CKS

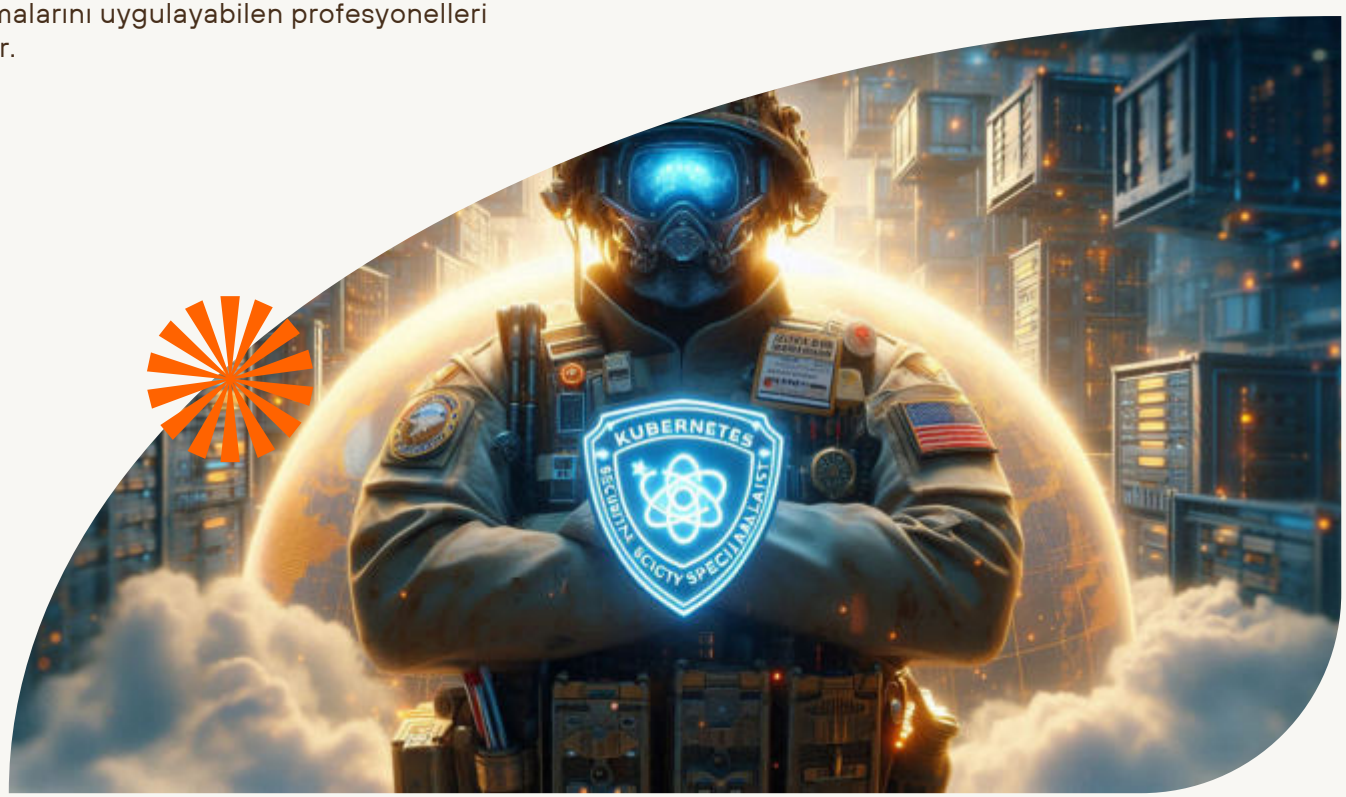
Certified Kubernetes Security Specialist (CKS), Cloud Native Computing Foundation (CNCF) ve Linux Foundation tarafından geliştirilen, Kubernetes kümelerini ve bu kümeler üzerinde çalışan uygulamaları güvenli hale getirme konusunda derinlemesine bilgi ve pratik becerilere sahip olduğunuzu kanıtlayan bir sertifikasyondur.

Bu sertifika, özellikle Kubernetes'in oluşturma, dağıtım ve çalışma zamanı (build, deployment and runtime) aşamalarında güvenlik en iyi uygulamalarını uygulayabilen profesyonelleri hedefler.

CKS

KURUMSAL ÖZEL EĞİTİM

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



Vebende

CKS EĞİTİMİ

Neden CKS Eğitimi?

1. **Artan Siber Güvenlik Tehditleri:** Günümüzün dijital dünyasında siber saldırılar giderek daha karmaşık hale geliyor. Özellikle bulut tabanlı ve kapsayıcılı uygulamalar, yeni güvenlik açıklarına yol açabiliyor. CKS, bu tehditlere karşı Kubernetes ortamlarını koruma konusunda uzmanlık sağlar.
2. **Yüksek Talep ve Kariyer Fırsatları:** Kubernetes, modern uygulama geliştirme ve dağıtımının temel taşı haline gelmiştir. Şirketler, Kubernetes altyapılarını güvenli bir şekilde yönetebilecek yetenekli profesyonellere büyük ihtiyaç duymaktadır. CKS sertifikası, bu alandaki uzmanlığınızı kanıtlayarak iş piyasasında sizi öne çıkarır ve daha iyi kariyer fırsatlarına kapı açar.
3. **Kapsamlı Güvenlik Bilgisi:** CKS, sadece temel Kubernetes bilgisi değil, aynı zamanda küme kurulumundan mikroservis güvenliğine, tedarik zinciri güvenliğinden çalışma zamanı izlemeye kadar geniş bir yelpazede güvenlik konularını kapsar. Bu sayede, Kubernetes güvenliğine bütünsel bir yaklaşımla hakim olursunuz.
4. **Pratik ve Uygulamalı Beceriler:** CKS sınavı, teorik bilgiden ziyade gerçek dünya senaryolarında problem çözme yeteneğinizi ölçen uygulamalı bir sınavdır. Bu, sertifikayı alan kişilerin sadece bilgiye sahip olmakla kalmayıp, aynı zamanda bu bilgiyi pratik olarak uygulayabildiğini gösterir.
5. **Kurumsal Güven ve İtibar:** CKS sertifikası, Cloud Native Computing Foundation (CNCF) ve Linux Foundation gibi sektörün önde gelen kuruluşları tarafından desteklenmektedir. Bu da sertifikanın sektörde yüksek bir itibara sahip olmasını sağlar. İşverenler ve müşteriler için, CKS sahibi bir profesyonel, güvenilir ve yetkin bir güvenlik uzmanı demektir.

EĞİTİM İÇERİĞİ

Modül 1: Kubernetes Güvenliğine Giriş ve Temel Mimari

Kubernetes Güvenliğine Giriş:
- Kubernetes güvenlik modelinin temel bileşenleri ve katmanları (Cluster, Node, Pod, Container seviyeleri).
- Güvenlik bağlamında Kontrol Düzlemi (Control Plane) ve Çalışan Düzlemi (Worker Plane) bileşenleri (API Server, etcd, Scheduler, Controller Manager, Kubelet, Kube-proxy, Container Runtime).

Yaygın Güvenlik Zafiyetleri ve Tehdit Vektörleri:
- Kapsayıcı kaçış senaryoları, API Server zafiyetleri, hassas veri ifşası, yetkisiz erişim, DDoS saldırıları.
- Güvenlik açıklarının neden uygulama geliştiricileri için önemli olduğu.

Güvenlik Bakış Açısıyla Kubernetes Mimarisi:
- Mimarinin güvenlik kararlarına nasıl etki ettiğini anlama (örneğin, etcd güvenliği, Kubelet güvenliği).
- Kapsayıcı izolasyonu ve güvenlik duvarları (NetworkPolicy) rolü.

Modül 2: Temel Kimlik Doğrulama ve Yetkilendirme (AuthN/AuthZ)

API Server'a Erişimde Kullanıcı ve Servis Hesaplarının Rolü:
- Kullanıcı hesapları (normal users) ve Servis Hesapları (Service Accounts) arasındaki farklar.
- Geliştiricilerin API ile etkileşimi ve güvenlik prensipleri.

Kubeconfig Dosyalarının Yapısı ve Güvenliği:
- kubeconfig dosyasının içeriği (cluster bilgisi, kullanıcı kimlik bilgisi, bağlam - context).
- kubeconfig dosyalarının güvenli saklanması ve paylaşımının önemi.

Kimlik Doğrulama (Authentication) Mekanizmaları:
- Sertifika tabanlı kimlik doğrulama.
- Token tabanlı kimlik doğrulama (Bootstrap Tokens, ServiceAccount Tokens).
- Webhook kimlik doğrulama.

Yetkilendirme (Authorization) Mekanizmaları:
- ABAC (Attribute-Based Access Control) - Kısa tanıtım.
- RBAC (Role-Based Access Control) - Temel prensipler ve önemi.
- Webhook yetkilendirme.

Modül 3: Role-Based Access Control (RBAC) Temelleri

RBAC Objeleri: Role, ClusterRole, RoleBinding, ClusterRoleBinding:
- Namespace scoped Role ve cluster-wide ClusterRole arasındaki fark.
- Yetkilendirme bağlama objeleri: RoleBinding ve ClusterRoleBinding.
- .NET uygulamanızın ihtiyaç duyacağı minimum yetkileri belirleme.

Kaynaklar, Fiiller (Verbs) ve API Grupları Kavramları:
- Kubernetes API kaynakları (Pods, Deployments, Services, Secrets vb.).
- İzin verilen eylemler (get, list, watch, create, update, delete, patch).
- API Grupları (apps, batch, rbac.authorization.k8s.io vb.).

Temel RBAC Politikaları Tanımlama ve Uygulama:
- kubectl ile temel Role ve RoleBinding tanımlama.
- .NET geliştirme süreçlerinizde karşılaşabileceğiniz yaygın RBAC senaryoları (Deployment, Service, Pod yönetimi izinleri).

Modül 4: Service Accounts ve Güvenlik Riskleri

Podların API Server ile Etkileşimi için Servis Hesapları Kullanımı:
- Podların kendi kimliği olarak ServiceAccount kullanımı.
- .NET uygulamasının Kubernetes API'ye erişmesi gerektiğinde ServiceAccount ayarları.

Otomatik Mount Edilen ServiceAccount Tokenlerinin Riskleri:
- Varsayılan olarak Podlara mount edilen Secret içerisindeki token.
- Token çalınması durumunda ortaya çıkabilecek güvenlik zafiyetleri.

Varsayılan (default) Servis Hesapları ve Güvenli Alternatifler:
- Her namespace'deki default ServiceAccount'un kullanım riskleri.
- Uygulamalara özel ServiceAccount oluşturma ve automountServiceAccountToken: false kullanımı.

Modül 5: Secrets Yönetimi Temelleri ve Güvenlik Hassas Verilerin (Şifreler, API Anahtarları)

Kubernetes Secrets Objeleriyle Yönetimi:
- Secret objelerinin yapısı ve base64 kodlama.
- .NET uygulamanızın veritabanı bağlantı stringleri, API anahtarları gibi bilgileri yönetme.

Secrets Objelerinin Varsayılan Depolama Mekanizması (etcd) ve Riskleri:
- etcd'deki Secret'ların şifrelenmemesi durumundaki riskler.
- etcd şifrelemesinin (Encryption at Rest) önemi.

Podlar İçinde Secrets Objelerini Kullanma Yöntemleri:
- Volume mount olarak Pod'a bağlama (spec.volumes, spec.containers.volumeMounts).
- Ortam değişkeni olarak kullanma (spec.containers.envFrom, spec.containers.env).
- Uygulama kodunuzda Secret verisine güvenli erişim.

Modül 6: ConfigMaps ve Güvenlik Riskleri

Yapılandırma Verilerinin ConfigMaps ile Yönetimi:
- Uygulama yapılandırmaları, ayar dosyaları (örn: .json, .xml, .yml) için ConfigMap kullanımı.
- .NET uygulamasının ayarlarını ConfigMap aracılığıyla dışarıdan yönetme.

Yanlış Yapılandırılmış ConfigMaps'in Ortaya Çıkarabileceği Hassas Veri Riskleri:
- Secret olması gereken bilgilerin yanlışlıkla ConfigMap içinde tutulması.
- Geliştirme veya test ortamlarındaki dikkatsiz yapılandırmaların riskleri.

ConfigMaps ve Secrets Arasındaki Ayrım:
- Hangi verinin Secret hangisinin ConfigMap olması gerektiği konusunda en iyi uygulamalar.

EĞİTİM İÇERİĞİ

Modül 7: Pod Güvenlik Bağlamı (Security Context) Temelleri

Pod ve Container Düzeyinde Temel Güvenlik Ayarları:
- runAsUser, runAsGroup, fsGroup ile süreçlerin hangi kullanıcı/grup kimliğiyle çalışacağının belirlenmesi.
- .NET uygulamanızın kapsayıcı içinde minimum yetkiyle çalışmasını sağlama.

Ayrıcalıklı (Privileged) Kapsayıcı Çalıştırmanın Riskleri ve Önlenmesi:
- privileged: true ayarının tehlikeleri ve host sisteme erişim.
- Bu ayarın kullanımından kaçınma veya katı kurallarla sınırlama.

Dosya Sistemi ve Kullanıcı Kimliği Tabanlı Yetkilendirmeler:
- Kapsayıcı içindeki süreçlerin dosya sistemi üzerindeki yetkilerini yönetme.
- Geliştiricilerin Pod manifestlerinde bu ayarları nasıl yapacağını anlama.

Modül 8: NetworkPolicy Temelleri

Podlar Arasındaki Ağ İletişimini Kısıtlama ve Kontrol Etme:
- Mikro segmentasyon prensibi ve NetworkPolicy'nin rolü.
- .NET mikroservislerinizin yalnızca belirli servislerle konuşmasını sağlama.

Ingress (Gelen) ve Egress (Çıkan) Trafik Kuralları:
- Pod'a kimlerin erişebileceğini (Ingress) ve Pod'un kimlere erişebileceğini (Egress) tanımlama.

Temel NetworkPolicy Tanımlama ve Uygulama Senaryoları:
- Label seçicilerle Pod gruplarını hedefleme.
- Basit bir web uygulaması ve veritabanı arasındaki iletişimi güvenli hale getirme örneği.

Modül 9: Kapsayıcı İmaj Güvenliği Temelleri

Güvenlik Açıkları İçeren İmajların Riskleri:
- Popüler imajlardaki bilinen zafiyetler (CVE'ler).
- Güvenlik açıkları içeren imajların üretim ortamına dağıtımının sonuçları.

İmaj Tarama (Image Scanning) Kavramı ve Önemi:
- Build pipeline'ınızın bir parçası olarak imaj taramayı ekleme.
- Farklı katmanlardaki zafiyetleri anlama.

Temel İmaj Güvenliği İyi Uygulamaları:
- Küçük ve minimal taban imajlar (Alpine, Distroless).
- Çalışma zamanında root kullanıcısı kullanmaktan kaçınma (USER direktifi).
- Çok aşamalı (multi-stage) buildler.

Modül 10: Audit Log Temelleri

Kubernetes API Server Üzerinde Gerçekleştirilen İşlemlerin Kaydı:
- Kimin, ne zaman, neyi, nerede yaptığının kaydını tutma.
- API isteklerinin detayları ve log seviyeleri.

Güvenlik Olaylarını İzleme ve Denetleme için Audit Log Kullanımı:
- Yetkisiz erişim denemeleri, hassas kaynaklara erişim gibi güvenlik olaylarını tespit etme.
- Bir ihlal sonrası adli analiz (forensics) için logların önemi.

Temel Audit Log Yapılandırması ve Okuma:
- kube-apiserver için temel Audit Policy ayarları.
- Logları okuma ve anlama (kubectl logs).

Modül 11: Detaylı RBAC Yapılandırması ve En İyi Uygulamalar

Minimal Yetkilendirme Prensibi (Least Privilege) ile RBAC Tasarlama:
- Kullanıcı ve ServiceAccount'lara yalnızca işlerini yapmaları için gereken minimum yetkileri atama.
- Developer, QA, CI/CD süreçleri için örnek minimal yetkilendirme politikaları.

Role Aggregation ve Rol Birleştirme Senaryoları:
- Birden fazla ClusterRole'ü otomatik olarak tek bir ClusterRole altında toplama (aggregationRule).
- Karmaşık yetkilendirme ihtiyaçlarını yönetme.

Kullanıcı ve Grup Bazlı RBAC Atamaları:
- Kullanıcıların (user) ve grupların (group) RoleBinding / ClusterRoleBinding ile yetkilendirilmesi.
- Kurumsal dizin servisleri (LDAP, Active Directory) ile entegrasyonun RBAC'a etkisi.

RBAC Politikalarının Test Edilmesi ve Denetlenmesi:
- kubectl auth can-i komutu ile bir kullanıcının belirli bir eylemi yapıp yapamayacağını kontrol etme.
- Mevcut RBAC yapılandırmasını denetleme araçları.

Modül 12: Pod Security Standards (PSS) ve Uygulama

PSS Modları (Privileged, Baseline, Restricted) ve Anlamları:
- Kubernetes topluluğu tarafından tanımlanan güvenlik profilleri.
- Her bir modun izin verdiği veya kısıtladığı güvenlik ayarları (runAsUser, capabilities, volume mounts vb.).

Namespace Seviyesinde PSS Uygulama ve Zorunlu Kılma:
- PodSecurity Admission Controller'ı kullanarak namespace'lere PSS modları atama.
- enforce, audit, warn modlarının farkları ve kullanım senaryoları.

PSS Kullanarak Podların Güvenlik Profilini Güçlendirme:
- Geliştiricilerin yazdığı Pod manifestlerinin PSS standartlarına uyumunu kontrol etme.
- Uygulamalarınız için hangi PSS modunun uygun olduğuna karar verme.

Modül 13: Gelişmiş NetworkPolicy Senaryoları

Karmaşık Label Seçiciler ile Detaylı Ağ Segmentasyonu:
- podSelector, namespaceSelector, ipBlock kombinasyonları ile hassas kurallar yazma.
- .NET uygulamalarınızın katmanları (WebAPI, Business Logic, Data Access) arasındaki iletişimi sadece gerekli portlar üzerinden kısıtlama.

Farklı Namespace'ler Arası İletişim Politikaları:
- Farklı takımların veya ortamların namespace'leri arasındaki güvenli iletişimi tanımlama.
- namespaceSelector kullanımı.

CIDR Blokları ve Harici Servis Erişim Kuralları:
- Cluster içindeki Podların dışarıdaki belirli IP aralıklarına veya servislere (veritabanı, API geçidi) erişimini kontrol etme (Egress kuralları).

NetworkPolicy Uygulayıcıları (CNI) ve Yetenekleri:
- Calico, Cilium, Flannel gibi farklı CNI eklentilerinin NetworkPolicy desteği ve ek güvenlik özellikleri (MTLS, eBPF tabanlı politikalar).

EĞİTİM İÇERİĞİ

Modül 14: Secrets Yönetiminde İleri Teknikler

Harici Secret Store Entegrasyonları:
- HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, GCP Secret Manager gibi harici sistemlerle entegrasyonun avantajları.
- Kubernetes'in harici Secret'lara nasıl eriştiği (Secret Store CSI Driver).
- Geliştiricilerin bu entegrasyonu uygulama koduna yansıtması.

Sealed Secrets gibi Araçlarla Hassas Veriyi Git'te Güvenle Saklama:
- Secret manifestlerinin Git gibi versiyon kontrol sistemlerinde şifreli olarak saklanması.
- Hassas verinin açıkta kalmasını engelleme.

Secrets Rotasyonu ve Yaşam Döngüsü Yönetimi:
- Secret'ların periyodik olarak değiştirilmesinin (rotate) önemi.
- Uygulama kesintisi olmadan Secret güncelleme stratejileri.

Modül 15: Güvenlik Bağlamı Derinlemesine: Seccomp ve AppArmor

Seccomp (Secure Computing Mode) Profilleri ve Sistem Çağrısı Kısıtlamaları:
- Kapsayıcıların çekirdekle etkileşimini sınırlama.
- Default (runtime/docker) Seccomp profilleri.
- Uygulamaların ihtiyaç duymadığı sistem çağrılarını engelleme.

AppArmor (Application Armor) Profilleri ve Program Davranışı Kısıtlamaları:
- Programların dosya erişimi, ağ kullanımı gibi davranışlarını profilleme.
- Ubuntu gibi dağıtımlarda AppArmor kullanımı.

Varsayılan Profillerin Kullanımı ve Özel Profil Yazımına Giriş:
- Pod manifestlerinde Seccomp ve AppArmor profillerini belirtme.
- Basit bir .NET uygulamasının ihtiyaç duyabileceği sistem çağrılarını/dosya erişimlerini belirleme.

Modül 16: Admission Controllers ve Güvenlik Politikaları

Kubernetes Kaynaklarının Oluşturulması veya Güncellenmesi Sırasında Politikaları Zorunlu Kılma:
- API Server'a gelen isteklerin kabul edilmeden veya kalıcı hale getirilmeden önce denetlenmesi/değiştirilmesi.
- Güvenlik politikalarının merkezi olarak uygulanması.

Mutating ve Validating Admission Webhook'ların Rolü:
- Kaynakları değiştiren (Mutating) veya sadece kontrol eden (Validating) webhook'lar.
- .NET uygulamalarınızın deployment manifestlerini belirli güvenlik standartlarına (örn: PSS, image policy) uyması için kontrol etme.

Yaygın Güvenlik Odaklı Admission Controller'lar:
- PodSecurity, AlwaysPullImages, LimitRanger, ResourceQuota vb.

Modül 17: Kubelet ve API Server Güvenliği Temelleri

Kubelet Yapılandırmasının Sertleştirilmesi:
- Güvenli portlar (--read-only-port=0), kimlik doğrulama (--authentication-token-webhook, --client-ca-file).
- Anonim isteklerin devre dışı bırakılması (--anonymous-auth=false).

API Server Güvenliği:
- API Server'ın dinlediği portlar (6443, 8080) ve ağ erişimi kısıtlamaları.
- Transport Layer Security (TLS) yapılandırması ve sertifika yönetimi.

Güvenli Kubelet ve API Server Etkileşimi:
- Node kimlik doğrulama (Node Authorizer) ve node restriction.

Modül 18: CIS Kubernetes Benchmarkları ve Uygulama

Center for Internet Security (CIS) Tarafından Tanımlanan Güvenlik Standartları:
- Kubernetes kümesinin kontrol düzlemi ve çalışan düzlemi bileşenleri için güvenlik ayarları.
- Konfigürasyon doğrulama adımları.

Kubernetes Kümesinin CIS Benchmarklarına Göre Denetlenmesi:
- Otomatik denetleme araçları kullanımı (kube-bench).
- Denetim çıktılarının yorumlanması ve kritik bulguların belirlenmesi.

Bulguların Yorumlanması ve Düzeltilmesi:
- Tespit edilen güvenlik açıklarını giderme adımları.
- Geliştiricilerin kendi ortamlarında uygulayabileceği konfigürasyon değişiklikleri.

Modül 19: Kapsayıcı Çalışma Zamanı Güvenliği (Runtime Security) - Temeller

Çalışan Kapsayıcılar İçindeki Şüpheli Aktiviteleri Tespit Etme:
- Bir saldırı anında veya sonrasında neler olduğunu anlama.
- Kapsayıcı içinde beklenmedik süreç çalıştırma, dosya erişimi, ağ bağlantısı gibi olaylar.

Dosya Erişimi, Süreç Çalıştırma, Ağ Bağlantıları Gibi Olayları İzleme:
- Kernel sistem çağrılarına (syscalls) dayalı izleme.
- Güvenlik olaylarının tespiti için gerekli altyapı.

Falco Gibi Araçlarla Çalışma Zamanı Kuralları Tanımlama ve Uyarı Oluşturma:
- Popüler runtime güvenlik aracı Falco'ya giriş.
- Temel Falco kuralları ve uyarı mekanizmaları.
- Uygulama davranışına özgü potansiyel tehditleri modelleme.

Modül 20: Vulnerability Scanning Araçları ve Süreçleri

Kapsayıcı İmajlarındaki Güvenlik Açıklarını Tarama Araçları:
- Trivy, Clair, Anchore gibi yaygın araçlar.
- Tarama araçlarının nasıl çalıştığı (katman analizi, veritabanı karşılaştırması).

Tarama Sonuçlarını Yorumlama ve Önceliklendirme:
- Kritik, yüksek, orta, düşük risk seviyeleri.
- Hangi güvenlik açıklarının öncelikli olarak kapatılması gerektiği.

Güvenlik Açığı Yönetimi Süreçleri:
- Tarama sonuçlarının CI/CD pipeline'ına entegrasyonu.
- Güvenlik açıklarının kapatılması ve güncel imajların dağıtımı süreci.

Modül 21: DevSecOps Temelleri ve Kubernetes Entegrasyonu

Güvenliğin Yazılım Geliştirme Yaşam Döngüsünün Her Aşamasına Entegrasyonu:
- Geliştirme, build, test, dağıtım ve operasyon aşamalarında güvenlik.
- "Shift Left" prensibi.

Kod Analizi (SAST), Bağımlılık Analizi (SCA), İmaj Tarama:
- Static Application Security Testing (SAST) ile .NET kodundaki güvenlik açıklarını bulma.
- Software Composition Analysis (SCA) ile kullanılan kütüphanelerdeki (NuGet paketleri) zafiyetleri tespit etme.
- Build sonrası imaj tarama.

CI/CD Pipeline'larına Güvenlik Adımları Ekleme:
- Güvenlik taramalarını otomatize etme.
- Güvenlik açıklarının belirli bir eşiği aşması durumunda build/deployment'ı durdurma (Security Gates).

EĞİTİM SÜRESİ

- 3 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. 3 günlük formatta 18 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 iceriğe cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun iceriğe 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

- **Linux & Komut Satırı Temelleri:** Linux ortamında rahatça çalışabilme ve temel terminal komutlarına hakimiyet (örn. ls, cd, cp, mv, rm).
- **Konteyner & Docker Bilgisi:** Konteyner kavramını, Docker'ın temel kullanımı (imaj oluşturma, çalıştırma) ve konteynerleştirme mantığını anlama.
- **YAML Okur Yazarlığı:** Kubernetes manifestlerini oluşturan YAML dosyalarını okuyabilme, anlayabilme ve basit düzenlemeler yapabilme.
- **Temel Uygulama Geliştirme Bilgisi:** Tercihen bir programlama dilinde (Python, Go, Java vb.) temel uygulama geliştirme deneyimi.

KATILIMCILARDAN BEKLENTİLERİMİZ

1. Temel Linux Bilgisi ve Komut Satırı Yetkinliği
Kubernetes'in Linux üzerinde çalışması sebebiyle, katılımcıların temel Linux komutlarına (ls, cd, cp, mv, rm, mkdir vb.) ve komut satırı (terminal) kullanımına hakim olmaları kritik önem taşır. Sınav tamamen terminal üzerinde ilerleyeceği için, Vim veya Nano gibi metin editörlerini etkin kullanabilmek de büyük bir avantaj sağlayacaktır.

2. Konteyner Temelleri ve Docker Bilgisi
Kubernetes'in konteyner orkestrasyonu aracı olması nedeniyle, katılımcıların Docker gibi bir konteyner platformunun temel kavramlarını (imajlar, konteynerler) anlamaları ve temel Docker komutlarını (docker run, docker ps) çalıştırabilmeleri beklenir. Bu bilgi, Kubernetes'teki Pod'lar ve Deployment'lar gibi yapıları kavramayı kolaylaştıracaktır.

3. Temel YAML Bilgisi
Kubernetes kaynak tanımları genellikle YAML formatında yazılır. Katılımcıların YAML'ın temel yapısını (girinti, anahtar-değer çiftleri, listeler) okuyup anlayabilmeleri ve basit YAML dosyalarını oluşturup düzenleyebilmeleri önemlidir.

4. Problem Çözme ve Analitik Düşünme Becerisi
CKA sınavının önemli bir kısmı sorun giderme (Troubleshooting) üzerine kuruludur. Katılımcıların karşılaştıkları problemleri analiz etme, mantıksal adımlar izleyerek kök nedeni bulma ve etkili çözümler üretme becerisine sahip olmaları beklenir. Resmi Kubernetes dokümantasyonunu (kubernetes.io/docs) etkin kullanabilme de bu süreçte hayati rol oynar.

5. Yoğun Pratik Motivasyonu
CKA sertifikasyonunda başarı, sadece dersleri takip etmekle değil, yoğun ve düzenli pratikle gelir. Katılımcılarımızın eğitimde verilen tüm uygulamalı laboratuvar çalışmalarını eksiksiz yapmaları, hatta kendi başlarına (Minikube, KIND gibi araçlarla) veya sınav simülatörleri (örn. Killer.sh) üzerinde bolca pratik yapmaya istekli olmaları en büyük beklentimizdir. Sınavın zaman kısıtlaması nedeniyle, hızlı ve doğru işlem yapma alışkanlığı kazanmak kritik öneme sahiptir.

HEDEF KİTLE

1. DevOps Mühendisleri
Uygulama dağıtımı, otomasyonu ve sürekli entegrasyon/sürekli teslimat (CI/CD) süreçlerinden sorumlu olan DevOps mühendisleri için CKA, Kubernetes ortamlarını daha etkin yönetme ve optimize etme yetkinliği kazandırır.

2. Sistem Yöneticileri (SysAdmins)
Sunucuların, ağ altyapısının ve uygulama platformlarının kurulumu, bakımı ve sorun gidermesinden sorumlu sistem yöneticileri, CKA ile konteynerize edilmiş iş yüklerini ve Kubernetes kümelerini yönetme becerilerini geliştirirler.

3. Bulut Mühendisleri
Genel bulut platformlarında (AWS, Azure, GCP) çalışan ve bulut tabanlı altyapıları tasarlayan, uygulayan ve yöneten mühendisler için CKA, bulut yerel (cloud-native) uygulamaları ve Kubernetes servislerini daha iyi yönetmelerini sağlar.

4. Yazılım Geliştiriciler (DevOps Rolüne Geçmek İsteyenler)
Konteyner ve Kubernetes dünyasına ilgi duyan, geliştirdikleri uygulamaları Kubernetes üzerinde dağıtmak ve yönetmek isteyen yazılım geliştiriciler için CKA, operasyonel süreçleri anlama ve DevOps rollerine geçiş yapma konusunda güçlü bir temel sunar.

5. IT Mimarları ve Çözüm Mimarları
Kurumsal IT altyapısını veya bulut çözümlerini tasarlayan mimarlar için CKA, Kubernetes'in mimari yapısını, güvenlik ve ölçeklenebilirlik prensiplerini derinlemesine anlayarak daha sağlam ve optimize edilmiş çözümler sunmalarına yardımcı olur.

6. Kubernetes Öğrenmek İsteyen Herkes
Yukarıdaki unvanlara sahip olmasa da, güçlü bir Linux temeli ve öğrenme motivasyonu olan herkes CKA eğitimine katılabilir. Özellikle kariyerini bulut bilişim ve modern altyapı yönetimi alanında şekillendirmek isteyen profesyoneller için CKA, sektörde aranan bir yetkinlik kapısıdır.

"

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

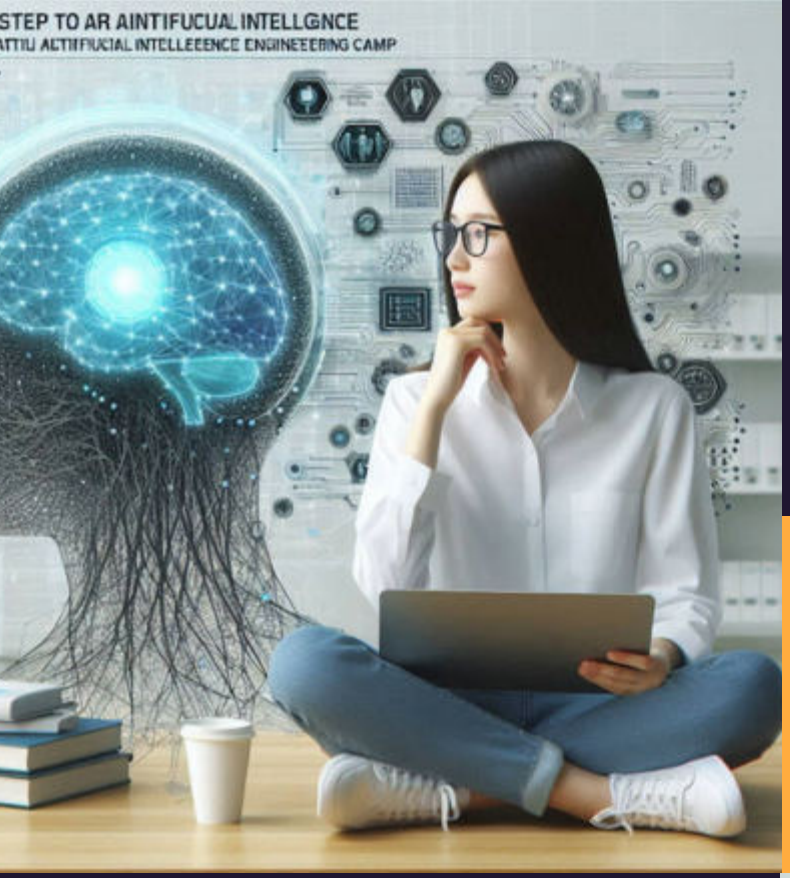
iletisim@vebende.com.tr

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 N0: 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.