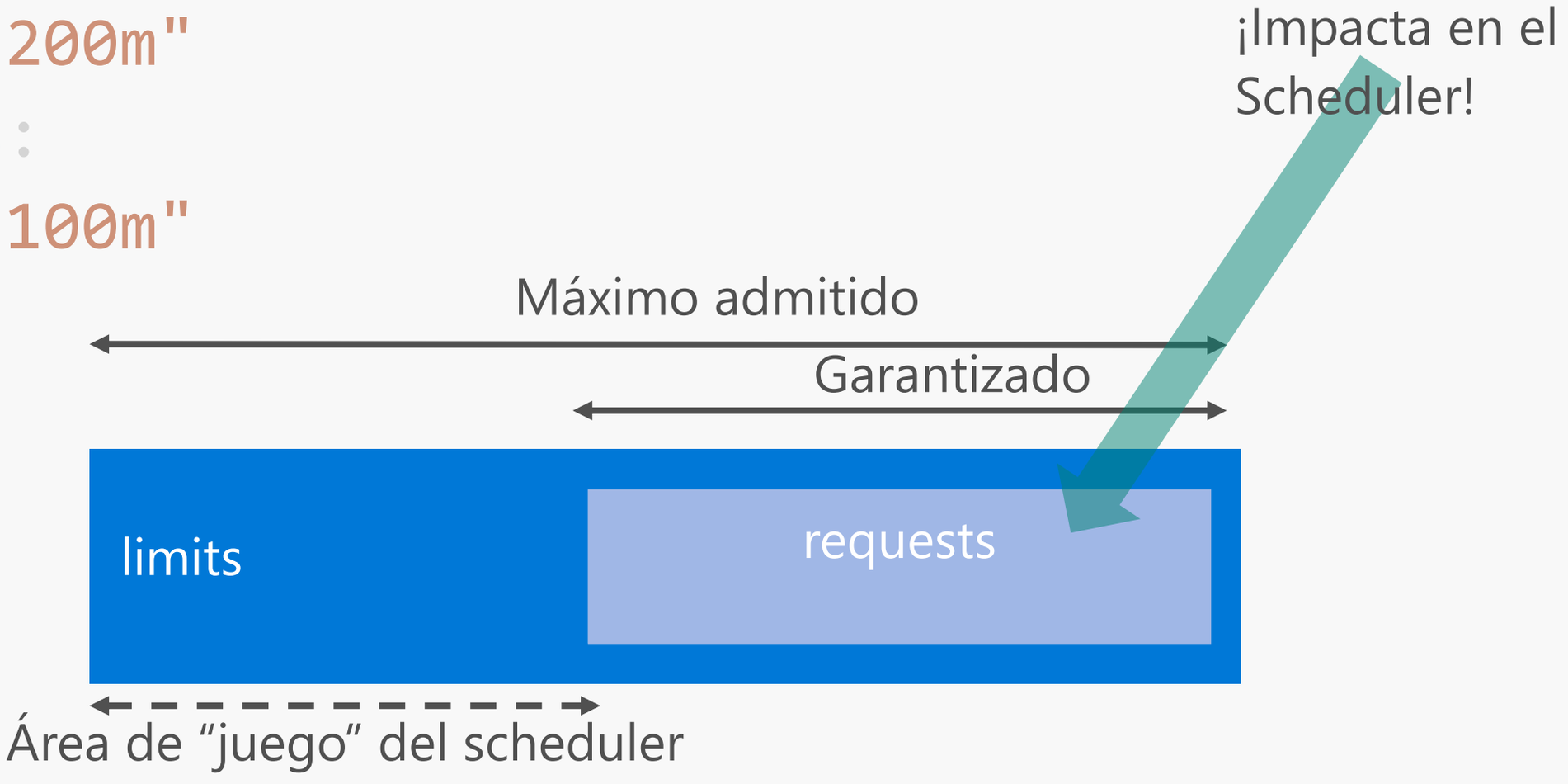
Se puede automatizar mediante el HPA

Horizontal pod scaling



Límites en los pods

```
resources:
  limits:
    cpu: "200m"
  requests:
    cpu: "100m"
```



Vertical pod scaling

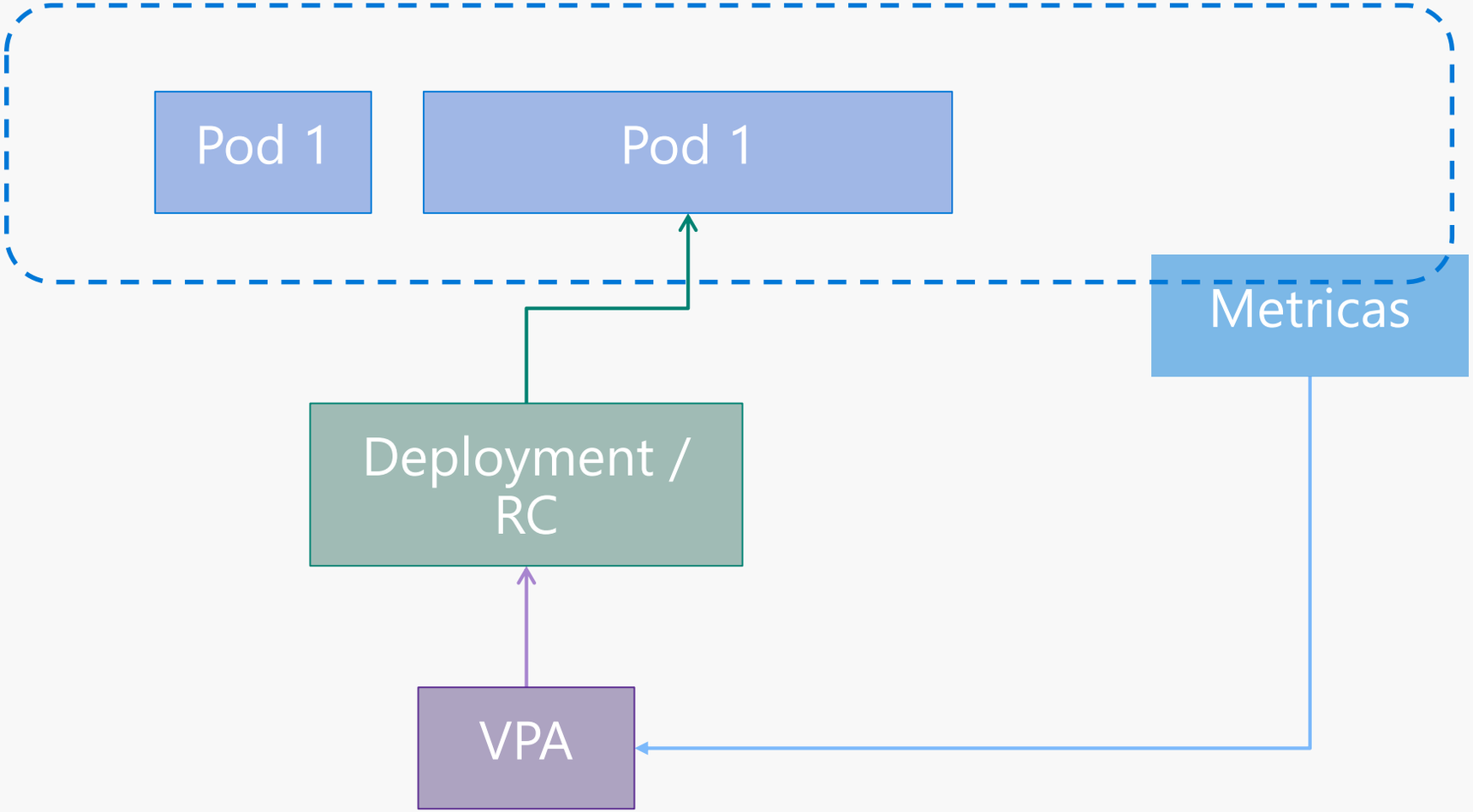
Si un pod alcanza determinados límites (CPU, memoria) es destruido y recreado con mayores límites disponibles.

Útil para pods *stateful* (como BBDDs)

Pero se puede usar también para *stateless*

Se puede automatizar mediante el VPA

Vertical pod scaling



Node scaling

Un nodo minion (worker) se puede agregar al clúster para añadir recursos computacionales

AKS lo permite (tanto añadir como eliminar nodos) sin interrupción de servicio

Se puede automatizar mediante el node autoscaler (en preview)

Nodos virtuales

ACI – Azure container instances

ACI es la **forma más fácil y rápida** de tener un contenedor corriendo en Azure

Un modelo *Serverless* aplicado a Azure

Ideal para escenarios Dev/Test

Aunque no para escenarios productivos (poca escalabilidad)

Virtual Nodes

AKS puede usar ACI para ejecutar **algunos** de sus pods

Para cada pod un ACI es creado

Este ACI existe solo mientras el pod está ejecutándose

Virtual nodos + HPA es ideal para workloads con escalabilidad altamente variable, a la que debe responderse en segundos

El secreto tras ACI se llama **virtual kubelet**

Asignar pods a nodos

Virtual nodes funciona agregando un nodo “fantasma” al clúster

Cada pod que termine ejecutándose en este nodo, se ejecutará en ACI

Este “nodo virtual” lleva el *taint*:

```
virtual-kubelet.io/provider=azure:NoSchedule
```

Taints y tolerations

Un *taint* es una etiqueta que se cuelga a un nodo. Tiene una clave, un valor y un efecto.

`virtual-kubelet.io/provider=azure:NoSchedule`

El efecto "NoSchedule" impide que cualquier pod se ejecute en el nodo a menos que tenga una *toleration* que lo permita

De este modo, evitamos que los pods se ejecuten en ACI: sólo los que tengan la toleration se podrán ejecutar en ACI

Taints y tolerations

Una *toleration* es la capacidad de un pod para soportar un determinado *taint*:

```
tolerations:
- key: virtual-kubelet.io/provider
  operator: Exists
```

Los pods soportan ser ejecutados en nodos que tengan el *taint* virtual-kubelet.io/provider

Node Selector

El escenario anterior permite a ciertos pods ejecutarse en el nodo virtual y evita que el resto de pod no lo hagan.

Pero no evita que los pods que se pueden ejecutar en el nodo virtual, puedan ejecutarse también en los nodos normales

NodeSelector es un mecanismo que permite a un pod ejecutarse en aquellos nodos que tengan, al menos, las etiquetas especificadas en el nodeSelector

Node Selector

El escenario anterior permite a ciertos pods ejecutarse en el nodo virtual y evita que el resto de pod no lo hagan.

Pero no evita que los pods que se pueden ejecutar en el nodo virtual, puedan ejecutarse también en los nodos normales

Node Selector es un mecanismo que permite a un pod ejecutarse en aquellos nodos que tengan, al menos, las etiquetas especificadas en el Node Selector

Node Selector

```
nodeSelector:
  kubernetes.io/role: agent
  beta.kubernetes.io/os: linux
  type: virtual-kubelet
```

```
λ  kubectl describe node virtual-node-aci-linux
Name:               virtual-node-aci-linux
Roles:              agent
Labels:             alpha.service-controller.kubernetes.io/exclude-balancer=true
                    beta.kubernetes.io/os=linux
                    kubernetes.io/hostname=virtual-node-aci-linux
                    kubernetes.io/role=agent
                    type=virtual-kubelet
```

Node Selector

Virtual nodes usa:

Taints: Para evitar que cualquier pod se ejecute en el nodo virtual

Tolerations: En los pods que pueden ejecutarse en el nodo virtual

Node Selector: Para forzar los pods a ejecutarse en el nodo virtual

Dev Spaces

Retos de “microservicios” – preguntas típicas

Debo ejecutar TODOS los microservicios en local?

Como puedo tener un ciclo iterative de desarrollo, prueba, despliegue de forma fácil (usando VS, CLI, …)

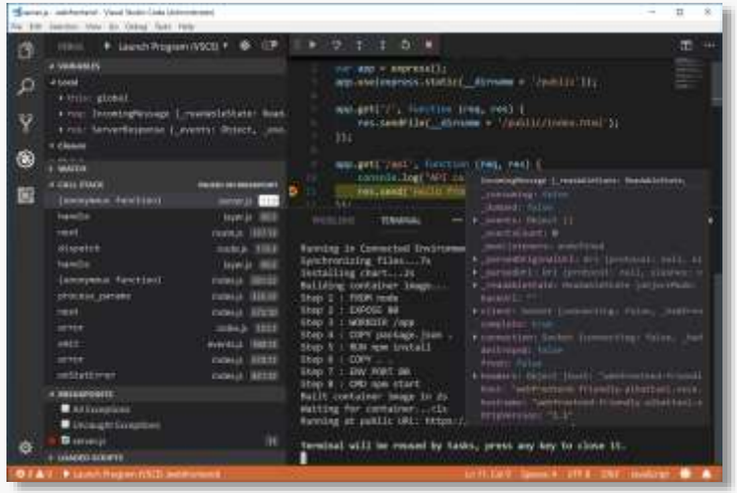
Como puedo ejecutar tests e2e con mi servicio actualizado en un orquestador compartido sin impactor al resto?

Azure Dev Spaces:

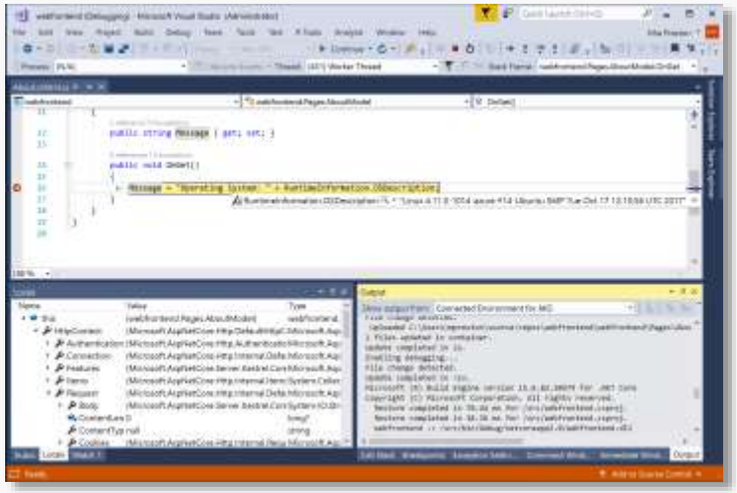
Herramienta **de desarrollo** para desarrollar iterativamente contra un AKS

Poco setup en local

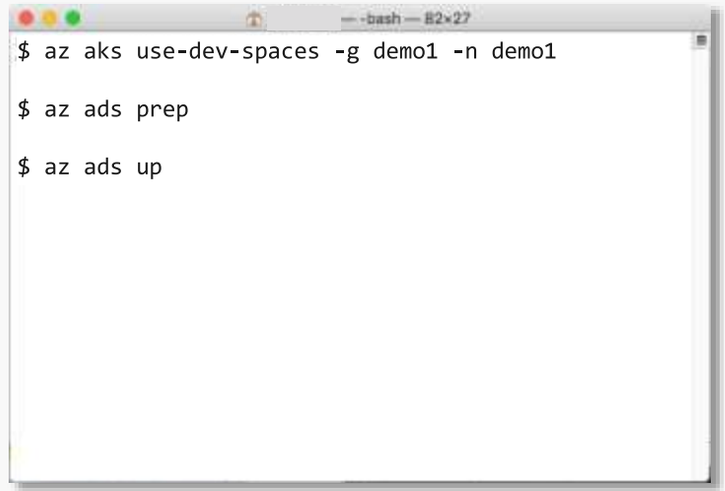
Usar los mismos assets en dev que en prod (helm charts)



Visual Studio Code



Visual Studio 2017+



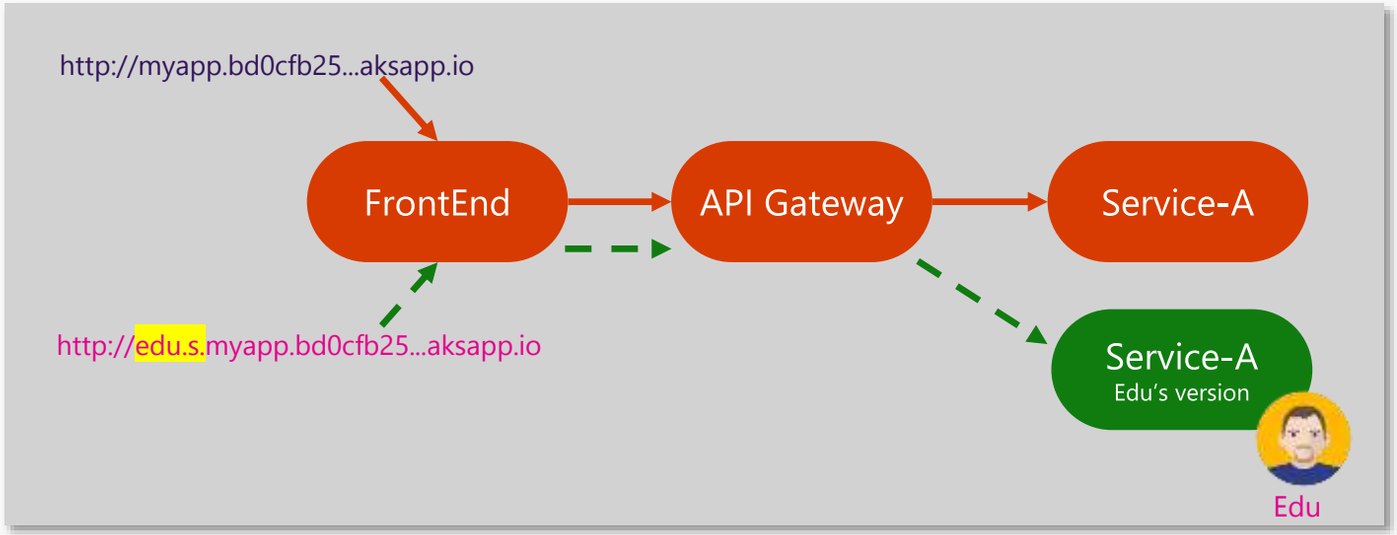
Command line

Azure Dev Spaces

Distintos miembros del equipo pueden trabajar **simultaneamente** en distintas partes del sistema compartiendo orquestador

Cada desarrollador obtiene "un espacio" de desarrollo independiente

Permite probar servicios **sin tener que desplegar** todas las dependencias



2019

Global Azure BOOTCAMP


2019
Global Azure
BOOTCAMP