Service Fabric & Containers

David Gristwood
david.gristwood@microsoft.com
@ScroffTheBad

Microsoft

Modernisation is a spectrum

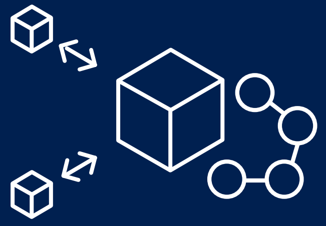
1

Traditional app

2

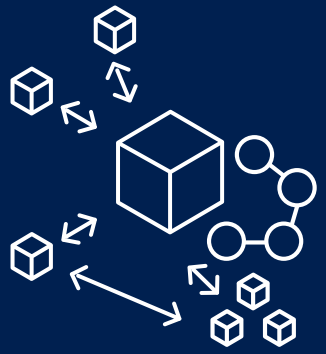
Monolith Hosted as guest executable or container

3



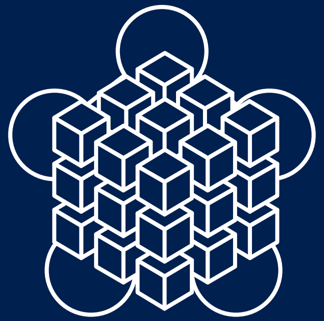
Existing Monolith + new microservices

4



Parts of existing monolith extracted

5



New or transformed microservices app

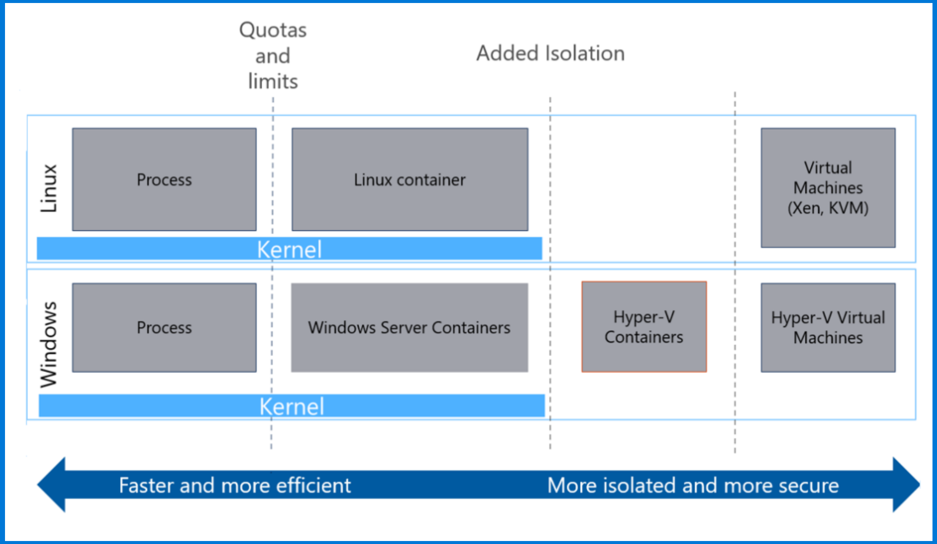
What does it mean for .NET dveelopers?

Service Fabric programming models:

- Low level API for stateless and stateful services
- Guest executables
- Containers

Service Fabric supports containers on Windows Server 2016

- Along with support for Hyper-V isolation mode



Ideal targets:

- ASP.NET MVC (when App Services not suitable)
- Traditional 3-tier apps
- Azure Cloud Services (Classic)

Either:

- Package into container images from pre-created IIS images
- Move source code to ASP.NET Core with lightweight Nano Server image

Note: Nano Server only available as container bases image as of version 1709

- Optimized for .NET Core applications
- Smaller than the Windows Server 2016 version
  - microsoft/aspnetcore nanoserver: **447 MB**
  - microsoft/aspnet windowsservercore: **13.5 GB**

Smilr and MVC

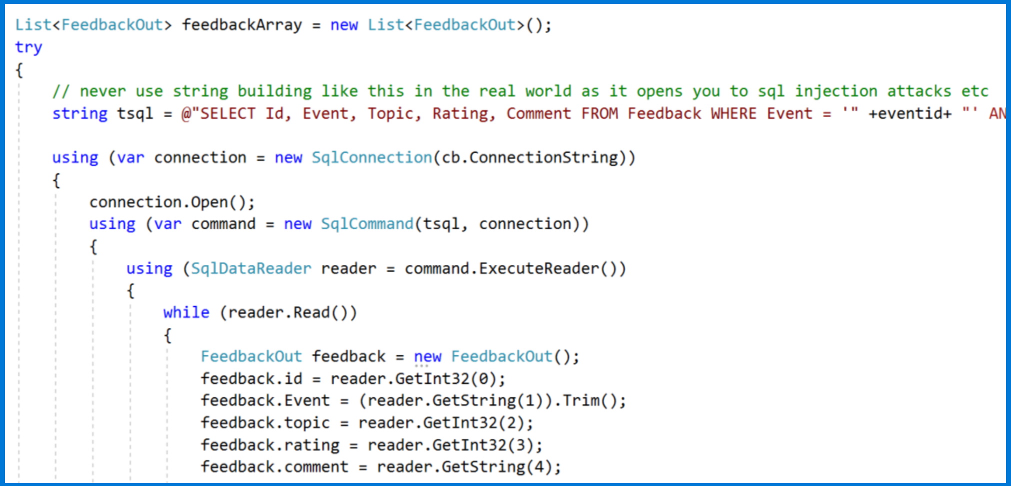
Classic ASP.NET app:

- C# MVC Web API
- ADO.NET to SQL Server

My changes:

- Recompiled C# to ASP.NET Core
    - Enabled use of smaller container images
- Moved data Azure SQL Database
    - No changes needed other than connections string

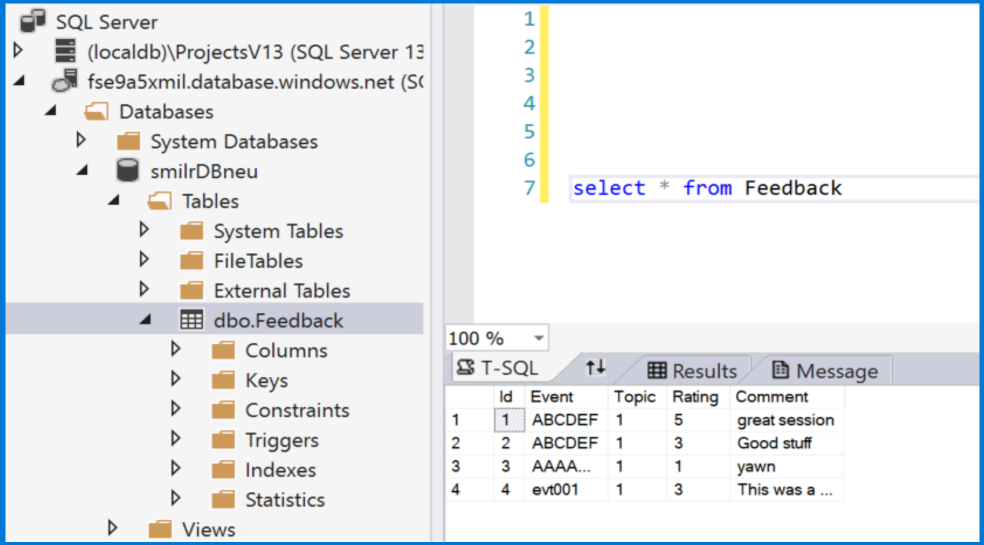
HTTP



```
List<FeedbackOut> feedbackArray = new List<FeedbackOut>();
try
{
    // never use string building like this in the real world as it opens you to sql injection attacks etc
    string tsql = @"SELECT Id, Event, Topic, Rating, Comment FROM Feedback WHERE Event = '" +eventid+ "' AN

    using (var connection = new SqlConnection(cb.ConnectionString))
    {
        connection.Open();
        using (var command = new SqlCommand(tsql, connection))
        {
            using (SqlDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    FeedbackOut feedback = new FeedbackOut();
                    feedback.id = reader.GetInt32(0);
                    feedback.Event = (reader.GetString(1)).Trim();
                    feedback.topic = reader.GetInt32(2);
                    feedback.rating = reader.GetInt32(3);
                    feedback.comment = reader.GetString(4);
```

ADO



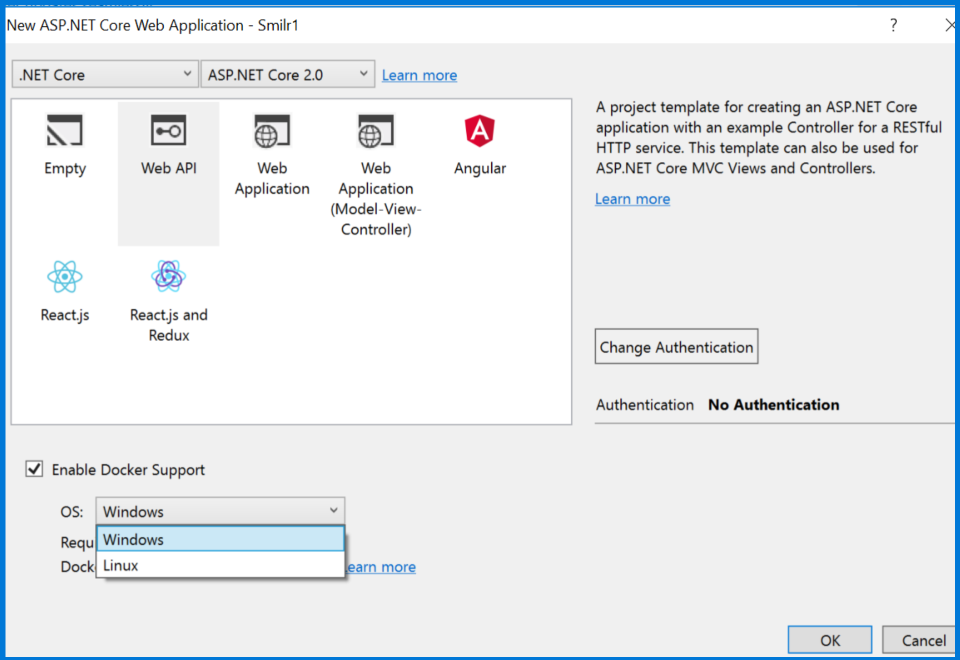
```
select * from Feedback
```

	Id	Event	Topic	Rating	Comment
1	1	ABCDEF	1	5	great session
2	2	ABCDEF	1	3	Good stuff
3	3	AAAA...	1	1	yawn
4	4	evt001	1	3	This was a ...

Visual Studio Support

Simplifies working with Docker

- Creates your Dockerfile as part of the project
- Uses containers for both build and deploy
  - microsoft/aspnetcore-build
  - microsoft/aspnetcore
- Pushes to Azure Container Registry



New ASP.NET Core Web Application - Smilr1

.NET Core | ASP.NET Core 2.0 | Learn more

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Learn more

Change Authentication

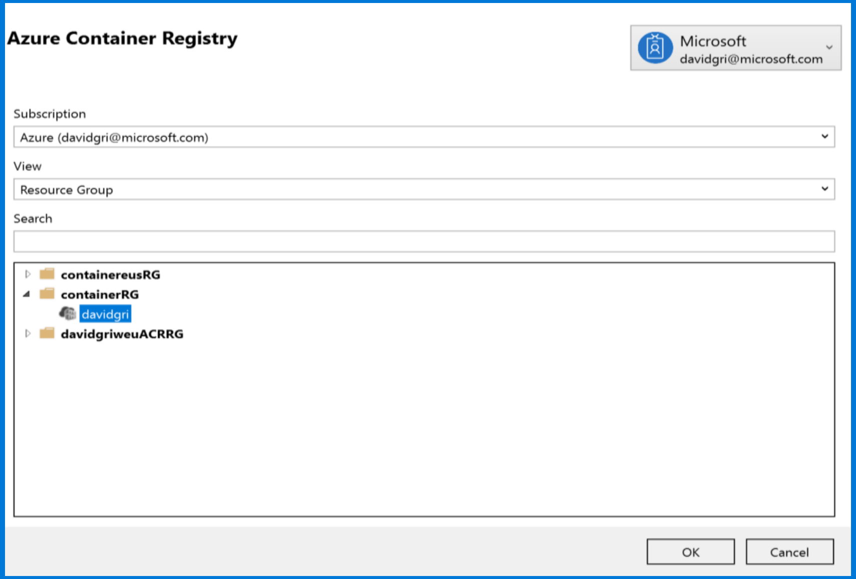
Authentication No Authentication

☑ Enable Docker Support

OS: Windows

Requ | Windows

Dock | Linux | earn more



Azure Container Registry

Microsoft davidgri@microsoft.com

Subscription: Azure (davidgri@microsoft.com)

View: Resource Group

Search:

- containereusRG
- containerRG
  - davidgri
- davidgriweuACRRG

Microsoft