Module 14

Hosting and Deployment

Module Overview

- On-Premises Hosting and Deployment
- Deployment to Microsoft Azure
- Microsoft Azure Fundamentals

Lesson 1: On-Premises Hosting and Deployment

- Web Servers
- Hosting ASP.NET Core Application
- Deploying to IIS
- File Providers

Web Servers

- Setting up the server is an important part of an ASP.NET Core MVC application and requires you to make important decisions

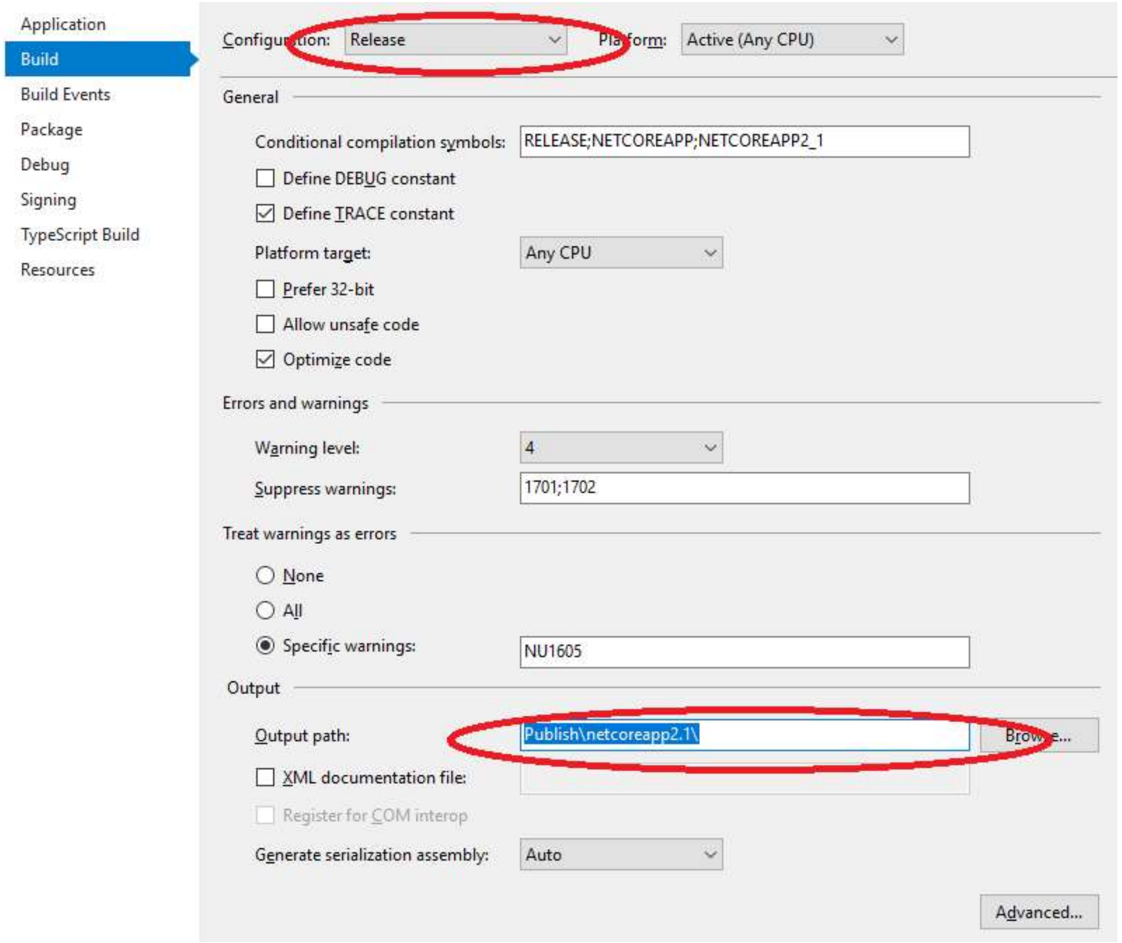
- By choosing Kestrel, you get:
  - A lightweight server that is fast
  - The ability to use reverse proxy
  - Cross-platform support

- By choosing HTTP.sys you get:
  - A robust framework with many prebuilt options
  - Windows-based authentication
  - Direct file transfer from the server

Hosting ASP.NET Core Application

- For hosting ASP.NET Core MVC applications, you need to decide:
- What configurations are needed as part of the publishing process
- The server infrastructure to use to host your application
- Whether to use a reverse proxy as part of your application
- Whether load balancing functionality is required

Setting Output Path for a Project



Configuration: Release Platform: Active (Any CPU)

General

Conditional compilation symbols: RELEASE;NETCOREAPP;NETCOREAPP2_1

□ Define DEBUG constant

☑ Define TRACE constant

Platform target: Any CPU

□ Prefer 32-bit

□ Allow unsafe code

☑ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings: 1701;1702

Treat warnings as errors

○ None

○ All

● Specific warnings: NU1605

Output

Output path: Publish\netcoreapp2.1\

□ XML documentation file:

□ Register for COM interop

Generate serialization assembly: Auto

Deploying to IIS

- Deploying an application requires a large investment at the start
- It becomes considerably easier on updates
- The first set up requires several steps:
- Update the ASP.NET Core application to work with IIS
- Set up IIS
- Create the IIS Web Site
- Deploy the application
- After the first deployment you can directly perform the deployment

File Providers

- **File Providers** allow us to interact with actual files in the project structure:
- **PhysicalFileProvider** interacts with files that are physically present in the project structure
- **ManifestEmbeddedFileProvider** interacts with files which are embedded within the application itself, allowing for added security at the cost of being unchangeable at run time
- **CompositeFileProvider** allows us to combine two or more providers and use them all with a single interface

Lesson 2: Deployment to Microsoft Azure

- What is Microsoft Azure?
- Benefits of Hosting in Microsoft Azure
- Deploying Web Application on Microsoft Azure
- Demonstration: How to Deploy a Web Application to Microsoft Azure
- Azure Deployment Strategy
- Debugging a Microsoft Azure Application

What is Microsoft Azure?

Cloud Service that makes computing resources available on demand and over the internet

Infrastructure-as-a-Service

- Compute resources available on demand
- Resources include:
  - Virtual machines
  - Persistent disks
  - Networking
  - Containers
- Scale up and down as per demand
- Pay only for what you use

Platform-as-a-Service

- Complete environment (OS, webservers and other necessary software) available on demand
- Provides auto-scaling, maintenance and monitoring of systems
- Ease of deployment
- Suitable for micro-services based architecture

Benefits of Hosting in Microsoft Azure

Benefits of Azure:

- **Efficiency**
  With PaaS, deploying and scaling application is very easy. This leads to efficient use or resources.
- **Elasticity**
  Ability to scale up to thousands of machines
- **Security**
  Common compliances and certifications in place. Also provides DDoS protection, threat protection, and information protection.
- **Cost**
  Pay as you use model means no up-front investment needed and hence more cost-effective.
- **Developer Tools**
  A wide variety of tools available for building and deploying your application automatically.

Deploying Web Application on Microsoft Azure

Azure App Service

- PaaS from Microsoft Azure, provides a managed platform to deploy and host your applications
- Offers auto-scaling, high availability, and load balancing
- Provides App Insights to monitor application performance
- Applications are hosted on Microsoft’s global infrastructure
- App Service is ISO, SCO, and PCI complaint
- App Service provides easy integration to other Azure services such as Storage, Active Directory etc.
- Integrates with various tools for continuous deployment

Demonstration: How to Deploy a Web Application to Microsoft Azure

- In this demonstration, you will learn how to:
- Deploy your application to Microsoft Azure App Service.
- Check the details of the deployed application in Azure Portal.

Azure Deployment Strategy

- Traditional Deployment Options
- FTP, CLI, Visual Studio – simple, but nor suitable for production grade applications
  - Require downtime and it is not easy to rollback in case of problems
- Deployment Slots
  - Provides additional environments similar to production environments
  - Applications can be deployed to these for testing or staging
  - Allows traffic routing for a percentage of the incoming traffic to one of the deployment slots to perform A/B testing
  - On successful testing, slots can be swapped
- Resource Templates
  - Ability to define resources needed for an application as a template
  - Re-create the entire stack from this template

Debugging a Microsoft Azure Application

Application Insights – ability to monitor applications running in App Service.

- Metrics – such as CPU usage, memory consumption, page views, performance of events etc
- Live Stream – ability to see key metrics in a streaming fashion.
- Analysis from Visual Studio
- Remote Debugging – ability to debug live applications from Visual Studio
- Server Explorer – Ability to manage Azure Services from Visual Studio

Lesson 3: Microsoft Azure Fundamentals

- Microsoft Azure Storage
- Demonstration: How to Upload an Image to Microsoft Azure Blob Storage
- Microsoft Azure SQL
- Design a Distributed Application by Using Microsoft Azure
- Design a Caching Strategy
- Security in Microsoft Azure

Microsoft Azure Storage

- Managed service providing storage that is highly available, secure, durable, scalable, and redundant.
- Type of storage :
  - **Azure Blob storage -** object-based storage; can be used to store image files, audio and video clips.
  - **Azure File Share -** fully managed file shares in the cloud
  - **Azure Queue storage -** service for storing large numbers of messages
  - **Azure Table storage -** service that stores structured NoSQL data in the cloud

Demonstration: How to Upload an Image to Microsoft Azure Blob Storage

- In this demonstration, you will learn how to:
- Create a storage account from the Azure Portal.
- Create a container from the Azure Portal.
- Upload an image from the Azure Portal.
- Connect to storage account from an application.
- Create a container from an application.
- Upload an image from a web application.

Microsoft Azure SQL

- Azure SQL Database – a fully managed SQL database
- Infrastructure management taken care of by Azure
- Scalability – Allows for dynamic scalability, i.e. increase/decrease infrastructure configuration without a downtime
- Availability – provides automatic backups, replication and failure detection
- Security and Compliance – provides data encryption at rest and in transit. Provides access control and tools to protect sensitive data.
- Intelligent insights and monitoring – provides automatic performance monitoring and tuning.

Design a Distributed Application by Using Microsoft Azure

- Distributed applications – ability to auto scale.
- Need for centralized session management
- Session-affinity
- Redis based session management.
- Need for asynchronous communication between components for easy scalability
- Azure Service Bus – integrated message broker
- WebJobs – ability to perform background tasks
- Azure Functions – run small functions on the cloud.
- Hybrid applications – run on Azure and on-premise datacenters
- Azure Stack – run Azure Services on datacenters in any location

Design a Caching Strategy

- Azure Cache for Redis
  - Redis cache as a service
  - Globally available, hence suitable for distributed applications
  - Useful for caching data within an application – session data, data that needs to be fetched from a database, etc.
- Azure CDN – Content Delivery Network
  - Global caching for static content such as html files, images etc.
  - Files are cached in edge servers across the globe
  - Files are served from closest location to consumers, thus improving performance

Security in Microsoft Azure

- Azure Key Vault – a secure store for keys and secrets
- All information within a vault is encrypted and stored.
- Useful to securely store application configuration information such as connection strings, passwords etc.
- Application access to Key Vault is through authentication by Azure Active Directory.
- Key Vault is backed by HSM.

Lab: Hosting and Deployment

- Exercise 1: Deploying a Web Application to Microsoft Azure
- Exercise 2: Upload an Image to Azure Blob Storage

Lab Review

- What are the advantages of deploying to Azure?
- When would you use Azure Blob storage?

Module Review and Takeaways

- Review Question
- Best Practice
- Common Issues and Troubleshooting Tips

Course Evaluation

- Your evaluation of this course will help Microsoft understand the quality of your learning experience.
- Please work with your training provider to access the course evaluation form.
- Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.