

Üretimde ne kullanıyoruz.

62 teknoloji, dört halka. Hangisini üretimde çalıştırdığımız, hangisini denediğimiz, hangisinden uzak durduğumuz.

Bu belge eclit.com/tr/radar sayfasının indirilebilir hâlidir.

SÜRÜM

2026.1

TEKNOLOJİ

62

GÜNCELLEME

2026-09-08

Bu radar nedir

Bir teknoloji radarı, bir şirketin hangi teknolojiyi hangi olgunlukta kullandığını açıkça yazdığı belgedir. Satın alma kararı veren bir okuyucu için pazarlama metninden daha kullanışlıdır: “bulut konusunda uzmanız” cümlesinin aksine, doğrulanabilir.

Halka bir puan değildir. Teknolojiye duyulan güveni değil, onu kendi operasyonumuzda hangi aşamada kullandığımızı gösterir. “Değerlendir” halkasındaki bir teknoloji kötü değildir; biz henüz üretime almamışızdır.

Dört halka

BENİMSE	Benimsendi — üretimde, varsayılan tercihimiz	34
DENE	Deneniyor — gerçek işlerde sınanıyor, henüz varsayılan değil	15
DEĞERLENDİR	İnceleniyor — izliyoruz, üretime almadık	12
BEKLET	Uzak duruluyor — yeni işlerde tercih etmiyoruz	1

Benimsendi — üretimde, varsayılan tercihimiz

Platformlar

Hybrid cloud

Hibrit bulut işletim modeli. Benimsendi çünkü sahadaki gerçek varsayılan bu: kurumların çoğunda regülasyon, gecikme ya da amortisman yüzünden yerelde kalması gereken bir katman var. Soru "buluta mı geçelim" değil, hangi iş yükünün nerede duracağı.

Kubernetes platforms

Üretim Kubernetes platformları. Benimsendi: modern iş yükünün omurgası ve kendi kendini onaran yapısı operasyon yükünü gerçekten azaltıyor. Karşılığı şu: cluster kendini onarır ama cluster'ı kimse onarmaz: platform ekibi olmadan Kubernetes bir çözüm değil, ikinci bir sistem.

Managed DR platform

Yönetilen felaket kurtarma ve yedekleme platformu. Dayanıklılık katmanı.

Multi-cloud architecture

Tek sağlayıcıya bağımlı olmayan mimari. Benimsendi, ama pazarlama anlamıyla değil: amaç her iş yükünü her bulutta çalıştırmak değil, çıkış maliyetini bilerek tasarlamak. Taşınabilirliği veri katmanı belirliyor, hesaplama katmanı değil.

Private cloud

VMware ve KVM üzerinde özel bulut. Benimsendi ve Eclit'in çekirdek hizmeti: veri yerelliği, öngörülebilir maliyet ve mevcut lisans yatırımının korunması, kurumların çoğunda bu katmanı yerinde tutuyor. Bulutla rakip değil, hibrit modelin bir yarısı.

ScaleEngine

Kurum içi otomasyon platformu. Benimsendi: MSP'nin tekrar eden işini yazılıma taşıma yönündeki kendi ürünümüz. Buradaki iddia teknoloji değil model; aynı işi elli müşteride elle yapmak yerine bir kez platforma yazmak.

Araçlar

Ansible

Yapılandırma denetimi ve dağıtım otomasyonu.

ArgoCD

Kubernetes üzerinde GitOps dağıtım omurgası. Benimsendi: istenen durum Git deposunda duruyor, cluster'daki gerçek durum ondan sapınca fark görünür oluyor ve geri alma bir commit. Elle kubectl ile yapılan değişikliğin bıraktığı "kim ne zaman değiştirdi" boşluğunu kapatıyor.

Elasticsearch

Arama ve analiz motoru; log analizi altyapımızın çekirdeği. Benimsendi: büyük log hacminde sorgu performansı ve olgun sorgu dili onu varsayılan yapıyor. Bilinmesi gereken maliyeti disk ve bellek; saklama süresi baştan tasarlanmazsa cluster büyümesi kontrolden çıkıyor.

ELK / OpenSearch

Log toplama, ayrıştırma ve görselleştirme katmanı. Benimsendi: toplama ve ayrıştırmayı motordan ayrı tutmak, arama motoru değişse bile boru hattının ayakta kalmasını sağlıyor. Merkezî log yönetimi hizmetimizin taşıyıcısı.

Enterprise backup stack

Kurumsal yedekleme yığını; felaket kurtarmanın güvenilirlik katmanı.

Foreman / Katello

Sistem yaşam döngüsü ve yama otomasyonu. Benimsendi: sunucu envanteri, içerik görünümüleri ve yama dağıtımını tek yerden yürütmek, yamanın hangi sunucuya ne zaman gittiğini denetlenebilir kılıyor. Yama ve zafiyet yönetimi hizmetimizin altyapısı.

GitLab

Kod, pipeline ve iş akışının tek merkezde toplanması. Benimsendi: üç ayrı araç yerine tek yerde durması, değişikliğin kodundan dağıtımına kadar izlenebilir bir zincir bırakıyor. Denetimde "bu değişiklik nereden geldi" sorusunun cevabı tek sistemde.

Grafana

Observability görselleştirmesinde standardımız. Benimsendi: metrik, log ve iz kaynaklarını tek panelde birleştirmesi, olay sırasında üç ayrı araç arasında gezinmeyi ortadan kaldırıyor. Nöbetteki mühendisin ilk açtığı ekran.

MongoDB

Esnek şema gerektiren iş yükleri için birincil doküman veritabanımız.

MySQL

İlişkisel veritabanı altyapısı olarak desteklenir ve işletilir. Benimsendi: olgun, iyi bilinen ve çoğu uygulamanın varsayılanı. İşletme tarafında dikkat ettiğimiz iki şey replikasyon gecikmesi ve yedeğin geri yüklenerek test edilmesi; ikisi de düzenli ölçülmezse sessizce bozuluyor.

Palo Alto

Kurumsal çevre güvenliği ve yeni nesil güvenlik duvarı. Benimsendi: uygulama farkındalıklı politika ve tehdit önleme olgun. İşletmede kritik nokta politika hijyeni; yıllar içinde biriken ve kimsenin silmeye cesaret edemediği kurallar, en iyi cihazı bile yavaşlatıyor.

PostgreSQL

Yeni ve modernize edilen sistemlerde varsayılan ilişkisel veritabanı tercihimiz.

Prometheus

Metrik toplama ve uyarı üretmenin standardı. Benimsendi: çekme modeli ve etiket tabanlı veri modeli, dinamik ortamlarda elle envanter tutmayı gereksiz kılıyor. Uzun vadeli saklama ayrı bir karar; Prometheus'un kendi diski arşiv için tasarlanmadı.

Prometheus & Grafana

Observability omurgası.

Proxmox / KVM stack

Maliyet açısından verimli bulut temeli.

Terraform

Altyapı standardizasyonu.

Veeam

Sanal makine ve bulut iş yükleri için çekirdek kurumsal yedekleme platformu.

VMware ecosystem

Kurumsal özel bulutta güvenilirlik.

Yöntemler

Blameless incident culture

Olay sonrası suçlu değil sistem hatası aramak. Benimsendi, çünkü teknik bir tercih değil ölçüm meselesi: suçlanacağını bilen mühendis olayı geç bildiriyor ve eksik anlatıyor. Kayıt bozulunca kök neden analizi de bozuluyor. Post-mortem disiplini bu yüzden kültürel değil operasyonel bir gereklilik.

Customer health scoring

Hizmet sağlığını ticket sayısı ile değil güvenilirlik ölçütleriyle izlemek. Benimsendi, çünkü ticket sayısı yanıltıcı bir gösterge: iyi çalışan bir ortamda da azalır, müşteri umudu kestiğinde de. Sağlık skoru olay sıklığı, çözüm süresi ve tekrar eden kök nedenlerden hesaplanıyor.

GitOps as Operational

Altyapı, yapılandırma, politika ve değişikliğin tek kaynaktan yürümesi. Benimsendi ve bu bizim için bir araç değil işletim modeli: ortamın istenen hâli depoda duruyor, sapma ölçülebiliyor, geri alma bir commit ve her değişikliğin bir yazarı ile gereksesi var.

IaC-first service provisioning

Elle kurulum kabul edilmiyor; her şey koddan. Benimsendi: elle kurulan sunucu bir yıl sonra kimsenin tam olarak hatırlamadığı bir yapılandırma taşıyor ve felaket anında yeniden kurulamıyor. Koddan kurulan sunucu ise tekrar üretilebilir; kurtarma planının ön şartı bu.

Reliability-first architecture

Altyapı kurmak yerine çalışır zaman teslim etmek. Benimsendi ve bu bir mimari tercihten çok sözleşme tercihi: teslim edilen şey sunucu değil erişilebilirlikse, tasarım kararları da ona göre veriliyor: tek nokta arıza, kurtarma süresi ve tatbikat baştan gündemde.

Service Ownership Model

L1/L2/L3 ayrımı yerine uçtan uca sahiplik. Benimsendi: seviyeler arası devir, olay süresinin çoğunu bağlamı yeniden anlatmakla geçiriyor. Sahiplik modelinde aynı takım hem çağırıyor hem kök nedeni görüyor, o yüzden aynı arıza ikinci kez gelmiyor.

Diller ve Çerçevesler

Bash

Alt seviye otomasyon ve sistem betikleme. Benimsendi ve öyle kalacak: her sunucuda kurulu olan tek dil bu, kurtarma kabuğunda çalışan da bu. Python omurga, bash ise Python'ın yüklenemediği ya da yüklenmesini bekleyemeyeceğiniz yerlerin dili.

Python

Otomasyon omurgası ve operasyon betikleme standardımız. Benimsendi: kütüphane ekosistemi, okunabilirliği ve ekipteki yaygınlığı onu varsayılan yapıyor. Bir otomasyonun ömrü boyunca en pahalı kısmı yazılması değil, iki yıl sonra başkası tarafından okunması.

REST API-first mindset

Her yeteneğin önce API olarak tasarlanması. Benimsendi: tek bir araca bağımlılığı azaltıyor ve aracı değiştirmek zorunda kaldığınızda entegrasyonun tamamını yeniden yazmaktan kurtarıyor. Portal ve otomasyon aynı API'yi kullanıyor, ikisi arasında fark kalmıyor.

YAML-driven infra definitions

Yapılandırma standardı olarak YAML. Benimsendi: araçlar arası ortak dil olması ve sürüm kontrolünde okunabilir fark üretmesi onu varsayılan yapıyor. Bilinen zayıflığı girinti duyarlılığı; o yüzden şemayla doğrulanmayan YAML üretime çıkmıyor.

Deneniyor — gerçek işlerde sınanıyor, henüz varsayılan değil

Platformlar

Internal Developer Platform

Mühendis deneyimi platformu. Denemede: geliştiricinin altyapıyı bilmeden ortam açabilmesi teslim hızını artırıyor, ama platform bir ürün gibi bakım isteyen bir şey; sahibi olmayan IDP altı ay sonra kimsenin güncellemediği bir katmana dönüşüyor.

Multi-tenant abstraction

Hizmetin platformlaşması için çok kiracılı soyutlama katmanı. Denemede: müşteri başına ayrı kurulum yerine ortak bir platform, tutarlılığı artırıyor. Zor kısmı izolasyon: bir kiracının yükünün diğerini etkilememesi, mimarinin en pahalı gereksinimi.

Open-Source AI Stack

Kurumsal iş yükleri için açık kaynak model ve çıkarım yığınlarını deniyoruz.

Self-service portals

Altyapının ürüne dönüşmesi. Denemede: müşterinin kendi ortamını görebilmesi, aylık rapor beklemekten çok daha güçlü bir şeffaflık. Portalın gösterdiği veri gerçek zamanlı değilse ters etki yapıyor; eski veri gösteren panel, hiç panel olmamasından kötü.

Araçlar

FinOps dashboards

Müşteriye şeffaf maliyet paneli. Denemede: harcamanın hangi iş yüküne ait olduğunu göstermek ancak etiketleme disiplini varsa mümkün. Etiketsiz bir ortamda panel "bilinmeyen" kalemini büyütmekten başka bir şey yapmıyor; o yüzden panelden önce etiketleme geliyor.

LLM-based ops assistant

Ticket'tan özete, yanıt ve runbook önerisine giden dil modeli asistanı. Denemede: tekrar eden L1 taleplerinde yanıt süresini kısaltıyor. Sınırı net; müşteri verisinin modele nasıl gittiği sözleşmeyle tanımlanmadan üretimde kullanılmıyor.

n8n

Birden fazla servisi asgari kodla birbirine bağlayan iş akışı otomasyonu. Denemede: iç süreçlerde hızlı sonuç veriyor. Kuralımız şu; müşteri ortamına dokunan hiçbir akış görsel bir araçta bırakılmıyor, orası kod ve sürüm kontrolü işi.

Voice AI support agents

Sesli yapay zekâ ile L1 destek otomasyonu. Denemede ve temkinli: basit, yönlendirme amaçlı çağrılarda işe yarıyor. Kurumsal destekte arayan kişi genelde zaten bir sorun yaşıyor ve yanlış anlaşılmanın maliyeti, kazanılan süreden yüksek. İnsana devir yolu her zaman açık kalıyor.

Zerto

Kritik iş yüklerinde sürekli veri koruma (CDP) ile felaket kurtarma.

Yöntemler

AI-assisted operations

Olay özeti çıkarma, kök neden önerisi ve runbook taslağı için dil modeli desteği. Denemede tutuyoruz çünkü gürültülü bir uyarı yığnında hızlanma gerçek, ama önerinin doğruluğu ortamın ne kadar iyi etiketlendiğine bağlı; kötü etiketlenmiş bir ortamda güvenle yanlış cevap veriyor. Nöbetçi mühendisin yerine değil, önüne

konuyor.

FinOps as discipline

Maliyet kontrolünü bir operasyon KPI'ı hâline getirmek. Denemede: bulut faturasını aylık şaşkınlıktan çıkarıp haftalık göstergeye çevirmek işe yarıyor, ama tasarrufun sahibi belirlenmeden yapılan raporlama okunmayan bir rapordan ibaret kalıyor.

Platform engineering

MSP modelinin platform mühendisliğine dönüşmesi. Denemede: her müşteriye özel kurulum yerine tekrar eden kalıpları platforma taşımak, kalite dalgalanmasını azaltıyor. Riski şu; platform, müşterinin gerçek ihtiyacını karşılamıyorsa herkes onun etrafından dolaşmaya başlıyor.

Self-service infrastructure

Mühendise API, müşteriye portal. Denemede: rutin taleplerin ticket açmadan karşılanması hem müşteriye hem nöbetçiye rahatlatıyor. Sınırı yetkilendirme belirliyor; self-servis, kimin neyi yapabileceği net tanımlanmadan açılırsa denetim izini bozuyor.

Diller ve Çerçeveler

Event-driven automation

Altyapının olaya tepki veren hâle gelmesi. Denemede: zamanlanmış işlerin yerine olay tetikli akışlar koymak tepki süresini kısaltıyor, ama olay akışı gözlemlenebilir değilse hata ayıklama zorlaşıyor; zamanlanmış iş çalışmadığında belli olur, olay tetikleyicisi sessizce kaybolur.

Go

Performans ve güvenilirlik gerektiren platform araçları için. Denemede: tek ikili dosya olarak dağıtılması operasyon araçlarında büyük kolaylık, ama ekipteki Python bilgisi Go bilgisinden fazla ve bir aracın bakımı yazılmasından uzun sürüyor. Dar bir alanda deneniyor.

DEĞERLENDİR · 12

İnceleniyor — izliyoruz, üretime almadık

Platformlar

AI inference platform

Kurum içi GPU havuzu ve çıkarım altyapısı. Değerlendirmede, çünkü asıl soru teknik değil ekonomik: veri kurumdan çıkamıyorsa ya da çıkarım hacmi sürekliyse kendi donanımın anlamlı oluyor, aksi hâlde bulut GPU daha ucuz kalıyor. Kararı hacim ve veri yerelliği belirliyor.

Edge workload platforms

Dağıtık altyapının uç noktalarını yönetmek. Değerlendirmede: üretim sahası ve perakende şubesi gibi yerlerde iş yükünün yerelde çalışması gerçek bir ihtiyaç, ama uç noktaların yönetimi merkezden yapılamıyorsa filo büyüdükçe operasyon maliyeti iş yükünün kendisini geçiyor.

OpenStack

Yüksek özelleştirme isteyen özel bulut kurulumları için. Değerlendirmede: yetenek tartışmasız ama işletme maliyeti de öyle; OpenStack bir ürün değil, sürekli bir mühendislik taahhüdü. Ancak ölçek ve özelleştirme ihtiyacı o taahhüdü haklı çıkardığında öneriyoruz.

Sovereign cloud patterns

Regülasyon odaklı bulut mimari desenleri. Değerlendirmede: KVKK, BDDK ve sektörel düzenlemeler verinin nerede duracağını belirliyor ve bu desenler o kısıtı mimariye çeviriyor. Alan hareketli: düzenleme değiştikçe desen de değişiyor, o yüzden erken sabitlemiyoruz.

Araçlar

AI log analysis engines

Log yığnında istatistiksel örüntü ve sapma tespiti. Değerlendirmede, çünkü bu araçlar bilinmeyen bir sorunu bulmakta iyi ama bilinen bir sorunu tekrar bulmakta pahalı; iyi yazılmış bir uyarı kuralı aynı işi bedava yapıyor. Var olan uyarı setinin yerine değil, üstüne düşünülüyor.

Policy as Code frameworks

OPA ve benzeri çerçevelerle uyum otomasyonu. Değerlendirmede: uyum kuralını belgeden koda taşımak, denetimde "bu kural uygulanıyor mu" sorusunu ölçülebilir hâle getiriyor. Zorluk kuralı yazmak değil, kural ihlalinde ne olacağına karar vermek.

Secrets lifecycle tools

Sır yönetiminin platform katmanında otomasyon. Değerlendirmede: sırrı bir kasada tutmak çözülmüş bir problem, döndürmek değil. Asıl iş sırrın süresi dolduğunda hangi sistemin etkileneceğini bilmek; envanteri olmayan bir kasa, düzenli aralıklarla kesinti üretiyor.

Yöntemler

AI maturity scoring

Müşteri ortamının teknik sağlığını yapay zekâ destekli puanlama. Değerlendirmede, çünkü puanın kendisi kolay, puanın SAVUNULABİLİR olması zor: müşteriye "altyapınız 62 puan" demek ancak o 62 puanın hangi ölçümden geldiğini gösterebiliyorsanız işe yarıyor. Şeffaf olmayan skor, olmayan skordan kötü.

Autonomous remediation

Uyarıdan düzeltmeye insan onayı olmadan geçen otomatik akışlar. Değerlendirmede ve acele etmiyoruz: dar kapsamlı, geri alınabilir düzeltmelerde (servis yeniden başlatma, disk temizleme) değeri açık, ama kapsam genişledikçe yanlış teşhisin maliyeti üstel artıyor. Önce insan onaylı adım, sonra otonom.

Predictive reliability

Olay gerçekleşmeden sinyali yakalamak. Değerlendirmede: disk dolması ve bellek sızıntısı gibi yavaş bozulmalarda öngörü çalışıyor, ani arızalarda çalışmıyor. Yanlış alarm oranı yüksek kalırsa nöbetçi mühendis uyarıya güvenmemeye başlıyor ve asıl kayıp o.

Diller ve Çerçeveler

AI orchestration frameworks

Ajan tabanlı iş akışlarının yönetimi ve zincirlenmesi. Değerlendirmede: alan hızlı değişiyor ve bugün seçilen çerçeve bir yıl sonra bakımsız kalabiliyor. Üretim operasyonuna sokmadan önce hata durumunda ne yaptığını, geri alınabilirliğini ve denetim izini görmek istiyoruz.

Declarative workflow DSLs

Runbook'un koda dönüştüğü bildirimsel iş akışı dilleri. Değerlendirmede: vaat açık: çalıştırılabilir runbook, hem belge hem otomasyon. Ama her DSL yeni bir öğrenme eğrisi ve yeni bir bağımlılık; nöbetteki mühendisin gece 03.00'te okuyamadığı bir dil, runbook olmaktan çıkıyor.

BEKLET · 1

Uzak duruluyor — yeni işlerde tercih etmiyoruz

Araçlar

Citrix NetScaler

Tescilli donanım ADC. Beklemede: cihazın kendisi yetenekli ama lisans modeli, yükseltme pencereleri ve donanıma bağlı ölçekleme, yazılım tanımlı yük dengeleme ile karşılaştırıldığında esnekliği sınırlıyor. Yeni kurulumlarda yazılım tanımlı ADC ve WAF öneriyoruz; mevcut kurulumları desteklemeye devam ediyoruz.

Bu belgenin sınırları

Bu radar **Eclit'in kendi operasyonunu** anlatır, genel bir teknoloji tavsiyesi değildir. Sizin ortamınızda doğru cevap farklı olabilir; bir teknolojinin bizde “beklet” halkasında olması onun kötü olduğunu değil, bizim iş yükümüze uymadığını söyler.

Halkalar ölçülmüş bir kıyaslamanın değil, mühendislik ekibinin ortak değerlendirmesinin sonucudur. Sürüm numarası ve tarih tam bu yüzden kapakta duruyor: bu belge bir anın fotoğrafıdır.

Her teknolojinin güncel değerlendirmesi ve ilgili yazılar: **eclit.com/tr/radar**