Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

ROCK YOUR CODE TOUR 2017 | CHICAGO

Microservices with Azure Container Service & Service Fabric

Vishwas Lele
@vlele
Applied Information Sciences

Agenda

- MicroServices and friends
  - Containers and Continuous Deployment (CD)
- Reference Application
- Azure Container Service
- Azure Service Fabric
- Summary

Microservices

- Single responsibility principle == strong cohesion and loosely coupled
- Failure of one microservice does not cascade to other parts
- Autonomous
- Scaled independently
- Deployed and updated independently
- Technology, language or platform agnostic
- Composable

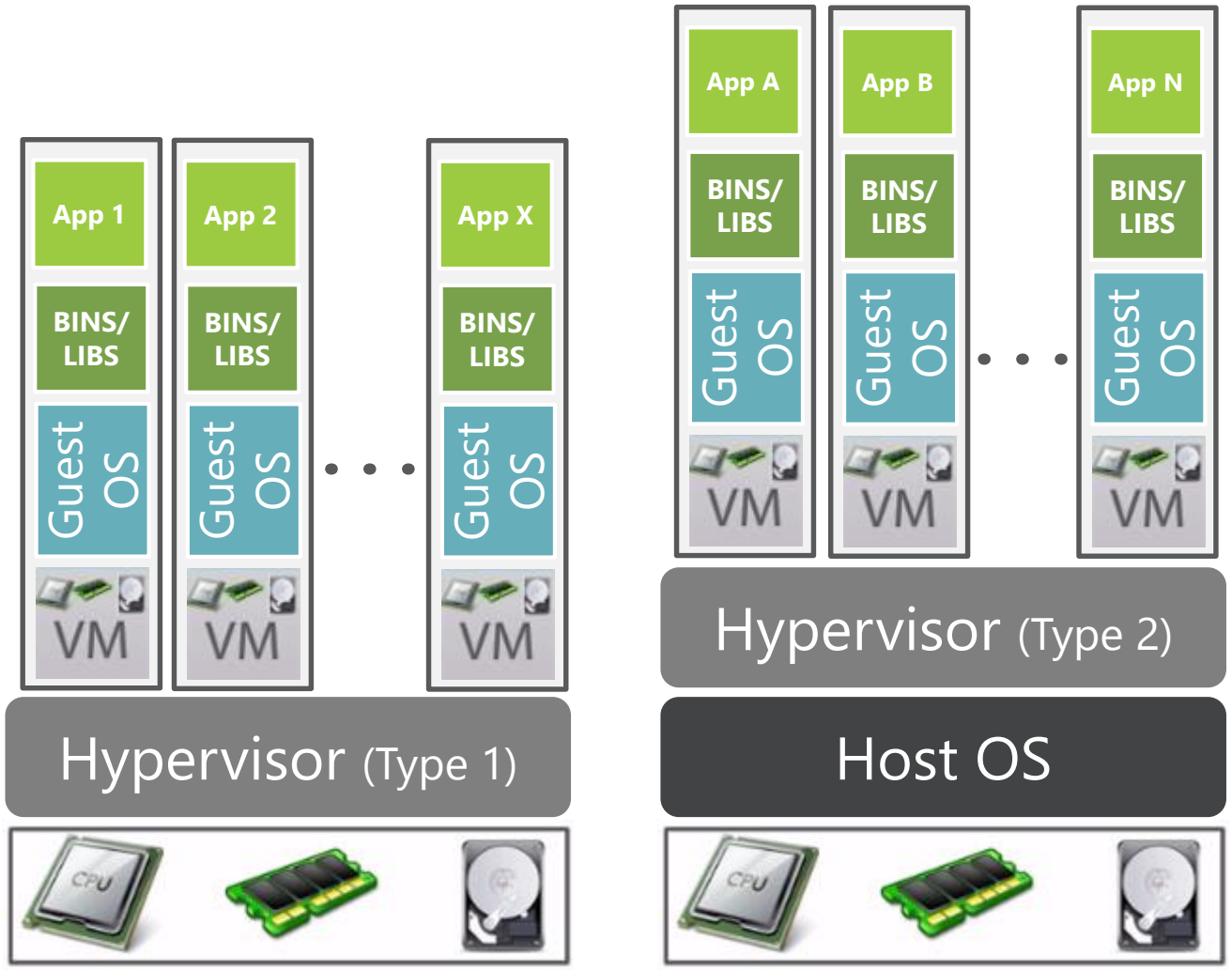
Containers

- Container - native OS construct light weight isolation
- Linux Container
  - Namespaces provided isolation
    - Scope user database, processes and even IP address
  - cgroups
    - Resource governors
  - Union File Systems
    - Mount several file systems at the same time
    - Each container can get its own read-write view

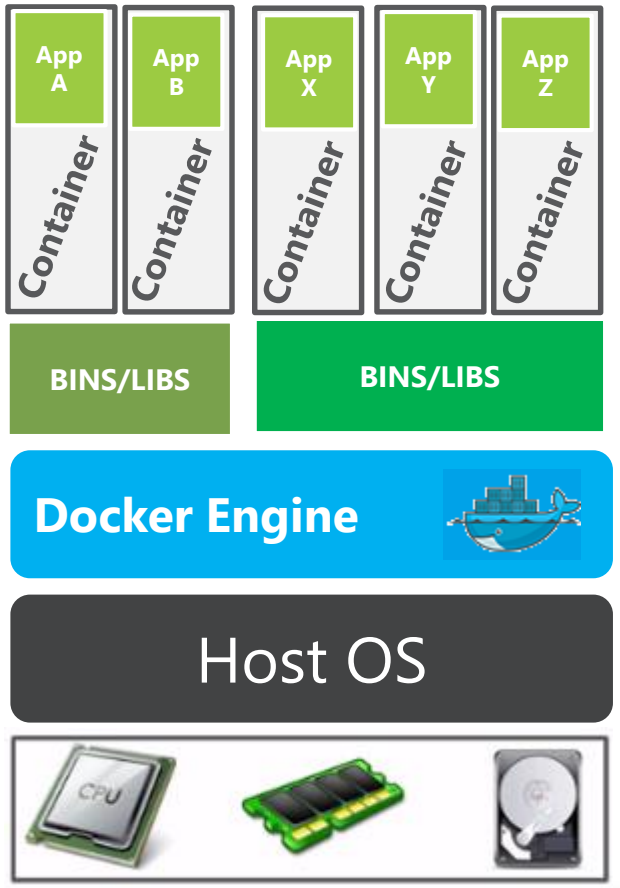
Docker

- Docker (name of the company and framework)
- Harnessed the OS container features to define "Docker Container"
  - Unified API and tooling to package applications
  - Holds everything needed to run an application
  - Can be started, stopped or moved
  - Based on a concept of image
    - In turn based on a collection of images made up of OS and standard components such as web server
- Docker Registries
  - Public and private

Virtual Machine Versus Docker Container



Virtualization



Containerization

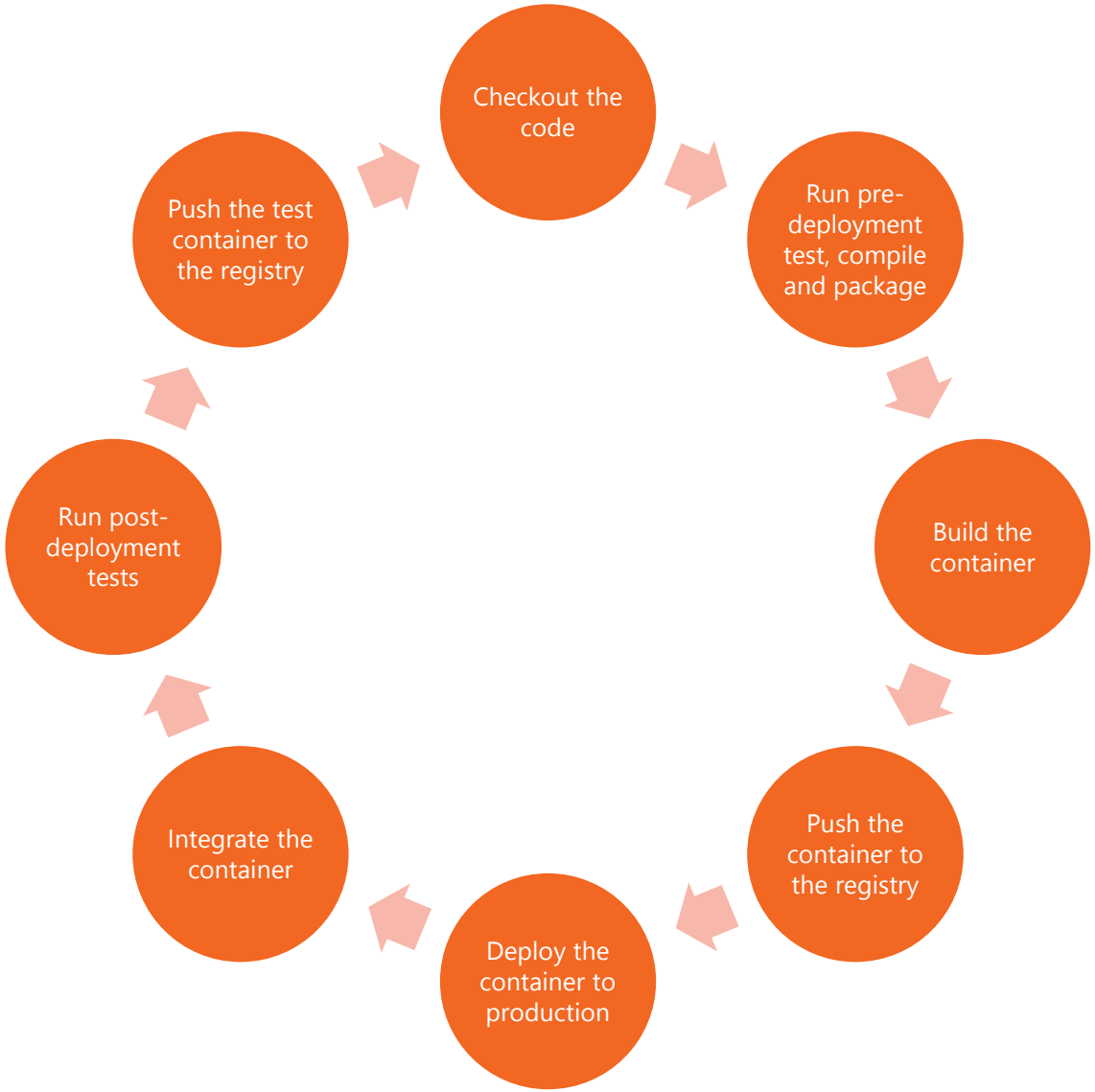
But Containers Aren't Enough!

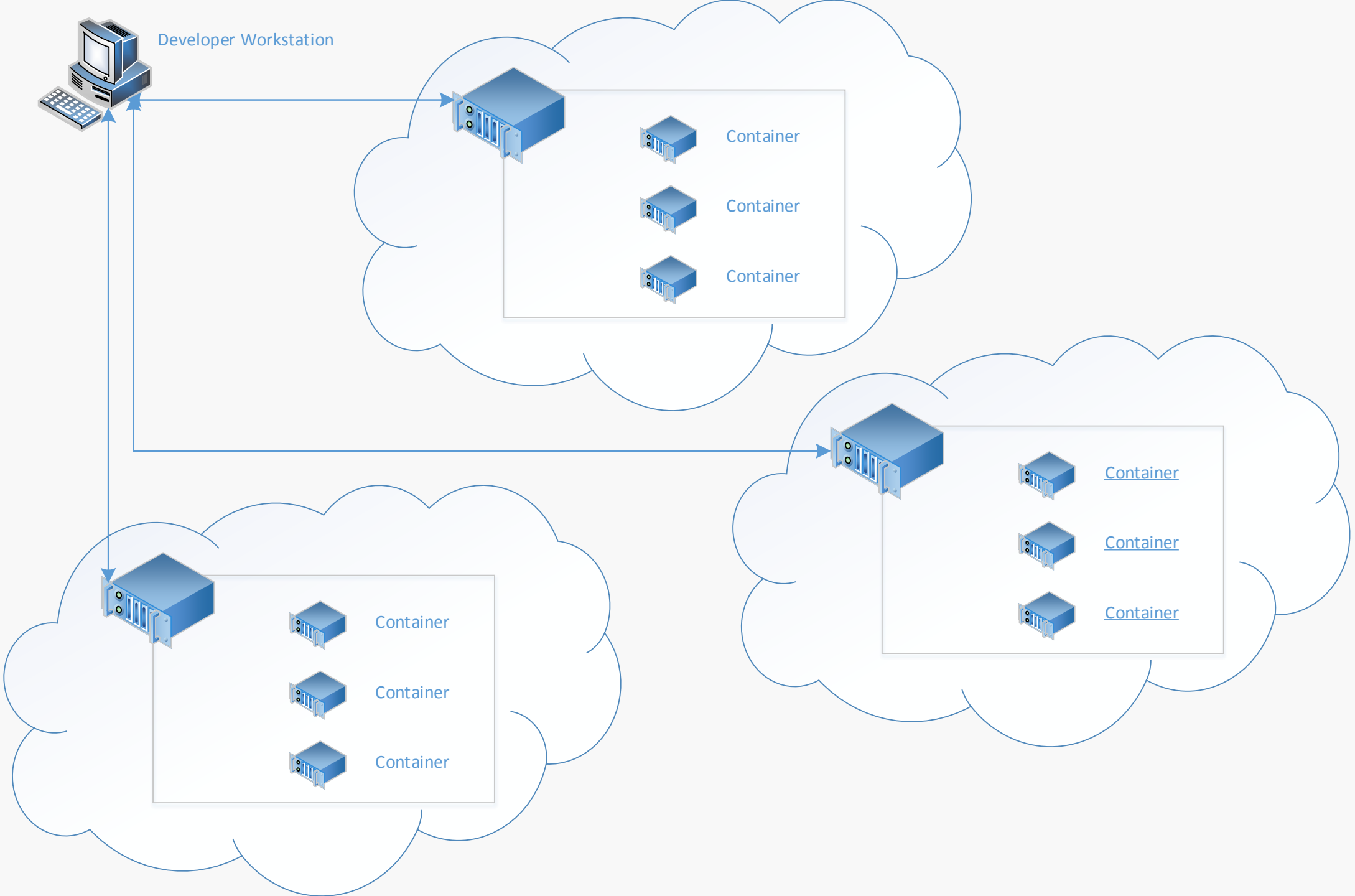
Running containerized "micro-services" in production requires ***much more*** than just Docker.

It requires a "Platform" that can do the following:

- Building and pushing Docker images to an image repository
- Pulling images, provisioning and scheduling containers
- Discovering and binding to services running as containers
- Containers discovering and binding to other containers
- Operating and managing services in containers

Docker Deployment Pipeline




Developer Workstation
Container
Container
Container
Container
Container
Container
Container
Container
Container

Tools

- Vagrant
  - Command line tool to create virtual machines though a hypervisor
- Ansible
- Configuration Management, Provisioning and Deployment (agentless)
- Jenkins
  - CI / CD

Demo

Reference App

Docker Swarm

Discovery Backends

Maintains a list of nodes in the cluster

Manager

Schedules and manages containers running on nodes

Nodes

Responsible for running containers

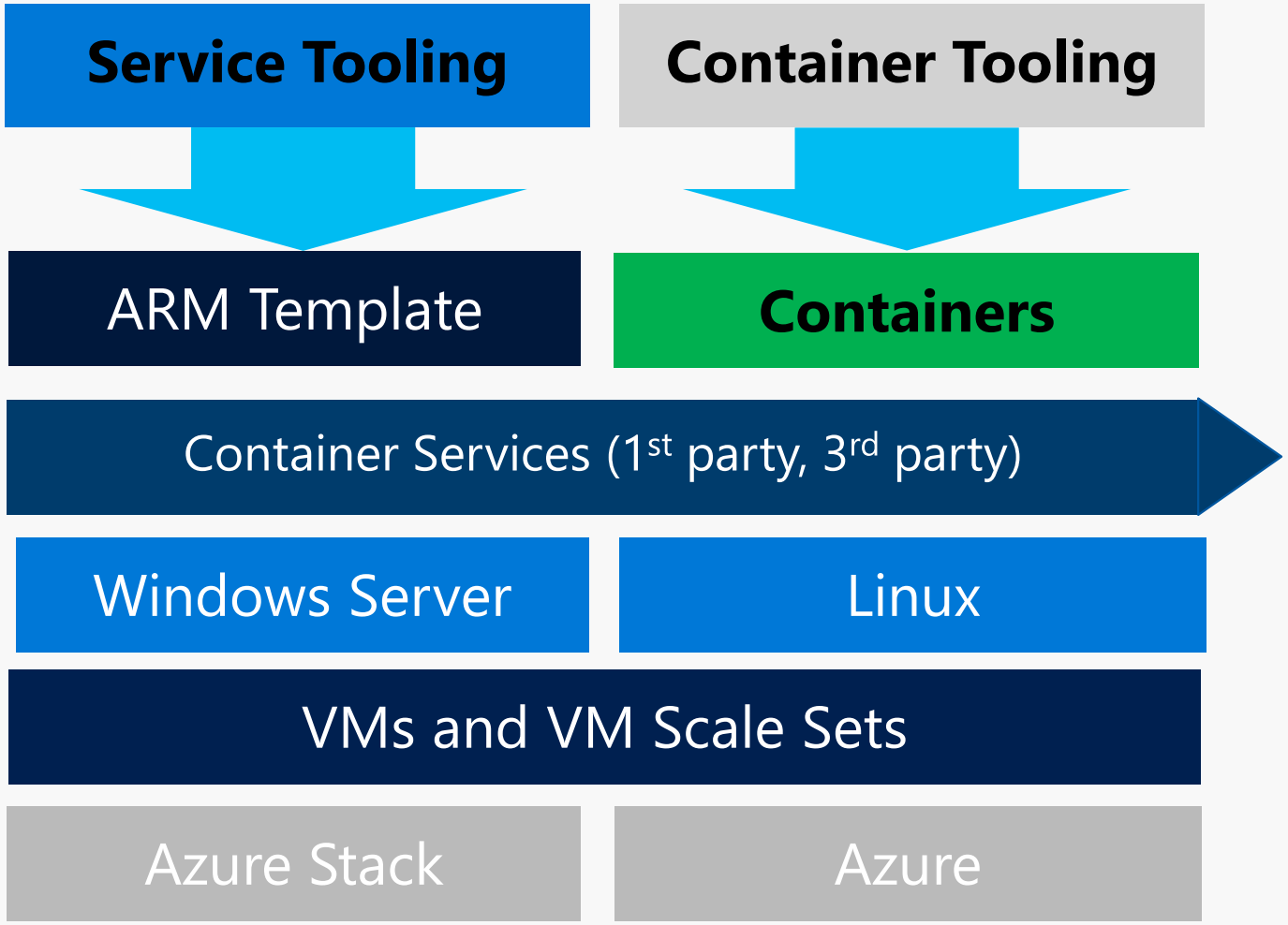
Strategies

Algorithm used for container scheduling; spread, binpack, random

Filters

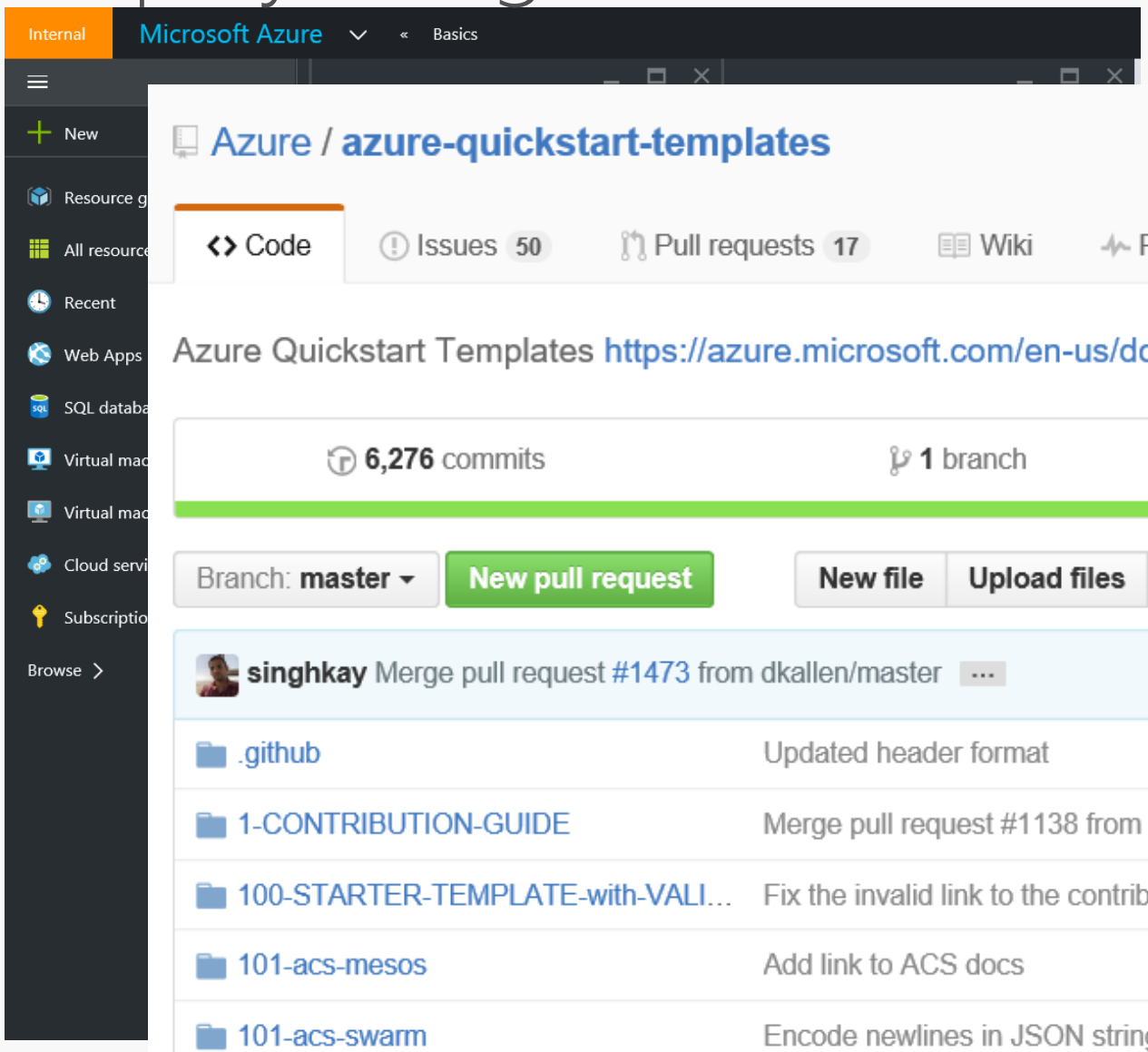
Used to control container placement on nodes

Azure Container Services

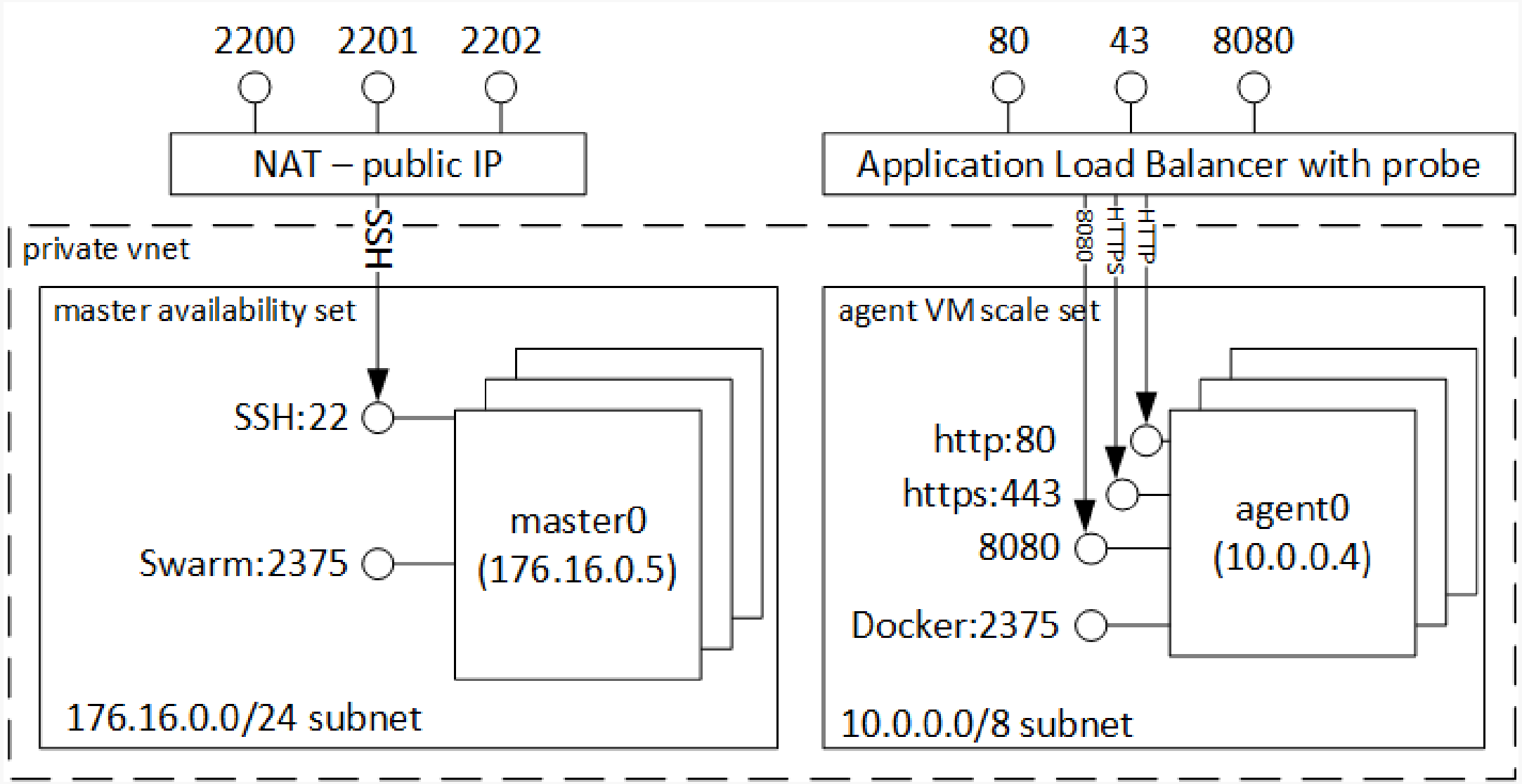


Layer	Supported Technologies
Configuration as Code	ARM, Dockerfile, Docker Compose, Marathon.json
Host cluster management	VM Scale Sets
Container orchestration	Docker Swarm, Chronos, Marathon, Apache Mesos
Monitoring	OMS, Statsd

Deploy using Portal or ARM



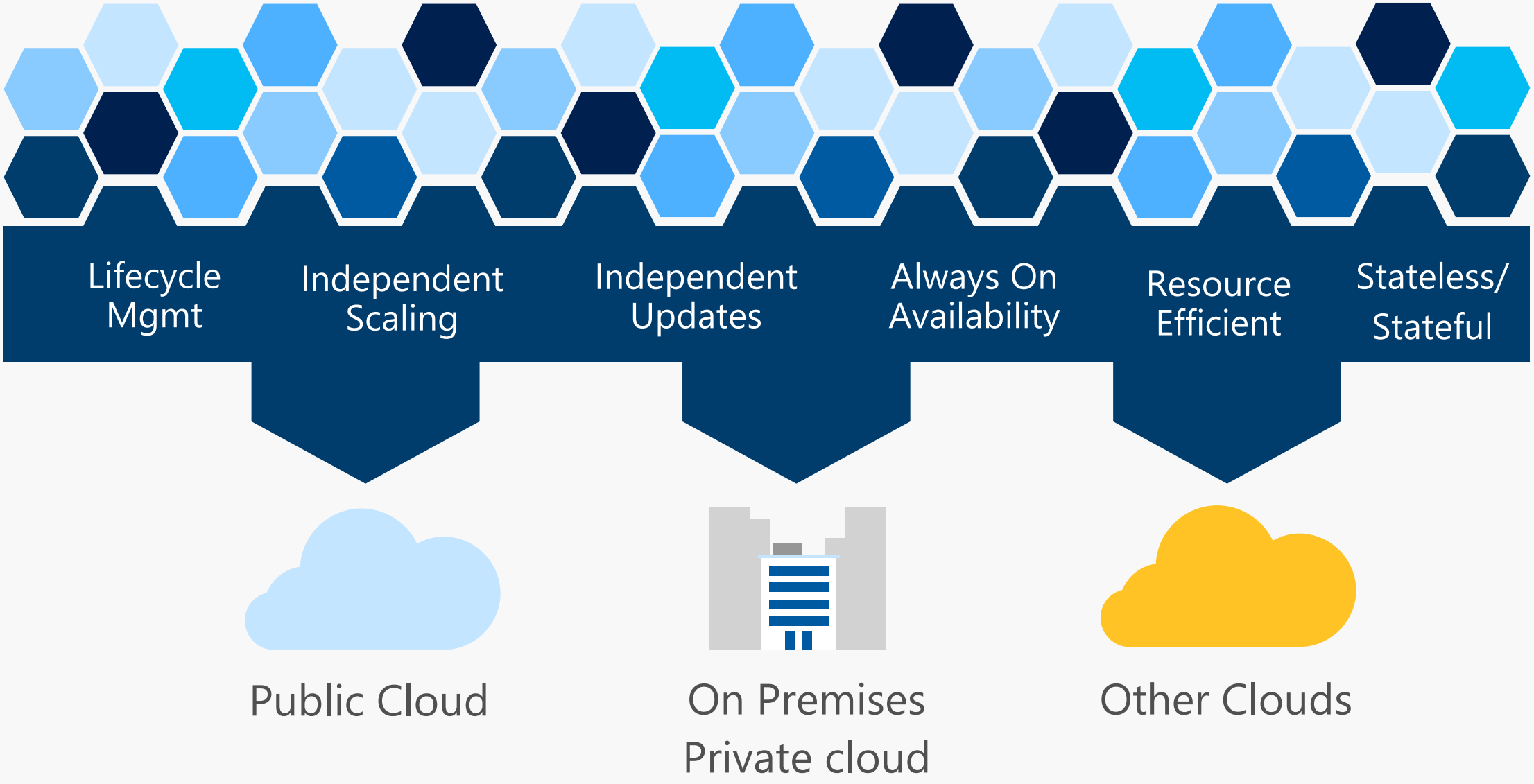
Azure Container Service Architecture (Swarm)



Demo

Azure Container Service

Azure Service Fabric

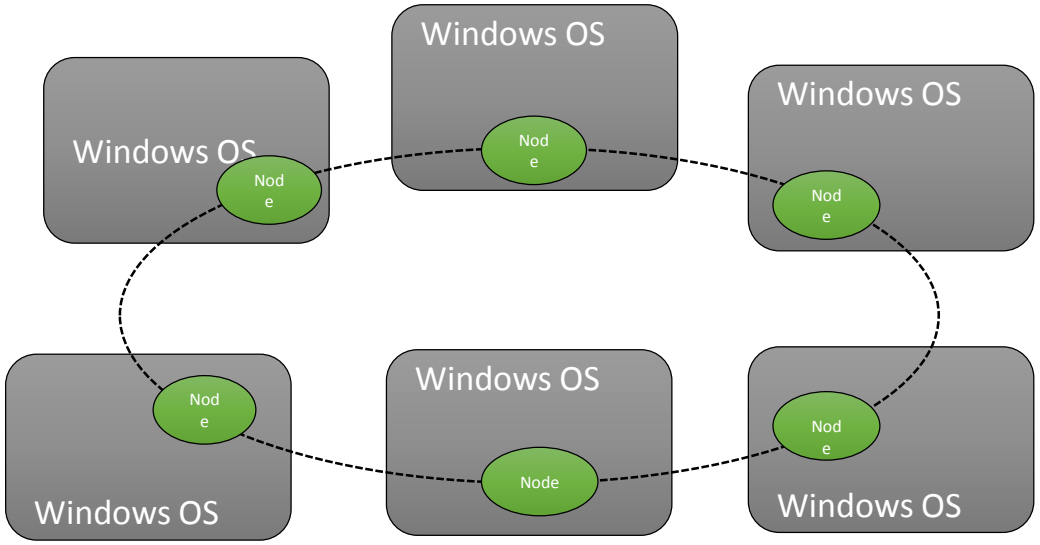


Cluster

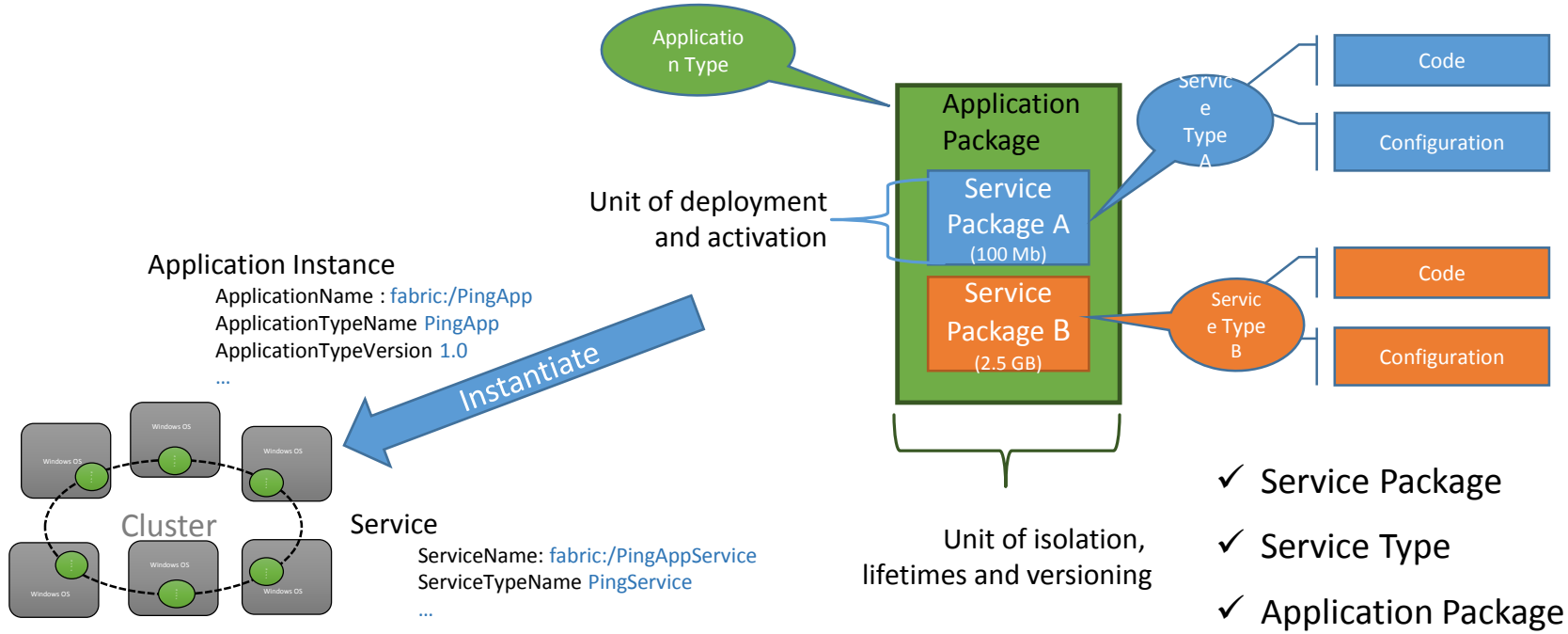
- A set of virtual or physical machine resources that are stitched together to form a highly available, scalable, and reliable infrastructure for applications and services.
- Cluster can scale to 1000s of machines
- Cluster can scale up or down
- Cluster is an infrastructure independent abstraction layer (Runs on Windows Server & Azure)

Node

- Addressable unit in the cluster that you associate properties like placement, fault domains etc
- 1:1 Node to OS instance mandated for production scenarios



Concept of an Application and Service



- ✓ Application Instance
- ✓ Service instance

- ✓ Service Package
- ✓ Service Type
- ✓ Application Package
- ✓ Application Type

Reliable Actor API

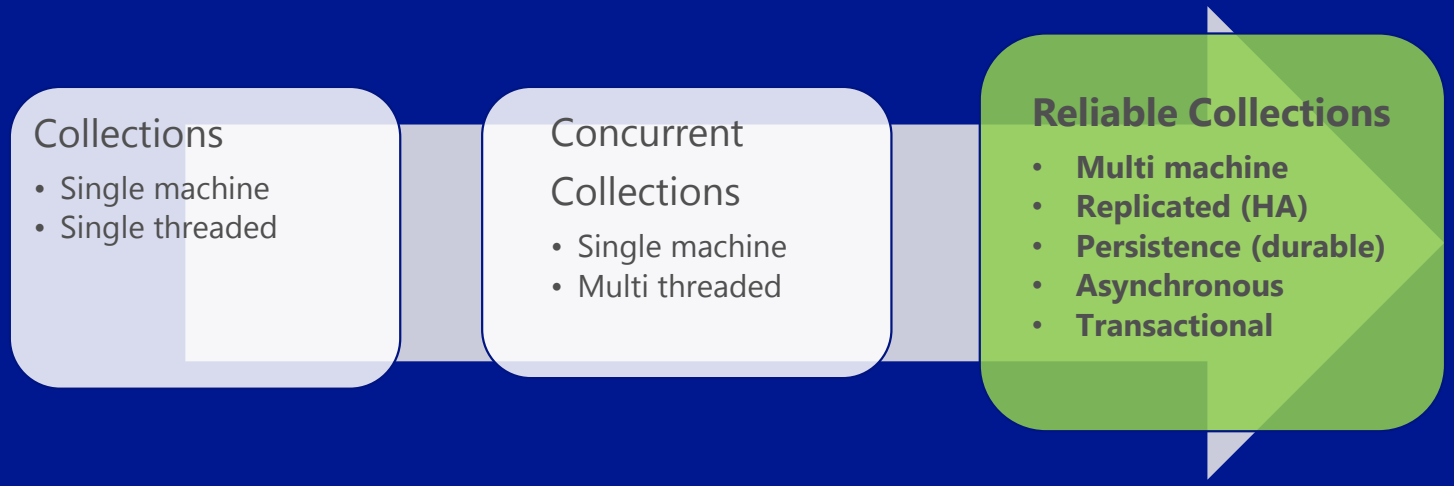
- Build reliable stateless and stateful objects with a virtual Actor Programming Model
- Suitable for applications with multiple independent units of state and compute
- Automatic state management and turn based concurrency (single threaded execution)

Reliable Services API

- Build stateless services using existing technologies such as ASP.NET
- Build stateful services using reliable collections
- Manage the concurrency and granularity of state changes using transactions
- Communicate with services using the technology of your choice (e.g WebAPI, WCF)

Reliable Collections

- Reliable collections make it easy to build stateful services.
- Evolution of the .NET collections for the cloud



Demo

Azure Service Fabric

When to choose

Option	Ability to customize the guest OS	Management Overhead (patching, capacity planning, adding nodes to the cluster)	DevOps (Integration with CI / CD) / Time to publish	Orchestration – Fine grained cluster management	Refactoring Needed	Compliance
AF				N/A		
ASF						
ASE						
ACS						
VMSS				N/A		

https://blog.appliedis.com/2017/06/02/azure-paas-options-when-to-use-what/