# KTD-001 - Bölüm 1

## Sunucuya SSH ile Güvenli Bağlantı Kurma

Versiyon: 1.0

Yazar: Kemal Tokmak

## Amaç

Bu bölümde Ubuntu 24.04 VPS sunucusuna SSH ile güvenli bağlantı kurmayı öğreneceksiniz.

## SSH Nedir?

SSH (Secure Shell), uzak sunuculara şifreli bağlantı sağlayan güvenli protokoldür.

## Gereksinimler

| Gereksinim | Açıklama |
|---|---|
| VPS | Ubuntu 24.04 LTS |
| Kullanıcı | ubuntu |
| Port | 22 |

## SSH Bağlantısı

```
ssh -v ubuntu@SUNUCU_IP
```

## Komut Açıklaması

ssh: SSH istemcisini başlatır
-v: Ayrıntılı çıktı
ubuntu: Varsayılan kullanıcı
SUNUCU_IP: Sunucu IP adresi

## Ekran Görüntüsü

## İlk Bağlantı Uyarısı

İlk bağlantıda 'Are you sure you want to continue connecting (yes/no)?' sorusuna yes yanıtı verilir.

## Sık Karşılaşılan Hatalar

Permission denied
Connection timed out

Connection refused
Host key verification failed

## Bu Bölümde Ne Öğrendik?

SSH bağlantısı kurmayı ve temel bağlantı sorunlarını öğrendiniz.

## Sonraki Bölüm

Bölüm 2 - Root Yetkisi Alma ve İlk Güvenlik Ayarları

# KTD-001 - Bölüm 2

## Root Yetkisi Alma ve İlk Güvenlik Ayarları

## Amaç

Bu bölümde root kullanıcısını, sudo kullanımını ve temel güvenlik ayarlarını öğreneceksiniz.

## Root Nedir?

Root, Linux sistemlerde en yüksek yetkiye sahip kullanıcıdır.

## Root Yetkisi Alma

Komut:
sudo -i

## Yetki Kontrolü

Komut:
whoami

Beklenen çıktı:
root

## Root Parolasını Değiştirme

Komut:
passwd

## Normal Kullanıcıya Dönüş

Komut:
exit

### Ekran Görüntüsü


### Güvenlik İpuçları

Root hesabını yalnızca gerektiğinde kullanın ve günlük işlemleri normal kullanıcı ile yapın.

### Sık Karşılaşılan Hatalar

Permission denied
Authentication failure
sudo: command not found

### Bu Bölümde Ne Öğrendik?

Root yetkisi almayı, doğrulamayı ve güvenli kullanımı öğrendiniz.

### Sonraki Bölüm

Bölüm 3 - Ubuntu Sistem Güncellemesi

## KTD-001 - Bölüm 3

### Ubuntu Sistem Güncellemesi

Versiyon: 1.0
Yazar: Kemal Tokmak

### Amaç

Bu bölümde Ubuntu sunucusunu güncelleyerek güvenli ve kararlı bir kurulum için hazırlayacağız.

### Neden Güncelleme Yapılır?

Güncellemeler güvenlik açıklarını kapatır, hata düzeltmeleri ve performans iyileştirmeleri sağlar.

### Paket Listesini Güncelleme

Komut:
apt update

## Yüklü Paketleri Güncelleme

Komut:

apt upgrade -y

## Tam Sistem Güncellemesi

Komut:

apt full-upgrade -y

## Kullanılmayan Paketleri Temizleme

Komut:

apt autoremove -y

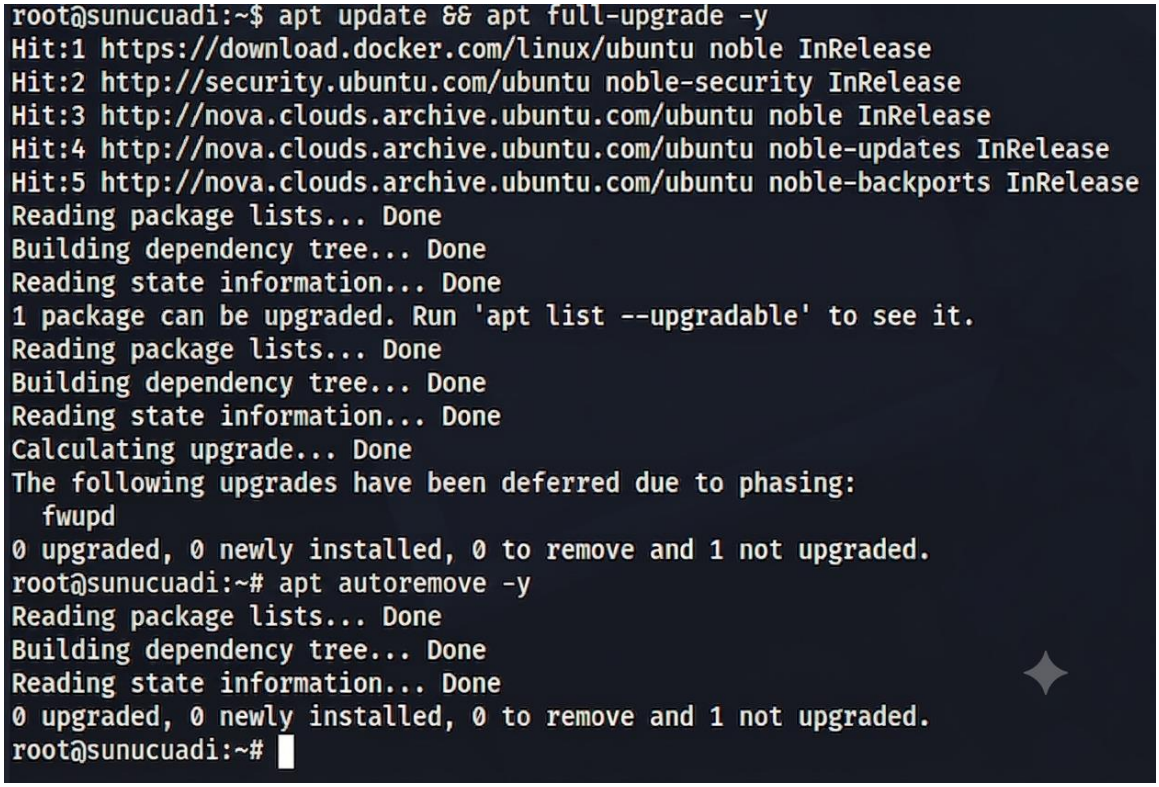
apt autoclean

## Yeniden Başlatma

Komut:

reboot

## Ekran Görüntüsü




## Sık Karşılaşılan Hatalar

Failed to fetch

Could not get lock /var/lib/dpkg/lock

Temporary failure resolving

### Gerçek Hayat Notu

Sunucu kurulumu sonrasında her zaman ilk işim sistemi tamamen güncellemektir. Böylece eski paketlerle kurulum yapmamış olurum.

### Bu Bölümde Ne Öğrendik?

Ubuntu paketlerini güncellemeyi, sistemi yükseltmeyi ve gereksiz paketleri temizlemeyi öğrendiniz.

### Sonraki Bölüm

Bölüm 4 - UFW Güvenlik Duvarı Kurulumu

## KTD-001 - Bölüm 4

### UFW Güvenlik Duvarı Kurulumu

### Amaç

Bu bölümde Ubuntu'nun yerleşik güvenlik duvarı UFW kurulacak ve temel kurallar yapılandırılacaktır.

### UFW Nedir?

UFW (Uncomplicated Firewall), Ubuntu için kolay yönetilebilir bir güvenlik duvarıdır.

### Kurulum

Komut:
apt install ufw -y

### SSH İzni

ufw allow OpenSSH

### HTTP İzni

ufw allow 80/tcp

### HTTPS İzni

ufw allow 443/tcp

### UFW'yi Etkinleştirme

ufw enable

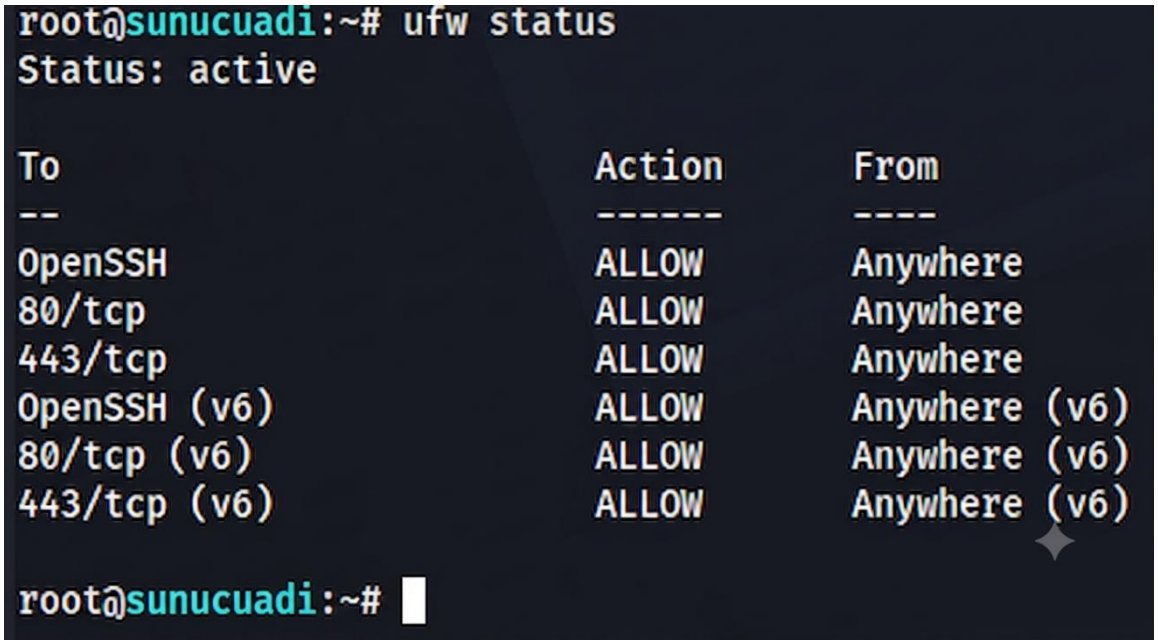
### Durum Kontrolü

ufw status

### Beklenen Çıktı

Status: active
22/tcp ALLOW
80/tcp ALLOW
443/tcp ALLOW

### Ekran Görüntüsü




### Sık Karşılaşılan Hatalar

SSH izni vermeden UFW'yi etkinleştirmek bağlantının kesilmesine neden olabilir.

### Gerçek Hayat Notu

UFW'yi etkinleştirmeden önce mutlaka OpenSSH kuralını ekleyin.

### Bu Bölümde Ne Öğrendik?

UFW kurulumu, temel kuralların eklenmesi ve güvenlik duvarının etkinleştirilmesini öğrendiniz.

### Sonraki Bölüm

Bölüm 5 - Fail2Ban Kurulumu

## KTD-001 - Bölüm 5

### Fail2Ban Kurulumu ve Yapılandırması

### Amaç

Bu bölümde Fail2Ban kurularak SSH servisi brute-force saldırılarına karşı korunacaktır.

## Fail2Ban Nedir?

Fail2Ban, başarısız giriş denemelerini izler ve belirlenen eşik aşıldığında ilgili IP adresini geçici olarak engeller.

## Kurulum

Komut:

apt install fail2ban -y

## Servisi Etkinleştirme

systemctl enable fail2ban

systemctl start fail2ban

## Durum Kontrolü

systemctl status fail2ban

## Servis Doğrulama

Çıktıda 'active (running)' ifadesi görülmelidir.

## Yapılandırma Notu

İleri düzey ayarlar için /etc/fail2ban/jail.local dosyası kullanılabilir. Varsayılan ayarlar başlangıç için yeterlidir.

## Ekran Görüntüsü


## Sık Karşılaşılan Hatalar

Unit fail2ban.service not found

inactive (dead)

Failed to start fail2ban

## Gerçek Hayat Notu

İnternete açık tüm Linux sunucularımda ilk kurduğum güvenlik yazılımlarından biri Fail2Ban'dır. SSH saldırılarını önemli ölçüde azaltır.

## Bu Bölümde Ne Öğrendik?

Fail2Ban'ın ne olduğunu, kurulumunu, etkinleştirilmesini ve servis durumunun kontrolünü öğrendiniz.

## Sonraki Bölüm

Bölüm 6 - Docker Engine Kurulumu

# KTD-001 - Bölüm 6

## Docker Engine Kurulumu

## Amaç

Bu bölümde Ubuntu 24.04 üzerine resmi Docker Engine kurulumu gerçekleştirilecektir.

## Docker Nedir?

Docker, uygulamaları container olarak izole şekilde çalıştırmayı sağlayan platformdur.

## Eski Paketleri Kaldırma

apt remove docker docker-engine docker.io containerd runc

apt install ca-certificates curl gnupg -y

## Docker GPG Anahtarı

Docker'ın resmi GPG anahtarı eklenir.

## Docker Deposunu Ekleme

Resmi Docker deposu apt kaynaklarına eklenir.

## Kurulum

apt update

apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

## Servisi Kontrol

systemctl status docker

## Sürüm Kontrolü

docker --version

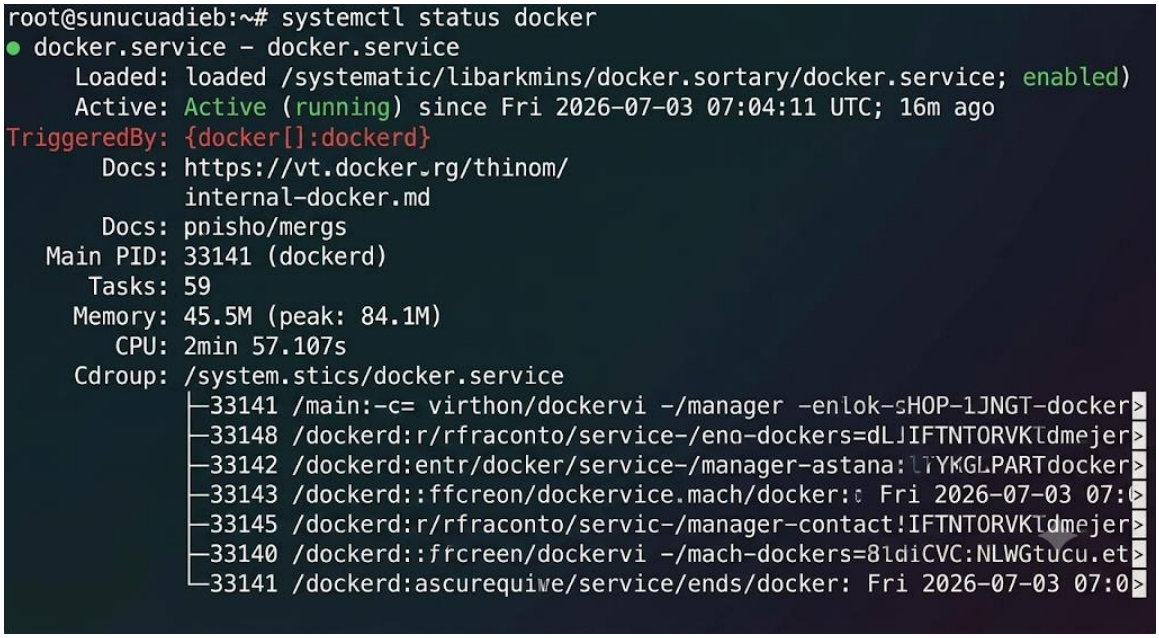
## İlk Test

docker run hello-world

## Beklenen Sonuç

Hello from Docker! mesajı görülmelidir.

## Ekran Görüntüsü




## Sık Karşılaşılan Hatalar

Unable to locate package

permission denied

Cannot connect to the Docker daemon

## Gerçek Hayat Notu

Her zaman Docker'ın resmi deposunu kullanın. Ubuntu deposundaki sürümler güncel olmayabilir.

## Bu Bölümde Ne Öğrendik?

Docker Engine kurulumunu, servis kontrolünü ve ilk container testini öğrendiniz.

## Sonraki Bölüm

Bölüm 7 - Docker Compose Plugin

# KTD-001 - Bölüm 7

## Docker Compose Plugin Kurulumu ve İlk Compose Projesi

## Amaç

Bu bölümde Docker Compose Plugin'in doğrulanması, kullanımı ve ilk docker-compose.yml dosyasının oluşturulması anlatılır.

## Docker Compose Nedir?

Docker Compose, birden fazla container'ı tek bir YAML dosyası ile yönetmeyi sağlayan araçtır.

## Compose Plugin Kontrolü

Komut:

docker compose version

## İlk Proje Klasörü

mkdir -p /opt/docker/web

cd /opt/docker/web

## docker-compose.yml Oluşturma

compose dosyası oluşturulur ve servisler burada tanımlanır.

## Örnek Compose Dosyası

```
services:
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
    restart: unless-stopped
```

## Servisi Başlatma

docker compose up -d

## Container Kontrolü

docker ps

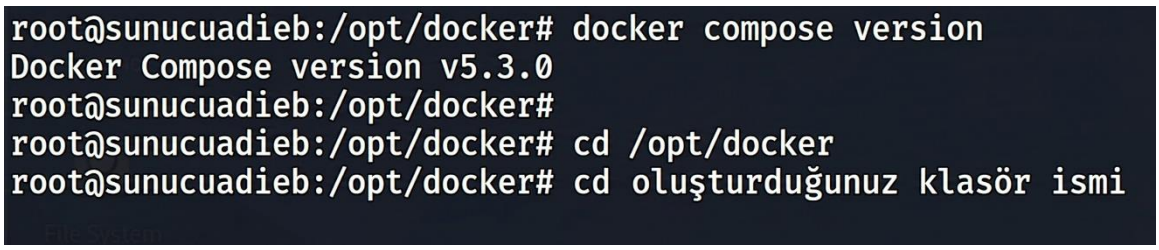
## Log Kontrolü

docker compose logs

## Servisi Durdurma

docker compose down

## Ekran Görüntüsü



## Sık Karşılaşılan Hatalar

docker: 'compose' is not a docker command
port is already allocated
YAML syntax error

## Gerçek Hayat Notu

Tek container yerine tüm projeleri Docker Compose ile yönetmek bakım ve güncellemeleri büyük ölçüde kolaylaştırır.

## Bu Bölümde Ne Öğrendik?

Docker Compose Plugin'i kullanmayı, ilk compose dosyasını oluşturmayı ve container yönetimini öğrendiniz.

## Sonraki Bölüm

Bölüm 8 - Nginx Web Sunucusunu Docker ile Yayına Alma

# KTD-001 - Bölüm 8

## Docker ile Nginx Web Sunucusunu Yayına Alma

## Amaç

Bu bölümde Docker Compose kullanarak Nginx web sunucusunu ayağa kaldıracak ve ilk web sitemizi yayınlayacağız.

## Nginx Nedir?

Nginx, yüksek performanslı açık kaynaklı bir web sunucusu ve reverse proxy yazılımıdır.

## Proje Dizin Yapısı

```
/opt/docker/web/
├── docker-compose.yml
├── html/
|   ├── index.html
|   ├── css/
|   └── js/
```

## HTML Klasörü Oluşturma

mkdir -p /opt/docker/web/html

## CSS ve JS Klasörleri

mkdir -p /opt/docker/web/html/css
mkdir -p /opt/docker/web/html/js

## docker-compose.yml Dosyası

```
services:
  nginx:
    image: nginx:latest
    container_name: nginx
    restart: unless-stopped
    ports:
      - '80:80'
    volumes:
      - ./html:/usr/share/nginx/html:ro
```

## Servisi Başlatma

```
docker compose up -d
```

## Container Kontrolü

```
docker ps
```

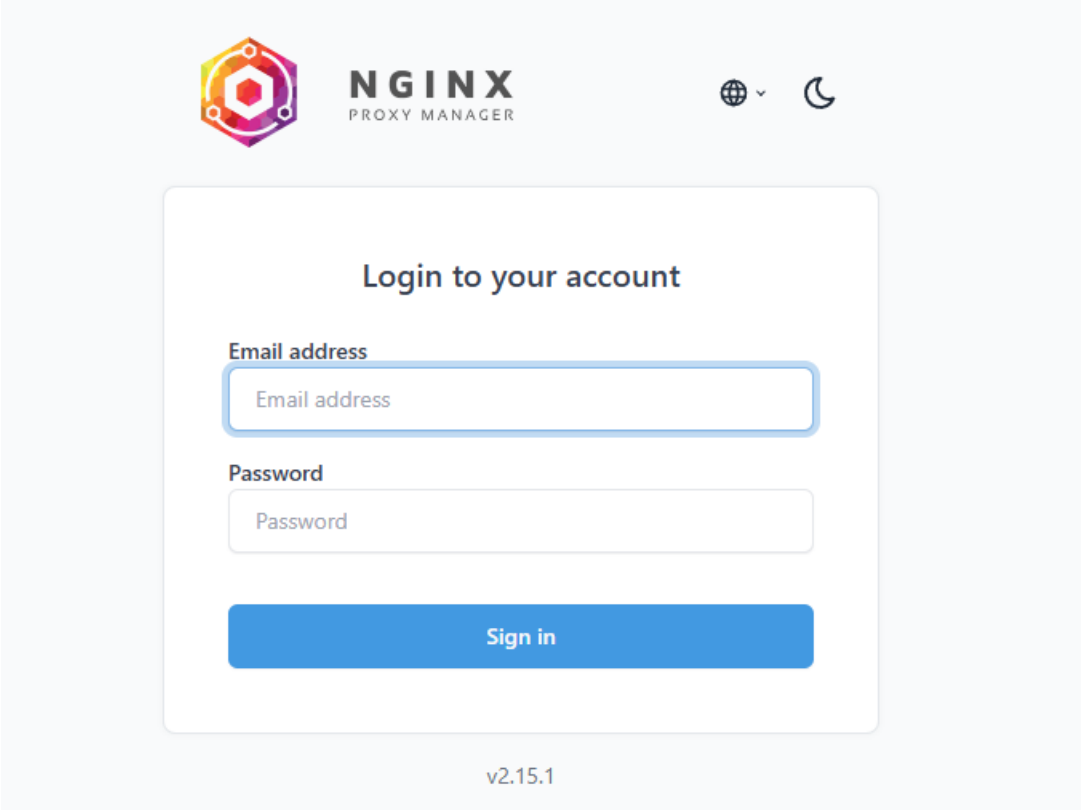
## Tarayıcı Testi

Tarayıcıdan http://SUNUCU_IP adresine giderek Nginx'in çalıştığını doğrulayın.

## İlk index.html

html klasörüne basit bir index.html dosyası ekleyin ve sayfanın açıldığını kontrol edin.

## Ekran Görüntüsü




## Sık Karşılaşılan Hatalar

404 Not Found
403 Forbidden
Port 80 already allocated
Volume mount hataları

## Gerçek Hayat Notu

Projelerde tüm web dosyalarını ayrı klasörlerde tutmak bakım, yedekleme ve güncelleme işlemlerini büyük ölçüde kolaylaştırır.

## Bu Bölümde Ne Öğrendik?

Docker Compose ile Nginx'i yayınlamayı, web dizinlerini bağlamayı ve ilk web sayfasını yayınlamayı öğrendiniz.

## Sonraki Bölüm

Bölüm 9 - Domain Bağlama ve DNS Yapılandırması

# KTD-001 - Bölüm 9

## Domain Bağlama ve DNS Yapılandırması

### Amaç

Bu bölümde alan adınızı VPS sunucunuza yönlendirecek ve DNS kayıtlarını yapılandıracaksınız.

### Domain ve DNS

Bir domain adı, DNS kayıtları sayesinde sunucu IP adresine yönlendirilir.

### Gerekli Bilgiler

Alan adı, VPS IP adresi ve DNS yönetim paneline erişim.

### A Kaydı

@ -> VPS_IP
TTL: Auto

### WWW Kaydı

www -> @ (CNAME) veya VPS_IP (A kaydı)

### DNS Yayılımı

DNS değişiklikleri birkaç dakika ile 24 saat arasında yayılabilir.

### Doğrulama

dig domain.com
nslookup domain.com
ping domain.com

### Nginx Testi

Tarayıcıdan http://domain.com adresini açarak sitenin çalıştığını doğrulayın.

### Ekran Görüntüsü


### Sık Karşılaşılan Hatalar

Yanlış IP adresi
DNS yayılımının tamamlanmaması
Cloud proxy kaynaklı sorunlar

### Gerçek Hayat Notu

DNS değişikliklerinden sonra önbelleği temizlemek ve yayılımı beklemek birçok sorunu çözer.

## Bu Bölümde Ne Öğrendik?

Domaini VPS'e yönlendirmeyi ve DNS kayıtlarını doğrulamayı öğrendiniz.

## Sonraki Bölüm

Bölüm 10 - SSL (Let's Encrypt) Kurulumu

# KTD-001 - Bölüm 10

## Let's Encrypt ile SSL Kurulumu

## Amaç

Bu bölümde web sunucusuna ücretsiz SSL sertifikası kurarak HTTPS erişimini etkinleştireceğiz.

## SSL Nedir?

SSL/TLS, istemci ile sunucu arasındaki trafiği şifreleyerek güvenli iletişim sağlar.

## Gereksinimler

Domain DNS kayıtları doğru yapılandırılmış ve web sitesi HTTP üzerinden erişilebilir olmalıdır.

## Certbot Kurulumu

Docker ortamında Nginx Proxy Manager, Certbot veya benzeri yöntemlerden biri kullanılabilir.
Bu dokümanda Certbot yaklaşımı anlatılır.

## 80 ve 443 Portları

UFW üzerinde 80/tcp ve 443/tcp portlarının açık olduğundan emin olun.

## Sertifika Alma

Örnek komut:
certbot --nginx -d domain.com -d www.domain.com

## Otomatik Yenileme

certbot renew --dry-run komutu ile otomatik yenileme testi yapılır.

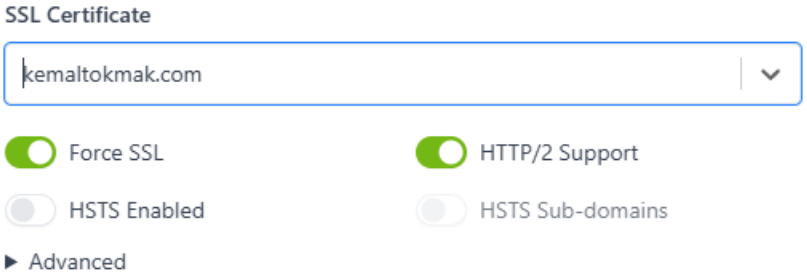
## HTTPS Testi

Tarayıcıdan https://domain.com adresini açın ve kilit simgesini doğrulayın.

## Doğrulama

SSL Labs veya tarayıcı sertifika bilgileri ile sertifikanın doğruluğunu kontrol edin.

## Ekran Görüntüsü




## Sık Karşılaşılan Hatalar

DNS yanlış yönlendirilmiş olabilir.
80 portu kapalı olabilir.
Let's Encrypt rate limit hatası alınabilir.

## Gerçek Hayat Notu

Sertifika aldıktan sonra HTTP isteklerini kalıcı olarak HTTPS'e yönlendirmek güvenlik ve SEO açısından önerilir.

## Bu Bölümde Ne Öğrendik?

Let's Encrypt ile ücretsiz SSL kurulumu, sertifika doğrulaması ve otomatik yenilemeyi öğrendiniz.

## Sonraki Bölüm

Bölüm 11 - Docker Üzerinde Nginx Reverse Proxy Yapılandırması

# KTD-001 - Bölüm 11

## Docker Üzerinde Nginx Reverse Proxy Yapılandırması

## Amaç

Bu bölümde Nginx'i reverse proxy olarak yapılandırarak alan adlarını Docker servislerine yönlendireceğiz.

## Reverse Proxy Nedir?

Reverse proxy, istemci isteklerini arka plandaki uygulama veya container'lara yönlendiren ara katmandır.

## Mimari

İstemci -> Nginx -> Docker Container

## Örnek Nginx Yapılandırması

```
server {
   listen 80;
   server_name domain.com www.domain.com;

   location / {
      proxy_pass http://app:3000;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header X-Forwarded-Proto $scheme;
   }
}
```

## Yapılandırmayı Test Etme

nginx -t

## Nginx'i Yeniden Yükleme

nginx -s reload veya docker restart nginx

## Bağlantıyı Doğrulama

Tarayıcıdan domain üzerinden uygulamanın açıldığını kontrol edin.

## Ekran Görüntüsü

[Nginx yapılandırması, nginx -t çıktısı ve çalışan uygulamanın ekran görüntülerini ekleyin.]

## Sık Karşılaşılan Hatalar

502 Bad Gateway
404 Not Found
Host çözümlenemiyor
Yanlış proxy_pass adresi

## Gerçek Hayat Notu

Her uygulama için ayrı server bloğu kullanmak bakım ve ölçeklenebilirlik açısından en doğru yaklaşımdır.

## Bu Bölümde Ne Öğrendik?

Nginx Reverse Proxy mantığını, temel yapılandırmayı ve doğrulama adımlarını öğrendiniz.

## Sonraki Bölüm

Bölüm 12 - Docker Container Yönetimi ve Log Analizi

# KTD-001 - Bölüm 12

## Docker Container Yönetimi ve Log Analizi

### Amaç

Bu bölümde Docker container'larını yönetmeyi, logları incelemeyi ve temel bakım işlemlerini öğreneceksiniz.

### Container'ları Listeleme

docker ps

docker ps -a

### Container Bilgileri

docker inspect <container_adi>

### Canlı Kaynak Kullanımı

docker stats

### Logları Görüntüleme

docker logs <container_adi>

docker logs -f <container_adi>

### Container'ı Durdurma ve Başlatma

docker stop <container_adi>

docker start <container_adi>

docker restart <container_adi>

### Container'a Bağlanma

docker exec -it <container_adi> /bin/bash

### Log Analizi

Hata mesajlarını, yeniden başlatmaları ve uygulama çıktılarını düzenli olarak kontrol edin.

### Ekran Görüntüsü

[Buraya docker ps, docker logs ve docker stats çıktılarının ekran görüntülerini ekleyin.]

### Sık Karşılaşılan Hatalar

Container sürekli yeniden başlıyor (restart loop)
Container bulunamadı

Permission denied
No such container

## Gerçek Hayat Notu

Üretim ortamında sorunların büyük bölümü docker logs çıktıları incelenerek kısa sürede tespit edilebilir.

## Bu Bölümde Ne Öğrendik?

Container yönetimi, log analizi ve temel Docker bakım komutlarını öğrendiniz.