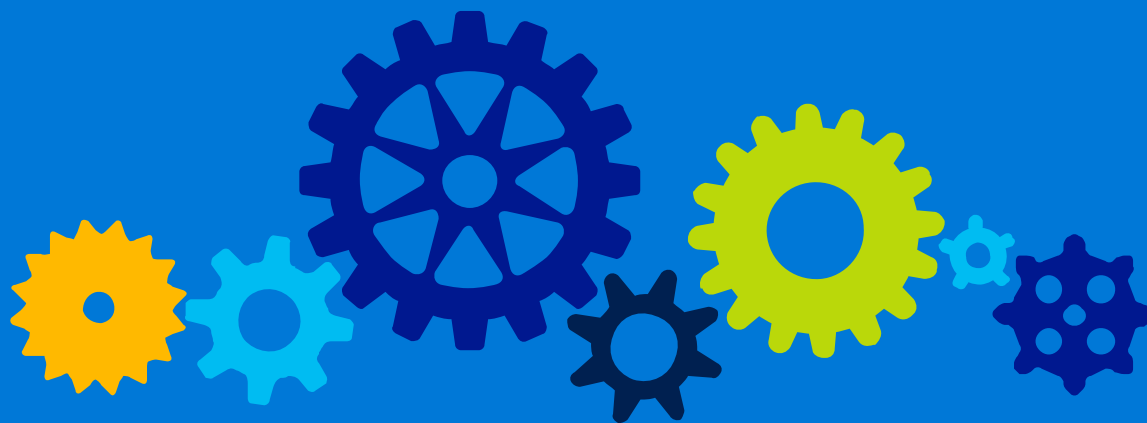




Azure Container Strategy

Ben Coleman
Cloud Solution Architect

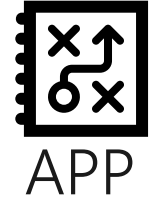
@BenCodeGeek



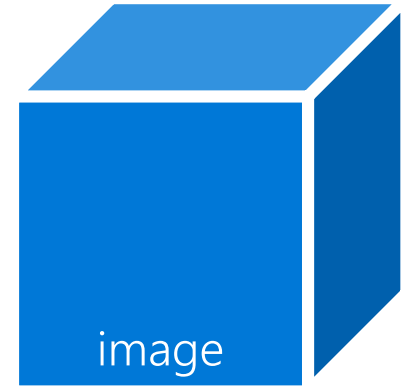
Intro To Containers

What Are Containers?

- A way to deploy your app and it's environment
- Immutable – build, deploy, replace
- Abstraction of where **services** are running
- Optimized for **fast deployment**, instancing & updating
- Services have complete **isolation** – can deploy conflicting stacks



APP

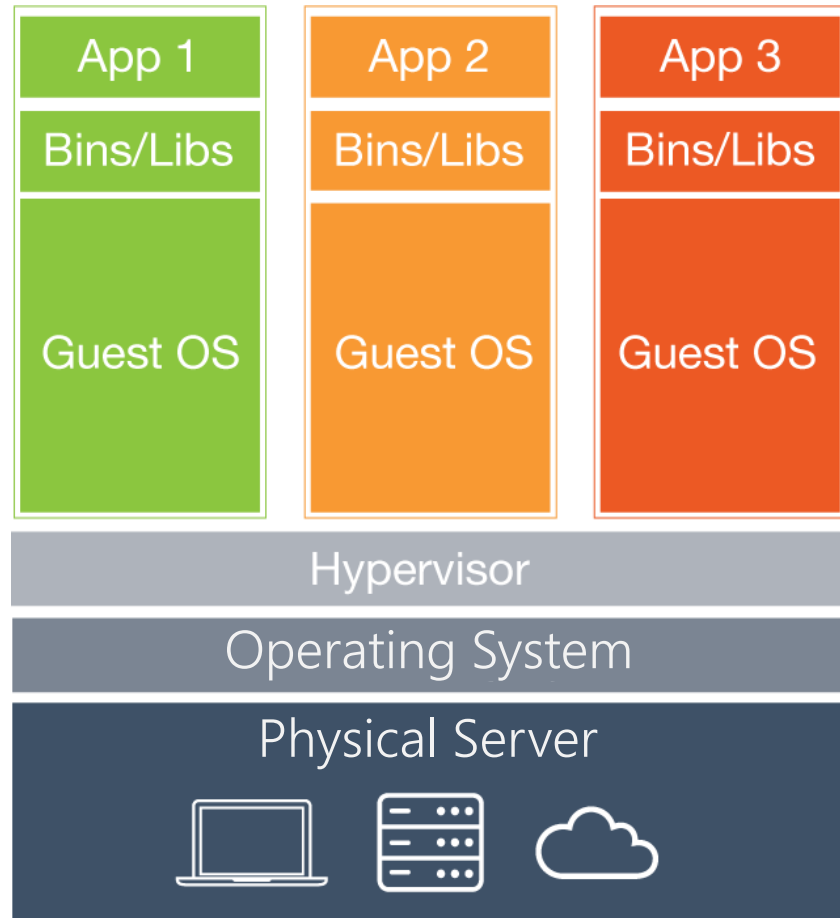


image



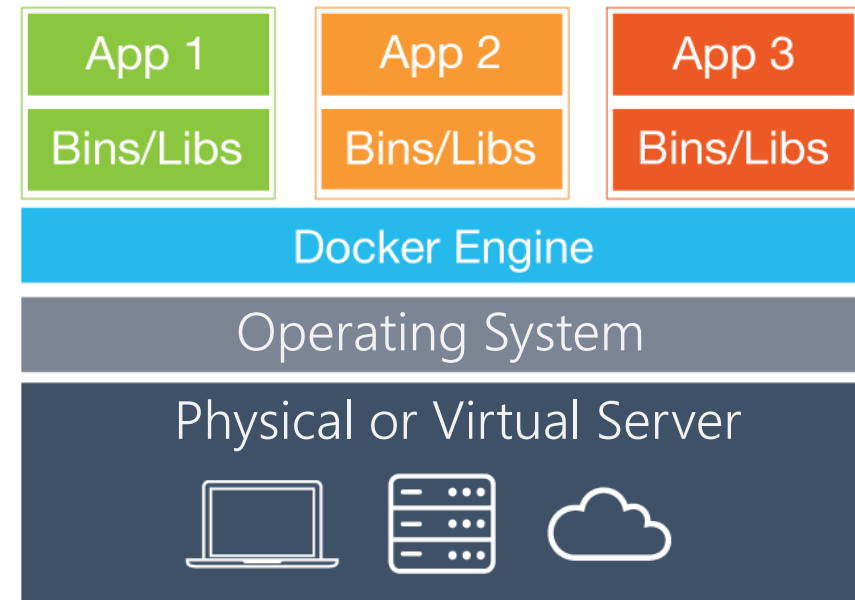
containers

Containers vs Virtual Machines



Virtual Machines

Containers are isolated, but **share OS** and, where appropriate, binaries/libraries, system calls, etc



Containers



Standardise config & packaging

- Remove dependency headaches



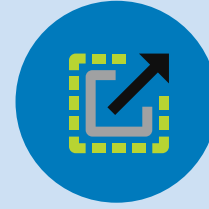
Simplify release & deployment

- \$ docker run



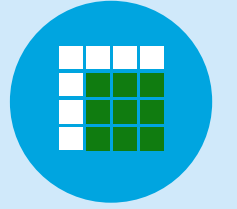
Run anywhere

- Hybrid cloud
Windows & Linux



Rapid deployment

- Seconds rather than minutes



Compute density & utilisation

- Physical Machine →
Many VMs →
Many Containers

OK Great, Why Use Containers?

Build, Ship, Run ... Any App ... Anywhere

From Dev



To Ops



Any App



Any OS



Windows



Linux

Anywhere



Physical



Virtual



Cloud

~~Use Cases~~ "Sweet Spots" for Containers



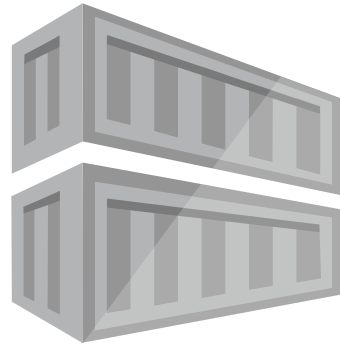
APP
MODERNIZATION



MICROSERVICES



APP
DISTRIBUTION



DEVOPS &
RELEASE PIPELINE



DEV & TEST



API
DEVELOPMENT

Containers & Azure

“Containers
Everywhere”

Containers as a
core unit of
deployment

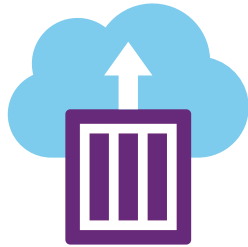


**Obligatory photo of
shipping containers**

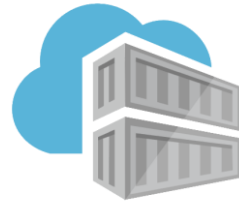
"Containers everywhere"



Azure Container Service



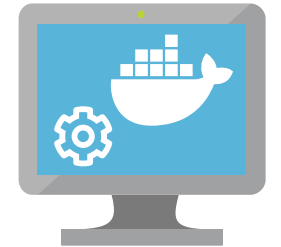
Azure Container Instances



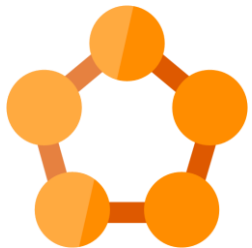
Azure Container Registry



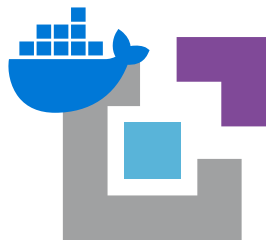
Web App for Containers



Docker Machine Driver



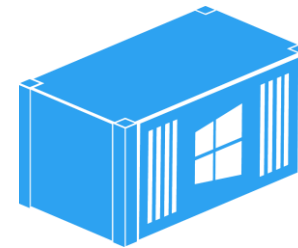
Azure Service Fabric



Docker VM Extensions



Azure DevTest Labs



Windows Server 2016



Azure Marketplace



Azure Container Service

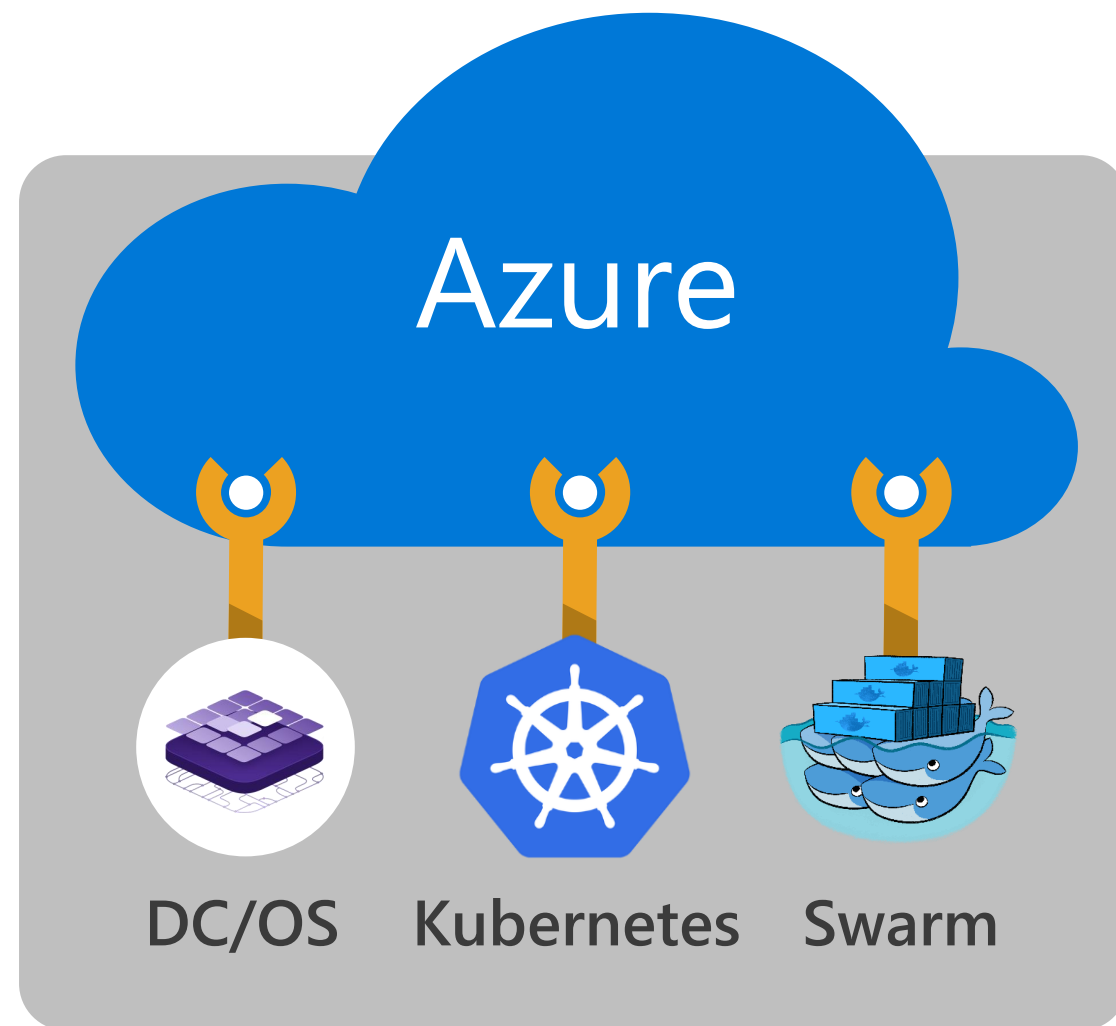
Optimized provisioning of

- Kubernetes
- DC/OS
- Docker Swarm

Use standard tooling and API support

Linux & Windows containers

Azure and Azure Stack



Azure Container Service



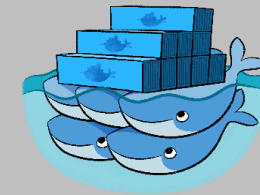
Kubernetes

- Google project
- "Production grade container orchestration"
- Basis of several PaaS platforms, e.g. OpenShift



DC/OS

- Mesosphere project
- Built on Apache Mesos
- Workload independent
- Often used with data workloads



Docker Swarm

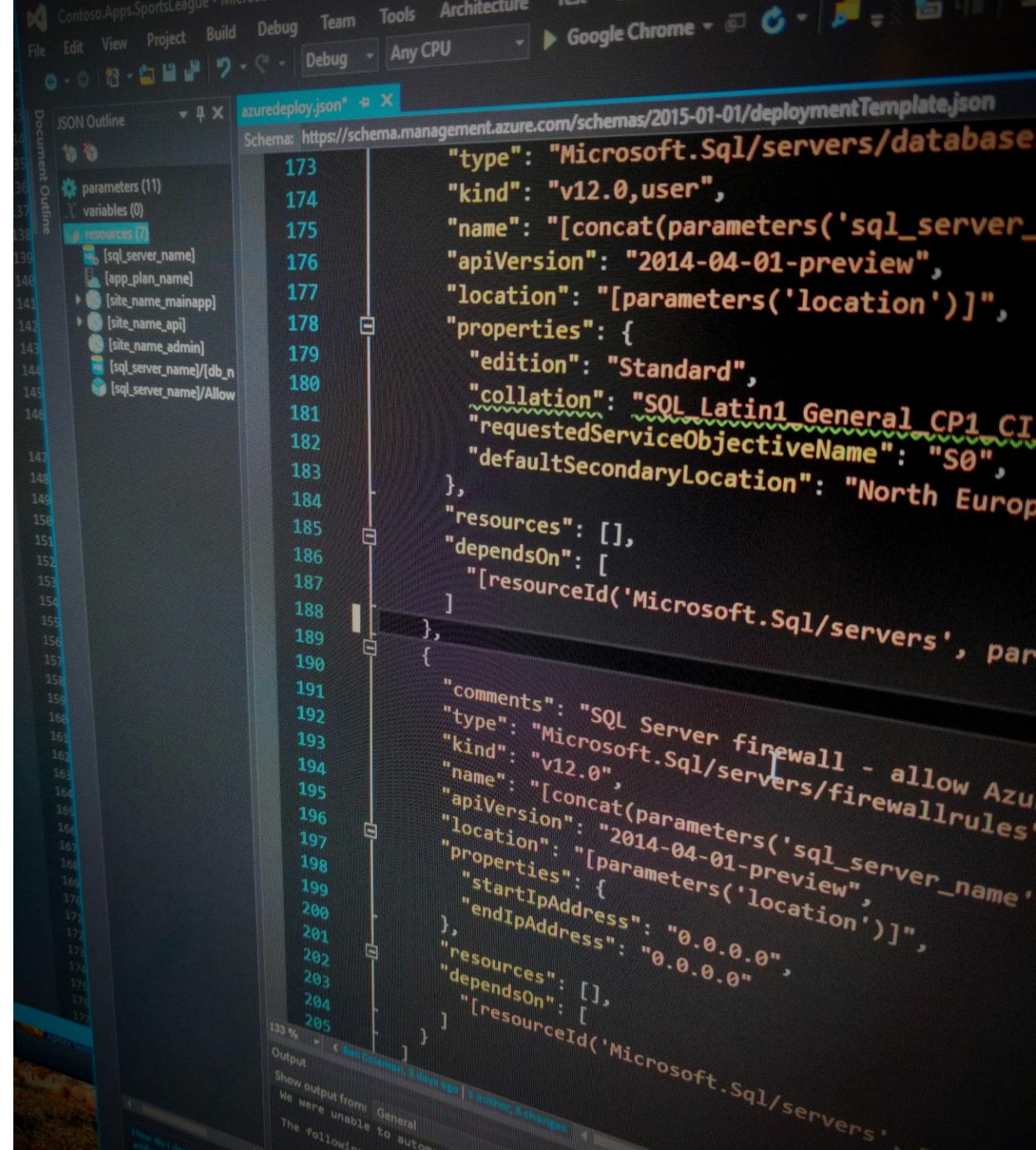
- Native Docker clustering
- Built into Docker engine
- Standard APIs, CLI tools
- Easy to manage

Azure

ACS Engine

- The deployment system behind Azure Container Service
- Open source
- Standard Azure Resource Manager (ARM) templating

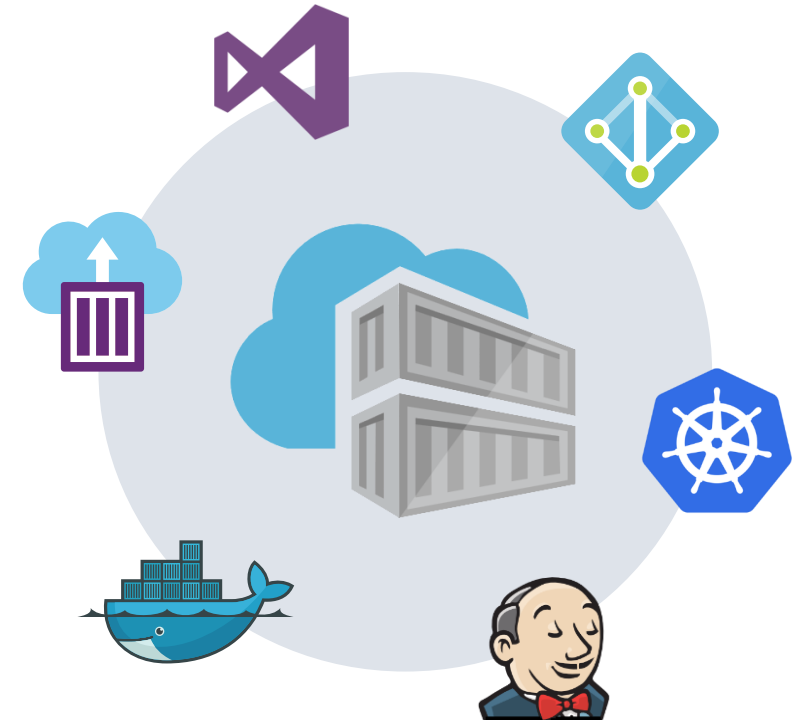
github.com/Azure/acs-engine



Azure Container Registry

Container registry as a service

- Secure private Docker v2 registry as a service
- Use **standard** Docker tools & APIs
- Backed with Azure Active Directory for **access management**
- **Webhooks** for integration & DevOps
- Fully managed & **resilient** storage

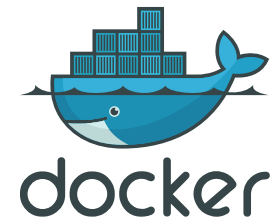


Azure Container Instances

"Containers As A Service"

- Simple - start any container in seconds
- Fully managed platform
- Containers as 1st class citizens
- No need to deploy or manage VMs

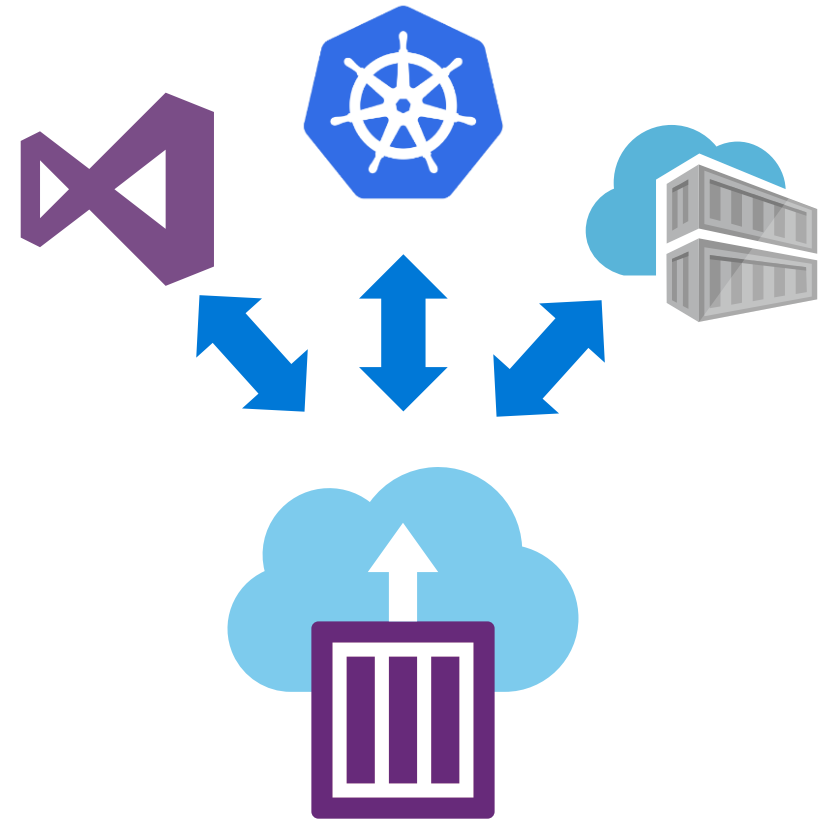
```
az container create  
--name my-container  
--image aci-helloworld  
--ip-address public
```



Azure Container Instances

PREVIEW

- Billed *per second*
- Choice of CPU / memory
- Persist volumes to Azure Storage
- Integration with other services
- Windows containers coming soon

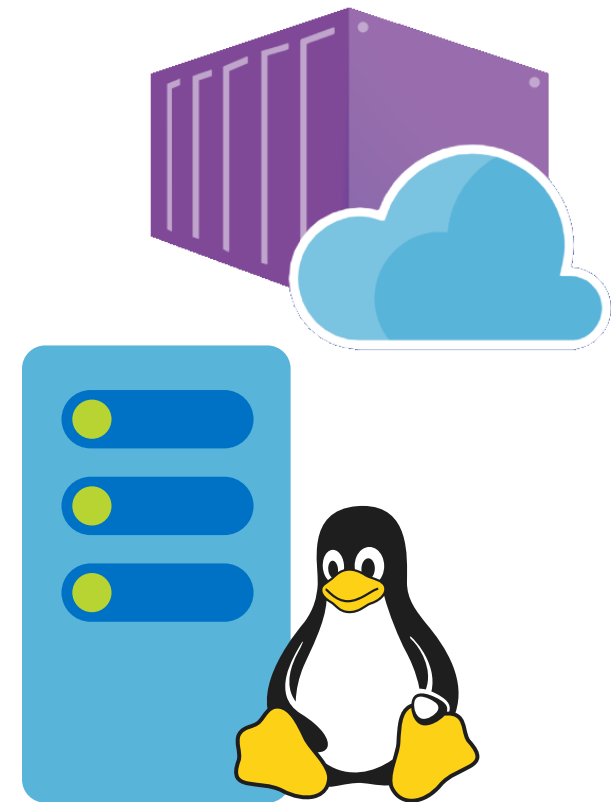


Azure Web Apps for Containers

All the benefits of PaaS + flexibility of containers

- Run **containerized web apps** in Azure App Service
- Use **provided stacks** - Node.js, PHP, .NET Core, Ruby
- Bring your **own image**
 - CI / CD
 - Public (Dockerhub) or other registry
 - Azure Container Registry
- Benefit from App Service's standard PaaS features:

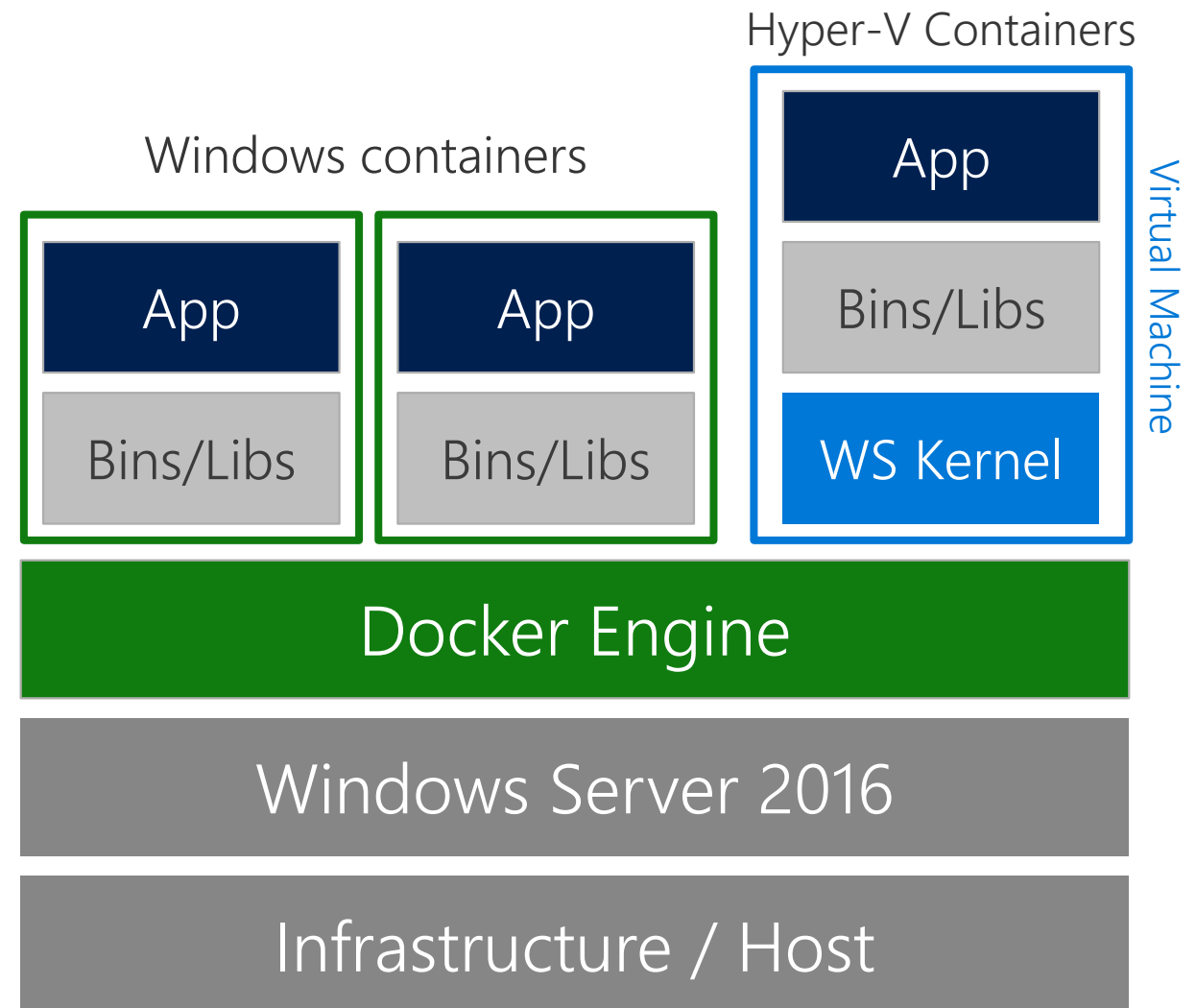
- Deployment Slots
- Autoscaling
- Continuous Deployment
- Backup
- SSL & custom domains
- Monitoring & diagnostics



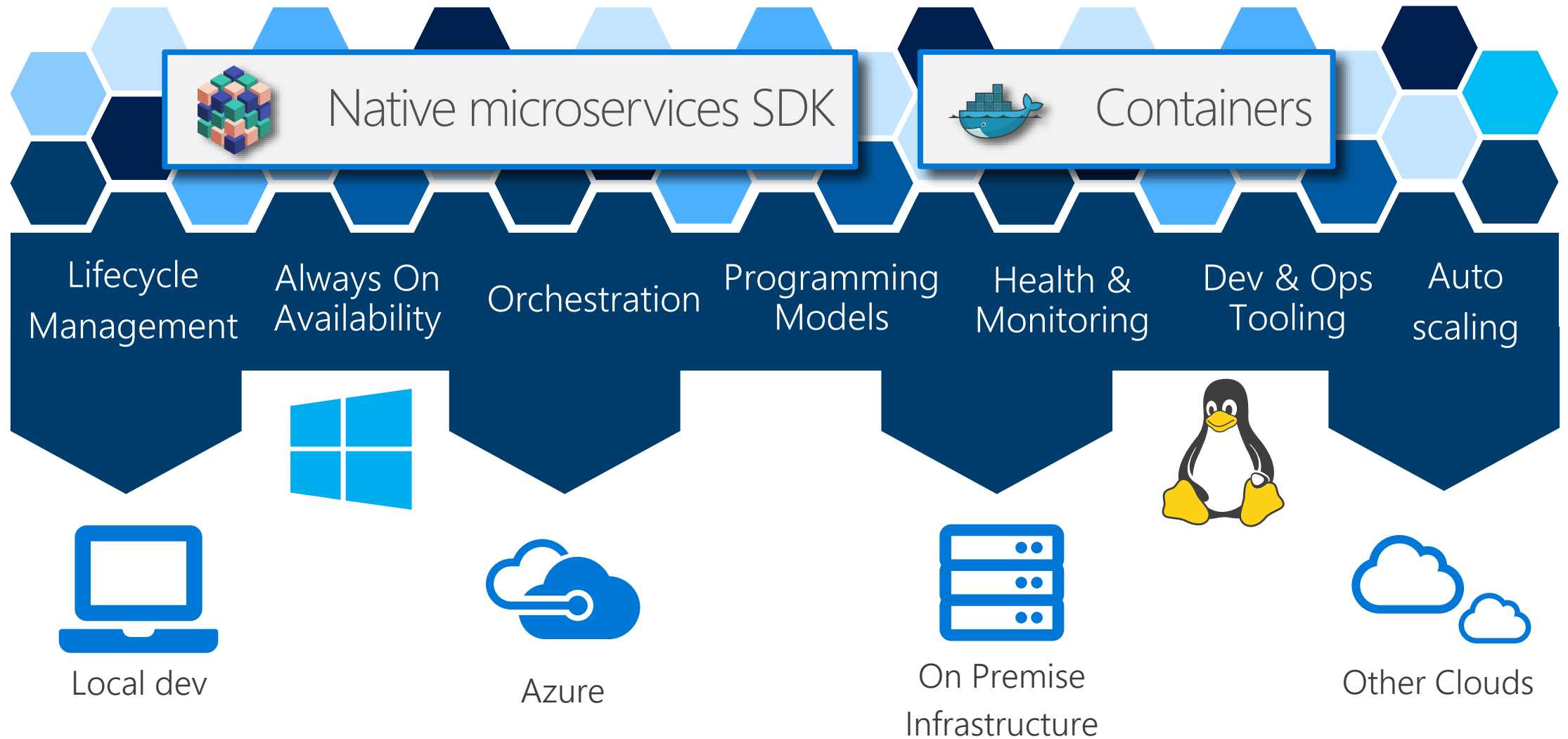
Server 2016 - Windows Containers

- Windows Containers
 - Native Windows containers
 - Powered by Docker Engine
(Same API and CLI etc)
 - No process isolation
- Hyper-V Containers
 - Same as Windows containers with added kernel & process isolation
 - Wrapped in a thin invisible VM

```
docker run --isolation=hyperv
```

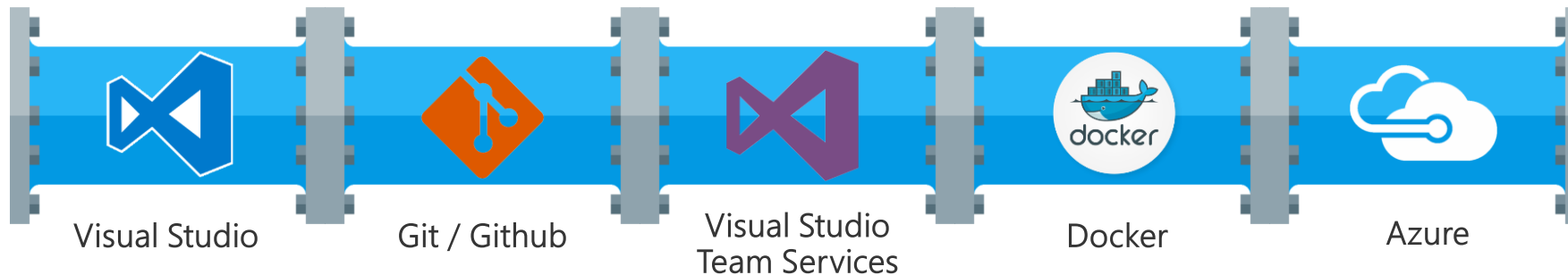
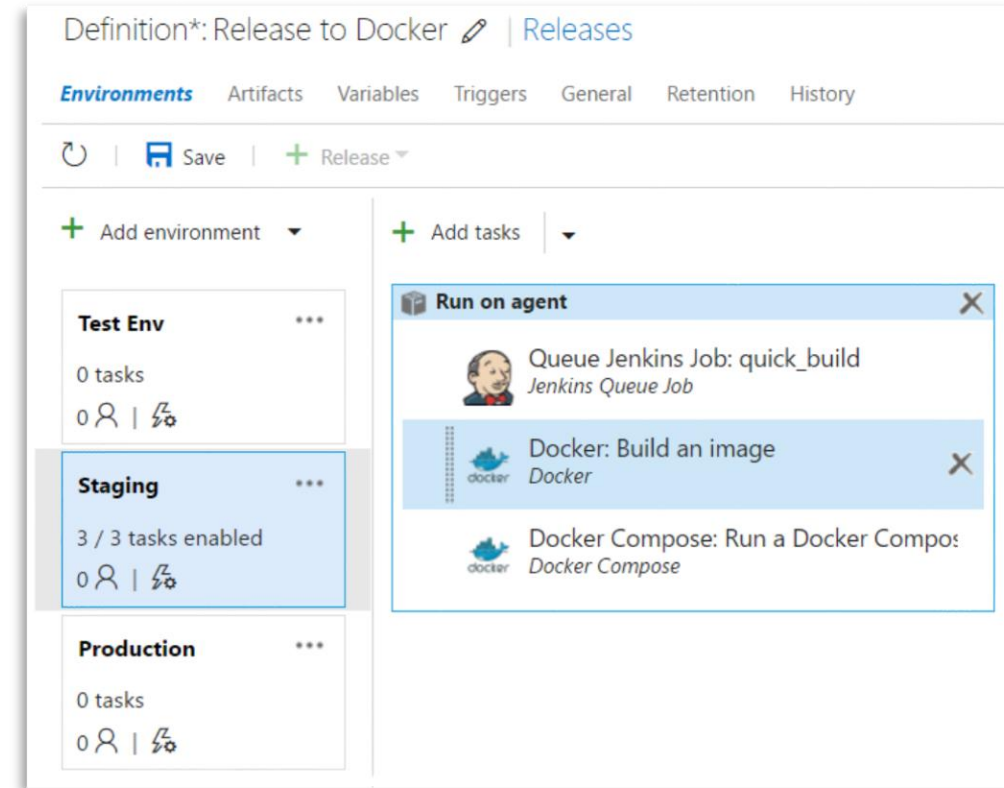


Service Fabric



Visual Studio Team Services

- Define a container based CI/CD pipeline
- Build and release automation
 - Build, run and manage images & containers
 - Hook into Docker Compose
 - Deploy natively to Azure Container Registry & Azure Container Service
- Combine Microsoft and Open Source tooling
- Utilise infrastructure as code automation with ARM templates



Azure Marketplace - Partner Offerings



Docker EE



Jelastic



Docker on Ubuntu



CoreOS



Helion Stackato



Pivotal Cloud Foundry



Aqua

Docker VM Extension

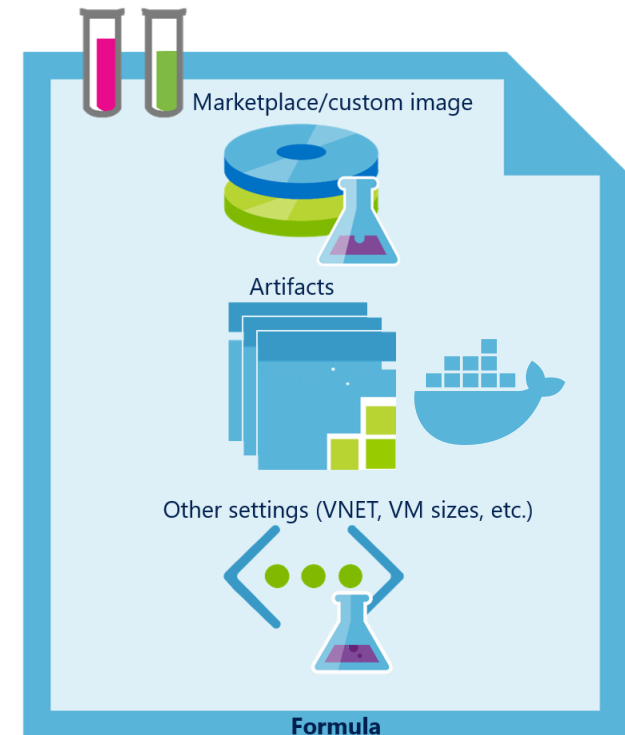
- Deploy Docker Engine + containers into any Linux VM
- The Docker VM Extension for use with template-based deployments via ARM (Azure Resource Manager)
- Supports Docker Compose declarations
- Role-based access, diagnostics, and post-deployment configuration.



```
{
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "name": "InstallAppServer",
  "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "DockerExtension",
    "typeHandlerVersion": "1.0",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "compose": {
        "appserver": {
          "image": "bencoleman/tomcat",
          "ports": [ "8080:8080" ],
          "volumes": [ "/myapps:/var/lib/tomcat7/webapps" ]
        }
      }
    }
  }
}
```

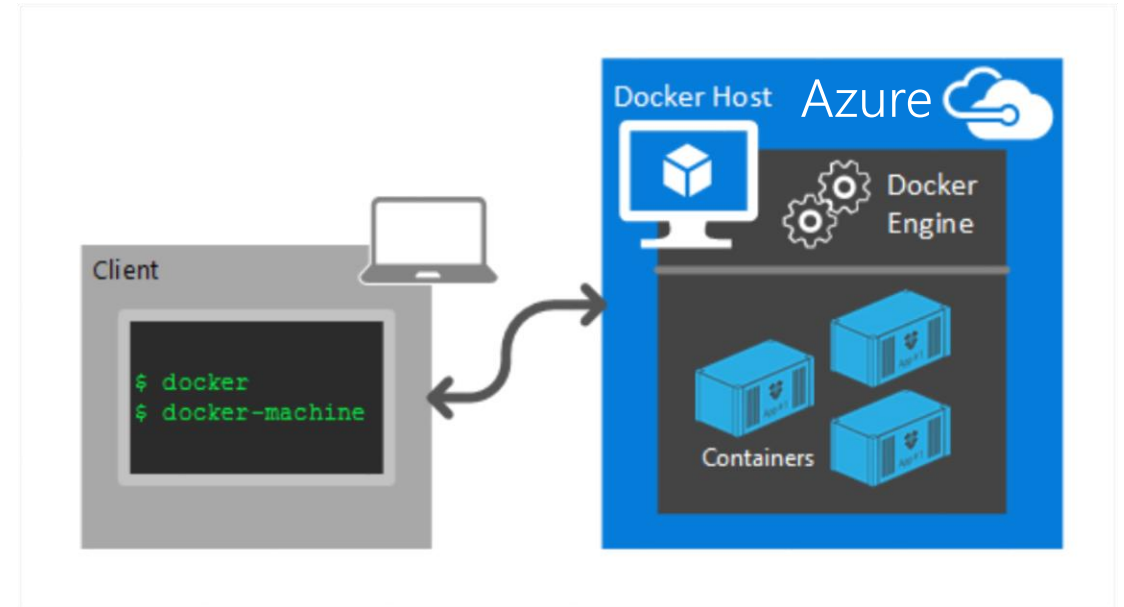
Azure DevTest Labs

- Self service for developers & testers - quickly provision dev & test environments
- Control spend with policies and auto-shutdown
- Build custom formulas from base images, artifact repositories & extensions
- Create Docker environments on standard Azure services, templates, VMs, Container Instance etc



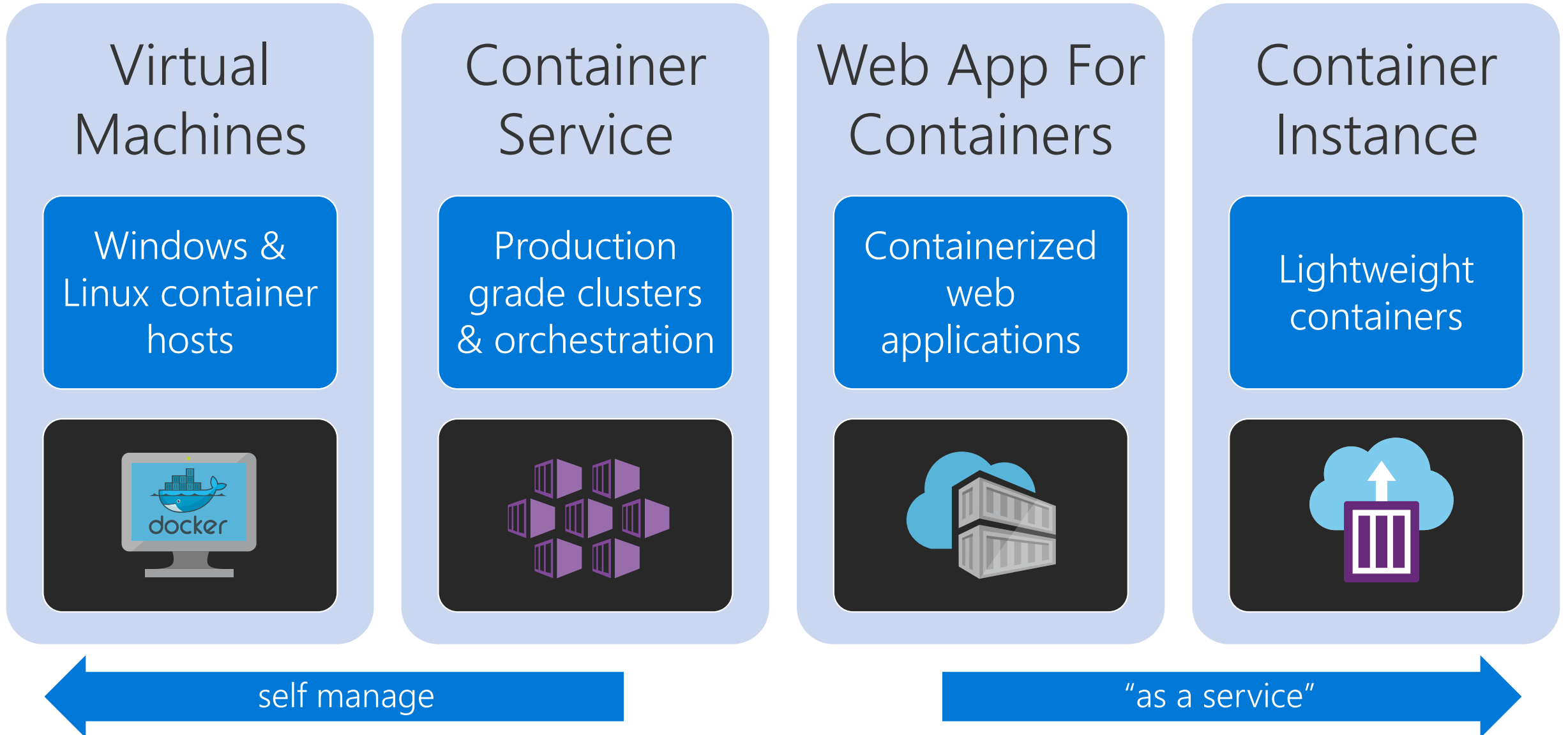
Docker Machine

- Docker Machine driver for Azure
- Build secure Linux Docker hosts directly in Azure
- Use native Docker CLI and client tools
- Remote management without Azure portal or APIs



```
$ docker-machine create -driver azure docker01
$ eval "$(docker-machine env docker01)"
$ docker run -d -p 8080:8080 tomcat
```


Azure Container Options - Summary

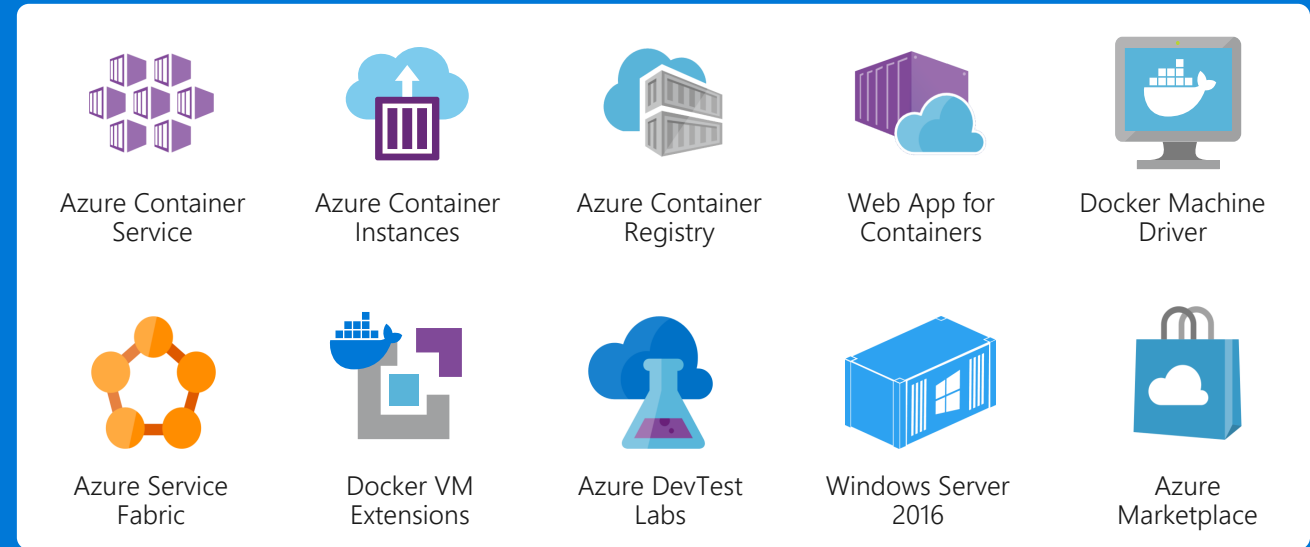


Summary

Breadth of options

- Bring your own
- Containers as 1st class citizens
- IaaS or PaaS
- Use open source and standard tools & APIs

CONTAINERS



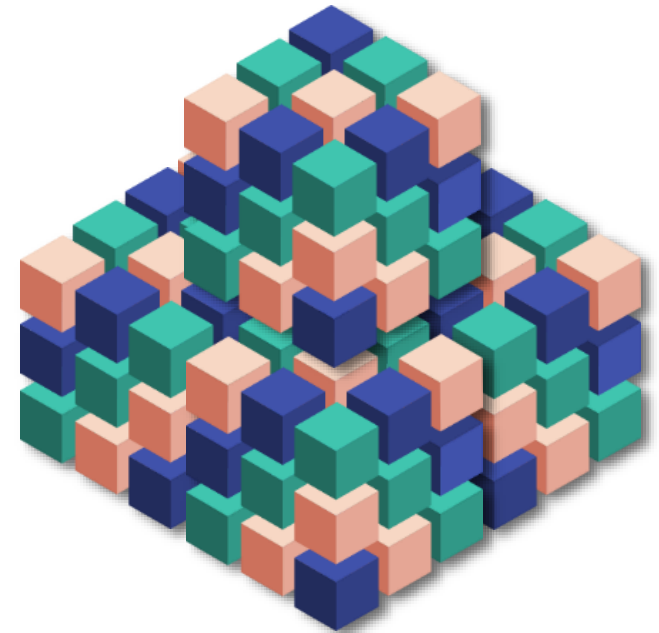
EVERYWHERE

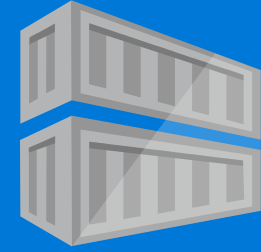
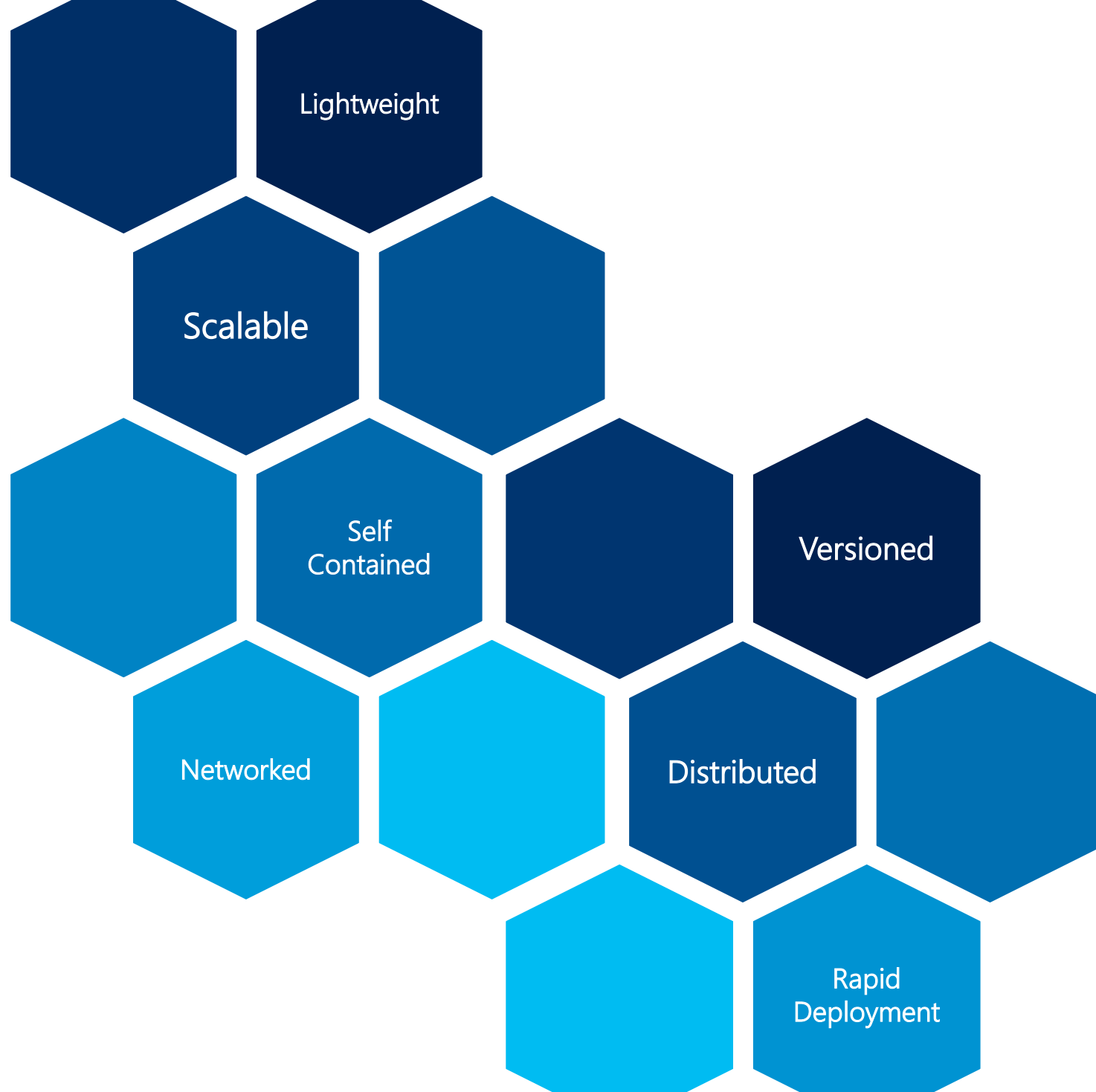


MICROSERVICES IS AN ARCHITECTURAL DESIGN PATTERN

Loosely coupled collection of small, autonomous services.

Each service is **self-contained** and should implement a **single** business capability.





Containers

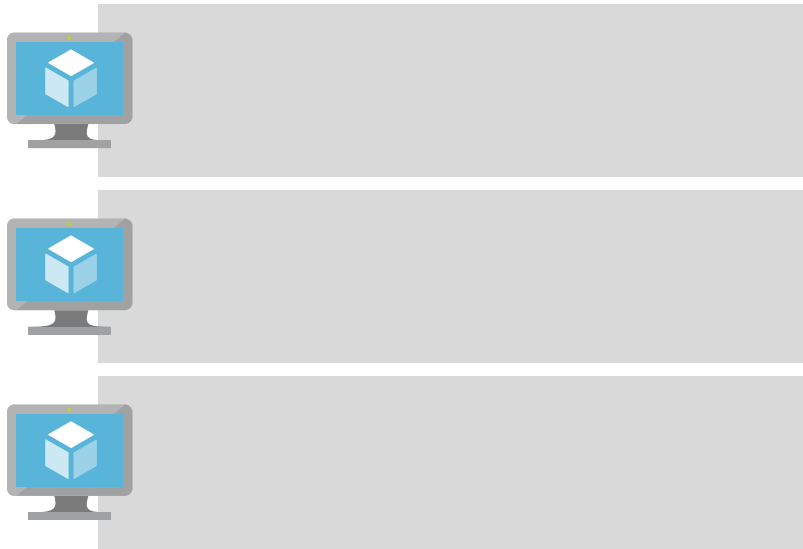
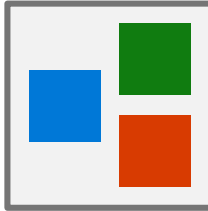
Natural Fit for
Microservices

Architecture and Deployment

Traditional Application

- Has its functionality within a few processes that are componentized with layers and libraries.
- Scales by deploying the whole app on multiple servers or VMs

App 1

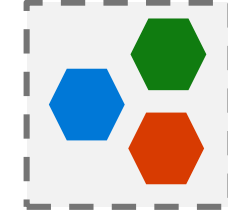


- Course grained scaling
- Deploy entire app stack each time
- Difficult resource optimization

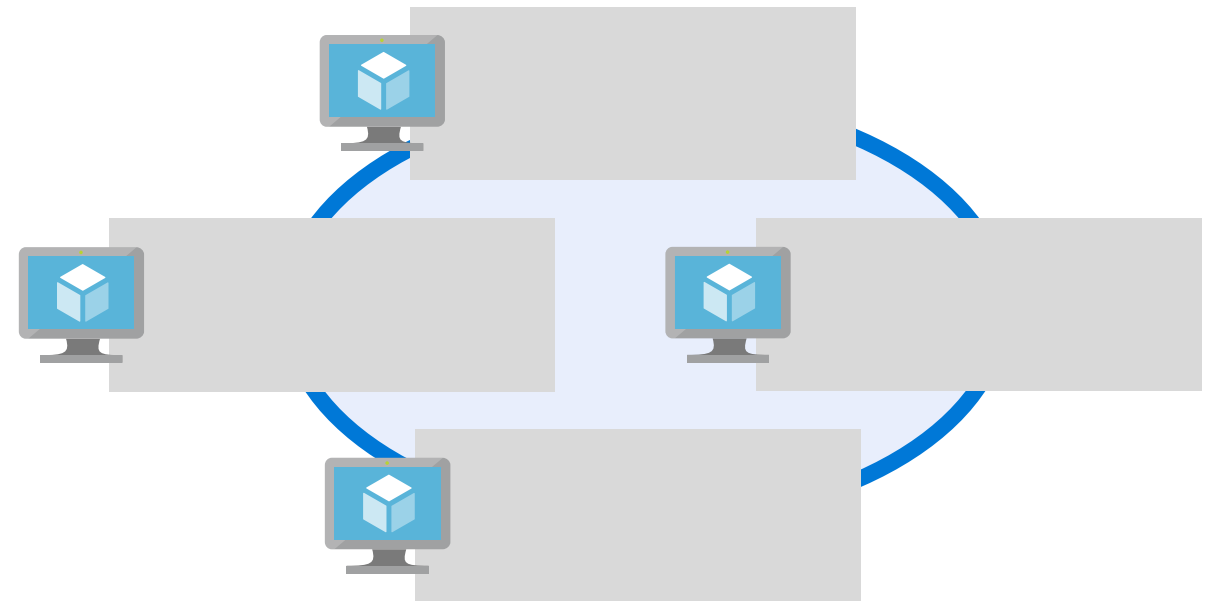
Microservices Application

- Application functionality segregated into separate smaller services.
- Scaled by deploying services independently with multiple instances across VM clusters

App 1



App 2



- Fine grained scaling
- Deploy individual services as needed

Example Microservices Architecture



Azure Container Service - Kubernetes

