Proje Raporu: Bağımsız, Güvenli ve Maliyetsiz Geliştirme Altyapısı Kurulumu

Proje Sahibi: Yusuf Turan
Tarih: 12.07.2025

1. Özet (Abstract)

Bu rapor, standart bir masaüstü bilgisayarın, modern DevOps prensipleri ve açık kaynaklı araçlar kullanılarak, güvenli, esnek ve maliyetsiz bir geliştirme ve barındırma sunucusuna dönüştürülmesi sürecini detaylandırmaktadır. Projenin temel motivasyonu, büyük veri setleri için GitHub gibi platformların getirdiği depolama limitlerini aşmak, web/mobil uygulamalar için hosting ve domain maliyetlerini ortadan kaldırmak ve statik IP adresi bulunmayan ev ağı koşullarında dahi internete açık servisler sunabilmektir. Bu hedeflere ulaşmak için **Docker** ile konteynerleştirme, **Nginx Proxy Manager** ile servis yönlendirme ve SSL yönetimi, **Cloudflare Tunnel** ile ise güvenli ağ erişimi ve origin IP gizliliği sağlanmıştır. Proje sonucunda, üzerinde birden çok servisin (web sitesi, Git sunucusu vb.) aynı anda ve güvenli bir şekilde çalışabildiği, ölçeklenebilir bir altyapı başarıyla hayata geçirilmiştir.

2. Giriş

Bilgisayar mühendisliği öğrencileri olarak, projelerimizi geliştirirken ve canlıya alırken çeşitli teknik ve finansal engellerle karşılaşmaktayız:

1. **Versiyon Kontrol Sistemi Limitleri:** GitHub gibi popüler platformlar, ücretsiz depolama alanını 2 GB ile sınırlamaktadır. Bu durum, makine öğrenmesi veya büyük veri setleri içeren projelerde ciddi bir engel teşkil etmektedir.
2. **Hosting ve Domain Maliyetleri:** Geliştirilen her bir web veya mobil uygulamanın backend'ini canlıya almak, ayrı bir hosting paketi ve potansiyel olarak ek domain/subdomain maliyetleri gerektirmektedir. Bu, özellikle deneme ve öğrenme amaçlı projeler için sürdürülebilir değildir.
3. **Ağ Kısıtlamaları:** Ev internet bağlantılarında genellikle statik bir Public IP adresi bulunmaz. Ayrıca, İnternet Servis Sağlayıcıları (ISS) maliyetleri düşürmek amacıyla sıkça CG-NAT (Carrier-Grade NAT) uygulamakta, bu da port yönlendirmeyi ve dışarıdan sunucuya erişimi imkansız kılmaktadır.

Bu projenin temel amacı, yukarıda belirtilen sorunları çözmek üzere, mevcut bir donanımı kullanarak aşağıdaki yeteneklere sahip bir ev sunucusu (homelab) altyapısı kurmaktır:

- Büyük boyutlu Git repolarını barındırabilecek özel bir Git sunucusu çalıştırmak.
- Birden çok web sitesi ve servisi tek bir IP adresi üzerinden, ek maliyet olmaksızın yayınlamak.
- Tüm servisler için otomatik ve ücretsiz SSL sertifikaları (HTTPS) temin etmek.
- Statik IP veya port yönlendirme ihtiyacını ortadan kaldırarak, ISS kaynaklı ağ kısıtlamalarını aşmak.
- Sunucunun gerçek IP adresini gizleyerek güvenliği en üst düzeye çıkarmak.
- **İşletim Sistemi:** Ubuntu Server 22.04 LTS

- **Konteynerleştirme:** Docker & Docker Compose
- **Reverse Proxy:** Nginx Proxy Manager (NPM)
- **Güvenli Ağ Erişimi:** Cloudflare Tunnel (cloudflared)
- **DNS Yönetimi:** Cloudflare

3. Kurulum ve Yapılandırma Adımları

1. **İşletim Sistemi Kurulumu:** Sunucu olarak belirlenen bilgisayara, kararlılığı, geniş topluluk desteği ve uzun süreli güvenlik güncellemeleri (LTS) nedeniyle **Ubuntu Server 22.04 LTS** kurulmuştur. Kurulum, gereksiz kaynak tüketimini önlemek amacıyla grafik arayüz olmadan (headless) gerçekleştirilmiştir.
2. **Temel Paketlerin Güncellenmesi:** Sistemin güvenliği ve kararlılığı için, kurulum sonrası ilk adım olarak apt paket yöneticisi ile tüm paketler en güncel sürümlerine yükseltilmiştir.

```
sudo apt update && sudo apt full-upgrade -y
```

3. **SSH Erişimi:** Sunucunun uzaktan yönetilebilmesi için openssh-server paketi kurulmuş ve SSH servisi aktif edilmiştir. Bu, tüm sonraki adımların yerel bir makineden, ağ üzerinden güvenli bir şekilde yapılmasını sağlamıştır.

1. **Docker Kurulumu:** Uygulamaları ve servisleri birbirinden izole, taşınabilir ortamlarda (konteyner) çalıştırmak için endüstri standardı olan Docker kurulmuştur. Ubuntu'nun kendi deposundaki potansiyel olarak eski sürüm yerine, en güncel özellikleri ve güvenlik yamalarını içeren Docker'ın resmi paket deposu sisteme eklenerek kurulum yapılmıştır.
2. **Docker Compose Kurulumu:** Birden çok konteynerden oluşan uygulamaları (örneğin, bir web sunucusu ve bir veritabanı) tek bir YAML dosyası üzerinden tanımlayıp yönetmeyi sağlayan docker-compose-plugin kurulmuştur. Bu, proje yönetimini deklaratif ve tekrarlanabilir hale getirmiştir.
3. **Kullanıcı Yetkilendirmesi:** Her docker komutu için sudo kullanma zorunluluğunu ortadan kaldırmak ve kullanım kolaylığı sağlamak amacıyla mevcut kullanıcı, docker grubuna dahil edilmiştir.

```
sudo usermod -aG docker $USER
```

4.

1. **Paylaşılan Docker Ağı:** Projedeki tüm servislerin (NPM, Cloudflared, web siteleri vb.) birbirleriyle kolayca iletişim kurabilmesi için, projelerden bağımsız, proxy adında harici bir köprü (bridge) ağı manuel olarak oluşturulmuştur.

   Generated bash
   ```
   docker network create proxy
   ```

   Bu yaklaşım, docker-compose'un her proje için ayrı ağlar oluşturmasının önüne geçerek, tüm servislerin tek ve ortak bir sanal ağda, birbirlerine servis isimleriyle (örneğin app) erişebilmesini sağlamıştır.

2. **Reverse Proxy Kurulumu (Nginx Proxy Manager):**
    - **Teknik Açıklama:** Reverse Proxy, dış dünyadan gelen tüm HTTP/HTTPS isteklerini (port 80/443) karşılayan ve isteğin yapıldığı alan adına (hostname) göre ilgili iç servise (konteyner) yönlendiren bir sunucudur. Bu sayede tek bir Public IP adresi arkasında yüzlerce web sitesi barındırılabilir.
    - **Uygulama:** Nginx Proxy Manager (NPM), bu işlevi kullanıcı dostu bir web arayüzü ve otomatik Let's Encrypt SSL sertifikası entegrasyonu ile sunduğu için tercih edilmiştir. Aşağıdaki docker-compose.yml dosyası ile proxy ağına dahil edilerek çalıştırılmıştır.

      Generated yaml
      ```
      # ~/docker-projects/npm/docker-compose.yml
      version: '3.8'
      services:
        app:
          image: 'jc21/nginx-proxy-manager:latest'
          restart: unless-stopped
          ports:
            - '80:80'
            - '443:443'
            - '81:81'   # Yönetim Arayüzü
          volumes:
            - ./data:/data
            - ./letsencrypt:/etc/letsencrypt
          networks:
            - proxy
      networks:
        proxy:
          external: true
      ```

1. **Domain ve DNS Yönetimi:** Proje için yusufturan.com.tr alan adı temin edilmiştir. Alan adının Nameserver (NS) kayıtları, domain'in alındığı firmadan (Registrar) Cloudflare'in sağladığı NS adreslerine yönlendirilerek, tüm DNS yönetimi Cloudflare'e devredilmiştir.
2. **Cloudflare Tunnel Mimarisi:**
    - **Teknik Açıklama:** Geleneksel port yönlendirme modelinin aksine Cloudflare Tunnel, sunucu içerisinden Cloudflare'in en yakın veri merkezine doğru giden (outbound) kalıcı ve şifreli bir tünel oluşturur. Bu model, sunucunun internete herhangi bir port açmasını gerektirmez, dolayısıyla sunucunun IP adresini tamamen gizler ve CG-NAT gibi ağ engellerini aşar. Dışarıdan gelen istekler önce Cloudflare tarafından karşılanır ve bu güvenli tünel üzerinden sunucuya iletilir.
    - **Uygulama:** Bu tüneli yönetecek olan cloudflared servisi, aşağıdaki docker-compose.yml dosyası ile, paylaşılan proxy ağına dahil edilerek çalıştırılmıştır. Gerekli olan tünel token'ı, Cloudflare Zero Trust panelinden temin edilmiştir.

      ```
      # ~/docker-projects/cloudflared/docker-compose.yml
      version: '3.8'
      services:
        cloudflared:
          image: cloudflare/cloudflared:latest
          restart: unless-stopped
      ```

```
    command: tunnel --no-autoupdate run --token <GİZLİ_TÜNEL_TOKENI>
    networks:
      - proxy
networks:
  proxy:
    external: true
```

3. **Tünel Yönlendirmesi (Public Hostnames):** Cloudflare Zero Trust panelinde, yusufturan.com.tr ve www.yusufturan.com.tr alan adlarına gelen isteklerin, tünel üzerinden http://app:80 servisine, yani Nginx Proxy Manager'ın genel trafik dinleme portuna yönlendirilmesi sağlanmıştır. Bu yönlendirme, Docker'ın dahili DNS çözümlemesi sayesinde servis ismi (app) kullanılarak yapılmıştır.

1. **Statik Site Konteyneri:** HTML/CSS ile hazırlanan kişisel portfolyo sitesini sunmak için hafif bir Nginx konteyneri kullanılmıştır. Proje dosyaları (index.html, style.css vb.), volumes aracılığıyla konteynerin web kök dizinine (/usr/share/nginx/html) bağlanmıştır.
2. **NPM Kural Tanımlaması:** NPM'in yönetim arayüzünden (http://<sunucu_yerel_ip>:81), yusufturan.com.tr alan adına gelen isteklerin, proxy ağı üzerindeki portfolio-web:80 servisine (statik site konteyneri) yönlendirilmesi için bir "Proxy Host" kuralı oluşturulmuştur.
3. **SSL Sertifikası:** Aynı kuralın SSL sekmesinde, "Request a new SSL Certificate" seçeneği kullanılarak Let's Encrypt'ten domain için ücretsiz SSL sertifikası alınmış ve "Force SSL" seçeneği ile tüm trafiğin HTTPS'e yönlendirilmesi sağlanmıştır.

1. **traceroute ve mtr Analizi:** Kurulan sistemin güvenliğini ve çalışma prensibini doğrulamak amacıyla traceroute ve mtr araçları kullanılarak yusufturan.com.tr adresine rota takibi yapılmıştır.
2. **Bulgular:** Analiz sonucunda, paketlerin izlediği yolun son durağının, kullanıcının ev IP adresi değil, **Cloudflare'e ait Public IP adresleri** (104.21.x.x gibi) olduğu gözlemlenmiştir. Bu, Cloudflare Tunnel'ın origin (kaynak) sunucu IP'sini başarıyla gizlediğini ve dış dünyaya karşı bir kalkan görevi gördüğünü somut olarak kanıtlamıştır. Ayrıca, rota takibinin ilk adımlarında görülen 10.x.x.x aralığındaki IP'ler, ISS tarafından uygulanan CG-NAT varlığını doğrulamış ve kurulan tünel altyapısının bu engeli başarıyla aştığını göstermiştir.

---

4. Sonuç

Proje, başlangıçta belirlenen tüm hedeflere başarıyla ulaşmıştır. Maliyetsiz ve açık kaynaklı araçlar kullanılarak, ev ağı koşullarında dahi çalışabilen, güvenli, esnek ve ölçeklenebilir bir sunucu altyapısı kurulmuştur. Cloudflare Tunnel entegrasyonu sayesinde, geleneksel yöntemlerle çözülmesi zor olan statik IP eksikliği ve CG-NAT gibi sorunlar etkili bir şekilde çözülmüş, aynı zamanda sunucu güvenliği en üst seviyeye taşınmıştır. Bu proje, sadece teorik bilgilerin pratiğe döküldüğü bir çalışma olmanın ötesinde, modern DevOps araç zinciri (toolchain) konusunda değerli bir uygulamalı deneyim sağlamıştır.

---

5. Gelecek Çalışmaları ve Potansiyel Geliştirmeler

Kurulan bu temel altyapı üzerine aşağıdaki servisler ve geliştirmeler eklenebilir:

- **Özel Git Sunucusu: Gitea** konteyneri kurularak git.yusufturan.com.tr adresi üzerinden özel ve limitsiz Git depolama hizmeti sağlanması.
- **Veritabanı Servisleri:** Projeler için **PostgreSQL** veya **MongoDB** gibi veritabanlarının ayrı konteynerler olarak çalıştırılması.
- **İzleme ve Gözetleme (Monitoring):** Sunucu ve servislerin durumunu anlık olarak takip etmek için **Prometheus** ve **Grafana** gibi araçlardan oluşan bir izleme yığını (monitoring stack) kurulması.
- **Yedekleme Stratejisi:** Kritik verilerin (Docker volume'leri, konfigürasyon dosyaları) rclone gibi bir araçla düzenli olarak şifrelenerek bir bulut depolama servisine (örn: Backblaze B2) yedeklenmesi.