

CLOUD DESIGN PATTERS

Modern yazılım geliştirme, bulutun dinamik ve ölçeklenebilir dünyasında şekilleniyor. Ancak bulutun sunduğu sınırsız potansiyeli tam anlamıyla kullanmak ve karşılaşılabilecek zorlukların üstesinden gelmek için doğru mimari kararları almak büyük önem taşır. İşte tam da bu noktada Cloud Design Patterns Eğitimi devreye giriyor!

Bu eğitim, bulut tabanlı uygulamalar tasarlarken karşılaşılan yaygın sorunlara kanıtlanmış, tekrar kullanılabilir çözümler sunan tasarım kalıplarını kapsamlı bir şekilde incelemektedir. İster mikroservisler, ister sunucusuz mimariler veya büyük veri çözümleri geliştiriyor olun, bu kalıplar sistemlerinizin güvenilirliğini, ölçeklenebilirliğini, performansını, yönetilebilirliğini ve maliyet etkinliğini artırmanıza yardımcı olacaktır.

CLOUD DESIGN PATTERS

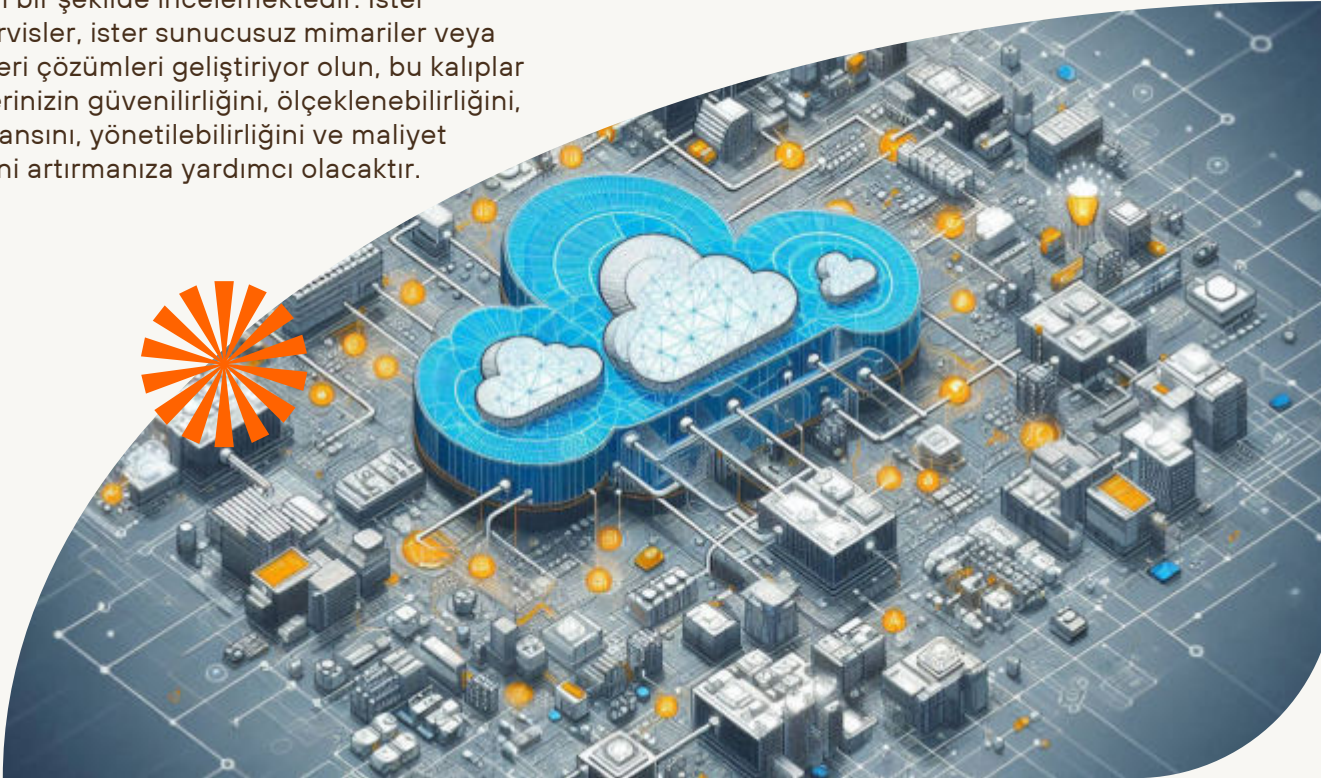
KURUMSAL ÖZEL EĞİTİM

Bu konuyu istediğiniz gibi özelleştirebilirsiniz. Kullandığınız teknolojiye özel uygulamalar ile anlatılmasını, projenize uygun içerik haline getirilmesini ...

CLOUD DESIGN PATTERS EĞİTİMİ

Neden Katılmalısınız?

- **Mimari Uzmanlık Kazanımı:** Bulut mimarisi sadece kaynakları kullanmaktan ibaret değildir; aynı zamanda sistemlerinizi dayanıklı, ölçeklenebilir ve performanslı hale getiren stratejik kararları içerir. Bu eğitimle, bulut ortamlarının kendine özgü zorluklarını aşmak için kanıtlanmış mimari desenleri öğrenecek, daha sağlam ve ileriye dönük sistemler tasarlama becerisi kazanacaksınız.
- **Sektörün En İyi Uygulamalarını Öğrenin:** Sektörde başarılı olmuş, büyük ölçekli ve kritik sistemlerde kendini kanıtlamış tasarım kalıplarını doğrudan öğrenme fırsatı bulacaksınız. Bu sayede, yaygın hatalardan kaçınacak ve projelerinizi daha ilk adımdan itibaren doğru temeller üzerine inşa edeceksiniz.
- **Pratik ve Uygulanabilir Bilgi:** Eğitimimiz sadece teorik bilgi aktarımıyla sınırlı kalmaz. Öğrenilen her kalıp, gerçek dünya senaryoları ve vaka çalışmaları üzerinden nasıl uygulanacağını gösterir. Bu sayede, teorik bilginizi pratik beceriye dönüştürerek, kendi projelerinizde doğrudan kullanabileceğiniz somut yetkinlikler edineceksiniz.
- **Kariyerinizde Fark Yaratın:** Bulut mimarisi ve tasarım kalıpları bilgisi, günümüzün rekabetçi teknoloji pazarında sizi öne çıkaran, aranan bir yetkinliktir. Bu eğitimle edineceğiniz uzmanlık, kariyerinizde yeni kapılar açmanıza ve daha stratejik roller üstlenmenize yardımcı olacaktır.
- **Maliyet ve Zaman Tasarrufu:** Doğru mimari kararlar almak, uzun vadede projenizin maliyetlerini düşürür ve geliştirme süreçlerini hızlandırır. Yanlış mimari seçimler, sonradan düzeltilmesi çok daha maliyetli ve zaman alıcı problemlere yol açabilir. Bu eğitimle, başından itibaren doğru adımları atarak bu tür riskleri minimize edeceksiniz.





Modül 1: Bulut Mimari Temelleri ve Dönüşüm

Bulut Bilişim Prensipleri ve Hizmet Modelleri:

- Bulut bilişimin temel felsefesi, avantajları ve dezavantajları.
- IaaS (Infrastructure as a Service), PaaS (Platform as a Service) ve SaaS (Software as a Service) modellerinin derinlemesine analizi ve .NET uygulamaları için seçim kriterleri.
- Bulut hizmetlerinin .NET uygulamalarına entegrasyonu ve yönetim modelleri (örneğin, Azure App Services, AWS Elastic Beanstalk).

Yaygın Bulut Sağlayıcılarının Temel Mimarileri:

- Azure, AWS ve GCP gibi önde gelen bulut sağlayıcılarının temel mimari bileşenleri ve .NET geliştiricileri için kritik servisleri.
- Bulut üzerindeki sanal makineler (VM), kapsayıcılar (Container) ve sunucusuz (Serverless) bilişim yaklaşımlarının .NET projelerindeki kullanım senaryoları.

Bulut Tabanlı Uygulama Geliştirmede Stratejik Dönüşüm:

- Monolitik uygulamalardan bulut-yerel mimarilere geçiş stratejileri ve bu süreçteki .NET best practices.
- Maliyet optimizasyonu, esneklik ve hızlı dağıtımın iş süreçlerine etkisi.

Modül 2: Dağıtık Sistem Kavramları ve Mikroservis Mimarisi

Dağıtık Sistemlerin Tanımı ve Temel Özellikleri:

- Monolitik yapılardan dağıtık sistemlere geçişin nedenleri ve sağladığı avantajlar.
- Dağıtık sistemlerin zorlukları (hata yönetimi, veri tutarlılığı, izleme) ve çözüm yaklaşımları.

Servis Tabanlı ve Mikroservis Mimarilerine Giriş:

- Mikroservis mimarisinin temel prensipleri, avantajları ve dezavantajları.
- .NET Core ile mikroservis uygulamaları geliştirme pratikleri ve örnekleri.

İletişim Modelleri: Senkron ve Asenkron

Mesajlaşma:

- HTTP/REST tabanlı senkron iletişim (API Gateway, Client-Side Load Balancing) ve .NET için HttpClient kullanımı.
- Mesajlaşma kuyrukları (RabbitMQ, Kafka, Azure Service Bus) ve asenkron iletişim desenleri.

Modül 3: Ölçeklenebilirlik Prensipleri ve Otomatik Ölçeklendirme

Uygulama ve Altyapı Ölçeklendirme Teknikleri (Scalability Primer):

- Yatay (horizontal) ve dikey (vertical) ölçeklemenin farkları, avantajları ve dezavantajları.
- .NET uygulamalarında stateless servis tasarımı ve yatay ölçeklendirme için hazırlık.

Otomatik Ölçeklendirme (Auto-Scaling Pattern) Mekanizmaları:

- Bulut sağlayıcılarının otomatik ölçeklendirme servisleri (Azure Scale Sets, AWS Auto Scaling Groups) ve yapılandırmaları.
- Metrik tabanlı (CPU, bellek, kuyruk uzunluğu) ve zamanlanmış ölçeklendirme kuralları.
- Hands-on: .NET Core Web API uygulamasının Azure App Service üzerinde otomatik ölçeklendirme yapılandırması.

Sistem Kapasite Planlama ve Performans Darboğazı Analizi:

- Yük testi ve stres testi araçları ile .NET uygulamalarının performansını ölçme.
- Profil çıkarma (profiling) ve darboğaz tespiti için .NET araçları (Application Insights, PerfView).

Modül 4: Kuyruk Tabanlı İş Akışları ve Hata Toleransı

Mesaj Kuyruklarının Dağıtık Sistemlerdeki Rolü (Queue-Centric Workflow Pattern):

- Asenkron işlem akışlarının faydaları: Yük dengeleme, bileşen bağımsızlığı, hata toleransı.
- Mesaj kuyrukları ile .NET worker servisleri arasında iletişim kurma.

Yük Dengeleme ve Geri Basınç (Backpressure) Yönetimi:

- Mesaj kuyrukları aracılığıyla iş yükünün dağıtılması ve sistemin aşırı yüklenmesini önleme.
- Hatalı mesajların yönetimi (dead-letter queues) ve yeniden deneme mekanizmaları.

.NET Ortamında Yaygın Mesajlaşma Çözümleri:

- Azure Service Bus, RabbitMQ, Kafka gibi popüler mesajlaşma sistemlerinin .NET Core ile entegrasyonu.
- Hands-on: Bir .NET Core producer/consumer uygulaması ile mesaj kuyruğu deseni uygulaması.

Modül 5: Temel Güvenlik ve OWASP Top 10

Yazılım Güvenliğinin Temel Prensipleri:

- Güvenli yazılım geliştirme yaşam döngüsü (SSDLC) ve güvenlik açıklarının önlenmesi.
- Temel güvenlik kavramları: Kimlik doğrulama, yetkilendirme, şifreleme, hashing.

OWASP Top 10 ve Yaygın Güvenlik Zafiyetleri:

- Injection, Broken Authentication, Cross-Site Scripting (XSS) gibi en kritik güvenlik açıklarının analizi.
- Hands-on: .NET uygulamalarında bu zafiyetlere karşı alınabilecek önlemler ve güvenli kodlama pratikleri.

Saldırı Yüzeyinin Azaltılması ve Risk Yönetimi:

- Minimal yetki prensibi, güvenlik duvarları, ağ segmentasyonu gibi stratejiler.
- Uygulama güvenliği testleri: SAST, DAST ve SCA.

Modül 6: Veritabanı Ölçekleme Temelleri ve Sharding

Veritabanı Dikey ve Yatay Ölçekleme Yaklaşımları:

- Veritabanı performansını artırma yöntemleri: İndeksleme, sorgu optimizasyonu, ön bellekleme.
- Dikey ölçeklemenin sınırları ve yatay ölçeklemenin avantajları.

Basit Veri Replikasyonu ve Parçalama (Sharding) Kavramına Giriş:

- Veri replikasyonunun amacı (yüksek erişilebilirlik, okuma ölçeklemesi) ve türleri.
- Database Sharding Pattern: Veritabanının yatay olarak bölünmesi, veri dağıtım stratejileri (range, hash, directory based).

İlişkisel ve NoSQL Veritabanlarının Ölçeklenebilirlik Perspektifi:

- SQL Server, PostgreSQL gibi ilişkisel veritabanlarının ölçeklendirme zorlukları ve çözümleri.
- Cosmos DB, MongoDB, Cassandra gibi NoSQL veritabanlarının ölçeklenebilirlik avantajları ve kullanım senaryoları.



Modül 7: Hata Toleransı ve Dayanıklılık Stratejileri

Dağıtık Sistemlerde Hata Tespiti, İzolasyonu ve Kurtarma:

- Sistem bileşenlerinin başarısızlık durumlarını yönetme.
- İzole süreçler ve izolasyon duvarları (Bulkhead) desenleri.

Düğüm Arızası Deseni (Node Failure Pattern) ve Sistem Kesintisizliği:

- Bir düğümün (node) arızalanması durumunda sistemin çalışmaya devam etmesini sağlayan stratejiler.
- Replikasyon, lider seçim mekanizmaları ve otomatik yeniden yönlendirme teknikleri.

Temel Dayanıklılık Desenleri:

- Yeniden Deneme (Retry Pattern): Geçici ağ hataları veya servis kesintileri durumunda otomatik yeniden deneme stratejileri.
- Devre Kesici (Circuit Breaker Pattern): Hata oranları yükseldiğinde servis çağrılarını durdurarak sistemin çökmesini engelleme.
- Hands-on: Polly kütüphanesi ile .NET Core uygulamalarında Retry ve Circuit Breaker desenlerinin uygulanması.

Modül 8: CI/CD Temelleri ve Otomasyon

Sürekli Entegrasyon (CI) ve Sürekli Teslimat (CD) Felsefesi:

- CI/CD'nin temel prensipleri, faydaları ve yazılım geliştirme sürecindeki rolü.
- Hızlı geri bildirim döngüleri ve kod kalitesini artırma.

Otomatik Derleme, Test ve Dağıtım Pipeline'larının Oluşturulması:

- Azure DevOps Pipelines, GitHub Actions veya Jenkins gibi araçlarla CI/CD pipeline'larının tasarlanması.
- .NET projeleri için otomatik birim (unit), entegrasyon (integration) ve kabul (acceptance) testleri.

DevOps Kültürü ve .NET Projelerine Uygulanması:

- Geliştirme ve operasyon ekipleri arasındaki işbirliğini artırma stratejileri.
- Infrastructure as Code (IaC) kavramına giriş ve dağıtım otomasyonu.

Modül 9: Gelişmiş Bulut Mimari Desenleri ve Mikroservis İletişimi

Mikroservisler Arası İletişim ve Koordinasyon Desenleri:

- API Gateway: Dış dünya ile mikroservisler arasındaki tek giriş noktası olarak API Gateway'in rolü ve .NET/Ocelot ile uygulama.
- Backends-for-Frontends (BFF): Farklı istemci türleri için optimize edilmiş API'ler oluşturma.

Service Discovery, Konfigürasyon Yönetimi ve Yük Dengeleme:

- Mikroservislerin birbirlerini bulma mekanizmaları (örn. Consul, Eureka) ve .NET entegrasyonları.
- Dağıtık sistemlerde merkezi konfigürasyon yönetimi (örn. Azure App Configuration, Consul).
- İstemci tarafı yük dengeleme (Client-Side Load Balancing) yaklaşımları.

Kapsayıcı Ortamlarda (Kubernetes) Dağıtık Desenlerin Uygulanması:

- Docker Compose ve Kubernetes üzerinde mikroservis uygulamalarını dağıtma ve yönetme.
- Hands-on: .NET Core mikroservis uygulamasının Kubernetes'e dağıtımı ve servis iletişimi.

Modül 10: Veri Tutarlılık Modelleri ve Dağıtık İşlemler

CAP Teoremi'nin Derinlemesine Analizi:

- Tutarlılık (Consistency), Erişilebilirlik (Availability) ve Bölümlenebilirlik (Partition Tolerance) arasındaki ödünleşimler.
- Dağıtık sistem tasarımında CAP teoreminin önemi ve pratik çıkarımları.

Nihai Tutarlılık (Eventual Consistency Primer) Desenleri:

- Dağıtık sistemlerde verilerin zamanla tutarlı hale gelmesi ve güçlü tutarlılık yerine erişilebilirlik önceliği.
- Hands-on: .NET Core ve NoSQL veritabanları ile nihai tutarlı bir sistemin tasarımı ve uygulanması.

Dağıtık İşlemlerin Yönetimi ve Saga Deseni:

- İki fazlı commit (2PC) gibi geleneksel yaklaşımların sınırlılıkları.
- Mikroservis mimarisinde uzun süren dağıtık işlemler için Saga deseninin uygulanması.
- Hands-on: Bir Saga deseni uygulamasının .NET Core ve mesaj kuyrukları ile modellenmesi.

Modül 11: Büyük Veri İşleme Desenleri ve Gerçek Zamanlı Akışlar

Dağıtık Veri İşleme Algoritmaları ve Map-Reduce Pattern:

- Büyük veri kümelerini paralel olarak işlemek için Map-Reduce deseni ve çalışma prensibi.
- Verilerin parçalanarak işlenmesi (map) ve birleştirilmesi (reduce) aşamaları.

Gerçek Zamanlı Veri Akışı (Data Streaming) ve Olay Odaklı Mimariler:

- Kafka, Azure Event Hubs gibi akış platformları ile olay odaklı mimarilerin oluşturulması.
- .NET ile gerçek zamanlı veri akışlarının işlenmesi ve analiz edilmesi.

Bulut Tabanlı Büyük Veri İşleme Servisleri ve .NET Entegrasyonları:

- Azure Data Lake Analytics, AWS EMR gibi servislerin büyük veri işleme senaryolarında kullanımı.
- Hands-on: .NET Core uygulaması ile basit bir Map-Reduce benzeri işlemin simülasyonu ve bulut servisi entegrasyonu.

Modül 12: Çoklu Kiracılık (Multitenancy) Uygulamaları

SaaS Çözümlerinde Çoklu Kiracılık Mimari Yaklaşımları (Multitenancy Pattern):

- Tek bir uygulama örneğinin birden fazla müşteriye (tenant) hizmet vermesi.
- Kiracı izolasyon seviyeleri: Ayrı veritabanı, ayrı şema, paylaşımlı şema.

Kiracı Bazlı Kimlik Doğrulama, Yetkilendirme ve Veri Ayırımı:

- Çoklu kiracılı uygulamalarda kimlik ve erişim yönetimi stratejileri.
- Her kiracının verilerinin güvenli ve izole bir şekilde tutulması.
- Hands-on: .NET Core ve Entity Framework Core ile çoklu kiracılık desteğinin uygulanması.

Kaynak Tahsisi ve Maliyet Optimizasyonu Stratejileri:

- Kiracı başına kaynak tahsisi ve kullanımına göre maliyetlerin izlenmesi.
- Paylaşımlı altyapı kullanımının maliyet avantajları.



Modül 13: Gelişmiş Güvenlik ve Kimlik Yönetimi

Valet Key Pattern ve Kısıtlı Süreli Erişim

Anahtarları:

- Kullanıcıların bulut depolama hizmetlerine doğrudan ve güvenli bir şekilde erişmesini sağlama.
- .NET'te Azure Storage Shared Access Signatures (SAS Token) veya AWS S3 Presigned URLs oluşturma ve kullanma.

API Güvenliği: OAuth 2.0, OpenID Connect ve JWT Kullanımı:

- Modern API'ler için kimlik doğrulama ve yetkilendirme standartları.
- Hands-on: .NET Core Web API'lerinde JWT tabanlı kimlik doğrulama ve yetkilendirme uygulama.

Siber Güvenlik Riskleri ve DevSecOps

Prensiplerinin Entegrasyonu:

- Güvenliği yazılım geliştirme sürecinin her aşamasına entegre etme.
- Güvenlik açığı tarama araçları ve otomatik güvenlik testleri.

Modül 14: Ağ Gecikmesi ve Performans Optimizasyonu

Ağ Gecikmesinin Kaynakları ve Performansa Etkisi (Network Latency Primer):

- Gecikme türleri (propagation, transmission, processing delay) ve dağıtık sistemlerdeki etkileri.
- Gecikme ölçümleme ve analiz yöntemleri.

İçerik Dağıtım Ağları (CDN Pattern) ve Gelişmiş Önbellekleme Stratejileri:

- Statik içeriğin kullanıcıya en yakın lokasyondan sunulması ve sayfa yükleme sürelerinin kısaltılması.
- CDN yapılandırması ve .NET Core uygulamalarında önbellekleme (in-memory, distributed caching) stratejileri.

Kaynak Co-location (Colocate Pattern) ve Bölgesel Dağıtım Avantajları:

- Sık iletişim kuran bileşenlerin aynı fiziksel veya mantıksal bölgede konuşlandırılması.
- Ağ gecikmesini azaltma ve performans artırma stratejileri.
- Hands-on: .NET Core mikroservisleri için co-location stratejilerinin simülasyonu

Modül 15: Dağıtık Sistemlerde İzleme ve Gözlenebilirlik

Merkezi Loglama, Metrik Toplama ve Uyarı Sistemleri:

- Dağıtık sistemlerde log toplama (ELK Stack, Azure Log Analytics) ve analizi.
- Metrik toplama (Prometheus, Grafana) ve uyarı mekanizmaları.
- Hands-on: .NET Core uygulamalarında Serilog/NLog ile loglama ve Application Insights ile metrik toplama.

Dağıtık İzleme (Distributed Tracing) ve Korelasyon ID'leri:

- Mikroservisler arası istek akışını takip etme ve hata tespiti.
- OpenTelemetry, Zipkin, Jaeger gibi araçlarla dağıtık izleme uygulama.

Uygulama Performans Yönetimi (APM) Araçları:

- Dynatrace, New Relic gibi APM araçlarının dağıtık sistemlerde performans izleme ve optimizasyonundaki rolü.
- .NET uygulamaları için APM entegrasyonları.

Modül 16: Felaket Kurtarma ve Yüksek Erişilebilirlik

Aktif-Aktif ve Aktif-Pasif Çoklu Bölge Dağıtım Mimarileri (Multisite Deployment Pattern):

- Bir uygulamanın birden fazla lokasyonda (region) çalıştırılmasıyla yüksek erişilebilirlik ve felaket kurtarma.
- Aktif-Aktif (tüm bölgelerin trafiği eş zamanlı işlemesi) ve Aktif-Pasif (birincil bölgenin aktif olması) yaklaşımlarının incelenmesi.

RPO (Kurtarma Noktası Hedefi) ve RTO (Kurtarma Süresi Hedefi) Belirleme:

- Felaket durumunda veri kaybı toleransı ve servis geri dönüş süresi hedeflerinin belirlenmesi.
- İş sürekliliği ve felaket kurtarma planlarının oluşturulması.

Küresel Yük Dengeleme, DNS Tabanlı Trafik Yönetimi:

- Azure Traffic Manager, AWS Route 53 gibi servislerle küresel trafik yönlendirme stratejileri.
- Hands-on: .NET Core uygulaması için çoklu bölge dağıtım senaryosu ve trafik yönlendirme testi.

Modül 17: Sistem Dayanıklılığı ve Hata Yönetimi Derinlemesine

Aşırı Yükleme Deseni (Busy Signal Pattern), Rate Limiting ve Throttling:

- Sistem aşırı yük altında olduğunda yeni istekleri nazikçe reddetme ve kullanıcıya bildirme.
- API'lara gelen istekleri sınırlama (rate limiting) ve kullanıcı bazında kontrol (throttling).
- Hands-on: .NET Core Web API'de Rate Limiting ve Busy Signal pattern uygulaması.

Chaos Engineering (Kaos Mühendisliği) Prensipleri:

- Sistemlerin dayanıklılığını test etmek için kasıtlı olarak hatalar ve kesintiler oluşturma.
- .NET uygulamalarında Chaos Engineering uygulamaları ve araçları (Chaos Monkey).

Gelişmiş Hata Tolerans Mekanizmaları:

- Bulkhead (Bölme Duvarı) deseni ile kaynak izolasyonu.
- Fallback (Geri Dönüş) deseni ile başarısız servis çağrıları için alternatif davranışlar tanımlama.

Modül 18: Gelişmiş Veri Parçalama ve Yönetimi

Hibrit Veri Mimarileri ve Veri Yerleşimi (Data Locality) Optimizasyonları:

- Farklı veri türleri ve erişim desenleri için farklı veritabanı teknolojilerinin kombinasyonu.
- Verinin işleneceği konuma yakın tutulmasının performans avantajları.

Çok Kiracılı Veritabanı Parçalama Stratejileri:

- Kiracılar arası veri izolasyonu ve performans için dinamik parçalama.
- Database Sharding Pattern için gelişmiş anahtar stratejileri ve re-sharding.

Veri Göçü, Yeniden Parçalama ve Parçalanmış Veritabanı Bakımı:

- Mevcut veritabanlarının parçalı mimariye taşınması.
- Artan veri hacmi ve yük ile shards'ın yeniden dağıtılması.
- Parçalanmış veritabanlarının düzenli bakımı ve optimizasyonu.



Modül 19: Bulut Yerel Güvenlik Mimarileri

Güvenli Yazılım Geliştirme Yaşam Döngüsü (SSDLC) ve DevSecOps Kültürü:

- Güvenliğin tüm geliştirme ve operasyon süreçlerine entegrasyonu.
- Otomatik güvenlik taramaları ve sürekli güvenlik izleme.

Güvenlik Testleri: SAST, DAST, SCA ve Penetrasyon Testleri:

- Statik Uygulama Güvenlik Testi (SAST), Dinamik Uygulama Güvenlik Testi (DAST).
- Yazılım Bileşen Analizi (SCA) ve üçüncü parti bağımlılıkların güvenliği.
- Penetrasyon testleri ve zafiyet değerlendirmesi.

Kimlik ve Erişim Yönetimi (IAM) için Gelişmiş Bulut Stratejileri:

- Azure Active Directory, AWS IAM gibi bulut tabanlı kimlik yönetim sistemlerinin derinlemesine incelenmesi.
- Rol tabanlı erişim kontrolü (RBAC) ve politikalarla en küçük ayrıcalık prensibi.

Modül 20: Gelişmiş CI/CD ve Otomasyon

GitOps ve Infrastructure as Code (IaC) Prensipleriyle Altyapı Yönetimi:

- Git repository'sini altyapının tek gerçek kaynağı olarak kullanma.
- Terraform, Azure Resource Manager (ARM) şablonları, AWS CloudFormation ile altyapının kod olarak yönetimi.

Mavi/Yeşil Dağıtım (Blue/Green Deployment) ve Kanarya Sürümü (Canary Release):

- Kesintisiz dağıtım stratejileri ve risk azaltma teknikleri.
- Yeni sürümlerin küçük bir kullanıcı grubuna aşamalı olarak sunulması.

Sıfır Kesintiyle Dağıtım ve Geri Alma Mekanizmaları:

- Canlı sistemlerde kesintisiz güncelleme stratejileri.
- Hatalı dağıtımlarda hızlı ve güvenli geri alma (rollback) yetenekleri.
- Hands-on: .NET Core uygulamasının Blue/Green veya Canary dağıtımı için bir CI/CD pipeline'i oluşturma.

Modül 21: Modern Dağıtık Sistem Desenleri ve Karşılaştırmalar

CQRS (Command Query Responsibility Segregation) ve Event Sourcing Desenleri:

- Okuma ve yazma işlemlerini ayırma (CQRS) ve olayları tek gerçek kaynak olarak kullanma (Event Sourcing).
- Bu desenlerin .NET Core ve DDD (Domain-Driven Design) ile entegrasyonu.

Sunucusuz (Serverless) Mimariler, FaaS (Fonksiyon-as-a-Servis) ve Kullanım Senaryoları:

- Azure Functions, AWS Lambda gibi sunucusuz platformların avantajları ve kısıtları.
- Hands-on: .NET Core ile sunucusuz bir fonksiyon uygulaması geliştirme ve buluta dağıtma.

Dağıtık Önbellekleme (Distributed Caching) ve Veri Tutarlılık Desenleri:

- Redis, Memcached gibi dağıtık önbellekleme çözümleri.
- Cache-Aside, Write-Through, Write-Behind gibi önbellekleme desenleri ve .NET Core uygulamalarında uygulanması.

Modül 22: Yüksek Performanslı .NET Core Uygulama Geliştirme

.NET Core'da Bellek Yönetimi ve Garbage Collection Optimizasyonları:

- Bellek sızıntılarını (memory leaks) tespit etme ve önleme.
- Garbage Collector'ın çalışma prensipleri ve performans için ayarları.

Asenkron Programlama ve Paralel İşleme Teknikleri:

- async/await kullanımı, Task Parallel Library (TPL) ile paralel işlemler.
- Yüksek eşzamanlılık (concurrency) ve tepkisellik (responsiveness) sağlama.

Performans Profil Çıkarma, Benchmarklama ve Optimizasyon Araçları:

- Visual Studio Profiler, BenchmarkDotNet gibi araçlarla .NET uygulamalarının performansını analiz etme.
- Kodu optimize etme teknikleri ve best practices.
- Hands-on: .NET Core kodunda performans darboğazları bulma ve optimize etme çalışması.

Modül 23: Bulut Bilişim Maliyet Optimizasyonu

Bulut Kaynaklarının Verimli Kullanımı ve Maliyet Yönetimi Stratejileri:

- Gereksiz kaynak kullanımını tespit etme ve azaltma.
- FinOps prensipleri ve bulut maliyetlerini yönetme yaklaşımları.

Gelişmiş Maliyet İzleme, Bütçeleme ve Uyarı Sistemleri:

- Bulut sağlayıcılarının maliyet yönetim araçlarını (Azure Cost Management, AWS Cost Explorer) kullanma.
- Bütçe belirleme ve maliyet aşımalarında otomatik uyarılar.

Rezervasyonlar, Spot Instance'lar ve Tasarruf Planları ile Maliyet Düşürme:

- Uzun süreli taahhütlerle indirimli fiyatlardan yararlanma.
- Kesintiye uğrayabilen ancak daha ucuz olan Spot Instance'ların kullanımı.
- Bulut sağlayıcılarının sunduğu tasarruf planlarını değerlendirme.

EĞİTİM SÜRESİ

- 7 Gün
- Ders Süresi: 50 dakika
- Eğitim Saati: 10:00 - 17:00

Eğitim formatında eğitimler 50 dakika + 10 dakika moladır. 12:00-13:00 saatleri arasında 1 saat yemek arasındaki verilir. Günde toplam 6 saat eğitim verilir. 7 günlük formatta 42 saat eğitim verilmektedir.

Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katılırlar. Ayrıca farklı remote çalışma araçları da eğitmen tarafından tüm katılımlara sunulur. Katılımcılar bu araçları kullanarak eğitimlere katılırlar.

Eğitim içeriğinde github ve codespace kullanılır. Katılımcılar bu platformlar üzerinden örnek projeler oluşturur ve eğitmenle birlikte eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Eğitim yapay zeka destekli kendi kendine öğrenme formasyonu ile tasarlanmıştır. Katılımcılar eğitim boyunca kendi kendine öğrenme formasyonu ile eğitimlere katılırlar. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.

EĞİTİM YÖNTEMİ

Eğitimlerimiz, pratik ve uygulamalı öğrenmeyi merkeze alır. Her teorik konuyu, bilginin pekişmesi ve gerçek dünya becerilerinin gelişmesi için yoğun laboratuvar senaryolarıyla destekleriz. Katılımcılar, öğrendiklerini kendi elleriyle uygulayarak kalıcı bilgi edinirler. Ayrıca, sınavlara hazırlık sürecinde resmi dokümantasyonun etkin kullanımı ve sınav simülasyonları ile katılımcıların hem bilgi hem de zaman yönetimi becerilerini en üst seviyeye çıkarmayı hedefleriz. Öğrenme süreci boyunca sürekli geri bildirim ile katılımcıların gelişimini yakından takip eder, eksik noktalarını belirleyerek kişiselleştirilmiş destek sunarız.

KATILIMCILARDAN BEKLENTİLERİMİZ



- Temel Bulut Kavramları ve Servis Bilgisi Bulut tasarım kalıpları, genel bulut prensipleri üzerine inşa edilir. Katılımcıların IaaS, PaaS, SaaS gibi temel bulut hizmet modellerini, sanal makineler, depolama, ağ iletişimi, veritabanları gibi yaygın bulut servislerini (örneğin AWS, Azure veya GCP'den herhangi biri üzerinde genel bir aşinalık) anlamaları beklenir. Bu temel bilgi, kalıpların bulut ortamında nasıl uygulandığını daha kolay kavramayı sağlayacaktır.
- Yazılım Mimarisi ve Geliştirme Temelleri Eğitim, yazılım mimarisi prensipleri ve geliştirme süreçleriyle yakından ilişkilidir. Katılımcıların mikroservisler, API'ler, dağıtık sistemler gibi kavramlara aşina olmaları ve genel yazılım geliştirme süreçlerine dair temel bir anlayışa sahip olmaları beklenir. Bu, tasarım kalıplarının mevcut veya yeni geliştirilecek sistemlere entegrasyonunu anlamayı kolaylaştıracaktır.
- Problem Çözme ve Analitik Düşünme Becerisi Tasarım kalıpları, belirli mimari problemleri çözmek için geliştirilmiştir. Katılımcıların karşılaştıkları sistem gereksinimlerini veya mevcut mimari zorlukları analiz etme, bu zorluklara neden olan kök sorunları tespit etme ve potansiyel çözümleri değerlendirme yeteneğine sahip olmaları önemlidir. Bu, doğru tasarım kalıbını seçmek için kritik bir beceridir.
- Aktif Katılım ve Tartışmaya Açıklık Tasarım kalıpları, farklı senaryolara farklı şekillerde uygulanabilir. Katılımcıların ders içi tartışmalara aktif olarak katılmaları, kendi deneyimlerini paylaşmaları, soru sormaktan ve farklı yaklaşımları sorgulamaktan çekinmemeleri beklenir. Bu interaktif yaklaşım, kolektif öğrenmeyi ve farklı bakış açılarının keşfini zenginleştirecektir.
- Uygulamalı Yaklaşım Motivasyon Eğitim, kalıpların teorik tanımının ötesinde, gerçek dünya örnekleri ve senaryolar üzerinden uygulamalarına odaklanır. Katılımcılarımızın verilen örnek olayları ve pratik senaryoları anlamaya, hatta varsa kendi projelerindeki mimari sorunlara kalıpları nasıl uygulayabileceklerini düşünmeye istekli olmaları en büyük beklentimizdir. Bu, bilgiyi pratiğe dökme becerisini geliştirecektir.

HEDEF KİTLE

Bu eğitim, bulut tabanlı sistemlerin mimarisi ve tasarımı konusunda uzmanlaşmak isteyen, aşağıdaki rollere sahip veya bu rollere geçiş yapmayı hedefleyen profesyoneller için idealdir:

1. Bulut Mimarları (Cloud Architects) Karmaşık bulut çözümlerini tasarlayan, altyapı stratejilerini belirleyen ve teknoloji seçimleri yapan mimarlar için bu eğitim, tasarımlarını kanıtlanmış kalıplarla güçlendirerek daha dayanıklı, ölçeklenebilir ve maliyet etkin mimariler oluşturmalarına yardımcı olur.
2. Yazılım Mimarları (Software Architects) Uygulama düzeyinde mimari kararlar alan ve yazılım sistemlerinin genel yapısını tasarlayan yazılım mimarları için bulut tasarım kalıpları, uygulamalarını bulutun dinamik yapısına uygun hale getirmelerini ve dağıtık sistemlerin zorluklarını aşmalarını sağlar.
3. Kıdemli Yazılım Geliştiriciler Bulut tabanlı uygulamalar geliştiren ve kodlama standartlarının ötesinde mimari düşünme becerilerini geliştirmek isteyen kıdemli geliştiriciler için bu eğitim, yazdıkları kodun daha büyük sistem içindeki rolünü anlamalarına ve daha sağlam bileşenler tasarlamalarına olanak tanır.
4. DevOps ve Site Reliability Engineering (SRE) Uzmanları Uygulama dağıtımı, operasyonları, izleme ve otomasyondan sorumlu DevOps ve SRE uzmanları için bulut tasarım kalıpları, operasyonel verimliliği artıracak, hata toleransını yükseltecek ve sistemlerin çalışma zamanı davranışlarını optimize edecek mimarileri anlamalarını ve uygulamalarını sağlar.
5. Çözüm Mimarları ve Danışmanlar Müşteri ihtiyaçlarına yönelik bulut tabanlı çözümler tasarlayan ve danışmanlık hizmeti veren profesyoneller için bu eğitim, farklı senaryolara uygun en iyi mimari yaklaşımları belirleme ve müşterilerine stratejik değer katma yetkinliği kazandırır.
6. Bulut Bilişim Alanında Uzmanlaşmak İsteyen Herkes Yukarıdaki unvanlara sahip olmasa da, temel bulut bilgisine sahip ve kariyerini bulut mimarisi ve ileri seviye bulut mühendisliği alanında şekillendirmek isteyen, problem çözmeye ve analitik düşünmeye hevesli tüm teknoloji profesyonelleri bu eğitime katılabilir.

"

Kurumsal size özel eğitimler hazırlıyoruz. Her eğitim yeni bir heyecan.

Vebende A.Ş.

Kurumsal Terzi Usulü Butik Eğitimler.

Size özel hazırlanan seminer, danışmanlık, eğitim ve hizmetlerimizle yüksek verim elde edin. Paranızı boşa harcamayın. Zaman çok değerli.

Her Eğitim Yeni Bir Heyecan

2000 yılından günümüze devam eden eğitim heyecanı. Uluslararası tecrübe, proje geliştirme deneyimleri, danışmanlıklar ve arge mühendislik deneyimlerimizi ülkemize sunmak için yeni bir konsept tasarladık. Sizi dinliyor, takımınıza uygun özel içerikler ile hazırlanmış eğitimler hazırlıyoruz. Her eğitim özel bir çalışma, içerik üretimi, uygulama örnekleri hazırlıkları, sunumlar hazırlamayı gerektiriyor. Aynı eğitimi yönetim kadrosuna farklı, teknik ve mimar ekiplerinize farklı içerikler ile hazırlıyoruz. Her eğitim yeni bir macera ve heyecan.



Bizimle İletişime Geçin

iletisim@vebende.com.tr

www.vebende.com.tr

İzmir - Türkiye

Kurumsal Eğitimler

www.vebende.com.tr





İletişime Geçin



Whatsapp: 0542 5505704

Tel: 0532 512 7811



iletisim@vebende.com.tr



www.vebende.com.tr



Adalet Mah. Manas Bulvarı
No:39 İç Kapı NO: 3107 Bayraklı
İZMİR



Sürekli Güncellenen Eğitimlerimizi Sitemizden
Takip Edin.

“

Kurumsal terzi usulu size özel
eğitimler ve uluslararası deneyime
sahip eğitmenler ile çalışmanın keyfi

”



Size Neler Sunuyoruz?

- Kitaptan okunan eğitimler sunmuyoruz. Sizin için özel hazırlanan içerikler ve uygulamalı eğitimler hazırlıyoruz.
- Her eğitim için sunumlar, materyaller, örnek senaryolar size özel hazırlıyoruz.
- Her eğitim için katılımcılara özel github repoları hazırlanıyor. Eğitimden sonra da katılımcılar hayat boyu kaynaklara ulaşmaya devam edebilsinler diye, **“katılımcılar ve eğitmenle birlikte yapılan tüm çalışmaları”** github repolarında saklıyoruz.



Misyonumuz

1

Ülkemizde dünyanın en güncel teknolojilerini kazandırmak. En güncel teknolojilerine hakim takımlarına katkı sağlamak.

2

Alışılmışın dışında en güncel teknolojileri deneyimli eğitmenler ve uygulamalar ile sunmak.

3

Siber güvenlik konusunda hassas çalışmalar sunarak, çağımızın büyük kabusu siber tehditler konusunda deneyimli bilgili takımların oluşmasına katkı sağlamak.