Running Containers on Azure Service Fabric

Mark Heath
MICROSOFT AZURE MVP

@mark_heath https://markheath.net

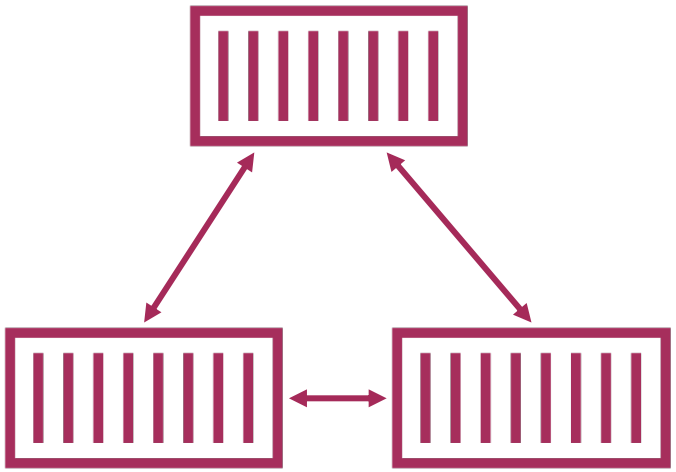
Azure Container Instances (ACI)

Azure Web App for Containers

Azure Service Fabric

Azure Kubernetes Service (AKS)

Challenges of Microservices



Deployment

Health monitoring

Scaling out to multiple instances

Service to service communication

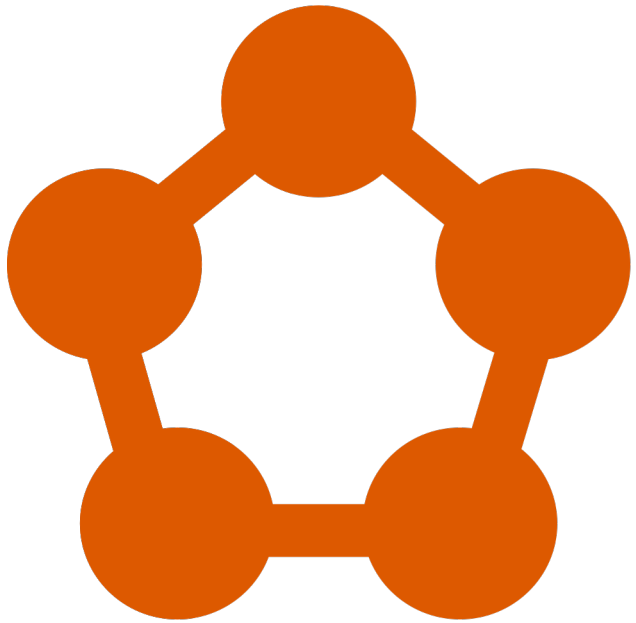
Upgrades

Recover from hardware failures

Orchestrators can help us

- Azure Service Fabric

Azure Service Fabric



An “application platform”

- Scalable and reliable microservices

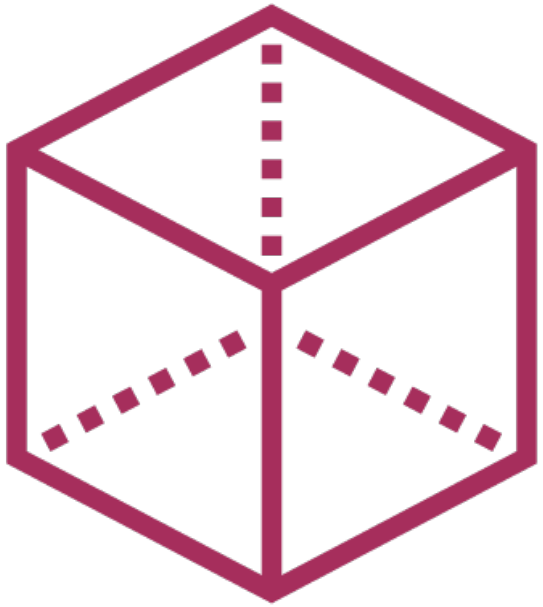
Hosting options

- On-premises or other cloud providers
- Development laptop
- Azure

Cluster

- Monitors service health

Programming Models



Stateful services

- Co-locate compute and data
- Reliable collections

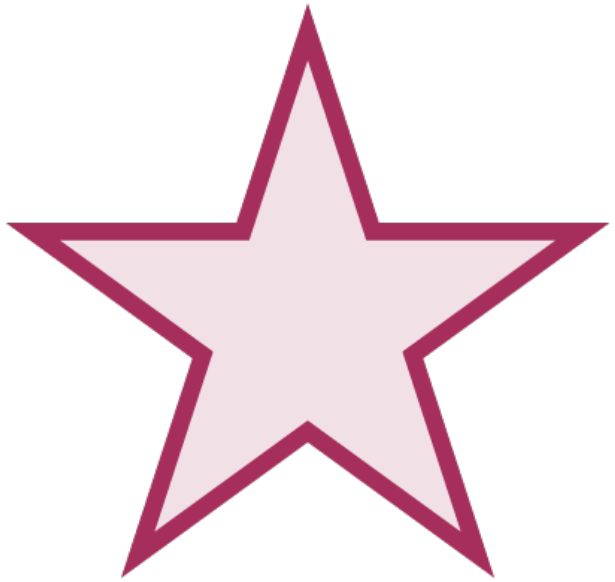
Stateless services

- Web APIs or executables
- Containers

Linux and Windows containers

- Constrain RAM and CPU allocation
- Docker Compose YAML support

Service Fabric Benefits



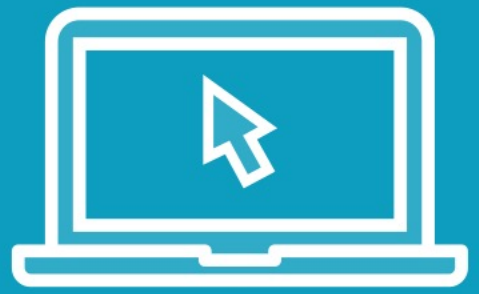
Powers many key Azure services

- e.g. Cortana, Skype, Cosmos DB & Power BI

Why choose Service Fabric?

- Microservices applications
- Windows containers
- Ability to deploy outside Azure
- Orchestration features

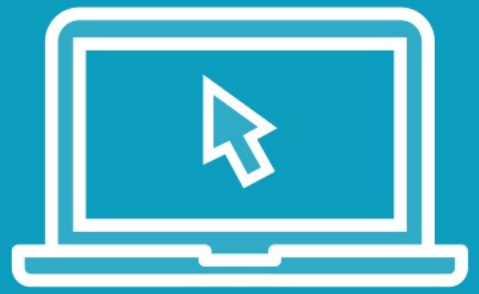
Demo



Setting up an Azure Service Fabric development environment

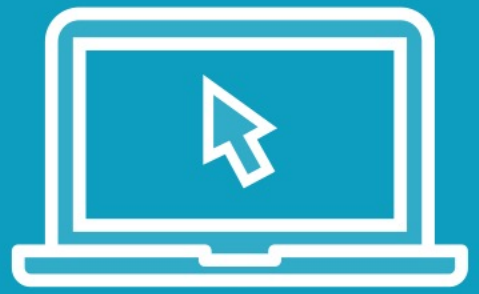
- Service Fabric tools

Demo



Creating a Service Fabric cluster in the Azure portal

Demo



Deploy an application to a Service Fabric cluster

Summary

Azure Service Fabric

- Powerful orchestration platform
- Windows and Linux container support
- Runs in many environments
- Define applications with manifest XML
- Service Fabric Explorer dashboard

Up next:
Running Containers on Azure Kubernetes Service