

Bir sunucuda birden fazla projeyi alıřtırma

- İlk olarak, "sudo -u postgres psql "ile postgresle baėlandı
- CREATE DATABASE myproject; veri tabanı oluřturdum (her proje farklı veri tabanı oluřturdum)
- "\q" postsqlden ıkıř yaptım
- Her projeye ait bir sanal ortam oluřturdum ve bu sanal ortamlara gunicorn'u indirdim (pip install django gunicorn).
- Her projenin "settings.py" dosyasına "STATIC_ROOT = os.path.join(BASE_DIR, 'static/')" satırını ekleyerek, statik dosyaların nereye toplanacaėını belirttim.
- "migrations", "migrate", "createsuperuser" ve "collectstatic" komutlarını alıřtırdım.
- Her proje iin farklı bir port kullanmak iin "sudo ufw allow 8000" komutunu alıřtırdım.
- Projeyi kontrol etmek iin "gunicorn --bind 0.0.0.0:8000 myproject.wsgi" komutunu alıřtırdım (statik dosyalar alıřmayacaktır).
- "sudo nano /etc/systemd/system/gunicorn.socket" komutunu kullanarak "gunicorn.socket" adında bir dosya atı

```
[Unit]
Description=tubitak socket

[Socket]
ListenStream=/run/tubitak.sock

[Install]
WantedBy=sockets.target
```

- Bu adımları diėer projem iin de aynı řekilde uyguladım, ancak soket adını farklı belirledim.
- "sudo nano /etc/systemd/system/gunicorn.service" komutunu kullanarak "gunicorn.service" adında bir dosya atı

- Dosyanın içine aşağıdaki metni yazdım bunu iki proje içinde yaptım buradaki bilgileri sucunuza ve projenize göre ayarlayınız

```
[Unit]
Description=erp daemon
Requires=erp.socket
After=network.target

[Service]
User=harun
Group=www-data
WorkingDirectory=/home/harun/erp
ExecStart=/home/harun/erp/venv/bin/gunicorn \
    --access-logfile - \
    --workers 3 \
    --bind unix:/run/erp.sock \
    django_erp.wsgi:application

[Install]
WantedBy=multi-user.target
```

- “sudo systemctl start gunicorn.socket sudo systemctl enable gunicorn.socket” komutlarını kullanarak soketi çalıştırdım
- “sudo systemctl status gunicorn.socket” bu komutla soketin durumunu kontrol ettim çalışıyor mu diye
- “sudo systemctl daemon-reload “ bu komut ile değişiklikleri yükledim
- sudo systemctl restart gunicorn” bu komutla beraber gunicornu yenide başlattım
- “sudo nano /etc/nginx/sites-available/myproject “ bu komutla bir ngnix dosyası oluşturdum ve içine aşağıdaki metni yazdım

```

server {
    listen 80;

    server_name harun.novipool.com;

    location / {
        proxy_pass http://unix:/run/gunicorn.sock; # İlk projenin Gunicorn sunucusunun adresi ve portu
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /static/ {
        alias /home/harun/baris/static/; # Tubitak projesinin static dosyalarının yolu
    }
}

server {
    listen 80;

    server_name efe.novipool.com;

    location / {
        proxy_pass http://unix:/run/erp.sock; # İkinci projenin Gunicorn sunucusunun adresi ve portu
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
    location /static/ {
        alias /home/harun/erp/static/; # ERP projesinin static dosyalarının yolu
    }
}

server {
    listen 80;

    server_name s3.novipool.com;

    location / {
        #proxy_pass http://unix:/run/minio.sock;
        proxy_pass http://89.252.131.133:35857; # İkinci projenin Gunicorn sunucusunun adresi ve portu
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

- bu metin ile domaine gelen istekleri oluşturduğumuz soketlere yönlendiriyoruz
- “sudo nginx -t” nginx dosyasını test ediyoruz
- “sudo systemctl restart nginx” nginx yeniden başlatıyoruz
- minioyu ise nohup ile arka tarafta çalıştırıyoruz