

# **MODERN WEB ALTYAPILARI ve SİSTEMLERİN ÇALIŞMASI**

Herkese merhabalar. Bu yazımda sizlere modern web altyapılarında kullanılan teknolojileri ve sistemlerin nasıl çalışıklarını bu çalışma yapısındaki zayıf noktaları ve bu noktalar nasıl engelleneceği gibi konuları ele alacağım.

Bilgisayar açılığı süreden itibaren nasıl bir süreç izliyor oradan başlayacağım.

## **IP ADRESİ TAHSİSİ:**

Bilgisayar açıldığı anda eğer Wi-fi bağlantısı aktifse internete çıkmak için bir kimliğe ihtiyacı olur. Bunun için MAC adresi (bu adres tüm cihazlara özgü bir ID değeridir.) Router (Modem)'a bir paket yollar. Der ki ben buradayım ve internete çıkma için bana bir kimlik lazım. Bu bağlantıyı DHCP sağlar. DHCP daha sonra yanıt olarak boşta olan bir IP adresi değeri gönderir. Böylelikle artık bilgisayaramız internete çıkmaya hazırır.

Bu durumu biraz teknik bilgilerle anlatayım.

- İlk olarak bilgisayar ilk açıldığı anda, işletim sistemi ağ bağlantısı başlatır.
- Bilgisayar, ağa bağlı bir DHCP sunucusu olup olmadığını kontrol etmek için DHCPDISCOVER isimli bir broadcast (yayın) paketi gönderir.
- Bu paket ağa bağlı bütün cihazlara yöneltılır.
- Daha sonra ağda bulunan DHCP sunucusu, gelen bu DHCPDISCOVER isteğini alır.
- Kullanılabilir boşta olan IP değerini DHCPOFFER yanıt olarak gönderir.
- Bu IP değeri aynı zamanda subnet mask (alt ağ maskesi), default gateway (varsayılan ağ geçidi) bilgileri içerir.
- Yanıt bilgisayarımızın MAC adresine yönlendirilir. Böylelikle doğru cihaz ile iletişim kurulmuş olur.
- Daha sonra bilgisayarımız bu DHCPOFFER değerini kabul ettiğini söylemek için DHCPREQUEST isimli bir paket gönderir.
- DHCP'de DHCPACK paketi göndererek bu talebi onayladığını dile getirmış olur.

*Bu mekanizma ile DHCP ile IP değeri ataması gerçekleşmiş olur.*

Özet olarak:

DHCPDISCOVER: Bilgisayar ağdaki DHCP sunucusuna istek yollar.

DHCPOFFER: DHCP sunucusu IP adresi gönderir.

DHCPREQUEST: Bilgisayar IP adresini onayladığını söyler.

DHCPACK: DHCP IP adresini kesin olarak atar.

## **ARP PROTOKOLÜ:**

Bilgisayar DHCP ile IP adresini alır ve ancak interne çıkabilecek duruma tam anlamıyla gelmiş sayılmaz. IP adresi değeri ile birlikte default gateway'e de sahiptir.

- Artık bilgisayar default gateway (varsayılan ağ geçidi)'ın MAC adresini öğrenmek için ARP isteği yollar.
- ARP yanıtını Router (Modem) bilgisayara gönderir ve MAC adresi değerini söyler.
- Bu bilgi ARP tablosuna kaydedilir. Bilgisayar artık internete erişir durumdadır.

Bilgisayar, aynı ağ üzerinde bulunan başka cihazlar ile (yazıcı, başka bir PC...) iletişim kurmak istediğiinde de Router'ın MAC adresine ihtiyaç duyar.

Eğer ARP tablosunda bu değer varsa sorun yok. Yoksa MAC adresini öğrenmek için bilgisayar iletişim kurmak istediği cihaza sorular yöneltir.

- ARP Request: Bilgisayar, ağda bulunan tüm cihazlara broadcast ile "Bu Router bize IP adresi verdi ama Router'ın MAC adresi ne?" sorusunu sorar.
- ARP Reply: Hedef cihaz, kendi MAC adresini içeren yanıt gönderir.
- Bilgisayar bu yanıtı ARP tablosuna ekler. Artık MAC adresini biliyor.

ARP TABLOSU, bilgisayarın öğrendiği IP-MAC adresi eşleşmelerini tutan bir önbellektir. (cache) Bu sayede her seferinde tekrar istek göndermeye gerek kalmaz.

Bu sayede Router ile iletişim kurulmuş oldu ve artık internete çabasıızız.

*Peki ARP saldıruları neden gerçekleşir?*

ARP spoofing dediğimiz saldırısı; bir cihazımız var ve bunun IP adresi X olsun mesela. Aynı ağ üzerinde Y'de farklı bir IP adresine sahip başka bir cihaz olsun. Dışarıdan gelen bir cihaz X ile bağlantı kurmak istiyor. ARP tablosunda X ve Y IP adresleri kayıtlı. Eğer Y cihazı kendisini X cihazı olarak gösterirse, X 'e gelecek bütün bağlantılar Y cihazına gidecektir. MITM (Man In The Middle) saldırısı da böyle gerçekleşmiş oluyor.

## **DNS SORGULAMALARI:**

Evet her cihazın IP adresi olduğu gibi ziyaret ettiğimiz web sitelerinin de bir IP adresi tabiki de var. Biz bu IP adreslerini kullanarak aslında web sitelerini ziyaret etmiş oluyoruz. Yalnız ziyaret ettiğimiz tüm web sitelerinin IP adresini ezbere bilmemiz mümkün olmadığı için bu siteler domain alıyorlar. [DNS çözümlemesini DHCP yapar!!!!]

Domain dediğimiz şey tam olarak URL oluyor. Mesela Google.com'a gitmek istiyoruz. IP adresini yazmak yerine Google.com yazarak erişim sağlamış oluyoruz. Böylelikle erişim kolaylaşmış oluyor.

Domainler DNS serverlar sayesinde çalışır. Domain çözümleme işlemleri ilk olarak:

- Host dosyalarının kontrol edilmesi ile başlar. (127.0.0.1 = localhost)

- Host dosyası, sistemde manuel olarak atanmış IP-domain eşleşmelerini içerir.
- Yani bir siteye erişmek istiyorsak işletim sistemimiz önce host dosyasına bakar.
- Host dosyasında var ise doğrudan bağlantı kurulur.
- Eğer yoksa DNS çözümleme süreci başlar.
- Öncelikle DNS sorgusu için önbellek kontrolü yapılır. (cache)
- İşletim sistemi veya tarayıcı, daha önce DNS sorgusu yapmışsa DNS önbelleğinde saklanır.
- Eğer IP adresi önbellekte varsa direkt siteye erişilir.

Eğer önbellekte yoksa DNS çözümleme süreci başlar.

- Bilgisayar belirtilen domain'e sorgu gönderir.
- DNS sunucusu yanıtı kendi önbelleğinde tutuyorsa doğrudan yanıt verir. Yoksa kök DNS sunucularına başvurur.
- Öncelikle '.com', '.net' gibi alan adları için TLD sunucusuna sorgu yapılır.
- Örneğin, 'example.com' için '.com' TLD sunucusu, yetkili DNS sunucusuna yönlendirme yapar.
- Yetkili DNS sunucusu, domainin gerçek IP adresini barındıran sunucudur.
- Domaini IP adresine yönlendirir.
- Böylelikle siteye erişmiş oluruz. Aynı zamanda bu sonuç önbelleğe kaydedilir.

Bütün bu işlemler günümüz teknolojisinde çok kısa süreler içerisinde gerçekleşir.

## **TCP-/IP İLETİŞİMİ:**

Bilgisayarlar ağlar arası cihazların veri iletimi, TCP/IP protokolü ile gerçekleştirilir. TCP/IP bu paketlerin nasıl yönlendirileceğini, adreslenebileceğini ve taşınacağını belirleyen temel internet protokolüdür.

Bu süreç içerisinde NAT (Network Address Translation), kaynak/hedef IP ve port kullanımı, TCP üçlü el sıkışması (three handshake) gibi mekanizmalar işin içeresine girer.

### **NAT (Network Address Translation):**

NAT, özel bir yelen ağdaki LAN cihazlarının, tek bir genel IP adresi üzerinden internețe çıkışmasını sağlar. Özellikle IPv4 adreslerinin sınırlı olması nedeniyle kullanılır.

3 çeşit NAT türü vardır. Bunlar statik, dinamik, pat (port address translation) 'dur.

- STATİK NAT: Yerel IP adresi her zaman belirli bir genel IP'ye çevrilir. Bu durum web sunucularında kullanılır.

- DİNAMİK NAT: Kullanılabilir bir genel IP adresi havuzundan geçici olarak bir IP adresi atanır. İnternet Servis Sağlayıcıları (ISP) tarafından kullanılır.
- PAT NAT: En yaygın NAT türüdür. Birden fazla cihaz, aynı genel IP adresini farklı portlar üzerinden kullanır. Evlerde veya ofislerde, ofis ağlarında modem/routerların internete çıkışı için kullanılır.

### **IP/PORT KULLANIMI:**

Bir ağa veri iletimi sırasında IP adresleri ve port numaraları kullanılır. Her bir port farklı durumlar için kullanılır. Örneğin 3306 bir mysql portudur veya 80 portu bir http portudur gibi.

- Source IP (Kaynak IP): Veriyi gönderen cihazın IP'sidir.
- Destination IP (Hedef IP): Verinin gideceği cihazın IP'sidir.
- Source Port (Kaynak Port): Gönderen cihazın TCP/UDP port numarasıdır. (rastgele atanır.)
- Destination Port (Hedef Port): Hedef hizmetin port numarasıdır. (80 (http), 3306, ...)

Örneğin; bizim bir IP adresimiz var. 192.168.1.10 olsun. Bir websitesine gitmek istiyorum. Bu website IP olarak 10.0.0.10 kullanıyor olsun. Website https ise 8080 veya 443 portunu, http ise 80 portunu kullanır. Biz http sayfasına gitmek istiyoruz. Bunun için 10.0.0.10:80 adresine istek gönderiyoruz. Buradaki 80 porttur. Ve IP adresi bizden erişim için bu portun ne olduğunu istiyor. Bu sayede website ile sağlıklı bir şekilde iletişim kurmuş olduk.

### **TCP ÜÇLÜ EL SIKIŞMASI:**

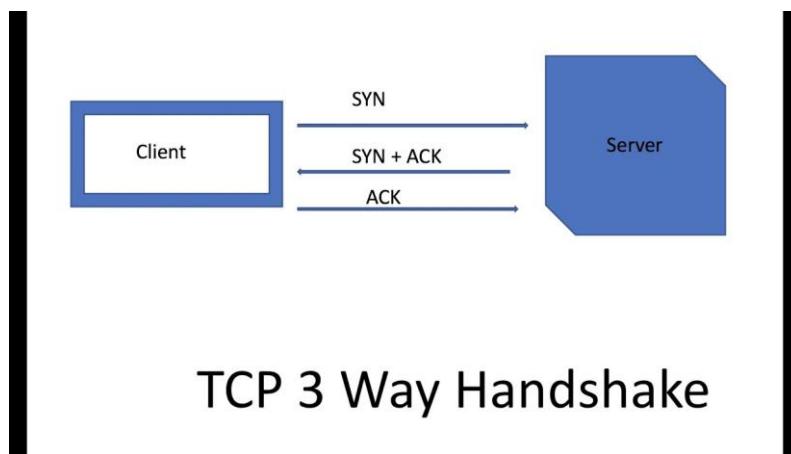
TCP, güvenilir ve kesintisiz bir şekilde bağlantı kurmak için Üçlü El Sıkışma yöntemini kullanır. Bu durumun aşamaları şu şekildedir:

- İlk başta SYN paketi istemci (client) tarafından gönderilir.
- Daha sonra sunucu bu paketi görerek bize SYN-ACK paketini tekrar geri döndürür.

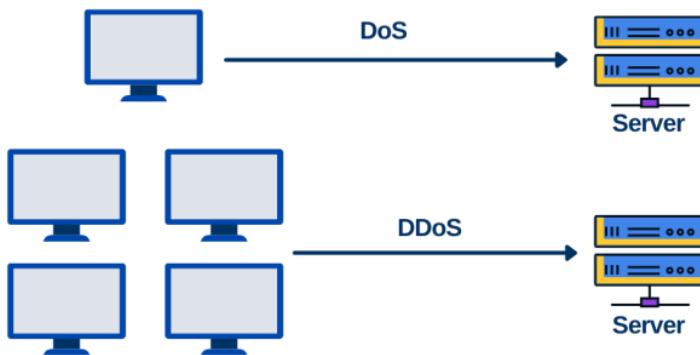
Bu paketler sayesinde bağlantı isteğinin kabul edildiğini bize bildirmiştir olur.

- Son olarak da ACK paketi istemci tarafından sunucuya gönderilir.

Bu durumlar sonucunda Üçlü El Sıkışması başarılı bir şekilde gerçekleşmiş oldu.



**ÖNEMLİ:** Eğer üçlü el sıkışmanın son aşamasında SYN-ACK paketi döndükten sonra biz sunucuya ACK paketi göndermezsek DOS saldırısı başlamış olur. Bu durumu üst üste sürekli yaparsak da sunucu karşılık alamadığı için bekleme durumunda olacak ve sistem DOS saldırısına maruz kalmış olur. Birden fazla IP yani başka cihazlarla da DOS saldırısı yapılrsa artık bu saldırının ismi DDoS saldırısı olacaktır.

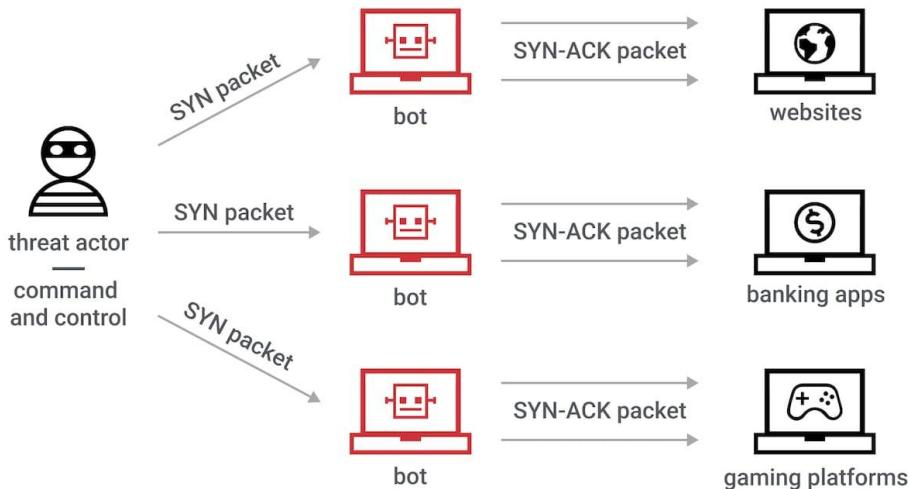


TCP bağlantı kapatma süreci:

- İstemci, sunucuya FIN paketi gönderir. (Bağlantıyı kapatmak istedığını söyler.)
- Sunucu, ACK paketi ile isteği kabul eder. (Fakat bağlantı hemen kapanmaz.)
- Sunucu, kendi FIN paketini gönderir. (Bağlantıyı sonlandırmaya hazır olduğunu söyler.)
- İstemci son olarak ACK paketini göndererek bağlantı tamamen kapanmış olur.

**AYRICA:** SYN Flood Attack, yani bağlantı kurmak istediğimizde SYN paketi gönderiyorduk. Bu isteği eğer çok fazla göndermeye başlarsak sunucu yine zarar görmeye başlayacaktır.

**ÇÖZÜM:** Rate limiting ile Aynı IP'den gelen paketleri sınırlıyoruz.



## WEB SUNUCUSU İLETİŞİMİ

### HTTP İSTEKLERİ:

HTTP (HyperText Transfer Protocol) istemci (client) ve sunucu (server) arasında veri alışverişi için kullanılır. HTTP, veriyi TCP üzerinden ileter. GET/POST süreci ise şu şekildedir:

- *GET İsteği*: Genellikle web sayfalarını veya API verilerini almak için kullanılır.

- URL içinde parametreler gönderilir.
- Önbelge alınabilir, genelde veri alma işlemlerinde kullanılır.
- Gövde kısmı içermez. (Body)

GET isteği form doldurma işlemlerinde kullanılmaz. Çünkü GET isteği URL üzerinden gösterir. Biz URL üzerinden giden verileri veya paketleri tutarsak (Burp Suite gibi araçlar ile) bunu kötüye kullanabiliriz. Bu yüzden GET isteği body kısmı içermez.

- *POST İsteği*: Sunucuya veri gönderirken kullanılır. Genellikle form gönderme işlemlerinde kullanılır.

- Kullanıcının “Gönder” tuşuna bastığı anda giden istek POST’tur.
- Parametreleri HTTP gövdesinde (body) taşıır.
- Önbelge almaz, bu risklidir.

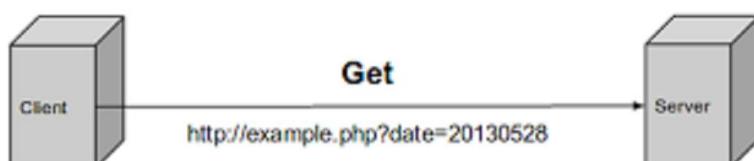
### HTTP ALTINDAKİ TCP İLETİŞİMİ:

HTTP istekleri TCP üzerinden taşınır. TCP, verinin bütünlüğünü bozmadan sıralı ve güvenli bir şekilde iletilmesini sağlayan protokoldür.

Bu durumda da TCP protokolü ile Üçlü El Sıkışma olayı gerçekleşir ve HTTP isteği iletilir.

Üçlü El Sıkışma işlemi tamamlandıktan sonra HTTP, GET/POST istekleri TCP bağlantısı üzerinden iletilir.

- HTTP isteği, TCP segmentleri içinde taşınır.
- Sunucu yanıt verince, TCP bunu güvenli şekilde istemciye ulaştırır.
- Eğer paket kaybı olursa, TCP tekrar iletim ile eksik verileri tamamlar.



## FIREWALL (GÜVENLİK DUVARI):

Ağ trafigini denetleyen ve belirli kurallar içerisinde gelen/giden paketleri engelleyen veya izin veren bir güvenlik sistemidir. Saldırılara karşı savunma mekanizması oluşturur.

*Çalışma Prensibi:*

- Kaynak IP adresi (Hangi IP'den geliyor?)
- Hedef IP adresi (Hangi IP'ye gidiyor?)
- Kaynak Port (Hangi bağlantı noktasından geliyor?)
- Hedef Port (Hangi bağlantı noktasına gidiyor?)
- Protokol (TCP, UDP, ...)
- Bağlantı durumu (Yeni mi, mevcut mu?)

Gelen ve giden tüm paketleri;

- IP adresi, portu ve protokolüne göre inceler.
- TCP bağlantılarının durumunu takip eder.
- Gelen istekleri analiz eder ve kötü amaçlı trafiği engeller. (Proxy Firewall)
- Web tabanlı saldırılara karşı koruma sağlar. (XSS, SQL Injection, ...) (WAF)

Filtremeler yapar. Zararlı bir durum görürse engel olmaya çalışır.

DDoS Koruması:

Yüksek miktarda trafik ile hedef sistemi devre dışı bırakma saldırısıdır. DDoS Saldırı Türleri:

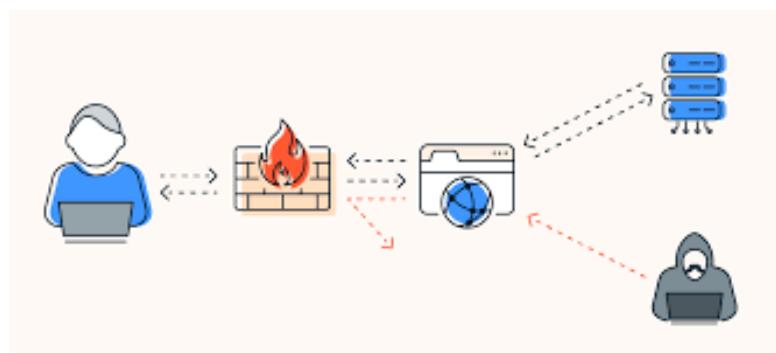
- SYN Flood: TCP bağlantılarını aşırı yükleyerek kaynak tüketir.
- UDP Flood: Hedef sunucunun işlem kapasitesini zorlar.
- ICMP Flood (Ping Flood): Çok miktarda ICMP isteği yollanır.
- HTTP Flood: Web sunucularını aşırı HTTP isteği ile meşgul eder.

Peki firewall bu durumu nasıl engeller?

Rate Limiting: Bu durum ile aynı IP adresinden çok fazla istek gelmesini engeller.

IP Kara Listeleme (Blacklisting): Şüpheli IP adresini banlar. O IP adresinden gelen tüm trafik engellenir.

**AYRICA; Firewall, NAT ile birlikte kullanılarak özel ağ güvenliği sağlanabilir.**



## REVERSE PROXY ve LOAD BALANCER:

Reverse Proxy ve Load Balancer, gelen istemci taleplerini birden fazla sunucuya yönlendiren ve web uygulamalarının performansını, güvenliğini ve ölçeklenebilirliğini arttıran kritik bileşenlerdir.

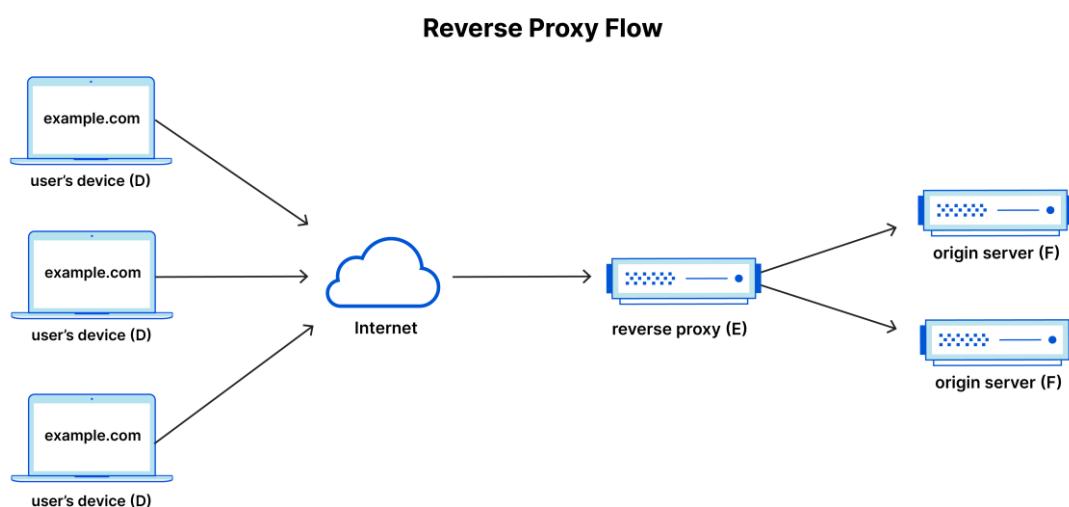
Örneğin, ÖSYM sınav sonuçlarının açıklandığı anda milyonlarca kullanıcı sınav sonucusunu öğrenmek için siteye giriş yapma isteği yolluyorlar. Bu durumda milyonlarca istek alan sunucu çöküyor ve yanıt veremez hale geliyor. Load balancer ve reverse proxyler de bu durumda meydana çıkıyorlar.

**REVERSE PROXY:** istemcilerden gelen HTTP/HTTPS isteklerini backend sunucularına yönlendiren bir sunucu türüdür.

- Kullanıcılar doğrudan backend sunucularına bağlanmaz, tüm trafik öncelikle reverse Proxy üzerinden geçer.
- Reverse Proxy, güvenlik ve yük dengeleme gibi işlemler yapar.

Aynı zamanda önbellekleme işlemleri ile de sunucu yükünü azaltır.

Reverse Proxy örneği olarak NGINX gösterilebilir.



**LOAD BALANCER:** Gelen ağ trafiğini birden fazla backend sunucusuna dağıtarak sistemin ölçeklenebilirliğini ve performansını arttıran bileşendir.

Avantajları:

- Yük Dağıtımı: Adından da anlaşılacağı gibi trafiği eşit şekilde dağıtarak sunucuların aşırı yüklenmesini öner.
- Arıza Tespiti: Çalışmayan sunucuları tespit eder.
- Oturum Yönetimi: Kullanıcı isteklerini aynı sunucuya yönlendirerek oturumu sürekli tutar.
- Güvenlik: DDoS saldırılarına karşı koruma sağlar.

Yöntemleri:

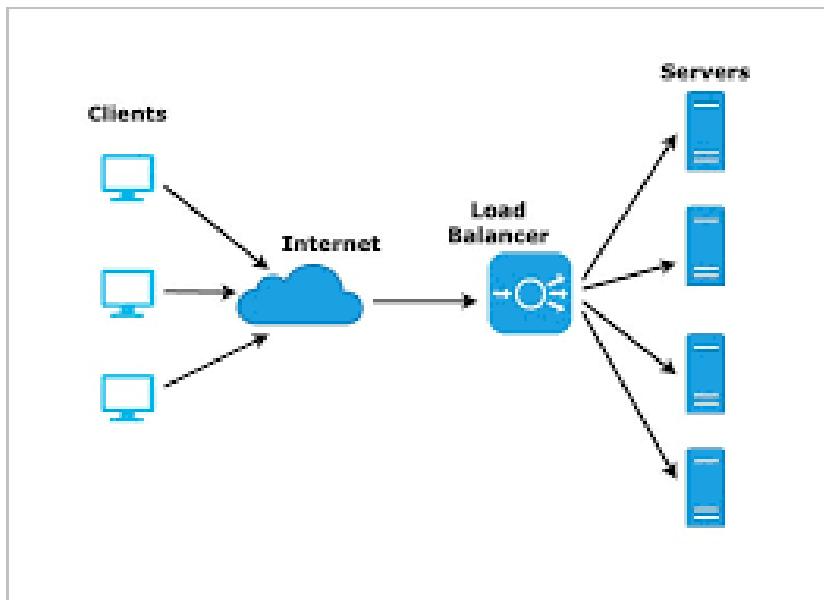
- Round Robin: İstekleri sırasıyla tüm sunuculara yönlendirir.
- Least Connections: En az bağlantıya sahip olan sunucuya gelen istekleri iletir.
- IP Hashing: Kullanıcının IP adresine göre belirli bir sunucuya yönlendirir.
- Weighted Round Robin: Sunuculara ağırlık (weight) atayarak yükü belirli sunuculara daha fazla yönlendirir.

#### REVERSE PROXY ve LOAD BALANCER İLE OTURUM (SESSIONS) YÖNETİMİ:

Web uygulamalarında kullanıcı oturumlarının yönetilmesi önemlidir. İstekler farklı sunuculara giderse oturum kaybı yaşanır.

Bunu önlemek için:

- Session Persistence kullanılır. (Oturum süresi boyunca aynı sunucuya bağlı kalmayı sağlar.)
- Session Data Replication ile oturum bilgileri tüm sunuculara kopyalanır.
- Redis veya Memcached gibi harici oturum yönetim sistemleri kullanılır.



## MODERN WEB ALTYAPISI

### MİKRO HİZMETLER ve VERİTABANI YÖNETİMİ:

*Veri tabanı yedekleme:* Veri tabanı yedekleme, veri kaybını önlemek ve felaket senaryolarına karşı koruma sağlamak için kritik bir adımdır.

Peki yedekleme türleri nelerdir?

- Tam yedekleme: Veri tabanının tamamının yedeklenmesidir.
- Artımlı yedekleme: Son tam yedeklemeden sonra değişen verilerin yedeği alınır. Geri yükleme süreci karmaşıktır.
- Fark yedekleme: Son tam yedeklemeden sonra değişen tüm verilerin yedeği alınır. Geri yükleme süreci artımlı yedeklemeye göre daha kolaydır.
- Anlık Görüntü (Snapshot): Veritabanının belirli bir andaki durumunun anlık görüntüsü alınır. Genellikle NoSQL sistemlerde kullanılır.

*Redis ile Session Yönetimi ve Replikasyon Stratejileri:*

Oturumlar (sessions), sunucu belleğinde tutulur. Ancak, birden fazla mikro hizmet ve yük dengeleme (load balancing) kullanıldığında paylaşılan bir session store gereklidir. Bu durum için Redis kullanılır.

Redis'in dağıtık sistemlerde kullanımı için master-slave replikasyonu, sentinel ve cluster yapıları kullanılır.



*Master-Slave Replikasyonu:*

- Verileri master'dan slave'e kopyalar.
- Slave sunucular sadece okumaya izin verir, master yazma işlemlerini yapar.

*Sentinel ile Yüksek Erişilebilirlik:*

- Sentinel, Redis master sunucusunu izler ve çökmesi durumunda bir slave'i yeni master olarak atar.
- Otomatik failover mekanizması sağlar.

*Cluster ile Yatay Ölçeklenebilirlik:*

- Veri otomatik olarak birden fazla düğüm (node) dağıtılr.
- Hash tabanlı bölümlendirme (sharding) sayesinde yük dengeleme yapılır.

## **STATİK DOSYALARIN YÖNETİMİ:**

Statik dosyalar (CSS, JavaScript, resimler, videolar, fontlar, ...) yönetimi için CDN (Content Delivery Network) kullanılır. Ve ayrıca dosya yükleme işlemi kullanılır. Peki bu CDN nedir?

CDN, statik içerikleri hızlandırmak ve yükü dağıtmak için dünya çapında konumlandırılmış sunucu ağıdır. Kullanıcılar, içeriği doğrudan orijinal sunucudan (origin server) almak yerine kendilerine en yakın CDN sunucusundan alırlar.

*Avantajları:*

- Daha Hızlı Yükleme Süresi
- Daha Az Sunucu Yükü
- DDoS Koruması
- Güvenilirlik ve Yüksek Erişilebilirlik
- Bant Genişliği Tasarrufu

CDN kullanan platformlara örnek olarak;

- Cloudflare CDN
- AWS CloudFront
- Google Cloud CDN
- Akamai CDN
- Fastly

Gibi dev markalar CDN kullanır.

*Dosya Yükleme Teknikleri:*

- HTML Form ile
- PHP ile
- AJAX ile (Asenkron Yöntem)
- Bulut Depolama Kullanımı (AWS S3, Google Cloud Storage)
- Node.js ile

Gibi teknikler kullanılır.



## SCABILITY (ÖLÇEKLENEBİLİRLİK):

Büyük ölçekli sistemlerde artan kullanıcı sayısı, yüksek trafik ve sistem güvenilirliği önemli faktörlerdir. Ölçeklenebilirlik (Scalability), sistemin artan yükü kaldırabilecek şekilde büyütülmesini ifade eder.

Ölçeklenebilirlik için *sunucu sayısını artırma, yük dengeleme (Load Balancing) ve failover (Hata Toleransı)* stratejileri kullanılır.

Ölçeklenebilirlik genellikle 2 farklı yöntemle sağlanır.

### Dik Ölçekleme:

- Mevcut sunucunun donanımını artırarak ölçeklendirme yapılır.
- CPU, RAM, disk kapasiyesi yükseltilir.
- Daha güçlü bir sunucuya geçmek anlamına gelir.

### Yatay Ölçekleme:

- Yeni sunucular ekleyerek sistemin genişletilmesi işlemidir.
- Trafik arttıkça yeni sunucular eklenir ve yük dengelenir.
- Genellikle mikro hizmetler ve dağıtık sistemler için uygundur.

## YÜK DENGELEME (LOAD BALANCER):

Yük dengeleme (Load Balancer), gelen ağ trafigini birden fazla sunucuya dağıtarak sistemin performansını ve erişilebilirliğini artırır.

### Çalışma Prensibi:

- ➔ Kullanıcıdan gelen istekler, Load Balancer üzerinden geçerek uygun bir sunucuya yönlendirilir.
- ➔ Dengeli yük dağılımı sayesinde bir sunucu aşırı yüklenmez.
- ➔ Sunucular arasında ölçeklenebilirlik ve hata toleransı sağlar.



## **FAILOVER ve HATA TOLERANSI:**

Ölçeklenebilir sistemlerde hata toleransı (Fault Tolerance) ve Failover stratejileri kullanılır.

### *Failover Mekanizması:*

- ❖ Bir sunucu çökerse, trafik çalışan diğer sunuculara yönlendirilir.
- ❖ Veritabanı replikasyonu ile veriler yedeklenir.
- ❖ Otomatik hata tespiti ve iyileştirme sağlanır.

### *Yüksek Erişilebilirlik (High Availability - HA) İçin Çözümler:*

- ◆ Sunucu Yedekleme (Backup Servers) → Bir sunucu çökerse yedeği devreye girer.
- ◆ Veri tabanı Replikasyonu → Master-Slave veya Master-Master sistemleri kullanılır.
- ◆ Konteyner ve Kubernetes Kullanımı → Servisler otomatik ölçeklenebilir.

### *Mikro Hizmetlerde Ölçeklenebilirlik:*

Mikro hizmet mimarisi, hizmetlerin bağımsız ölçeklenmesini sağlar.

- ❖ Her servis bağımsız çalıştığı için ihtiyaca göre ölçeklenebilir.
  - ❖ Container teknolojileri (Docker, Kubernetes) yaygın olarak kullanılır.
- 💡 Örnek: Kubernetes ile Otomatik Ölçekleme (Auto-Scaling)

