

# Enoca .Net Eğitim Task 2

**Konu:** Redis ile çeşitli verileri veritabanına gitmeye gerek kalmadan sorgulama

Talimatlar: Redis kullanarak basit bir cache yapısı kurmanız beklenmektedir, bu görev öğrenme amaçlı olduğu için öncelikle sayfanın altına belirtilen kaynaklar kısmındaki linklerden araştırma yapmanız gerekmektedir kaynakların yetersiz olduğunu düşündüğünüz noktada farklı kaynakları da kullanabilirsiniz

**Teknolojiler:** Asp.Net web API, Redis, StackExchangeRedis, Docker, RedisInsight

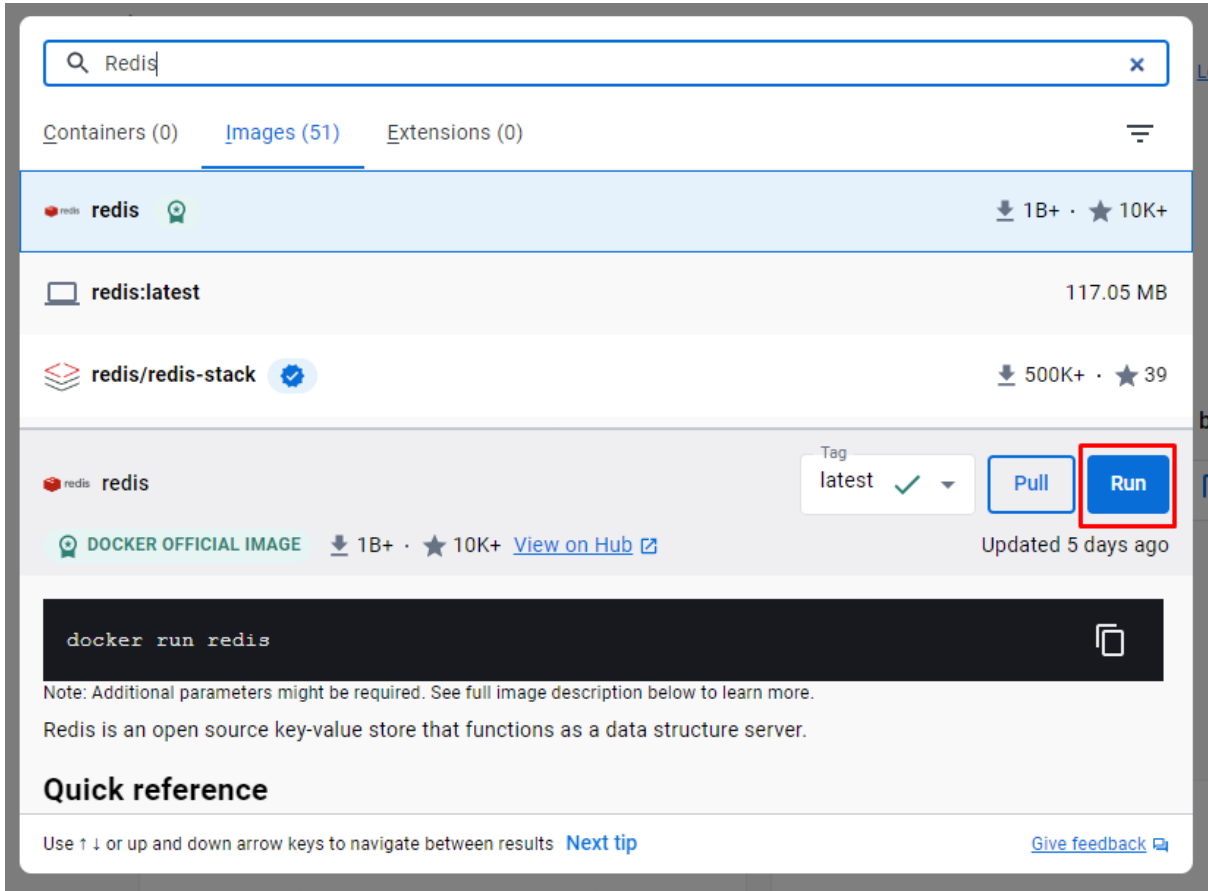
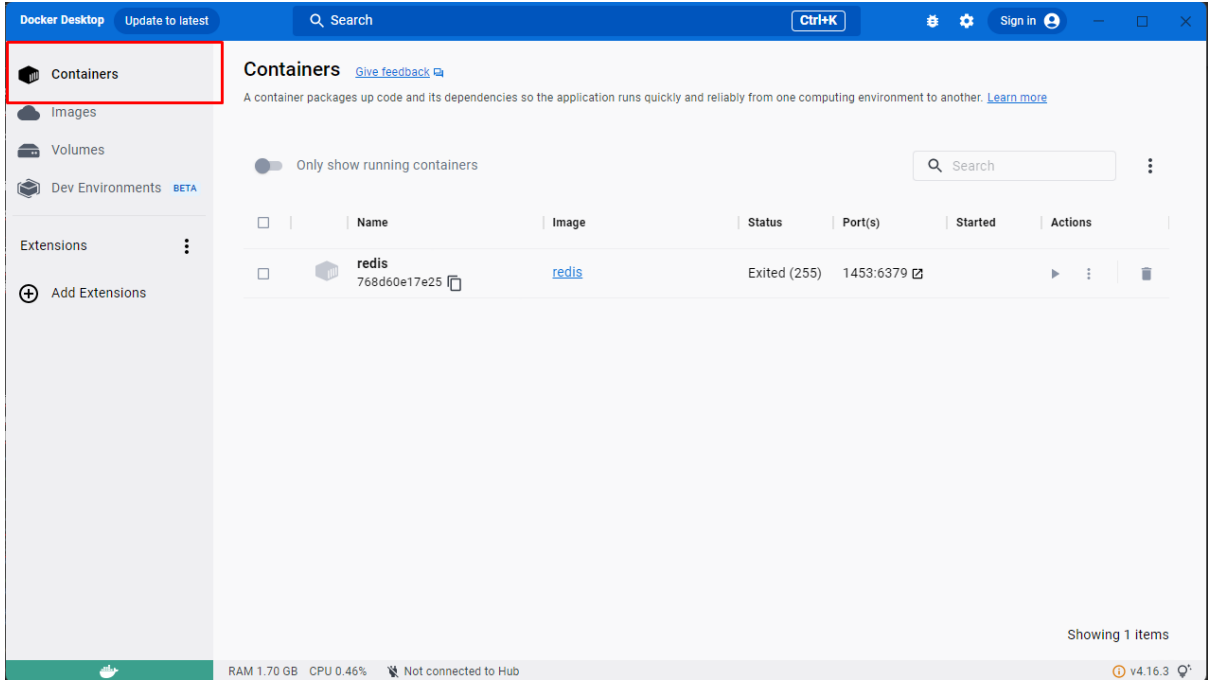
## **Yapılması Beklenenler:**

1. Api ile kullanıcıdan alınan uygulama konfigürasyon bilgilerini redis ile cache e kaydetme ve sonrasında bu verileri cache üzerinden okuyarak kullanıcıya gösterme
2. Kullanıcıdan alınan ve cache üzerinde tutulan verilerin farklı expire time larında güncellenmesini sağlama, `AbsoluteExpiration` ve `SlidingExpiration` işlemlerinin kullanılması
- 3.Yukardaki işlemlerin `DistributedCache` ve `InMemoryCache` olarak yapılması

## **Ön Bilgi:**

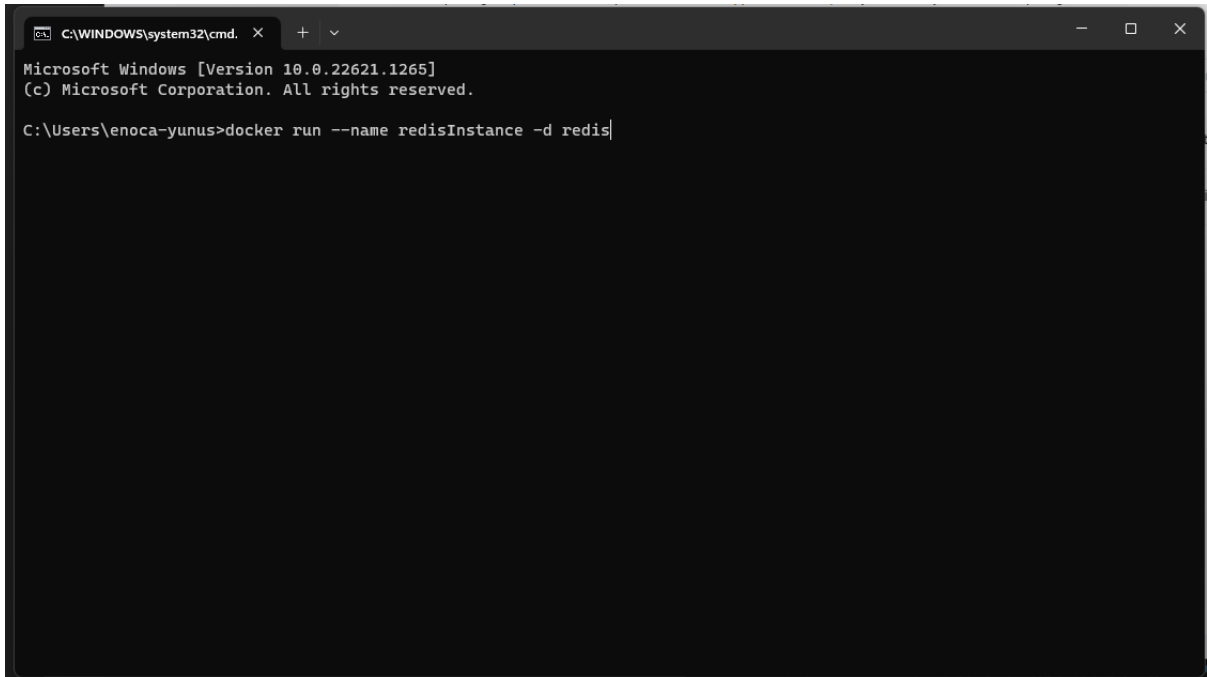
Cache nedir ne için kullanılır: Sunucu tarafında veritabanına sorgulama yapma sırasında oluşan maliyeti azaltmak amacıyla verilerin bir kısmını Ram bellek üzerine kaydedip Ram üzerinden okuma yapmamızı sağlayan mekanizma cache olarak adlandırılır. Cache `Distributed` ve `InMemory` olarak iki farklı yapıda kullanılabilir. Redis kullanabilmek için docker üzerinde ayağa kaldırılması gerekir bu işlem aşağıdaki reismlede anlatılmıştır.

# Enoca .Net Eğitim Task 2



Bir diğer redis container ayağ kaldırma yöntemi de cmd üzerinden `docker run - -name {container_adı} -d redis` komutudur

# Enoca .Net Eğitim Task 2



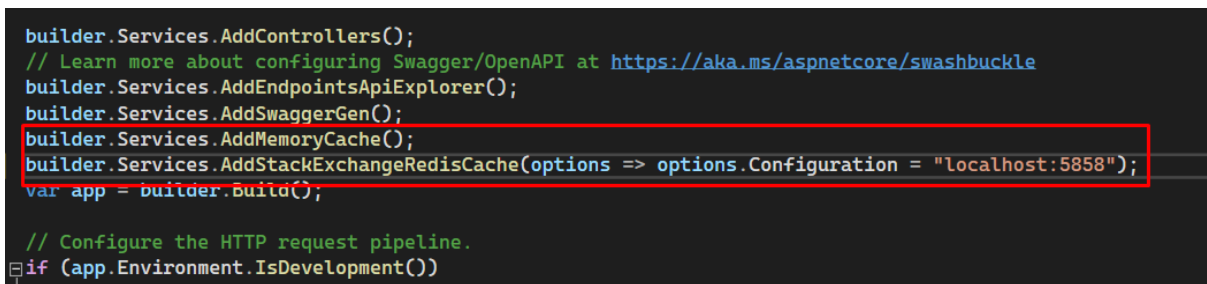
```
C:\WINDOWS\system32\cmd. X + -
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\enoca-yunus>docker run --name redisInstance -d redis|
```

Bu komutla redis varsayılan port olan 6379 ile çalışır bunu değiştirmek için aşağıdaki komut ile bir çalıştırma işlemi gerçekleştirilebilir

Docker run --name {container\_name} -p 5858:6379 -d redis => bu komut varsayılan 6379 portunu 5858 portuna bağlar ve biz 5858 portuna bağlanarak docker üzerinde oluşturulan redis container a erişim sağlamış oluruz.

.Net üzerinde Redis ile Cache kullanmak için [StackExchangeRedis](https://aka.ms/aspnetcore/swashbuckle) kütüphanesini kullanmak gerekir. Kütüphane kurulduktan sonra .Net 6 üzerinde program.cs dosyasında servisler içerisine kurulan paketin eklenmesi gerekir.



```
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddMemoryCache();
builder.Services.AddStackExchangeRedisCache(options => options.Configuration = "localhost:5858");
var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
```

Bu adımla birlikte projemize Redis implemente etme aşamaları tamamlanmış oldu Redisin yazılım içerisinde kullanımı için yardımcı kaynaklar kısmını takip edebilir veya kendiniz farklı kayaklardan araştırma yapabilirsiniz.

Yardımcı Kaynaklar:

<https://www.emrakin.com/2019/04/net-core-uygulamasnda-redis-cashe.html>

# Enoca .Net Eğitim Task 2

<https://www.gencayyildiz.com/blog/redis-yazi-serisi-13-asp-net-corede-session-icin-redis-cache-kullanimi/>

[https://www.youtube.com/watch?v=jx8KITTTq8Q&list=PLQVXoXFVVtp3\\_UIZu9qibcUzfm9ve3yVO&index=2](https://www.youtube.com/watch?v=jx8KITTTq8Q&list=PLQVXoXFVVtp3_UIZu9qibcUzfm9ve3yVO&index=2)

<https://www.youtube.com/watch?v=6HZVu3kGOrg>

<https://muratsuzen.github.io/posts/aspnet-core-redis-cache-kullanimi/>