



# Modernizing Applications with Containers and Orchestrators

Microsoft Services





# Module 5 – Container Orchestrators

Microsoft Services



# Agenda

- What Is Orchestration?
- Microsoft Offerings For Containers
- Additional Options For Orchestration

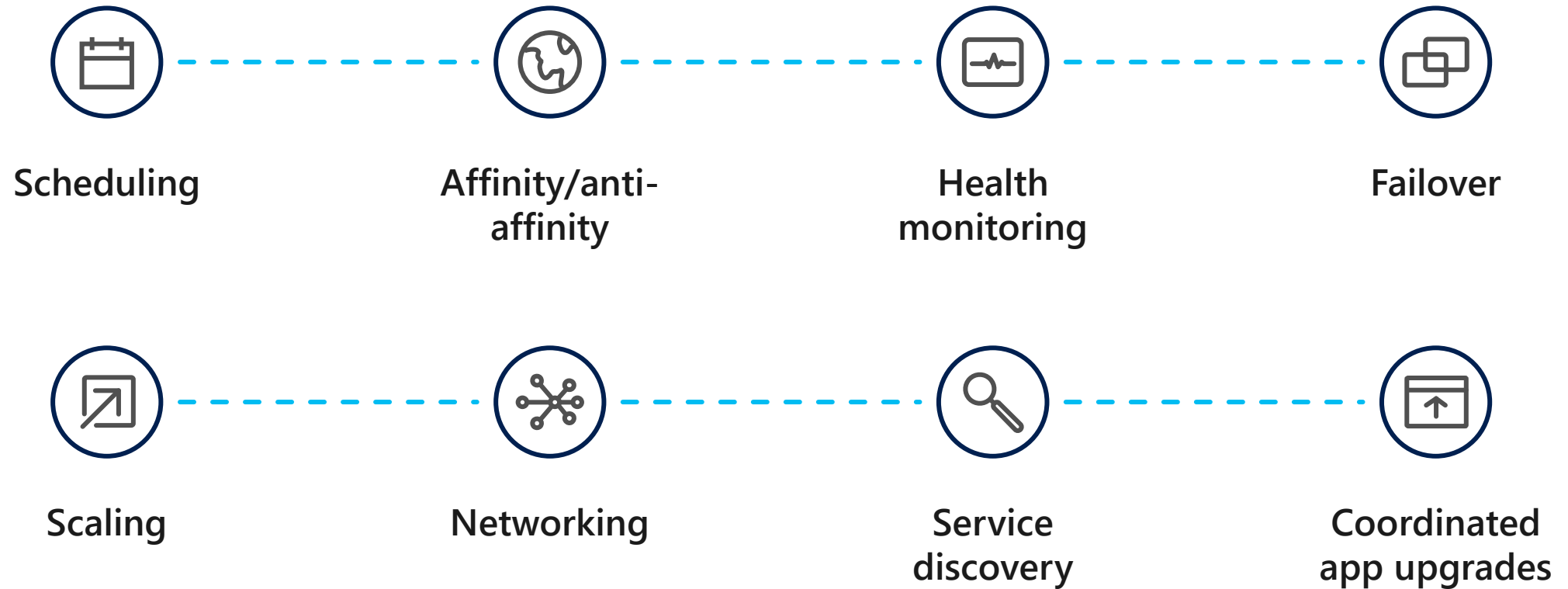


# Challenges Of A Containerized World

As application development has moved towards a container-based approach, the need to orchestrate and manage the inter-connected resources becomes important

- Load Balancing
  - Distributing traffic across containers at scale
- Naming and Discovery
  - How do containers or groups find one another?
- Logging and Monitoring
  - Keeping track of what containers are doing
- Debugging
  - Getting inside running containers
- Networking
  - Differentiating container networks from host networks at scale

# The Elements Of Orchestration



# Clustering Versus Orchestration

## Clustering

- Grouping “hosts”—either VMs or bare metal—and networking them together
- A cluster should feel like a single resource rather than a group of disparate machines

## Orchestration

- Managing and monitoring of the workloads running in your cluster
- Starting containers on appropriate hosts and connecting them
- May also include support for scaling, automatic failover, and node rebalancing

# Microsoft Offerings for Containers

IF YOU'RE LOOKING FOR THIS...	USE THIS
Scale and orchestrate containers using Kubernetes	<a href="#">Kubernetes Service</a>
Easily run containers on Azure with a single command	<a href="#">Container Instances</a>
Store and manage container images across all types of Azure deployments	<a href="#">Container Registry</a>
Develop microservices and orchestrate containers on Windows or Linux	<a href="#">Service Fabric</a>
Deploy web applications on Linux using containers	<a href="#">App Service</a>
Run repetitive compute jobs using containers	<a href="#">Batch</a>

# Azure App Service

Easily deploy and run container-based web apps at scale

## Accelerated outer loop



Tight integration w/ Docker Hub, Azure Container Registry



Built-in CI/CD w/ Deployment Slots



Intelligent diagnostics & troubleshooting, remote debugging

## Fully managed platform



Automatic scaling and load balancing



High availability w/ auto-patching



Backup & recovery

## Flexibility & choices



From CLI, portal, or ARM template



Single Docker image, multi container w/ Docker Compose

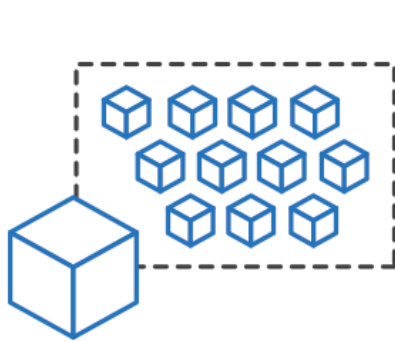


IntelliJ, , Jenkin, Maven Visual Studio family

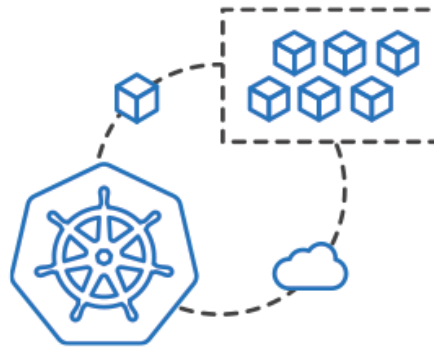


# Azure Container Instances (ACI)

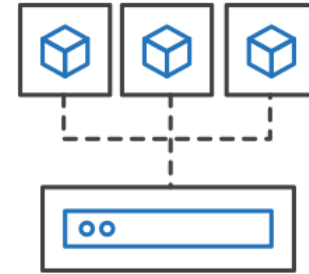
Easily run containers on Azure without managing servers



Run containers  
without managing  
servers



Increase agility  
with containers on  
demand



Secure applications  
with hypervisor  
isolation



# Azure Container Instances (ACI)

Great for:

- Isolated Windows and Linux containers
- Simple applications
- Task automation
- Build jobs
- Hypervisor-level security
- Custom sizes for CPU cores and memory
- Public IP connectivity
- Persistent storage
- Co-scheduled groups

NOT great for:

- Full container orchestration
- Service discovery across multiple containers
- Automatic scaling
- Coordinated application upgrades

# Demonstration: *Azure Container Instance*

Deploy `dwolters/pandoc-http` Image





HashiCorp  
Nomad



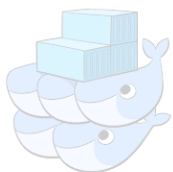
MESOS



RANCHER



OPENSIFT



# Kubernetes

- **Kubernetes** is "an open-source software for automating deployment, scaling, and management of containerized applications"
- **Kubernetes**, in Greek κυβερνήτης, means the Helmsman, or pilot of the ship
- Keeping with the maritime theme of **Docker** containers, **Kubernetes** is the pilot of a ship of containers



HashiCorp  
**Nomad**



**MESOS**



**RANCHER**



**OPENSIFT**



# Kubernetes

The de-facto orchestrator



**Portable**  
Public  
Private  
Hybrid  
multi-cloud

**Extensible**  
Modular  
pluggable  
Hookable  
composable

**Self-healing**  
Auto-placement  
auto-restart  
auto-replication  
auto-scaling



HashiCorp  
**Nomad**



MESOS



**RANCHER**



OPENSIFT



# Kubernetes

Empowering you to do more



Deploy your  
applications quickly  
and predictably

Scale your  
applications  
on the fly

Roll out  
new features  
seamlessly

Limit hardware  
usage to required  
resources only





HashiCorp  
**Nomad**



MESOS



**RANCHER**



OPENSIFT



# Azure Kubernetes Engine

Open source: <https://github.com/Azure/aks-engine>

- Easiest way to provision a self-managed Kubernetes cluster on Azure
- Leverages Azure Resource Manager (ARM), to help you create, destroy and maintain clusters provisioned with basic IaaS resources in Azure
- Allows you to customize Deployments
  - Deploying into existing virtual networks
  - Utilizing multiple agent pools



HashiCorp  
**Nomad**



MESOS



**RANCHER**



OPENSIFT



# Azure Kubernetes Service

Simplify the deployment, management, and operations of Kubernetes



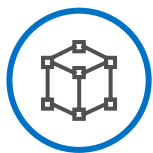
Deploy and  
manage Kubernetes  
with ease



Scale and run  
applications with  
confidence



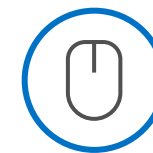
Secure your  
Kubernetes  
environment



Accelerate  
containerized application  
development



Work how you want  
with open-source  
tools & APIs



Set up  
CI/CD in a  
few clicks



HashiCorp  
Nomad



MESOS



RANCHER



OPENSIFT



# Azure Kubernetes Service

## Windows Support

- Windows support available in Preview on AKS  
=> Only Windows Server 2019 agent nodes
- AKS cluster with Windows Server feature will be composed of a Windows node pool and a Linux node pool
- Use **Node Selector** to specify on which nodes the workloads have to be deployed

# Demonstration: *Kubernetes Cluster in Azure Kubernetes Service*

Deploy Kubernetes clusters in Azure Kubernetes Service

Deploy NGINX container into Kubernetes cluster



