



# NGINX

Fatih Köprücü

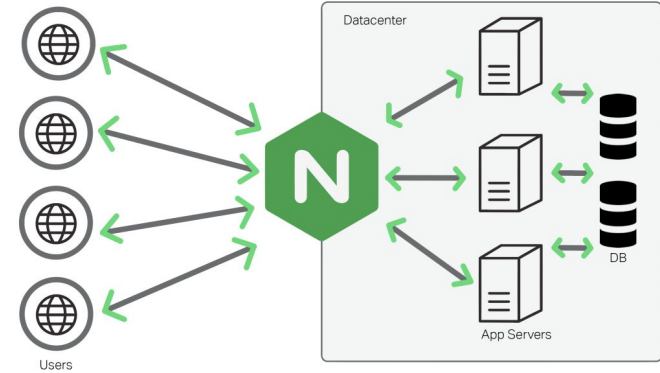


## İÇERİK

- NGINX Nedir?
- Kullanım Amaçları
- Avantajları ve Dezavantajları
- Proxy ve Reverse Proxy
- Load Balancer
- Reverse Proxy Olarak Kullanımı
- Load Balancer Olarak Kullanımı

## NGINIX Nedir?

- ❑ Web Server Yazılımı
- ❑ Açık kaynaklı
- ❑ 2004
- ❑ C10k Problemi



## Kullanım Amaçları



### NGINX Open Source

- Basic load balancer
- Content Cache
- Web Server
- Reverse Proxy
- SSL termination
- Rate limiting
- Basic authentication
- 7 metrics

### NGINX Plus

- + Advanced load balancer
  - + Health checks
  - + Session persistence
  - + Least time alg
- + Cache purging
- + High Availability
- + JWT Authentication
- + OpenID Connect SSO
- + NGINX Plus API
- + Dynamic modules
- + 90+ metrics



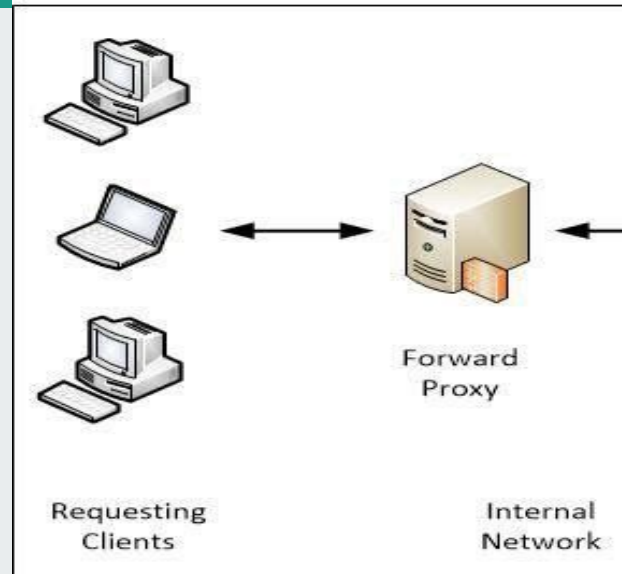
## Avantajları

- ★ Basit Kurulum
- ★ Verimli kaynak kullanımı
- ★ SEO desteği
- ★ Programlama dilleri ile uyumluluk
- ★ Çoklu bağlantı yönetimi

## Dezavantajları

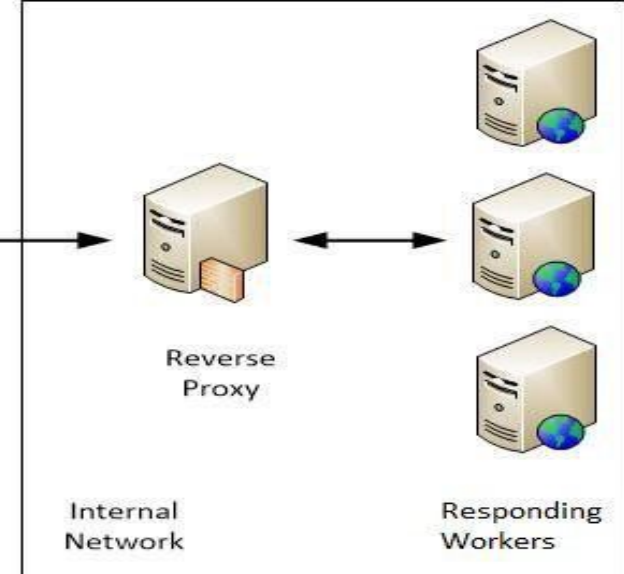
- Destek yetersizliği  
(Apache'ye oranla)
- Modul ve uzantı yetersizliği

## Proxy



- ★ IP Gizleme
- ★ Güvenlik
- ★ Cache

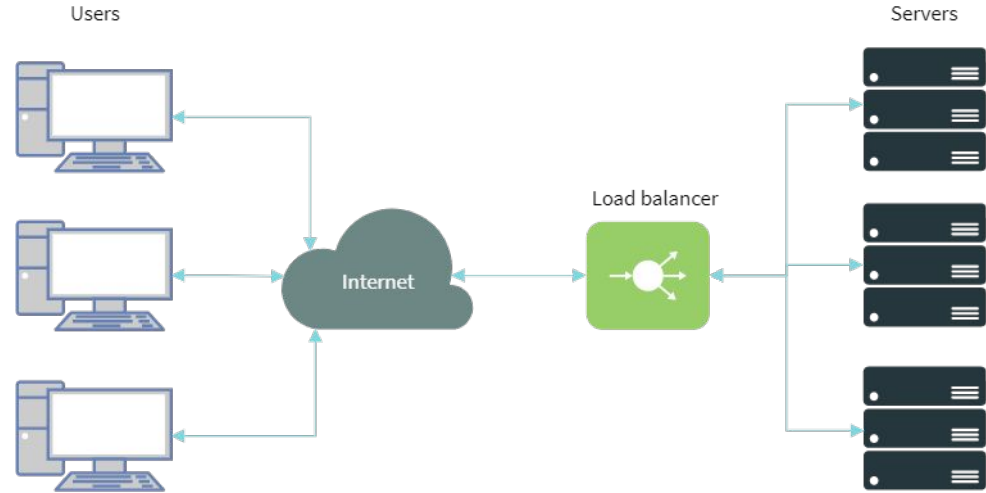
## Reverse Proxy



- IP Maskeleye
- Saldırılarına karşı koruma
- SSL
- Cache

## Load Balancer

- ❏ Round Robin
- ❏ Least Connections
- ❏ Ip Hash
- ❏ Ağırlıklı





## Reverse Proxy Olarak Kullanımı

```
server {  
    listen 80;  
  
    location / {  
        proxy_pass http://my_server;  
    }  
}
```





## Load Balancer Olarak Kullanımı

```
upstream backend {  
    server backend1.example.com;  
    server backend2.example.com;  
    server backend3.example.com;  
}
```

---

Teşekkürler...