Microservices

Lesson Objectives

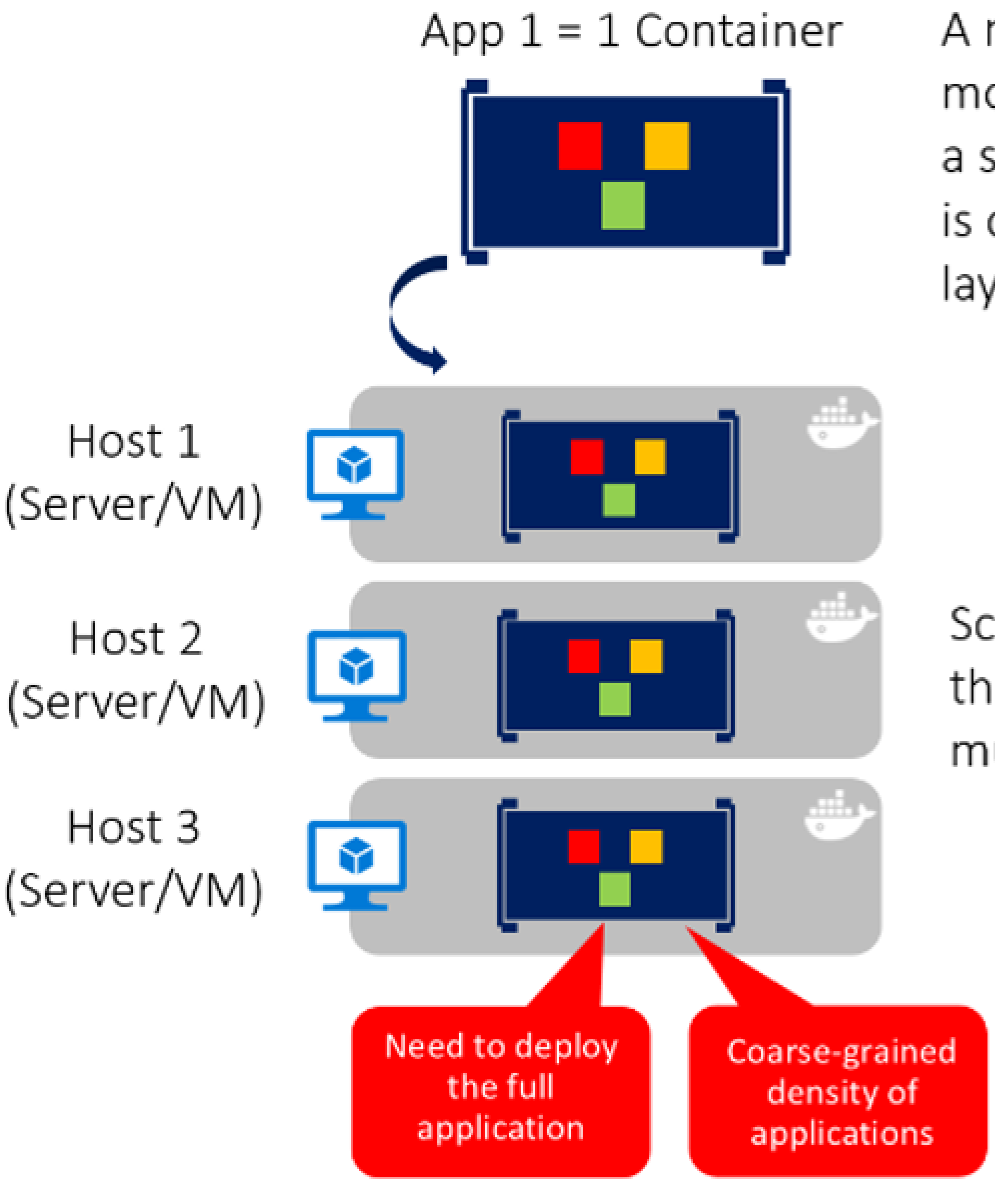
In this lesson, you will learn:

- Software required to create and deploy Micro Services
- Monolithic vs Microservices approach
- Introduction to Dot Net Micro Services
- Advantages of Micro Services
- Creating a Micro Service using VS2017

Software Required

- Windows 10 Pro Version
- Dockers [We can download from : https://www.docker.com/]. Dockers Community Edition is FREE.
- Visual Studio 2017 [which has built-in support for Docker]

Monolithic Application


App 1 = 1 Container
Host 1
(Server/VM)
Host 2
(Server/VM)
Host 3
(Server/VM)
Need to deploy the full application
Coarse-grained density of applications

A monolithic application has most of its functionality within a single process/container that is componentized with internal layers or libraries.

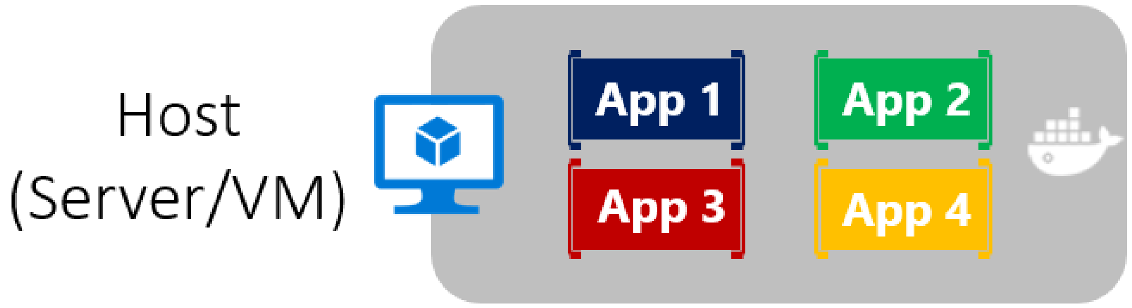
Scales out by cloning the app/container on multiple servers/VMs

Container Principle and Monolithic Approach

“A container does one thing, and does it in one process”

- A monolithic pattern / approach might conflict with the container principle.
- As the monolithic application grows, scaling it will be a problem.

Host running multiple apps/containers



Monolithic Approach :
Host running multiple apps, each app running as a container.

Microservices [Introduction]

Microservices is an approach where we develop smaller services and each service runs in its own process.

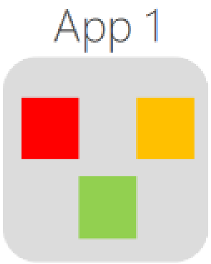
Benefits of Microservices:

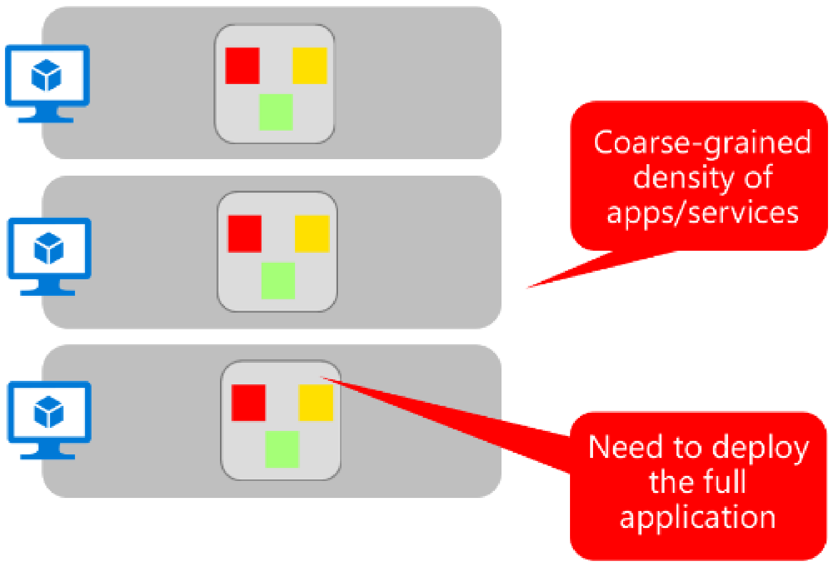
- ✓ Microservices are easier to develop, deploy and debug.
- ✓ Monitoring and health checks of the services and infrastructure.
- ✓ Microservices can be scaled easily and can be re-used across different applications / services.
- ✓ Microservices approach will follow the container principle
- ✓ Microservices work well with Containers like dockers.
- ✓ The biggest benefit is – Microservices are independent to each other. i.e., if one micro service goes down, there is a little risk of full application shutdown
- ✓ DevOps and CI/CD practices and infrastructure

Monolithic vs Microservices

Monolithic deployment approach

- A traditional application has most of its functionality within a few processes that are componentized with layers and libraries.
- Scales by cloning the app on multiple servers/VMs



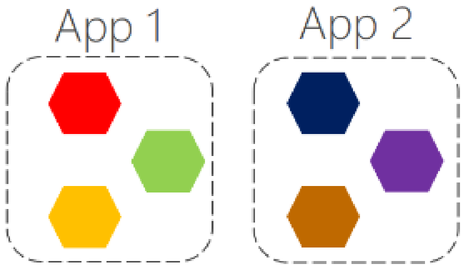


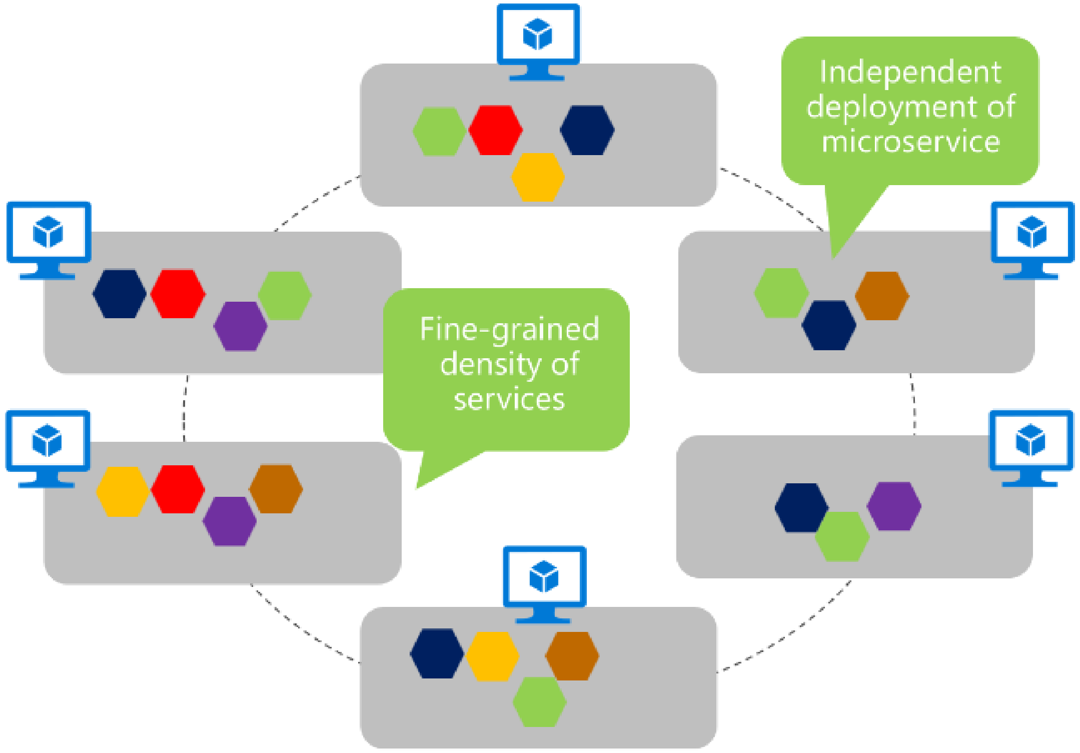
Coarse-grained density of apps/services

Need to deploy the full application

Microservices application approach

- A microservice application segregates functionality into separate smaller services.
- Scales out by **deploying each service independently** with multiple instances across servers/VMs





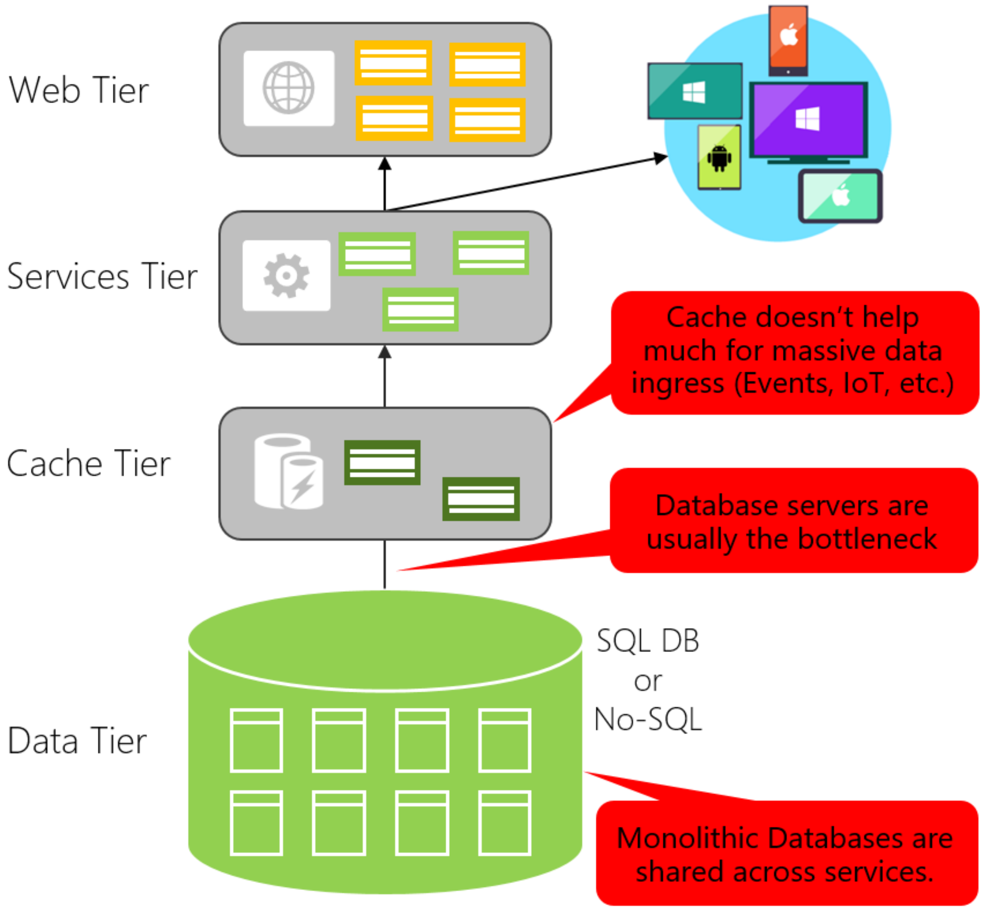
Independent deployment of microservice

Fine-grained density of services

Data sovereignty per microservice

Data in Traditional approach

- Single monolithic database
- Tiers of specific technologies



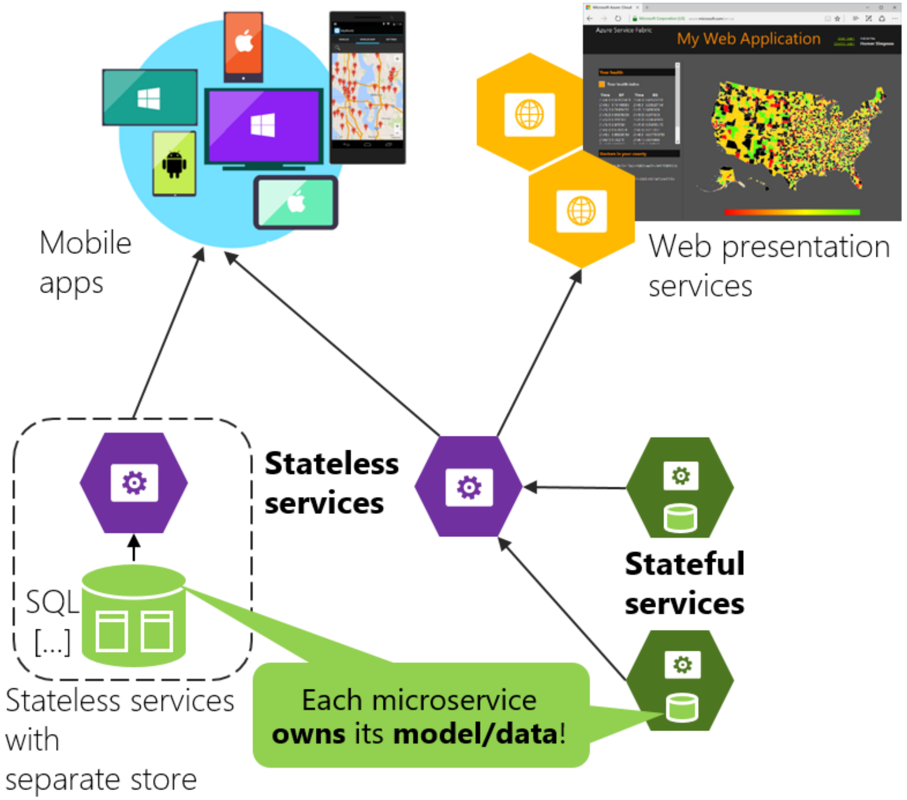
Cache doesn't help much for massive data ingress (Events, IoT, etc.)

Database servers are usually the bottleneck

Monolithic Databases are shared across services.

Data in Microservices approach

- Graph of interconnected microservices
- State typically scoped to the microservice
- Remote Storage for cold data



Each microservice **owns its model/data**!

Demo

Demo on how to create and consume a micro service using Visual Studio 2017

Summary

In this lesson, you have learnt about:

- Microservices and its benefits over Monolithic approach
  - Introduction to Microservices
  - Monolithic vs Microservices deployment approach
  - Creating a simple microservice
  - Data in traditional monolithic vs microservice approach

References

https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/container-docker-introduction/

https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/multi-container-microservice-net-applications/

People matter, results count.

Rightshore® is a trademark belonging to Capgemini.

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com

This message is intended only for the person to whom it is addressed. If you are not the intended recipient, you are not authorized to read, print, retain, copy, disseminate, distribute, or use this message or any part thereof. If you receive this message in error, please notify the sender immediately and delete all copies of this message.