

# High Availability, Load Balancing, and Replication

## High Availability nedir?

Bu cluster yapısı, herhangi bir serverda bir hata ortaya çıkarsa ilgili işlemi cluster'daki başka bir serverın devralmasıyla erişilebilirliği her hâlükârda sağlamayı amaçlar. Bu durumda client hiçbir şeyden etkilenmeden işlemlerini gerçekleştirir.

## Failover

Cluster'lara bağlı serverların herhangi birinde ortaya bir hata çıkarsa ilgili işlemi farklı bir serverın devralmasına, hatayla uğraşırken diğer servis kesintisiz olarak çalışmaya devam eder. Bu duruma "**failover**" denir.

## Load Balancing nedir?

Load balancing (LB) clusterlarda öncelikli amaç ağır yük altında çalışan servislerin yüklerinin birden fazla sunucu arasında dengelenmesi ve bu sayede performans artışı sağlanmasıdır.

## Replication nedir?

Replikasyon ile canlı veritabanının sorgulabilir bir kopyası oluşturulur. Raporlama amaçlı sorgular bu read-only sunucu üzerinde çalıştırılarak master sunucu üzerindeki yük azaltılmış olur.

# Docker üzerinde Replication kurulumu

[Buradan](#) compose dosyasını indirebilir veya bir **docker-compose.yml** dosyası oluşturup aşağıdaki kodu içine yazıyoruz.(Eğer indirirseniz dosyanın ismini **docker-compose.yml** olarak değiştiriniz.)

## docker-compose.yml

```
version: '2'

services:
  postgresql-master:
    image: 'bitnami/postgresql:latest'
    ports:
      - '5432'
    volumes:
      - 'postgresql_master_data:/bitnami/postgresql'
    environment:
      - POSTGRESQL_REPLICATION_MODE=master
      - POSTGRESQL_REPLICATION_USER=repl_user
      - POSTGRESQL_REPLICATION_PASSWORD=repl_password
      - POSTGRESQL_USERNAME=my_user
      - POSTGRESQL_PASSWORD=my_password
      - POSTGRESQL_DATABASE=my_database
  postgresql-slave:
    image: 'bitnami/postgresql:latest'
    ports:
      - '5432'
    depends_on:
      - postgresql-master
    environment:
      - POSTGRESQL_REPLICATION_MODE=slave
      - POSTGRESQL_REPLICATION_USER=repl_user
      - POSTGRESQL_REPLICATION_PASSWORD=repl_password
      - POSTGRESQL_MASTER_HOST=postgresql-master
      - POSTGRESQL_PASSWORD=my_password
      - POSTGRESQL_MASTER_PORT_NUMBER=5432

volumes:
  postgresql_master_data:
```

(Kullanıcı adı şifre gibi değerleri kendinize göre değiştirebilirsiniz.)

Bu işlemten sonra bu dosyanın bulunduğu konumda aşağıdaki komutu çalıştırıyoruz.

```
docker-compose up
```

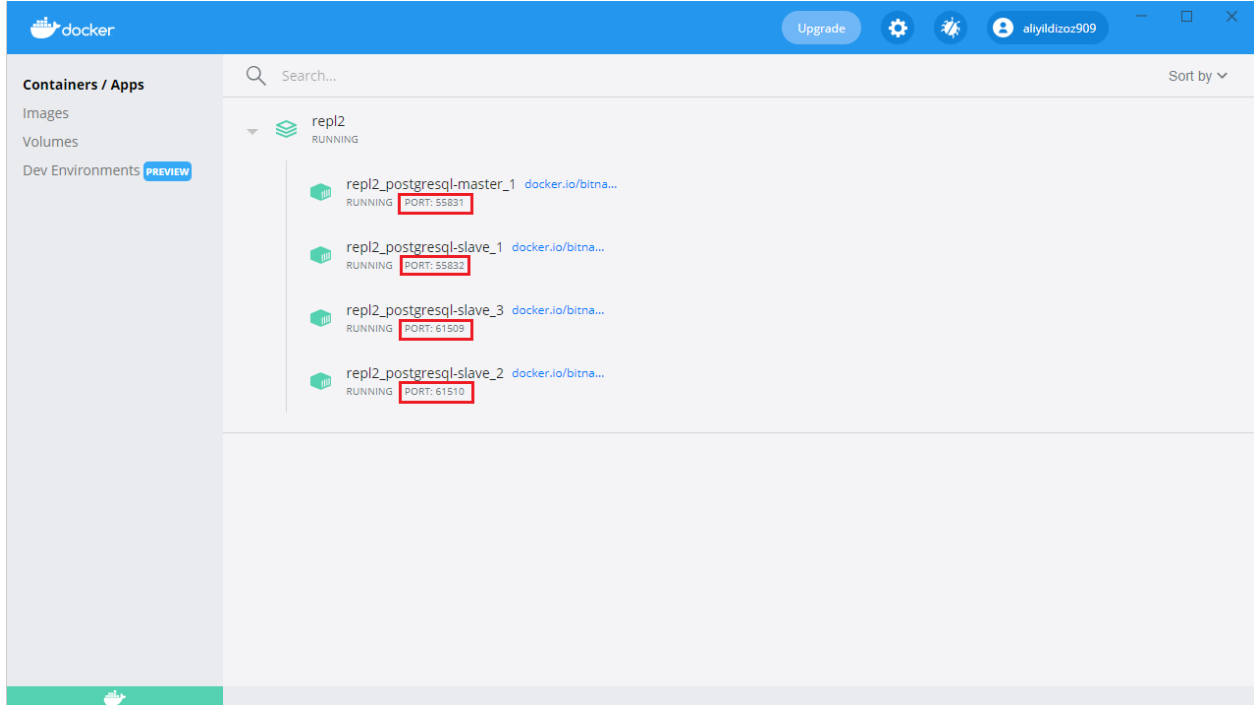
Bu şekilde sadece 1 master ve 1 slave serverı oluşacak.

**Daha fazla slave veya master ekleyerek ölçeklendirmek için aşağıdaki komutu aynı klasörde slave ve master sayısını girerek çalıştırınız.**

```
docker-compose up --detach --scale postgresql-master={master-sayısı} -  
-scale postgresql-slave={slave-sayısı}
```

## Serverlara bağlanmak

Master veya slave serverlarına bağlanmak için docker üzerinde onlara atanmış port numaralarını bilmemiz gerekiyor. Aşağıda docker desktop üzerinde çalışan serverların port numaraları yazıyor.



| Server-Adı | Port numarası |
|------------|---------------|
| Master     | 55831         |
| Slave-1    | 55832         |
| Slave-2    | 61509         |
| Slave-3    | 61510         |

**Not:** Ben burada bu örnek üzerinden bağlanacam sizde hangi portta çalışıyorsa öyle yapın.

### Master'a bağlanma

- `psql -h localhost -U postgres -p 55831`
- Password for user postgres:my\_password
- postgres=#

### Slave-1'e bağlanma

- `psql -h localhost -U postgres -p 55832`
- Password for user postgres:my\_password
- postgres=#

### Slave-2'ye bağlanma

- `psql -h localhost -U postgres -p 61509`
- Password for user postgres:my\_password
- postgres=#

### Slave-3'e bağlanma

- `psql -h localhost -U postgres -p 61510`
- Password for user postgres:my\_password
- postgres=#

**Not:** Slave serverları read-only olduğu için bu veri tabanlarında **kayıt**, **güncelleme** ve **silme** işlemleri yapamazsınız sadece **select** işlemleri yapılabilir. Eğer yaparsanız aşağıdaki hatayı alırsınız.

**ERROR:** cannot execute INSERT in a read-only transaction

# Kaynaklar

- [bitnami/bitnami-docker-postgresql\(github.com\)](https://github.com/bitnami/bitnami-docker-postgresql)
- [AtifCeylan-PostgreSQLdeOlceklenebilirlikVeYedeklilik.pdf](#)
- [PostgreSQL'de Uygulamalı. \(Streaming Replication\)](#)
- [PostgreSQL Streaming Replication Nedir ? | Azbucuk](#)