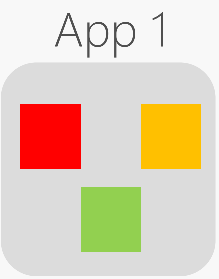
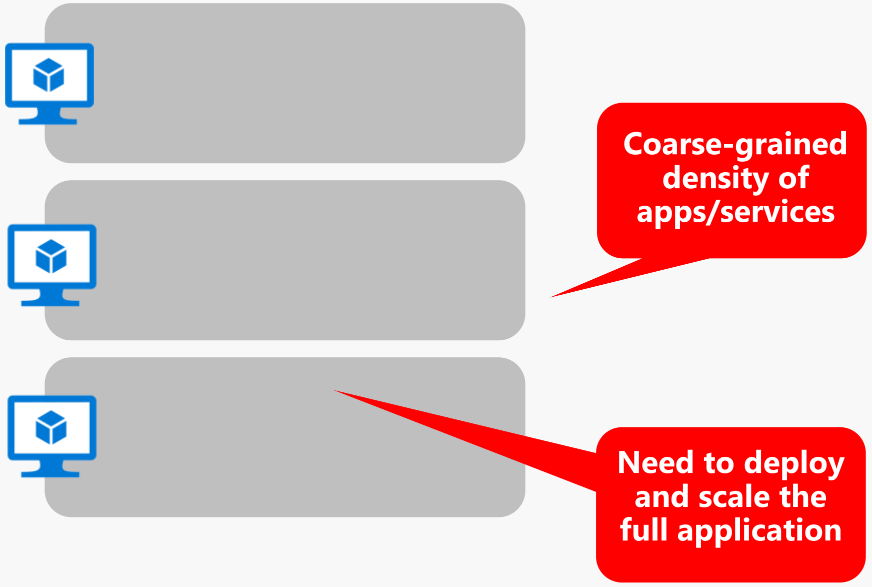
Azure Kubernetes Service

Stephan Schiller
Cloud Solution Architect
sschiller@microsoft.com

Traditional application approach

- A traditional application has most of its functionality within a few processes that are componentized with layers and libraries.
- Scales by cloning the app on multiple servers/VMs



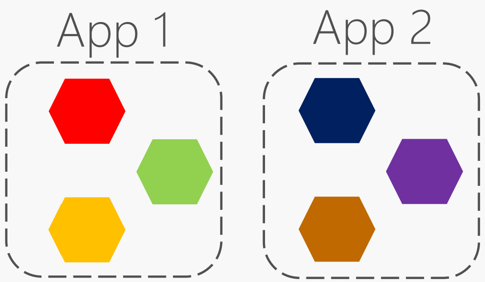


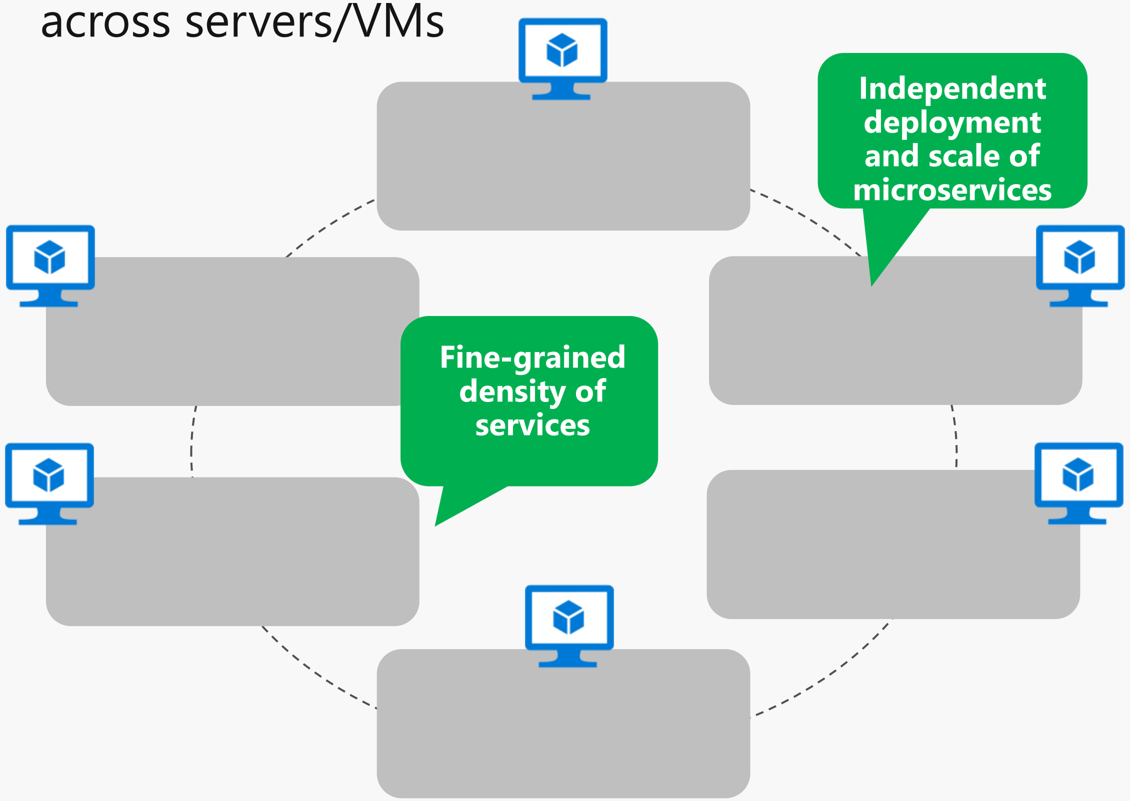
Coarse-grained density of apps/services

Need to deploy and scale the full application

Microservices application approach

- A microservice application segregates functionality into separate smaller services.
- Scales out by **deploying each service independently** with multiple instances across servers/VMs





Independent deployment and scale of microservices

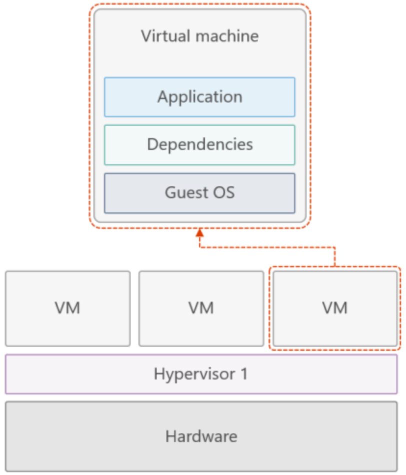
Fine-grained density of services

Customer’s pain points

- Delivering same functionality to multiple deployment environments
- Ensuring consistency and avoiding dependency hell
- Unable to migrate and scale apps while maintaining compatibility

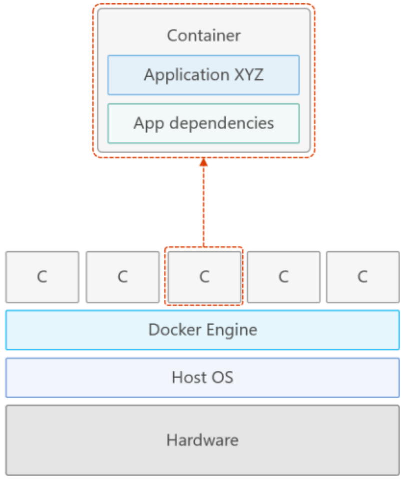
What are containers?

Virtualisation vs. Containerisation



- **Virtualisation**

- Each VM has independent, full OS
- Application startup is dependent on host OS startup (slow!)
- Full isolation
- Separate app frameworks
- Lower density

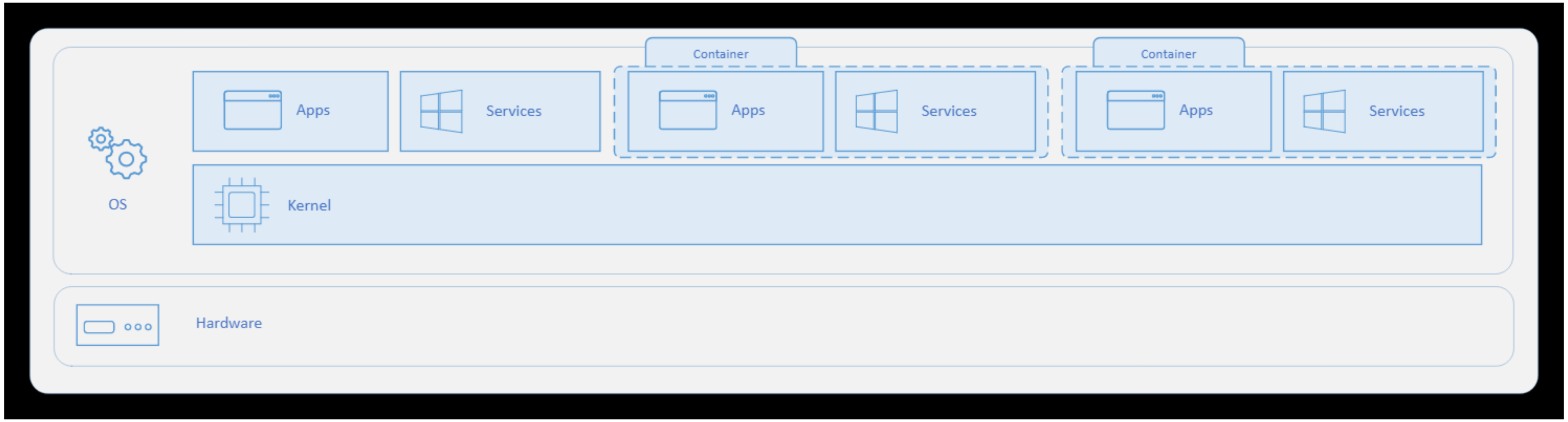


- **Containerisation**

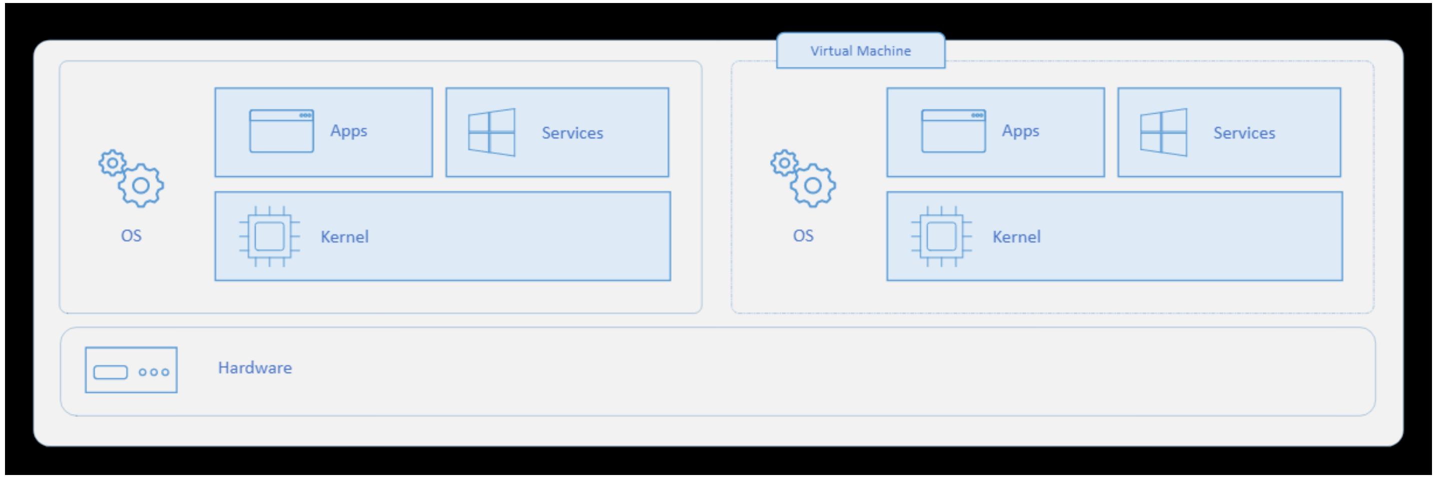
- Shared Host OS
- Near instant start-up
- Processes in containers are isolated
- Dependent app services and libraries are tied to container (layers)
- All containers on a host will share the same guest OS version

Difference between containers and virtual machines

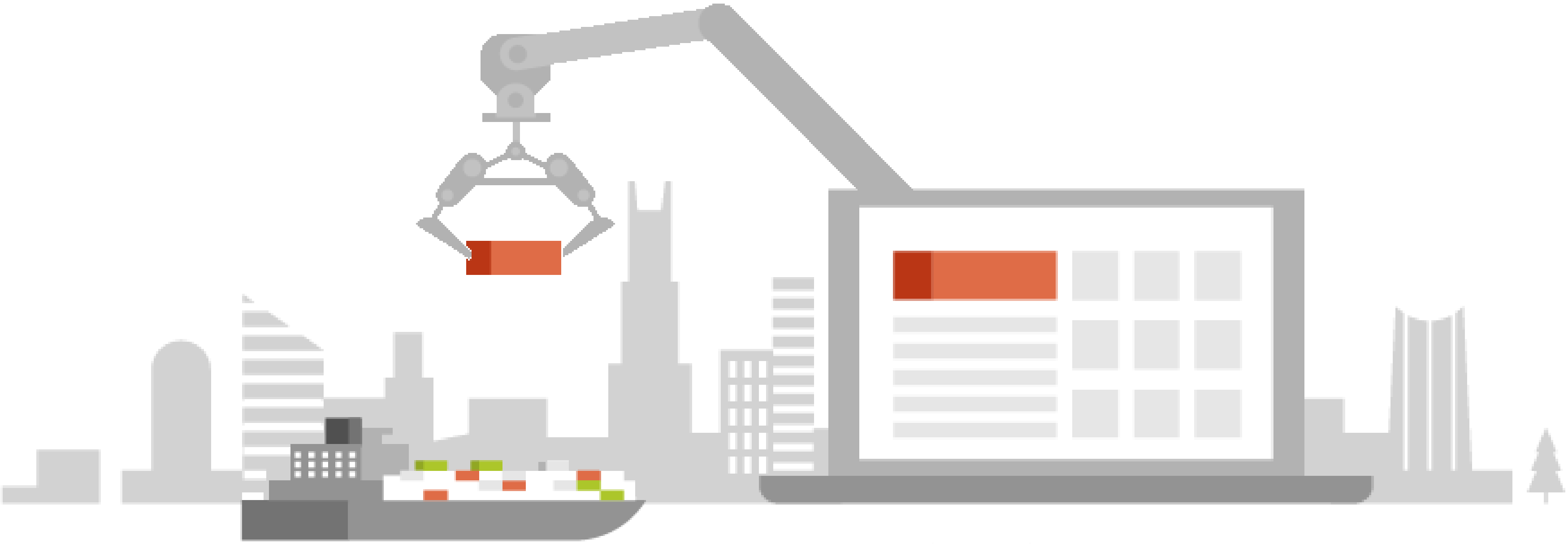
Container architecture



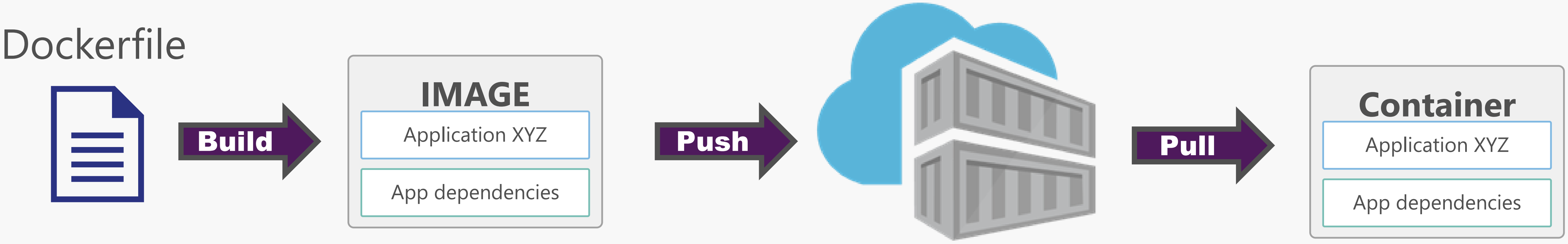
Virtual machine architecture



Docker – One example of a container engine



Container Lifecycle



Dockerfile

Build

IMAGE

Application XYZ

App dependencies

Push

Pull

Container

Application XYZ

App dependencies

Like a Container
that's "NOT running" yet

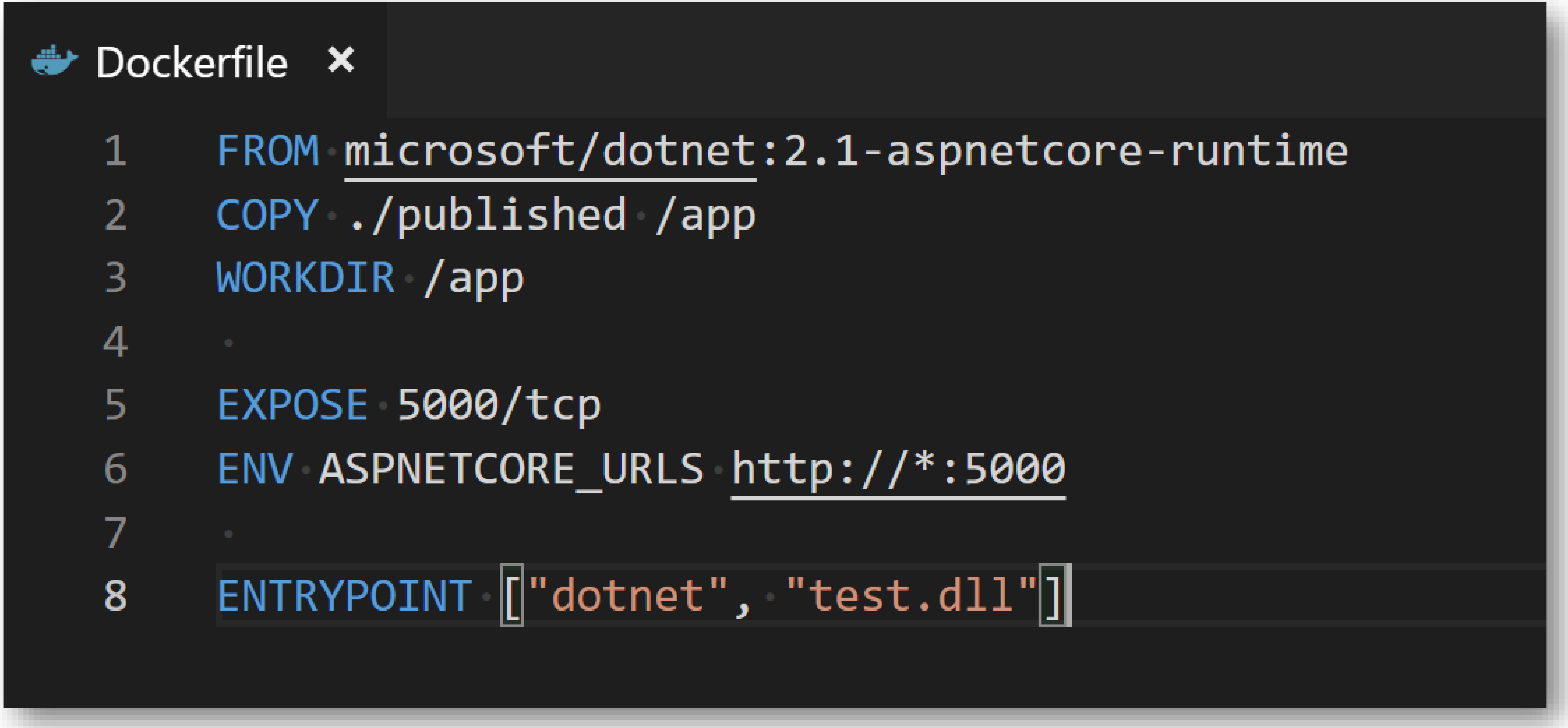
Repository, e.g.:

Azure Container Registry
-or-
Docker Hub
hub.docker.com

"A Running Image"

Dockerfiles

Recipes for building containers



```
FROM microsoft/dotnet:2.1-aspnetcore-runtime
COPY ./published /app
WORKDIR /app

EXPOSE 5000/tcp
ENV ASPNETCORE_URLS http://*:5000

ENTRYPOINT ["dotnet", "test.dll"]
```

The elements of **orchestration**

Scheduling

Affinity/anti-affinity

Health monitoring

Failover

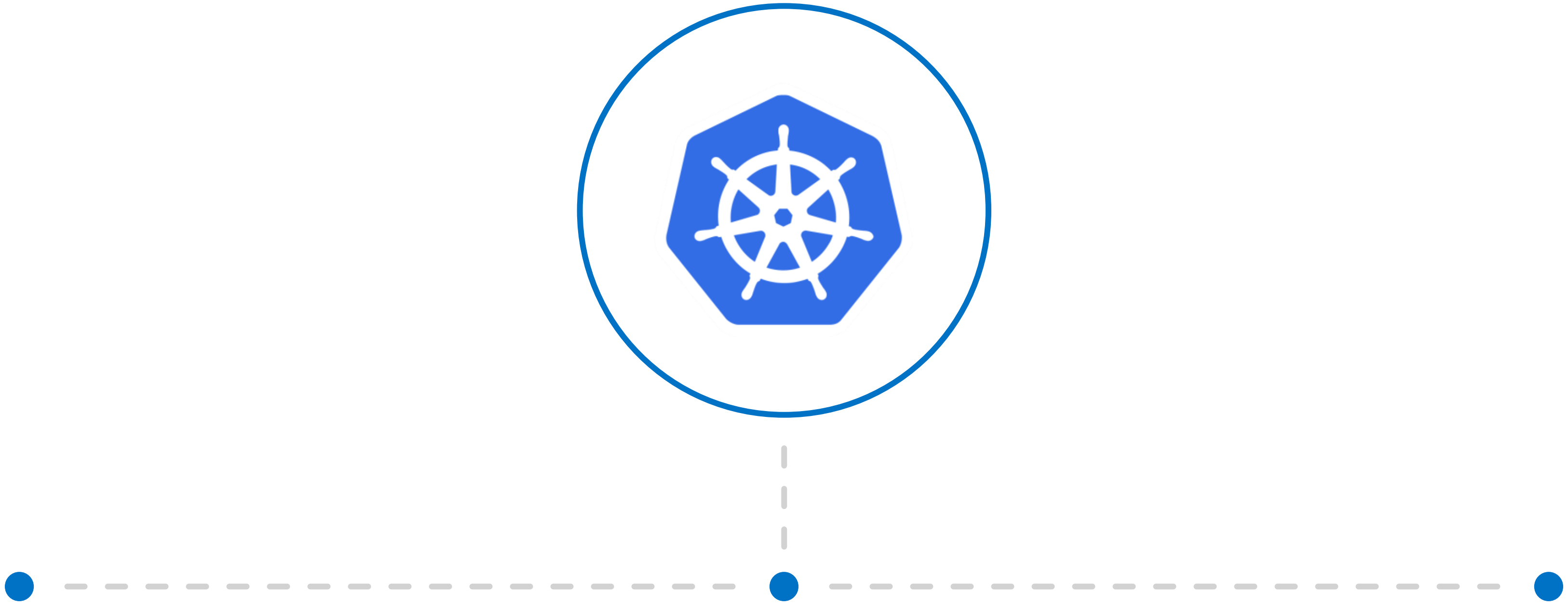
Scaling

Networking

Service discovery

Coordinated app upgrades

Kubernetes: the industry leading orchestrator



Portable

Public, private, hybrid, multi-cloud

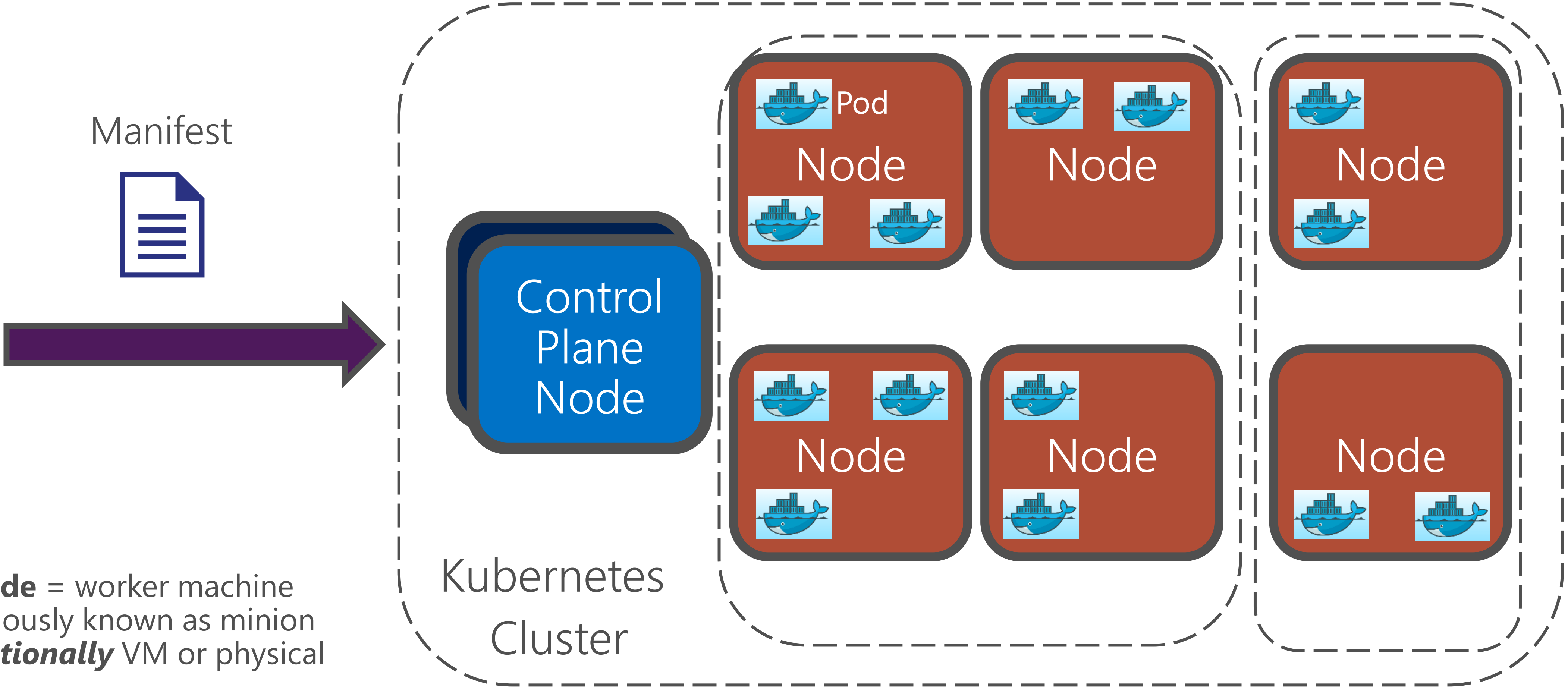
Extensible

Modular, pluggable, hookable, composable

Self-healing

Auto-placement, auto-restart, auto-replication, auto-scaling

Kubernetes Cluster



Node

Node = worker machine
Previously known as minion
Traditionally VM or physical

Pod = Basic K8S Building Block
“scale unit” of
one or more Containers
(but one is MOST common)

How Managed Kubernetes on Azure works

Automated upgrades, patches

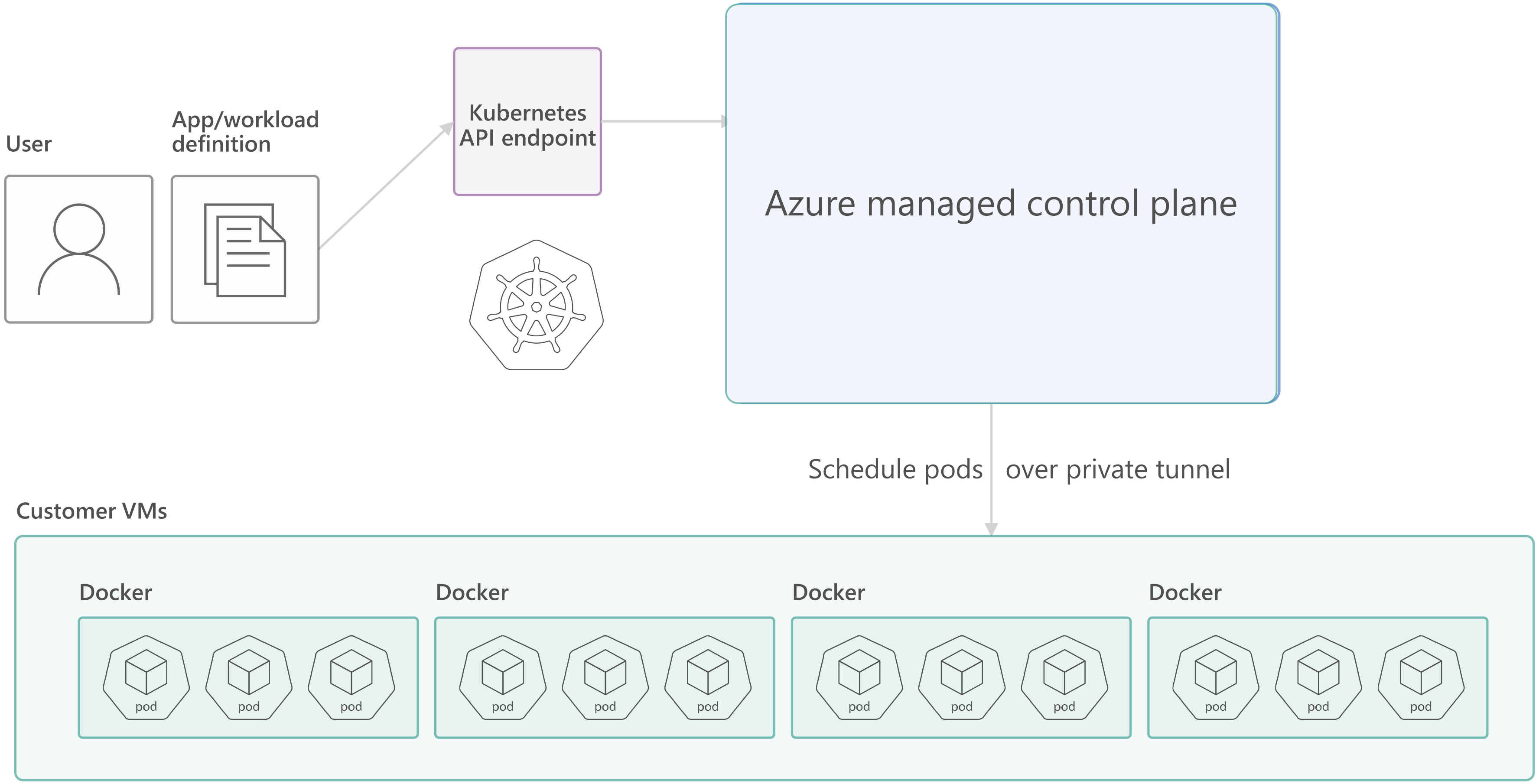
High reliability and availability

Easy and secure cluster scaling

Self-healing

API server monitoring

Control plane at no charge



From infrastructure to **innovation**

Managed Kubernetes empowers you to do more

Focus on your containers and code, not the plumbing of them.

Responsibilities	DIY with Kubernetes	Managed Kubernetes on Azure
Containerization	⬜	◪
Application iteration, debugging	⬜	◪
CI/CD	⬜	◪
Cluster hosting	⬜	🟦
Cluster upgrade	⬜	🟦
Patching	⬜	🟦
Scaling	⬜	🟦
Monitoring and logging	⬜	🟦

⬜ Customer

🟦 Microsoft

Work how you want with opensource tools and APIs

	Development	DevOps	Monitoring	Networking	Storage	Security
Take advantage of services and tools in the Kubernetes ecosystem	HELM DRAFT	Jenkins Terraform BRIGADE JFrog CODESHIP	Prometheus fluentd Grafana OPENTRACING DATADOG JAEGER	CNI Networking TIGERA	MAPR portworx	Twistlock aqua heptio HASHICORP RBAC
OR, Leverage growing Azure support	VS Code	Azure DevOps ARM	Azure Monitor	Azure VNET	Azure Storage	Azure Container Registry AAD Key Vault

Summary

- ✓ Container: content/hardware agnostic, efficient - light weight, more dense than VMs
- ✓ Kubernetes is an open-source container orchestrator that helps to handle containerized microservice architectures
- ✓ Azure Kubernetes Service: Managed K8s that removes complexity by handling management tasks (auto-upgrades, patching, self-healing …)
- ✓ AKS takes full advantage of all open-source solutions combined with a seamless integration in Azure services

Links

- Introduction to Azure Kubernetes Service - Azure Kubernetes Service | Microsoft Docs
- Kubernetes on Azure tutorial - Deploy a cluster - Azure Kubernetes Service | Microsoft Docs

Thank you!

ευχαριστώ Salamat Po متشکرم شكراً Grazie

благодаря ありがとうございます Kiitos Teşekkürler 谢谢

ขอบคุณครับ Obrigado شکریہ Terima Kasih Dziękuję

Hvala Köszönöm Tak Dank u Wel дякую Tack

Mulţumesc спасибо Danke Cám ơn Gracias

多謝晒 Ďakujem תודה நன்றி Děkuji 감사합니다

Microsoft Azure

Microsoft France • 39, quai du Président Roosevelt • 92130 Issy-Les-Moulineaux • France