

# About me



Current – Cloud architect at Eleks

Previous – Cloud Solution Architect at QiiO

E-doc at APU

MCS at MS Ukraine

MVP (5 times), Lead of SharePoint and IoT  
communities

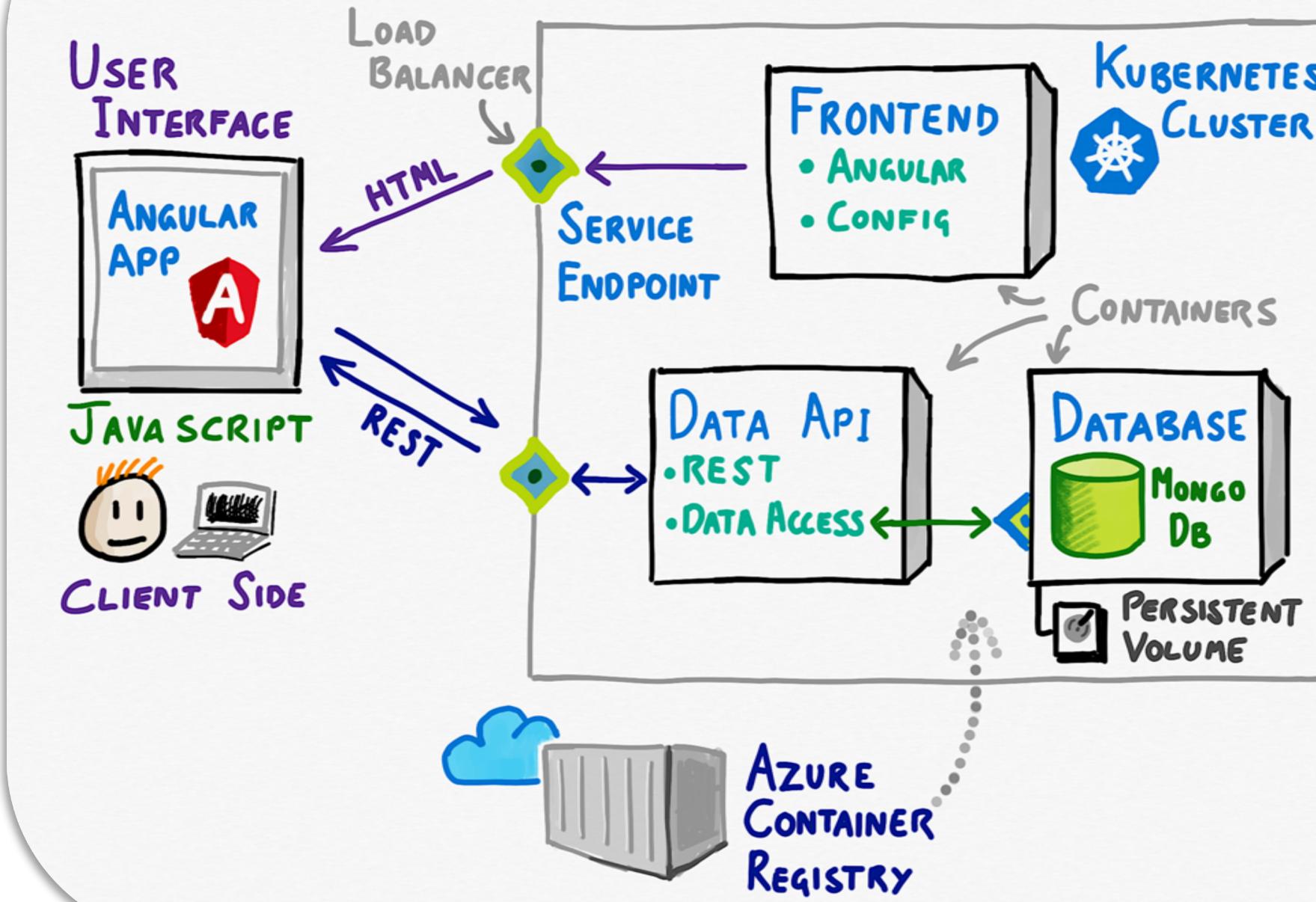
Facebook <https://www.facebook.com/sergey.belskiy>

Twitter [@sergey\\_belskiy](https://twitter.com/@sergey_belskiy)

Blog <https://medium.com/@sergiibelskyi>

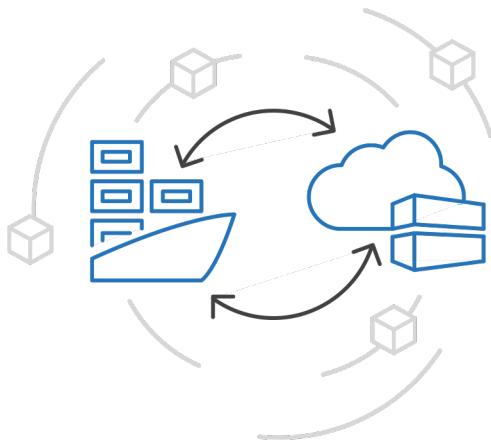
IoT community <https://www.facebook.com/groups/iot.ua/>

# KUBERNETES ARCHITECTURE

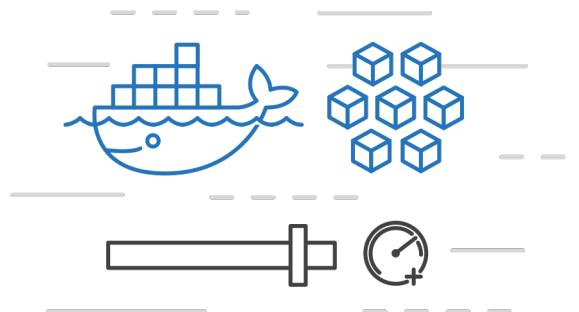


# Azure Container Instances (ACI)

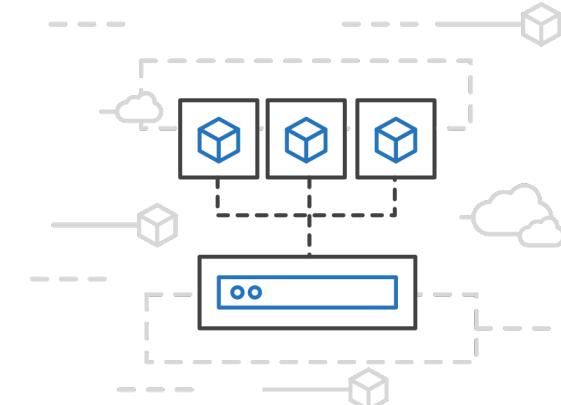
## Containers As A Service



Start using  
containers right away



Cloud-scale  
container capacity



Hyper-visор  
isolation

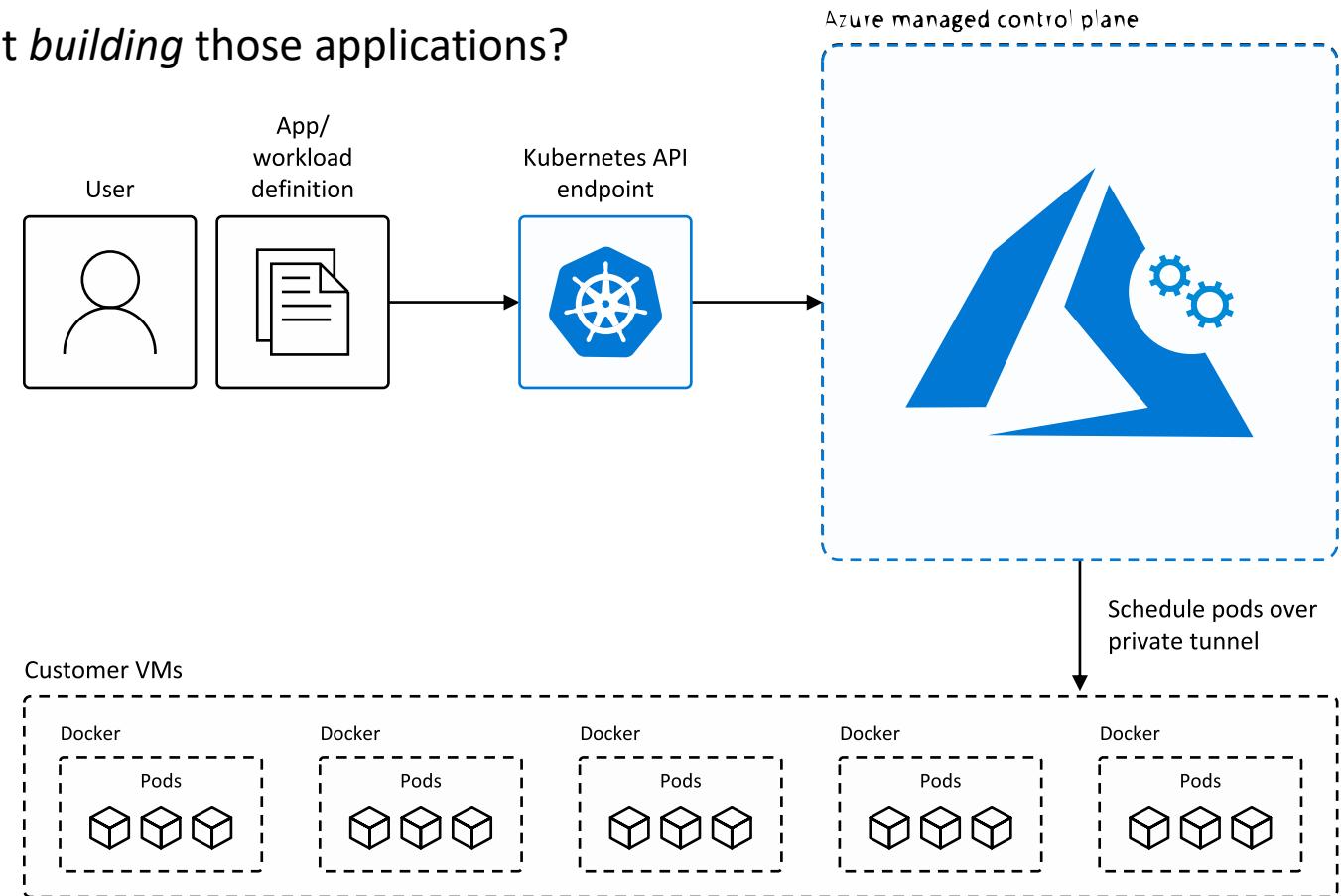
• ----- • ----- •

Per-second billing & completely serverless

# AKS means ...

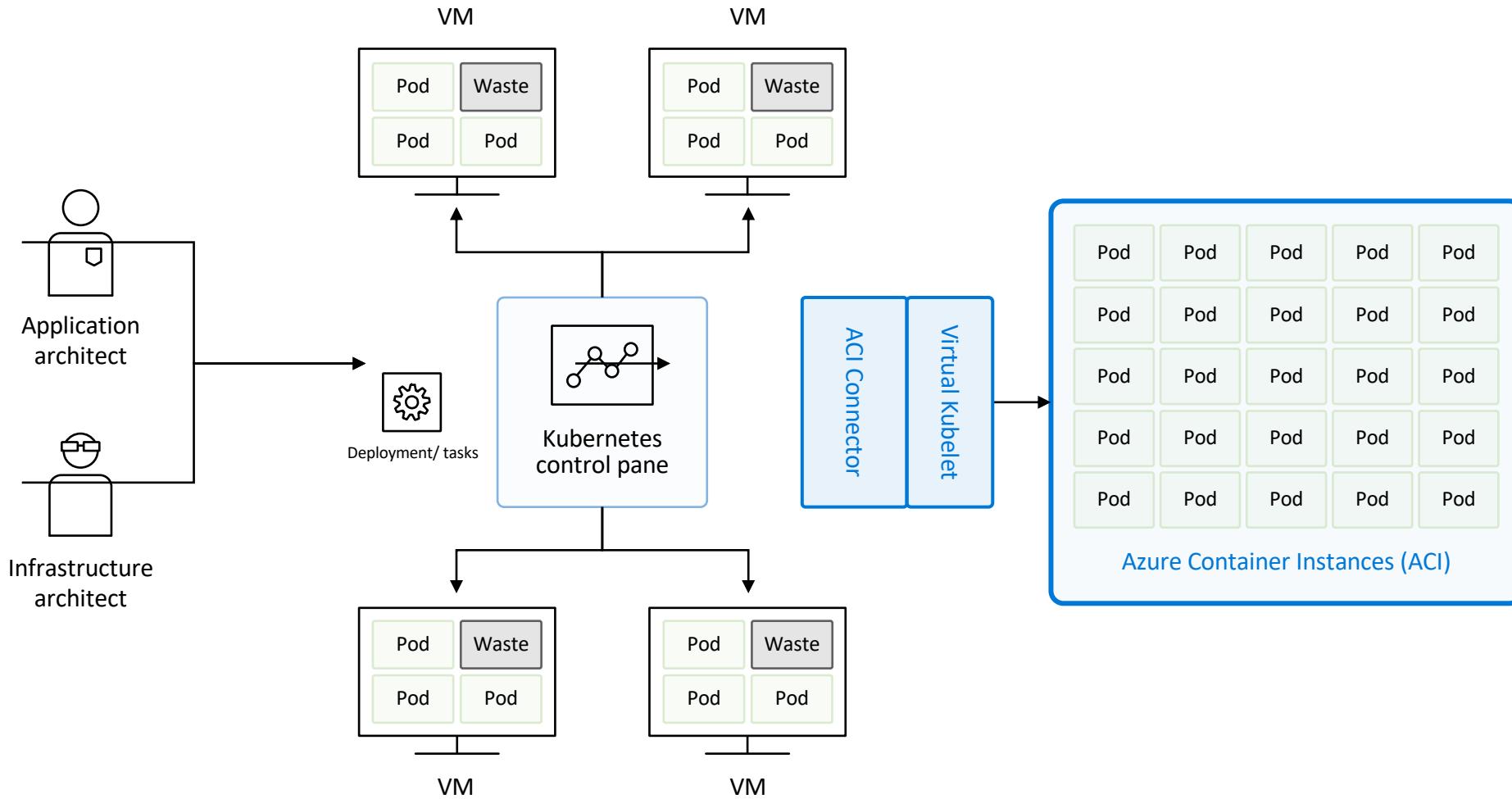
Kubernetes is great for deploying, running, and managing cloud-native applications...

...but what about *building* those applications?

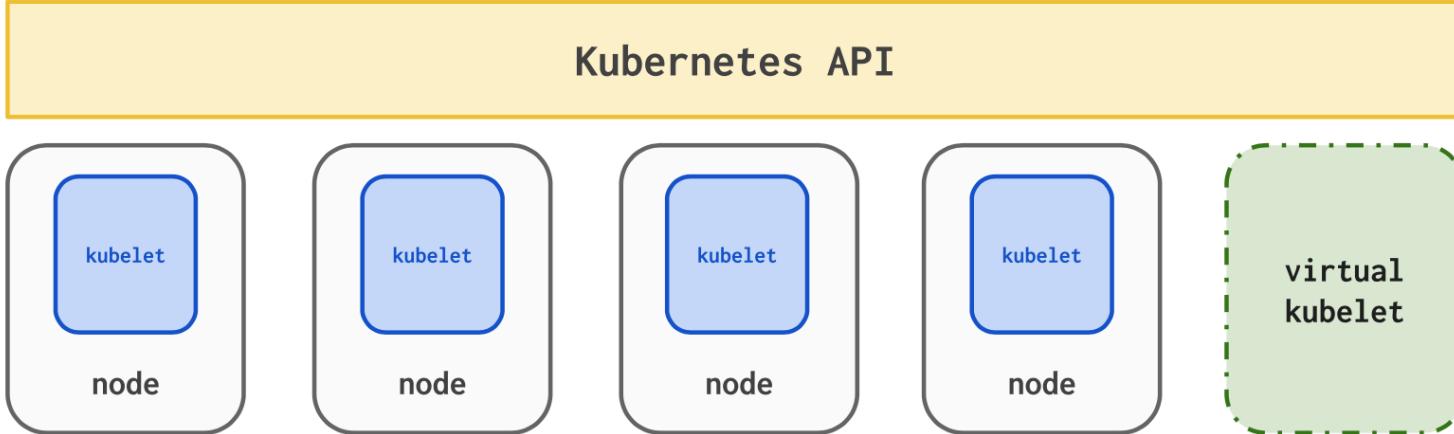


# Azure Container Instances (ACI)

## Bursting with the ACI Connector

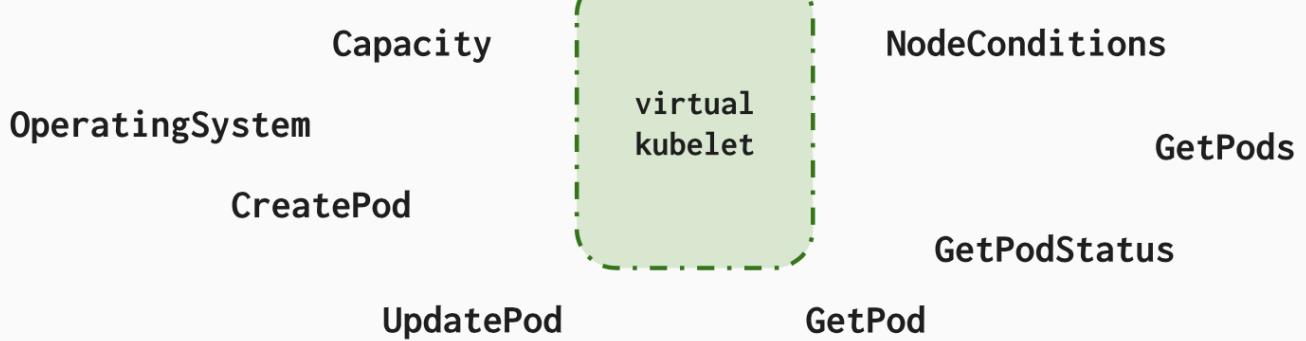


# Virtual kubelet

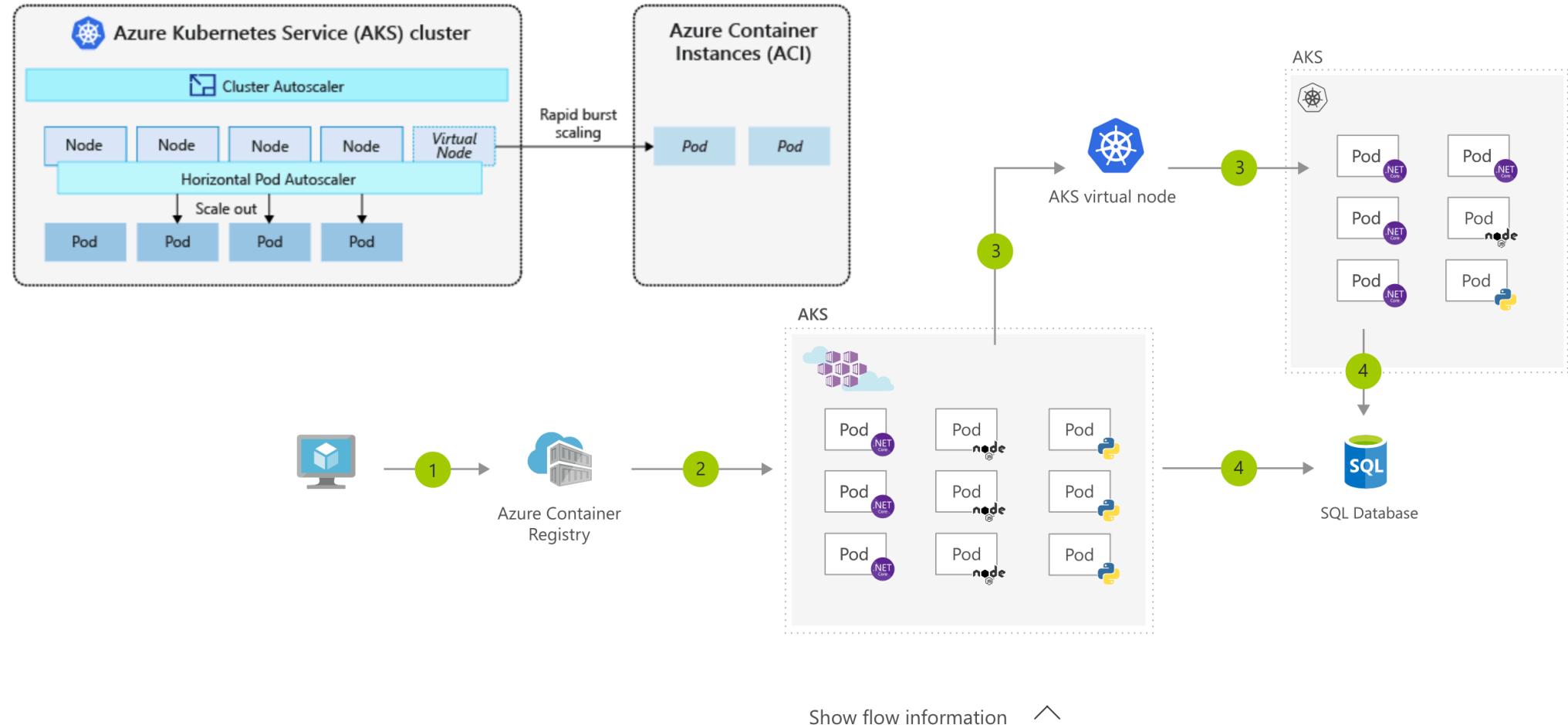


Typical kubelets implement the pod and container operations for each node as usual.

Virtual kubelet registers itself as a “node” and allows developers to deploy pods and containers with their own APIs.



# ACI in AKS



1 User registers container in Azure Container Registry

2 Container images are pulled from the Azure Container Registry

3 AKS virtual node, a Virtual Kubelet implementation, provisions pods inside ACI from AKS when traffic comes in spikes.

4 AKS and ACI containers write to shared data store

# Enable it

## Create Kubernetes cluster

Basics   Scale   Authentication   Networking   Monitoring   Tags   Review + create

Enable scaling features to allow flexible capacity and burstable scaling options within your cluster.

- **Virtual nodes** allow burstable scaling backed by serverless Azure Container Instances. [Learn more about virtual nodes](#)
- **VM scale sets** are required for a variety of scenarios including autoscaling and multiple node pools [Learn more about VM scale sets in AKS](#)

Virtual nodes ⓘ

Disabled   Enabled

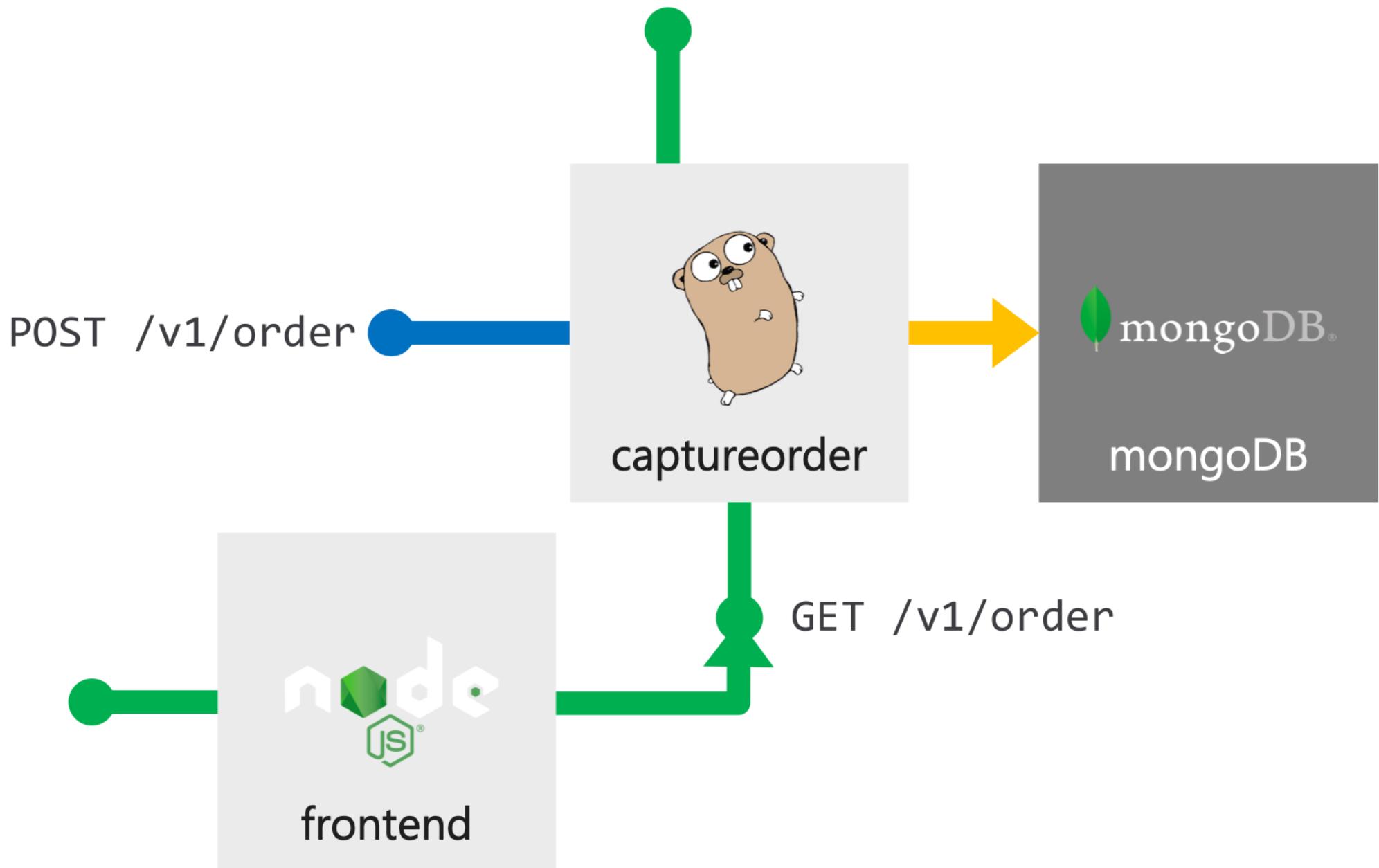
VM scale sets ⓘ

Disabled   Enabled

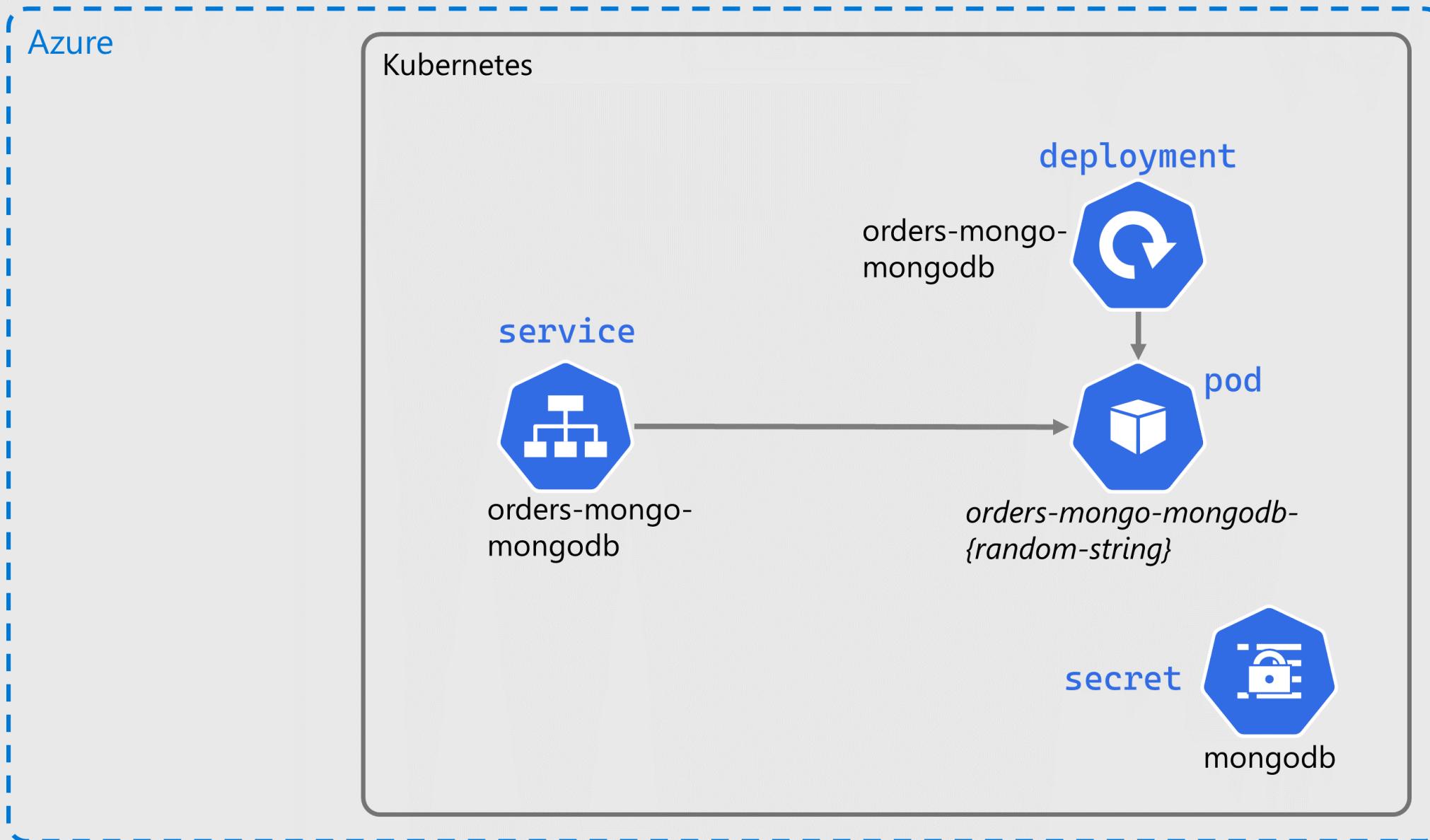


- VM scale sets are required for the following scenarios:
- \* Autoscaling
  - \* Multiple node pools

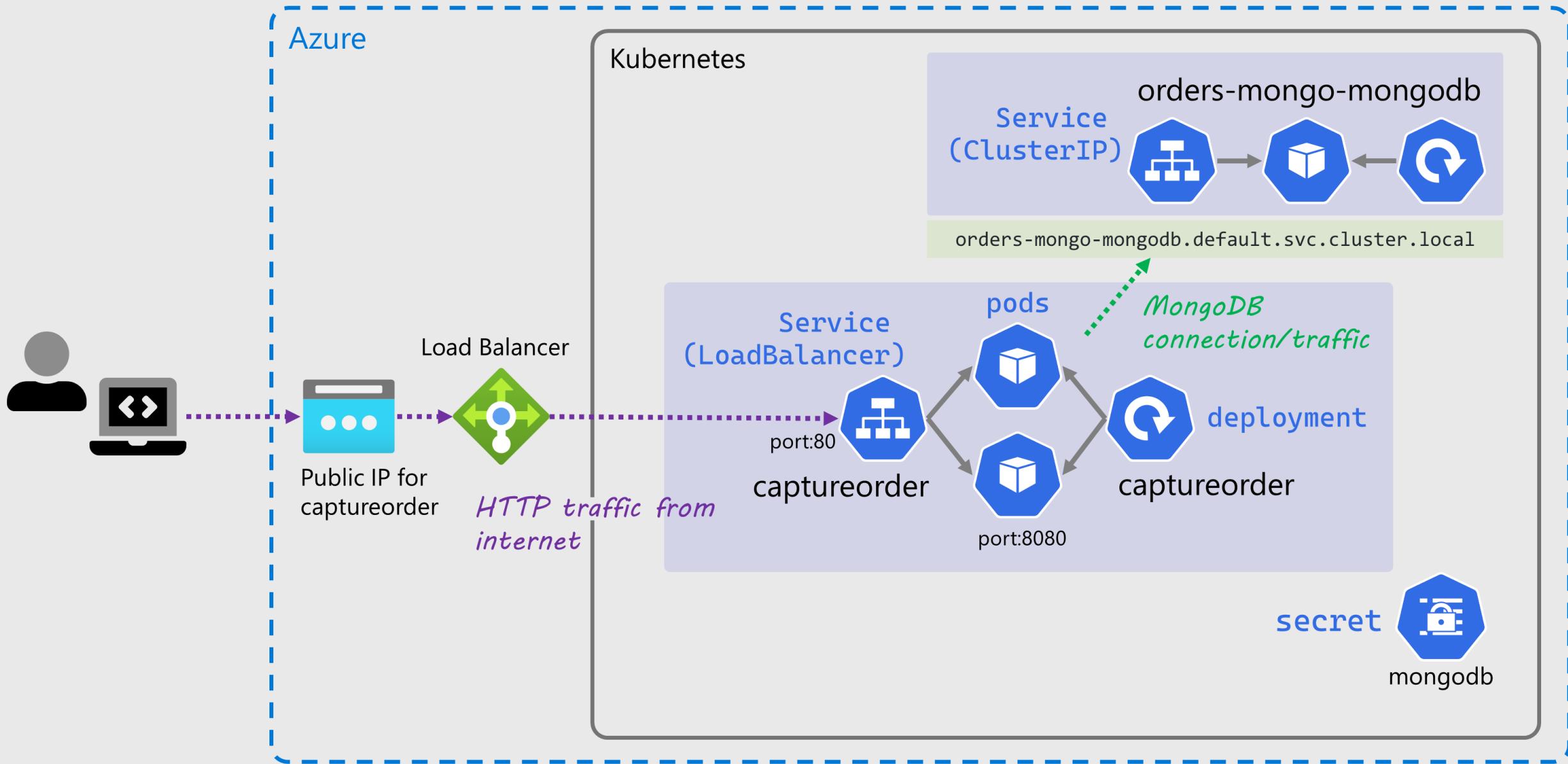
GET /healthz



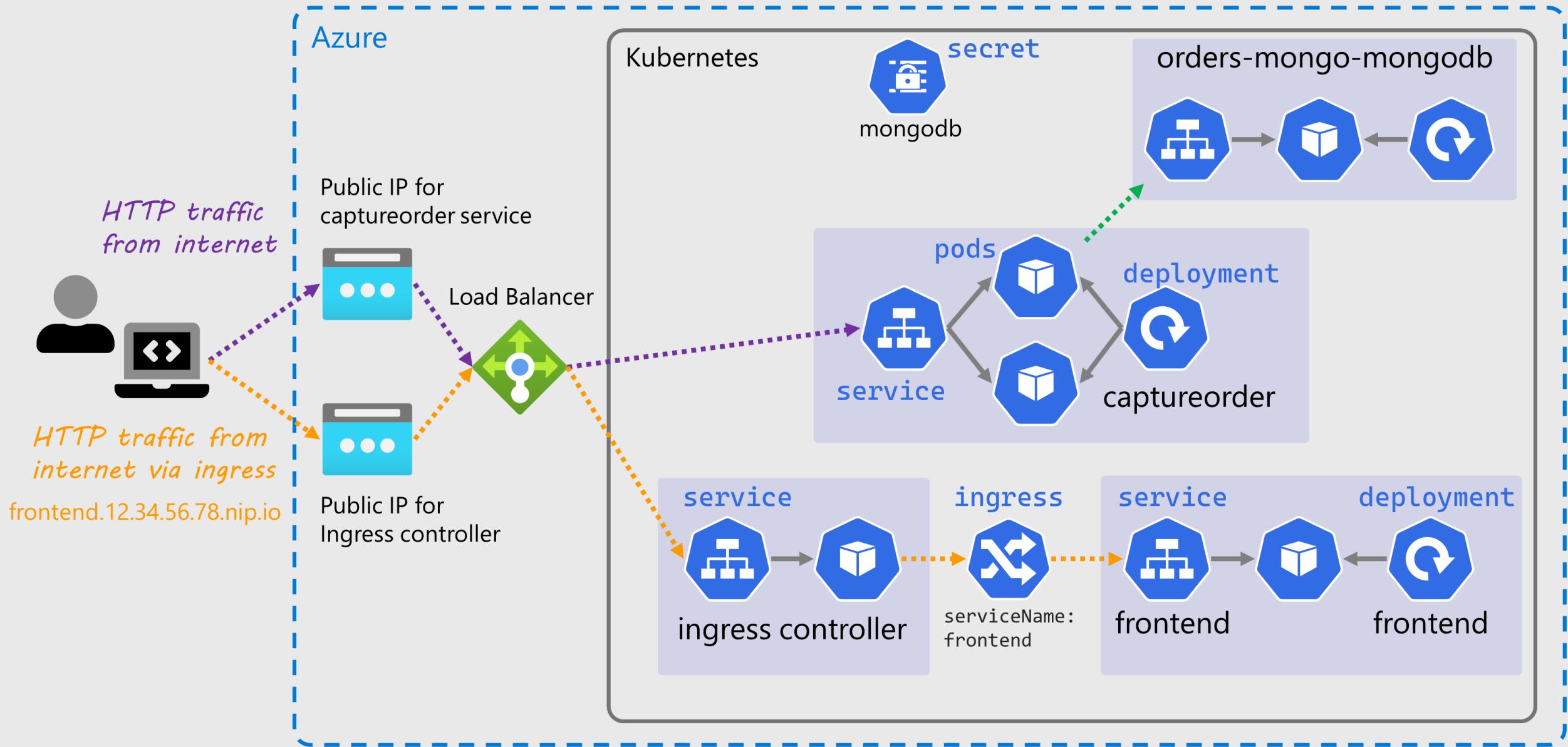
# Deploy MongoDB – Architecture



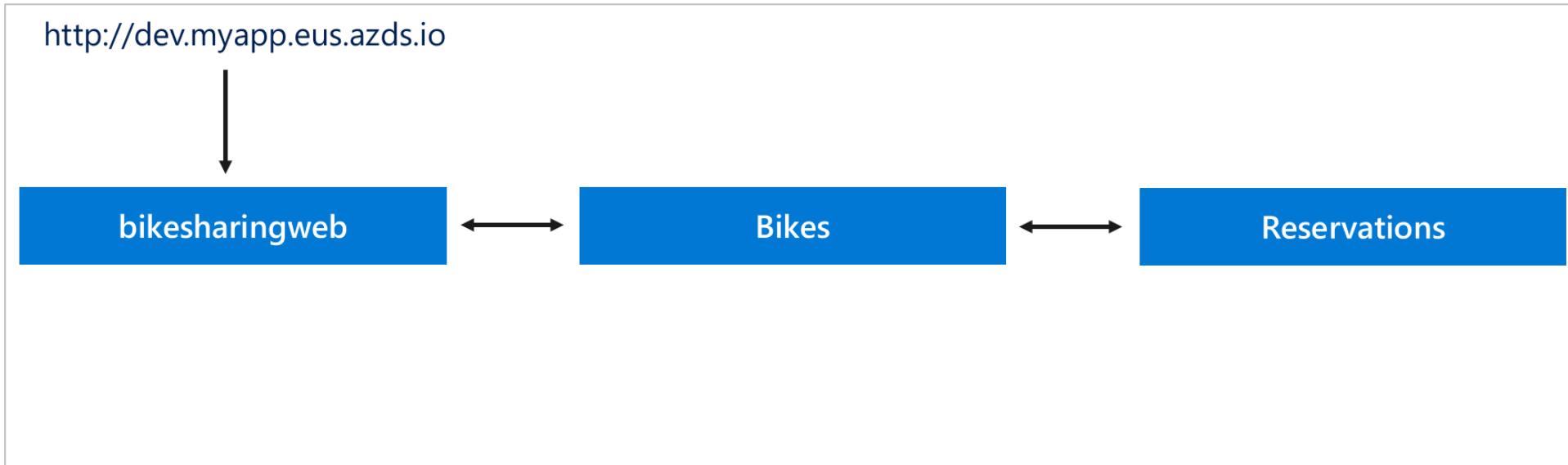
# Deploy the Order Capture API – Architecture



# Deploy the Frontend Using Ingress – Architecture



# Collaborate efficiently with Azure Dev Spaces



Share an AKS cluster for team development

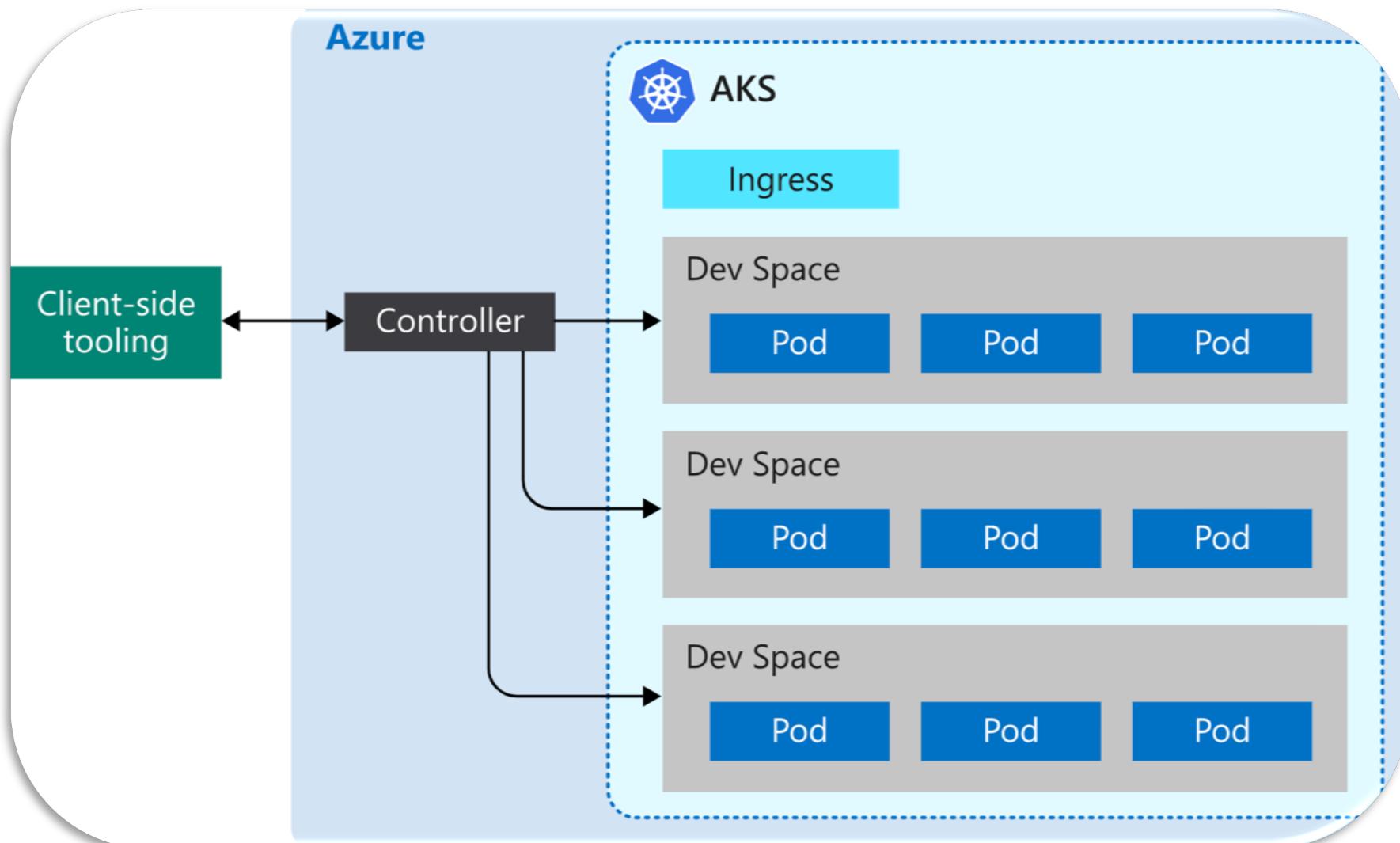
Test code end-to-end without replicating or simulating dependencies

Onboard new team members with minimal machine setup

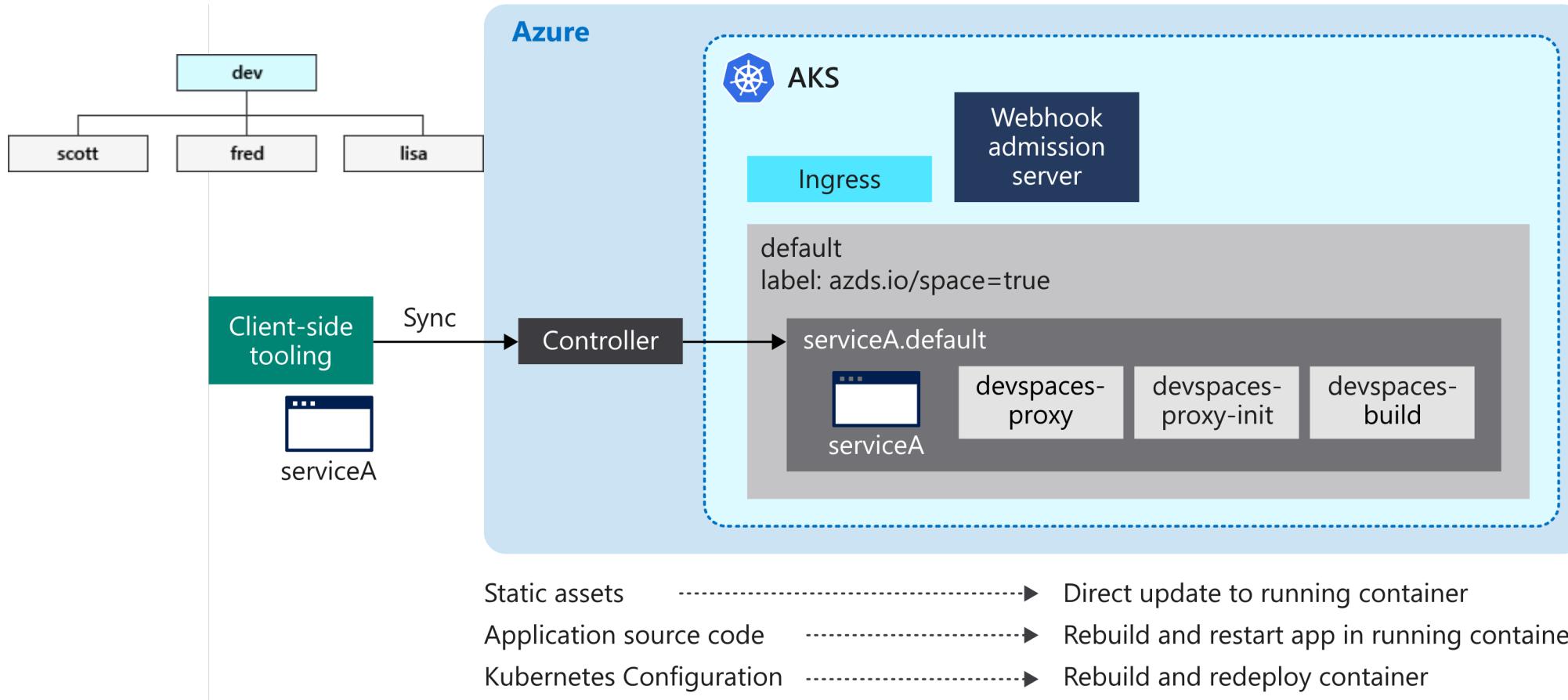
## Availability

East US, East US 2, Central US, South Central US, West US 2, North Europe, West Europe, UK South, Southeast Asia, Australia East, Canada Central, and Canada East regions

# How Azure Dev Spaces works



# Collaborate within a development team



## Resources

Get started with  
Azure Dev Spaces

<http://aka.ms/get-azds>

Get started with  
Kubernetes

<https://aksworkshop.io/#api>

The screenshot shows the Microsoft Azure Dev Spaces documentation page. The URL is <https://docs.microsoft.com/en-us/azure/dev-spaces/azure-dev-spaces>. The page has a sidebar with 'Overview' selected under 'Azure Dev Spaces'. It lists several sections: Quickstarts, Tutorials, How to, and Troubleshooting. To the right, there's a list of benefits:

- Generate Docker and Kubernetes configuration-as-code assets for you to use from development through to production.
- Share a managed Kubernetes cluster with your team and collaboratively work together. Develop your code in isolation, and do end-to-end testing with other components without replicating or mocking up dependencies.

## Get Started on Azure Dev Spaces

Please select a language-specific guide to get started:

- [Java \(VS Code & CLI\)](#)
- [.NET Core \(VS Code & CLI\)](#)
- [.NET Core \(Visual Studio 2017\)](#)
- [Node.js \(VS Code & CLI\)](#)

Below the documentation is a screenshot of the Visual Studio Code interface. It shows a Node.js application with code in server.js, a Chartyaml file for a Helm chart, and a terminal window displaying the output of a 'helm install' command. The terminal shows steps for installing a chart, building a container image, and running a command.