



**ARCHITECTING MICROSERVICES
WITH KUBERNETES, DOCKER, AND
CONTINUOUS INTEGRATION EĞİTİMİ
2 GÜN**



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	4
Introduction to Kubernetes	4
Kubernetes – From the Firehose	5
Docker Introduction.....	5
CI/CD with OpenShift, Jenkins, and Blue Ocean.....	5
Application Modernization	6
Security in Microservices.....	7

Eđitim Hakkında

“Architecting Microservices with Kubernetes, Docker, and Continuous Integration Eđitimi”, Windows platformunda microservices tabanlı uygulamaların geliştirilmesi sürecini öğretir. Bu eđitim, 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. Eđitim, 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?

Architecting Microservices with Kubernetes, Docker, and Continuous Integration Eđitiminde, aşağıdaki konuları öğreneceksiniz;

- Mikroservis mimarisinin temel kavramları ve faydaları
- Kubernetes ve Docker’ın mikroservis mimarisinde nasıl kullanılması gerektiđi
- Sürekli Entegrasyon (Continuous Integration) ve dağıtım (Continuous Deployment) süreçleri



- Mikroservis mimarisinin test ve monitöring süreçleri
- Kubernetes'in mikroservis dağıtımı, yapılandırması ve ölçeklendirmesi
- Mikroservis mimarisinin güvenliği ve güncelleme yönetimi

Ön Koşullar

Architecting Microservices with Kubernetes, Docker, and Continuous Integration Eğitiminin ön koşulları:

- Programlama dilleri (örneğin, Java, Python, Ruby vb.) ve web geliştirme deneyimi
- İşletim sistemi temel kavramları ve kabuk komutları
- Veri yapıları, algoritmalar ve programlama paradigmaları
- Agile yazılım geliştirme metodolojisi ve sürekli entegrasyon (Continuous Integration) kavramları

Kimler Katılmalı

Architecting Microservices with Kubernetes, Docker, and Continuous Integration Eğitimine aşağıdaki kişiler katılabilir:

- Yazılım geliştiricileri ve arhitekleri
- DevOps ve sürekli entegrasyon (Continuous Integration) profesyonelleri
- Kubernetes ve Docker kullanmak isteyen yazılım profesyonelleri
- Microservices mimarisi ve uygulamalarını geliştirmek isteyen yazılım profesyonelleri
- DevOps ve konteyner teknolojileri hakkında eğitim almak isteyen yazılım profesyonelleri.

Outline

Introduction to Kubernetes

- What is Kubernetes
- What is a Container
- Container – Uses
- Container – Pros
- Container – Cons
- Composition of a Container
- Control Groups
- Namespaces
- Union Filesystems
- Popular Containerization Software
- Microservices
- Microservices and Containers / Clusters
- Microservices and Orchestration



- Microservices and Infrastructure-as-Code
- Kubernetes Container Networking
- Kubernetes Networking Options
- Kubernetes Networking – Balanced Design
- Summary

Kubernetes – From the Firehose

- What is Kubernetes?
- Container Orchestration
- Kubernetes Basic Architecture
- Kubernetes Detailed Architecture
- Kubernetes Concepts
- Cluster and Namespace
- Node
- Master
- Pod
- Label
- Annotation
- Label Selector
- Replication Controller and Replica Set
- Service
- Storage Volume
- Secret
- Resource Quota
- Authentication and Authorization
- Routing
- Registry
- Using Docker Registry
- Summary

Docker Introduction

- What is Docker
- Where Can I Run Docker?
- Docker and Containerization on Linux
- Linux Kernel Features: cgroups and namespaces
- The Docker-Linux Kernel Interfaces
- Docker Containers vs Traditional Virtualization
- Docker as Platform-as-a-Service
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Competing Systems
- Docker Command-line
- Starting, Inspecting, and Stopping Docker Containers
- Summary

CI/CD with OpenShift, Jenkins, and Blue Ocean

- What is OpenShift
- OpenShift Online



- OpenShift Origin
- OpenShift Architecture
- OpenShift Origin Installation
- OpenShift CLI
- OpenShift CLI (Contd.)
- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins
- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- Installing Jenkins as a Windows Service
- Different types of Jenkins job
- Configuring Source Code Management(SCM)
- Working with Subversion
- Working with Git
- Build Triggers
- Schedule Build Jobs
- Polling the SCM
- Maven Build Steps
- Jenkins / OpenShift Pipeline
- Jenkins / OpenShift Pipeline Output
- Installing Jenkins Plugins
- The Blue Ocean Plugin
- Blue Ocean Plugin Features
- New modern user experience
- Advanced Pipeline visualizations with built-in failure diagnosis
- Branch and Pull Request awareness
- Personalized View
- OpenShift Pipeline Output
- Creating OpenShift Blue Ocean Pipeline
- Summary
- Chapter 5. Operational Readiness
- What is Operational Readiness
- Telemetry
- End-to-end Requirements Traceability
- Log Strategy
- Monitoring Strategy
- Runbooks
- Summary

Application Modernization

- What is Application Modernization
- Typical App Modernization Projects
- Why Modernization?
- Goals for Application Modernization
- Modernization Process
- Modernization in a Nutshell



- Modernization in a Nutshell – Analyze
- Modernization in a Nutshell – Rationalize
- Modernization in a Nutshell – Modernize
- Modernization in a Nutshell – Supervise
- Twelve-factor Applications
- Twelve Factors, Microservices, and App Modernization
- 12-Factor Microservice Codebase
- 12-Factor Microservice Dependencies
- 12-Factor Microservice Config
- 12-Factor Microservice Backing Services
- 12-Factor Microservice Continuous Delivery
- 12-Factor Microservice Processes
- 12-Factor Microservice Data Isolation
- 12-Factor Microservice Concurrency
- 12-Factor Microservice Disposability
- 12-Factor Microservice Environment Parity
- 12-Factor Microservice Logs
- 12-Factor Microservice Admin Processes
- Monolithic revisited
- Monolithic vs. Microservices
- Maintaining State in App Modernization
- Cloud Service Fabric
- Summar

Security in Microservices

- Why Microservice Security?
- Security Testing in Microservices
- Security Topology
- Authorization and Authentication
- J2EE Security Refresh
- Role-based Access Control in a Nutshell
- Claim-based Access Control in a Nutshell
- Sharing Sessions
- Session Cookie
- JSON Web Token (JWT)
- Spring Security
- Summary