



MICROSERVICES DEVELOPMENT IN WINDOWS EĞİTİMİ

2 GÜN



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	5
Microservice development.....	5
REST services	5
Getting started with asp.net core.....	5
Serverless architecture with Azure.....	6
Azure Container Service	6
Azure container service w/ Kubernetes	6
Kubernetes: from the firehose	6
Docker: Bring on the Whale	6
Microsoft Docker Tooling.....	7
Operational Readiness.....	7
Application Modernization.....	7
Security in microservices.....	7

Eđitim Hakkında

“Microservices Development in Windows Eđitimi”, Windows platformunda microservices tabanlı uygulamaların geliştirilmesi sürecini öğretir. Bu eğitim, microservices tabanlı uygulamaların geliştirilmesinde gereken temel konuları kapsar.

Eđitim, microservices mimarisi ve Windows ortamında nasıl kullanılacağını, veri yönetimi, hata toleransı, mesajlaşma ve diđer temel kavramları öğretir. Katılımcılar, örnekler ve projeler aracılığıyla Windows'ta microservices geliştirme becerilerini geliştirirler.

Eđitim ayrıca, uygulamaların microservices tabanlı geliştirilmesi sürecinde Windows'un nasıl kullanılabileceđini de öğretir. Katılımcılar, Windows'un büyük veri işleme, real-time analiz, hata toleransı ve diđer özelliklerini öğrenirler. Ayrıca, Windows'un microservices tabanlı uygulama geliştirmede nasıl kullanılabileceđini de anlarlar.

“Microservices Development in Windows Eđitimi”, microservices tabanlı uygulama geliştirme sürecinin tümünü kapsar. Katılımcılar, modern uygulamalar geliştirmeye başlamadan önce ihtiyaç duyacakları temel becerileri kazanırlar. Eğitim, örnekler, pratik uygulamalar ve projeler yoluyla öğrenmeyi sağlar ve katılımcıların microservices tabanlı uygulama geliştirmeye başlamalarına yardımcı olur.

Eđitim programı, microservices mimarisi ve Windows'un temelleriyle başlar. Katılımcılar, Windows ile veri yönetimi, mesajlaşma sistemlerinin yönetimi ve hata toleransı gibi temel kavramları öğrenirler. Ayrıca, Windows'un microservices tabanlı modern uygulamaların geliştirilmesi ve yönetilmesi konusunda nasıl bir rol oynadığına dair bilgi sahibi olurlar. Bu bilgiler, katılımcıların uygulama geliştirme sürecinde ihtiyaç duyacakları temel yapı taşlarını oluşturur.

Eđitimde, microservices mimarisi ve Windows'un temel özellikleri ve bileşenleri üzerinde duruyoruz. Bu, katılımcılara real-time veri işleme, büyük veri akışları ve hata toleransı gibi yetenekleri kazandırır. Konu akışları, veri yönetimi, ve dağıtık sistemlerin yönetimi gibi konular işlenir.

Son olarak, Windows ile bir uygulamanın nasıl microservices tabanlı geliştirileceđi hakkında bilgi veriyoruz. Bu süreç, uygulamanın son testlerini yapmayı, Windows ile veri yönetimini, ve en sonunda uygulamanın microservices tabanlı geliştirilmesini içerir. Bu bilgiler, katılımcıların uygulamalarını Windows ile başarılı bir şekilde microservices tabanlı geliştirmelerine yardımcı olur.

Neler Öğreneceksiniz?

Microservices Development in Windows Eđitiminde, aşağıdaki konuları öğreneceksiniz:

- Microservices mimarisi ve tasarım konseptleri
- Windows üzerinde microservices uygulamalarının geliştirilmesi
- API'lerin yapılandırılması ve kullanılması
- Veri deposu entegrasyonu ve veritabanı yapılandırma



- Microservices güvenliđi ve güvenlik katmanları
- Çoklu platform desteđi
- Uygulamanın test edilmesi ve dağıtılması
- Microservices'in yönetimi ve devam eden bakımı

Eđitim, teorik konuların yanı sıra, pratik uygulamalar ile de desteklenir ve katılımcılar kod yazma becerilerini geliştirme fırsatı bulurlar.

Ön Koşullar

Microservices Development in Windows Eđitiminin ön koşulları:

- Programlama dilleri: Katılımcıların .NET veya C# gibi bir programlama dili ile önceki deneyime sahip olması beklenebilir.
- Uygulama Geliştirme: Katılımcıların uygulama geliştirme kavramları ve prosedürleri hakkında temel bilgi sahibi olması gerekir.
- Veritabanı Teknolojileri: Veritabanı konseptleri ve SQL gibi bir veritabanı diline hakim olmak faydalı olabilir.
- Web Geliştirme: HTML, CSS ve JavaScript gibi web geliştirme dilleri hakkında temel bilgi sahibi olmak faydalı olabilir.

Bu koşullar eđitimin verimli bir şekilde tamamlanması için önerilen koşullardır ve eđitimin verildiđi seviyeyi ve kapsamını belirleyen faktörlerdir.

Kimler Katılmalı

Microservices Development in Windows Eđitimine;

- Uygulama Geliştiricileri: Uygulama geliştirme konularında deneyimli ve Microsoft Windows platformu üzerinde uygulama geliştirmeyi amaçlayan geliştiriciler.
- Sistem Mimarları: Sistem mimarisi ve microservices tasarımı konularında uzmanlaşmak isteyen ve Microsoft Windows platformu üzerinde sistem tasarımı yapmak isteyen mimarlar.
- DevOps Uzmanları: DevOps süreçlerinde microservices kullanımını ve Microsoft Windows platformu üzerinde dağıtım yapmayı amaçlayan DevOps uzmanları.
- Yöneticiler: Uygulama geliştirme ve DevOps ekibinin işletme süreçlerini yönetmek isteyen ve Microsoft Windows platformu üzerinde uygulama geliştirme konuları hakkında fikir sahibi olmak isteyen yöneticiler.



Outline

Microservice development

- The Microservices Architecture Design Principles
- Decentralized Processing
- Business Domain-Centric Design
- Crossing Process Boundary is Expensive!
- Designing for Failure
- Fault Injection During System Testing
- Distributed Transactions
- Managing Distributed Services with Docker & Kubernetes (OpenShift)
- Microservices and their relationship to the front-end
- Rich Client Applications
- Single Page Applications (SPA)
- The Building Blocks of a Fault-tolerant Application
- Example of Microservices in Their Purest Form: AWS Lambdas
- Example of Traditional Enterprise Application Architecture
- Example of Microservices Architecture
- Summary

REST services

- Many Flavors of Services
- Understanding REST
- Principles of RESTful Services
- Create
- Retrieve
- Update
- Delete
- Client Generated ID
- SOAP Equivalent Examples
- JSON
- REST vs SOAP Communication
- Restful API Design
- Versioning
- Documentation
- Security
- Additional Resources
- Summary

Getting started with asp.net core

- Getting Started with ASP.NET Core
- Configuration and Middleware Pipeline
- Controllers and Views
- RESTful Services with Web API
- Data Access with Entity Framework Core
- Summary



Serverless architecture with Azure

- What is a Serverless Architecture?
- Azure Functions
- WebJobs SDK
- WebJobs Core
- WebJobs Extensions
- Azure App Service
- Azure Resource Manager
- Understanding Azure Functions usage

Azure Container Service

- Azure Service Fabric
- Microsoft Application Insights
- Microsoft Operations Management Suite
- Summary

Azure container service w/ Kubernetes

- What is Kubernetes?
- What Is a Container?
- Microservices and Orchestration
- Microservices and Infrastructure-as-Code
- Kubernetes Container Networking
- Summary

Kubernetes: from the firehose

- Masters
- Nodes
- Pods
- Namespaces
- Resource Quota
- Authentication and Authorization
- Routing
- Registry
- Storage Volumes
- Microservices, Linking, and Catalogs
- Summary

Docker: Bring on the Whale

- What is Docker
- Docker ecosystem
- Docker concepts
- Docker Architecture
- Microservice encapsulation
- Secure microservices using Docker
- Agility, Portability, and Control
- .NET Core or .NET Framework for Docker
- Summary



Microsoft Docker Tooling

- Using Visual Studio Tools for Docker
- Configuring your local environment
- Using Docker Tools in Visual Studio 2015
- Using Docker Tools in Visual Studio 2017
- Using Windows PowerShell commands in a DockerFile to set up Windows Containers
- DevOps workflow for a Docker application
- Inner-loop development workflow
- Source-Code Control integration and management with Visual Studio Team Services and Git
- Build, Continuous Integration, and Test with Visual Studio Team Services and Docker
- Continuous Delivery, Deploy
- Run and manage
- Monitor and diagnose
- Summary

Operational Readiness

- Monitoring, Logging, and Resource Management
- Monitoring Compute Resources
- Checking Pod Health
- Logging
- Secrets & Vaulting
- Summary

Application Modernization

- What is Application Modernization?
- Typical App Modernization Projects
- Why Modernize?
- Goals for Modernization
- Twelve-factor Application Microservices
- Maintaining State
- Cloud Service Fabric
- Summary

Security in microservices

- Role-based
- Claim-based
- Session Cookie
- HTTP Basic
- JWT
- Summary