# Microsoft Azure Developer: Deploying and Managing Containers

## INTRODUCING CONTAINERS ON AZURE

**Mark Heath**
MICROSOFT AZURE MVP

@mark_heath https://markheath.net

# Course Overview

What are Docker containers?

How can we deploy them to Azure?

How can we run containers locally?

How can we create and publish container images?

What are our options for hosting in Azure?

- Azure Container Instances
- Azure Web Apps for Containers
- Azure Service Fabric
- Azure Kubernetes Service (AKS)

# Course Overview

What are Docker containers?

How can we deploy them to Azure?

How can we run containers locally?

How can we create and publish container images?

What are our options for hosting in Azure?

How can we secure containers?

Azure Container Instances (ACI)

Azure Web Apps for Containers

Azure Service Fabric

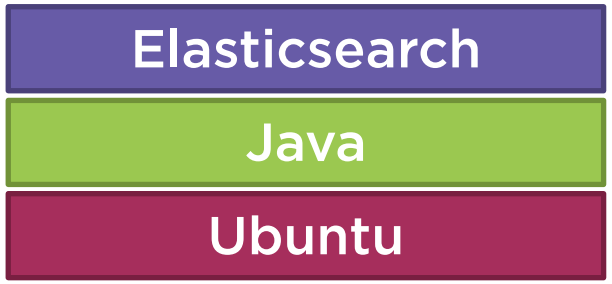
Azure Kubernetes Service (AKS)

# Docker Basics

## Images

Image = application + dependencies

Layered




Built from a Dockerfile

Tagged - e.g. markheath/myapp:1.4

Publish to a "container registry"

- Docker Hub
- Azure Container Registry

## Containers

An instance based on an image

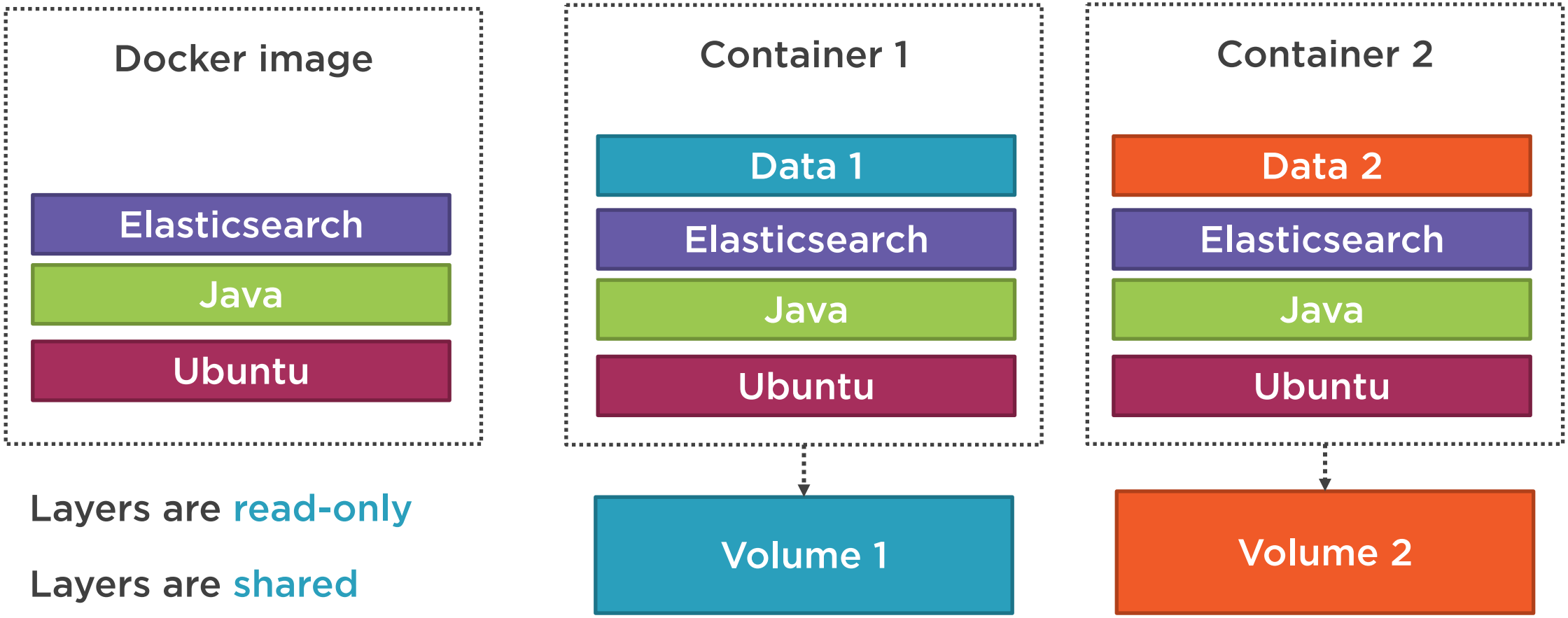
Run on a "Docker host"

- Works the same everywhere
- Provides memory and CPU
- Publish ports
- Disk access (image only)

Can be stopped and restarted

Multiple containers from a single image

# Data Storage



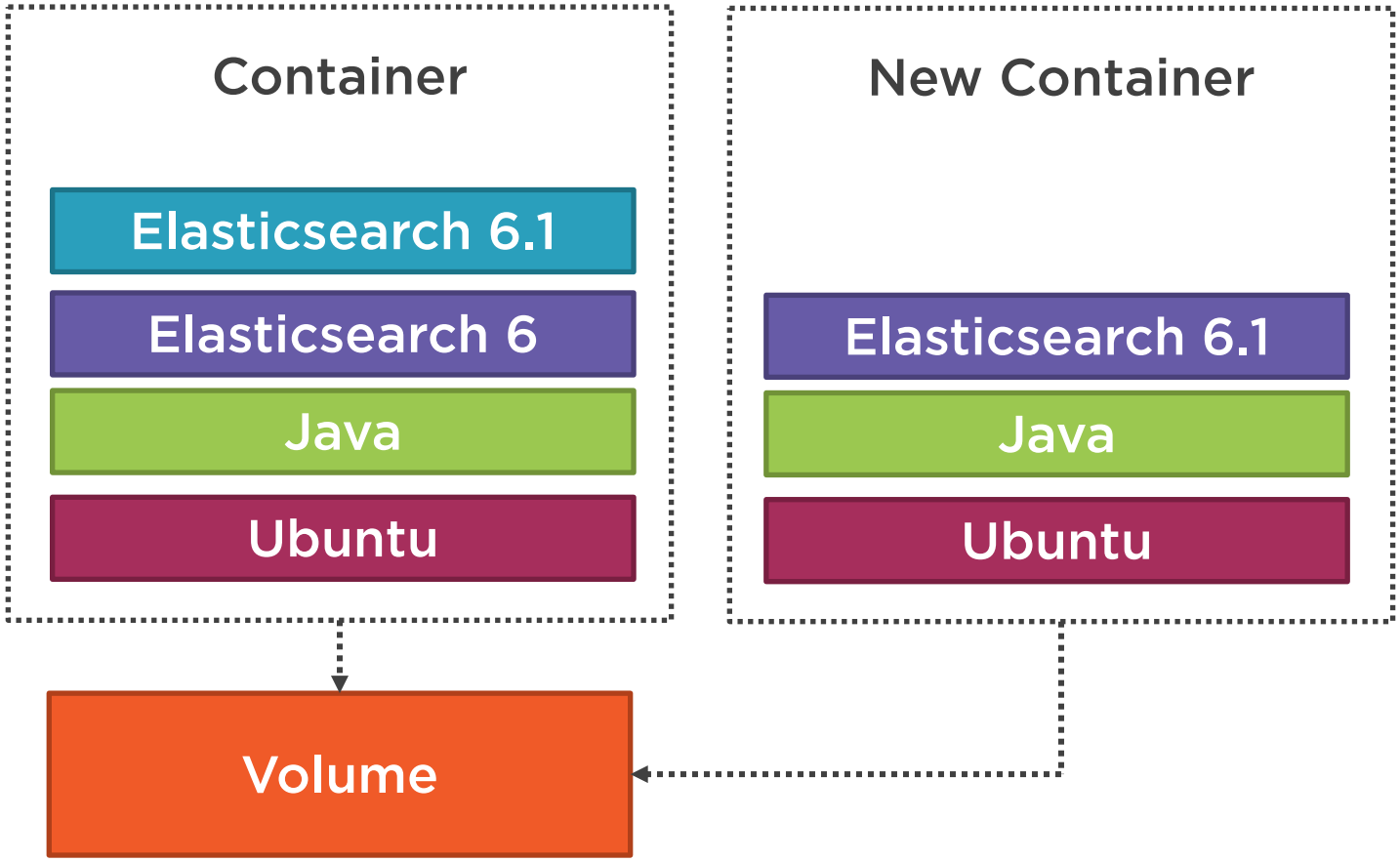

Layers are read-only

Layers are shared

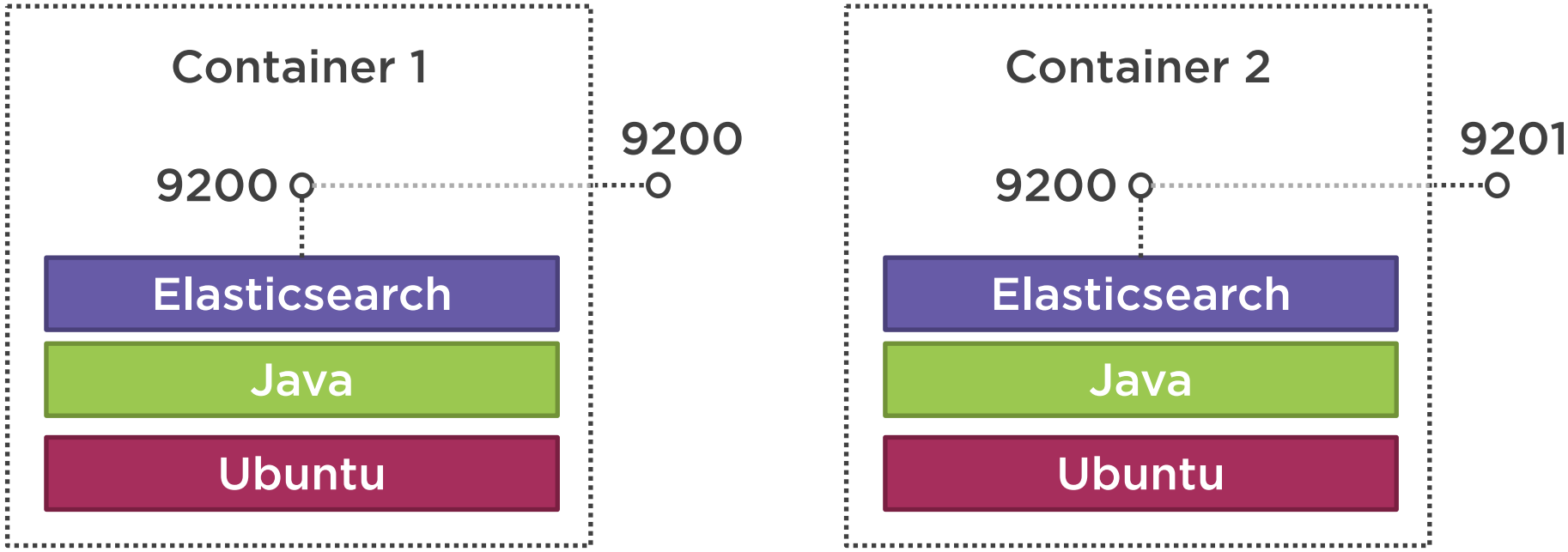
Use volumes to persist data

Mount volumes to a path in the container – e.g. /var/lib/mysql

# Upgrades


Container
Elasticsearch 6.1
Elasticsearch 6
Java
Ubuntu
New Container
Elasticsearch 6.1
Java
Ubuntu
Volume

# Network



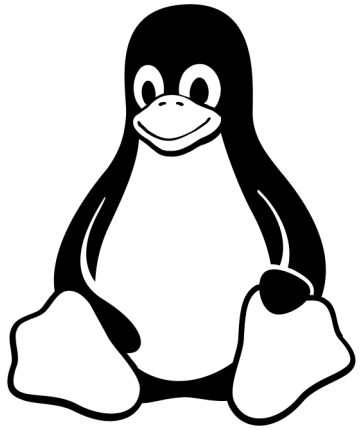

Containers can publish a port

Docker hosts can map a port

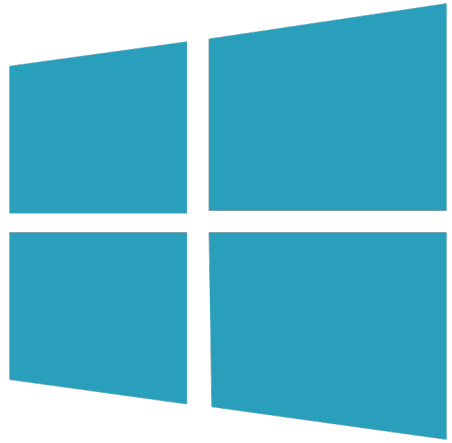
# Environment Variables


Containers have their own environment variables

Linux



Windows



Windows Server 2016 and newer supports Windows containers

Windows Subsystem for Linux (WSL 2)

# Docker Benefits

## Containers

- Isolation
- Portability
- Efficient
- Fast start
- Disposable
- Minimal attack surface area

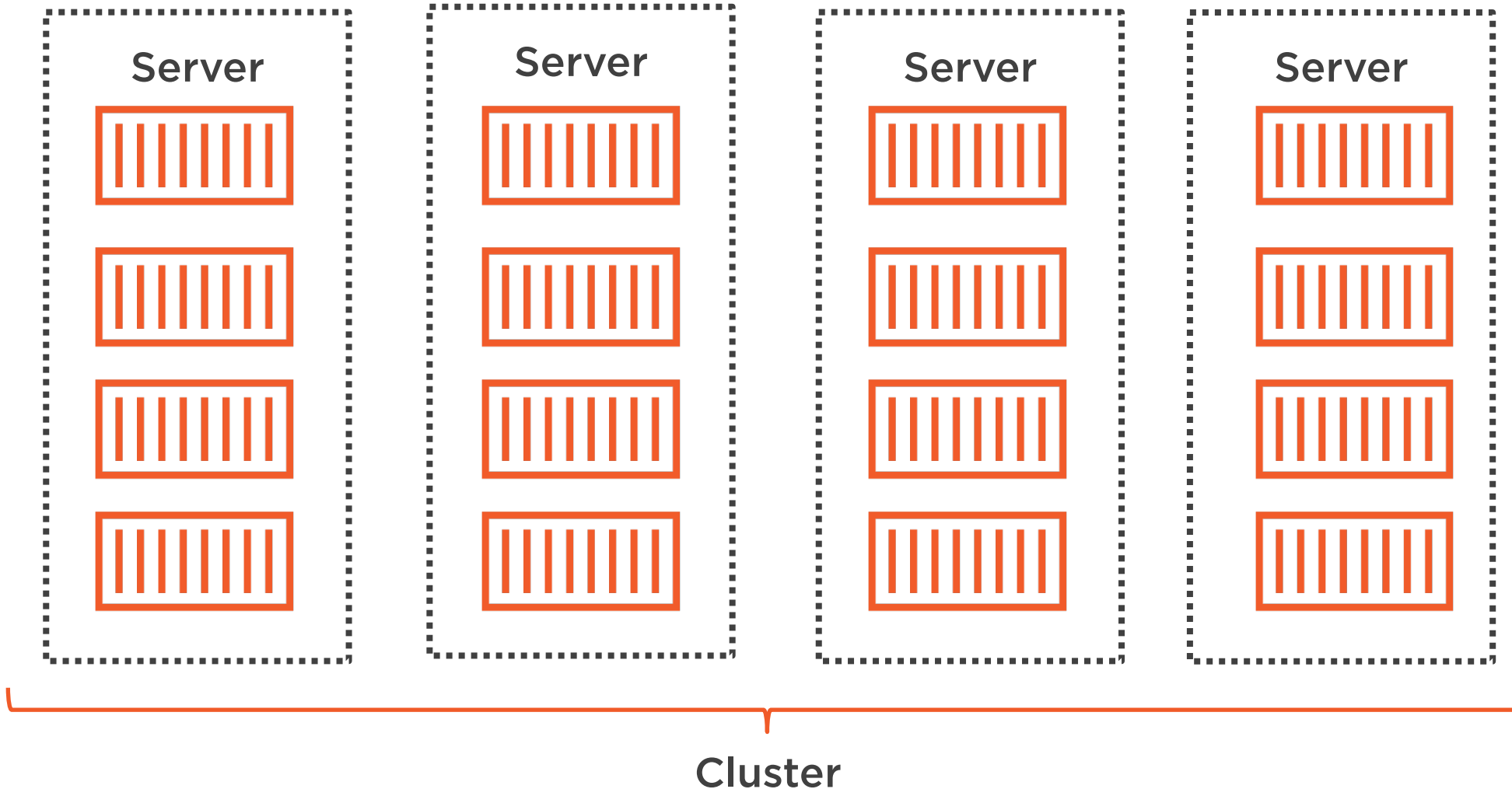
## VMs

- Strong isolation
- Portability
- Resource hungry
- OS boot times
- Require patching
- OS needs hardening

# Security

- Isolation
- Minimal attack surface area
- Vulnerability scanning
- Image signing

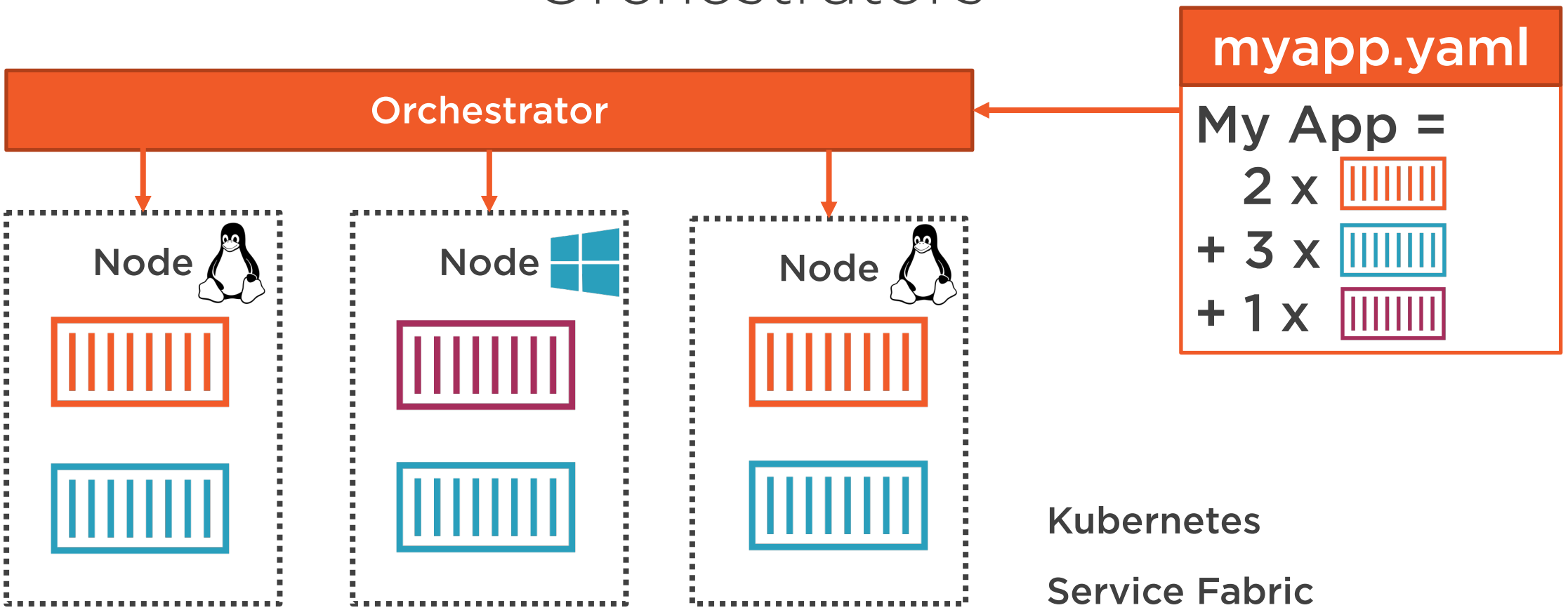
# Managing Multiple Containers




**Challenges:**

- Deploying
- Scaling
- Monitoring
- Networking
- Upgrading
- Repairing

# Orchestrators




Kubernetes

Service Fabric

# Hosting Containers in Azure

| Virtual Machines (IaaS) |
| --- |
| Linux |
| Windows Server |
| Kubernetes |

| Azure Container Instances (ACI) |
| --- |
| Serverless |
| Fast and easy |
| Per-second billing |

| Azure Web Apps for Containers |
| --- |
| Great for web apps |
| Custom domains |
| Auto-scaling |

| Azure Service Fabric |
| --- |
| Scalable orchestration platform |
| Powers key Azure services |
| Multiple programming models |

| Azure Kubernetes Service (AKS) |
| --- |
| Managed Kubernetes cluster |
| Just specify node count |
| Open source tooling |

# Summary

**Docker basics**

- Containers
- Images
- Orchestrators

**Docker benefits**

- Isolation
- Efficiency
- Portability
- ...and more

**Containers in Azure**

- Azure Container Instances (ACI)
- Azure Web Apps for Containers
- Azure Service Fabric
- Azure Kubernetes Service (AKS)

# Up next:
# Running Containers Locally with Docker