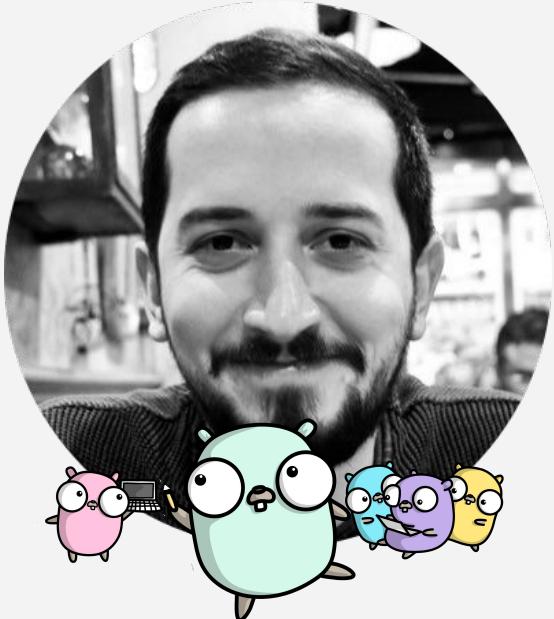


# Go ile 1.5M Throughput



# Hüseyin Güner

2013 - 2017 Sakarya Üniversitesi - Bilgisayar Mühendisliği

2018 Trendyol - Software Engineer

 [gunerhuseyin](#)

 [huseyin.guner](#)

 [huseyin-guner](#)

 [husguner](#)

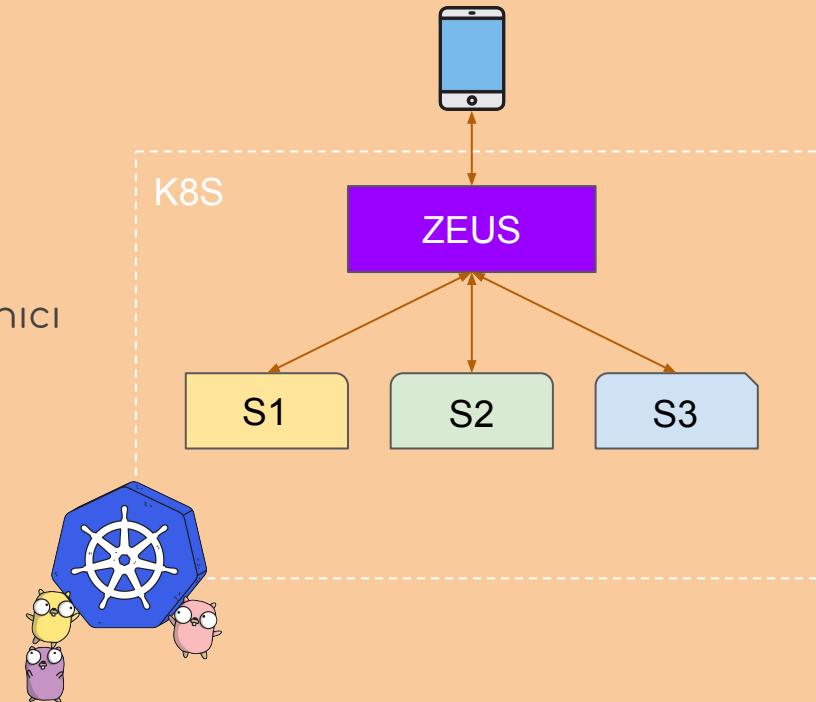
 [huseyin.guner@trendyol.com](mailto:huseyin.guner@trendyol.com)

# Ajanda



# Trendyol'da BFF Olmak

- Kullanıcıların %70'i mobil uygulama kullanıyor
- Günlük ortalama 130000 kullanıcı
- Kampanya döneminde ortalama 500000 kullanıcı
- Dakikada ortalama 1.14M istek
- 50+ servis



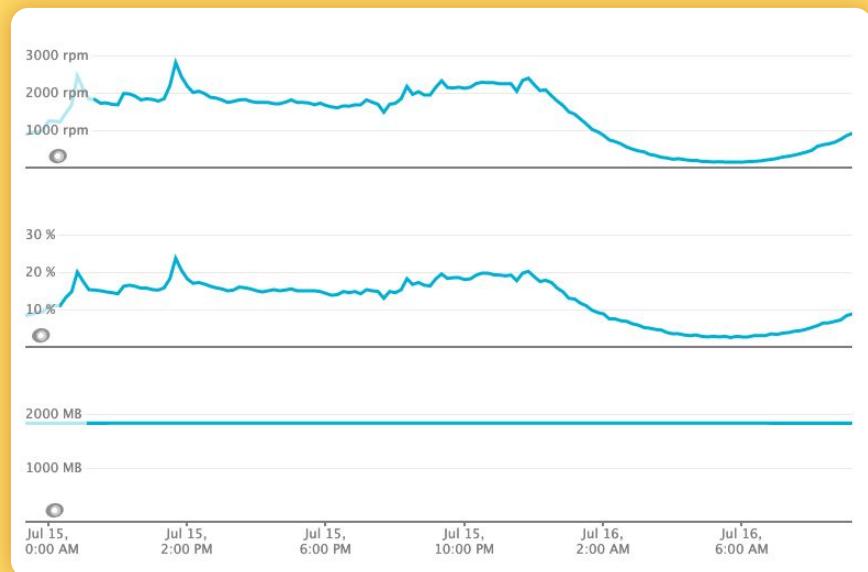
# JAVA ile Yaşadığımız Zorluklar

1 Yüksek kaynak tüketimi

2 Ayağa kalkış zamanı

3 Autoscale senaryolarını uygulayamamak

4 Eş zamanlı istekler



Trendyol Group

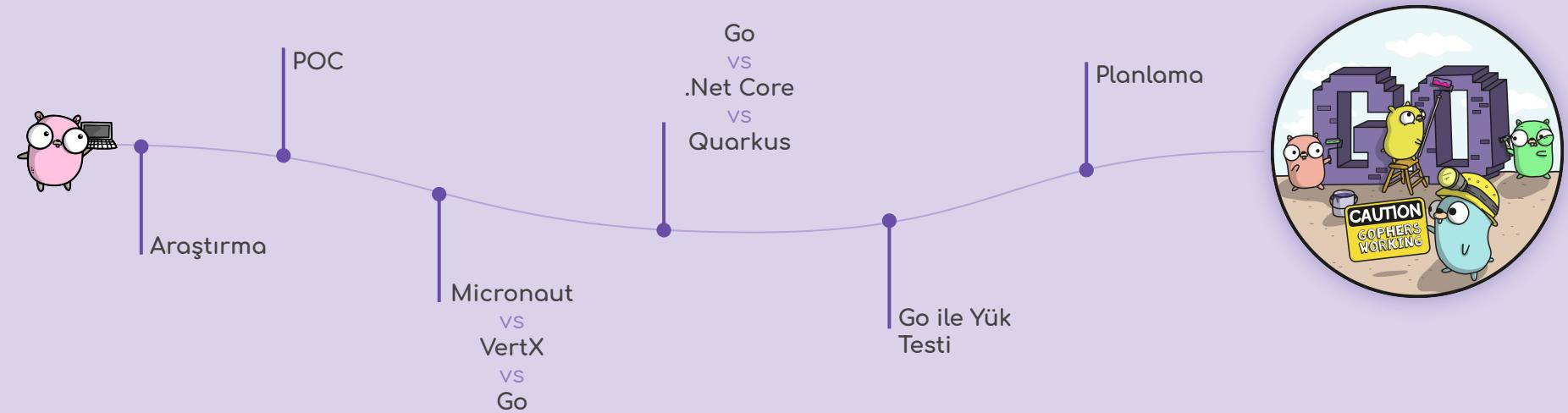
trendyol.com

trendyol  
tech

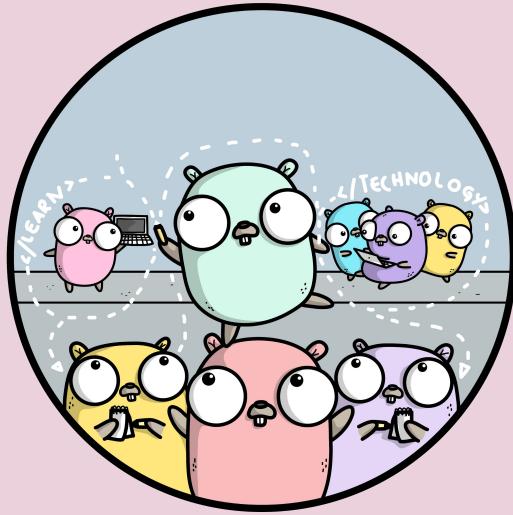
trendyol  
express

dolap

# Go Replatforminge Karar Vermek



# Go Öğrenme ve Geliştirme Serüveni



# Kaynaklar



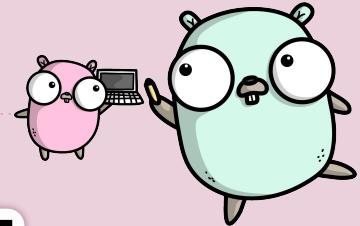
Tour Of  
Go



Learn Go  
Programming



Go 101



Go by  
Example



Awesome  
Go



RunGo

# Http Server

```
package main

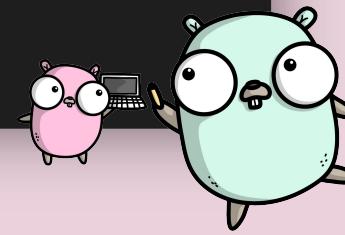
import (
    "net/http"
    "github.com/labstack/echo/v4"
    "github.com/labstack/echo/v4/middleware"
)

func main() {
    // Echo instance
    e := echo.New( )

    // Routes
    e.GET("/", hello)

    // Start server
    e.Logger.Fatal(e.Start(":8080"))
}

// Handler
func hello(c echo.Context) error {
    return c.String(http.StatusOK, "Hello, World!")
}
```



# Reverse Proxy

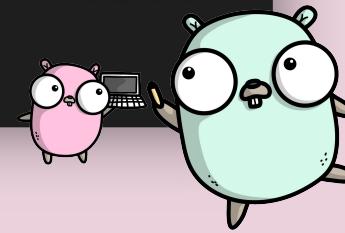
```
package main

import (
    "net/url"
    "github.com/labstack/echo"
    "github.com/labstack/echo/middleware"
)

func main() {
    e := echo.New()

    // Setup proxy
    url, err := url.Parse("http://gophercon.ist")
    if err != nil {
        e.Logger.Fatal(err)
    }
    targets := []*middleware.ProxyTarget{
        {
            URL: url,
        },
    }
    e.Use(middleware.Proxy(middleware.NewRoundRobinBalancer(targets)))

    e.Logger.Fatal(e.Start(":8080"))
}
```



# Benchmark

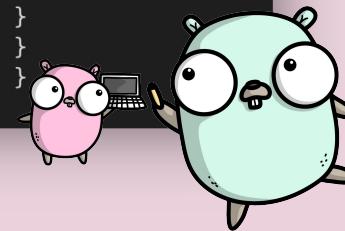
```
package user

import (
    "testing"
)

func Fib(n int) int {
    if n < 2 {
        return n
    }
    return Fib(n-1) + Fib(n-2)
}

func benchmarkFib(i int, b *testing.B) {
    for n := 0; n < b.N; n++ {
        Fib(i)
    }
}

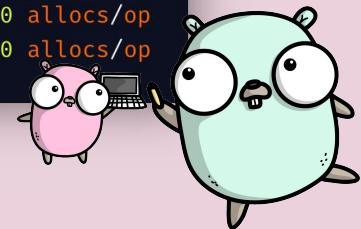
func BenchmarkFib1(b *testing.B) { benchmarkFib(1, b) }
func BenchmarkFib2(b *testing.B) { benchmarkFib(2, b) }
func BenchmarkFib3(b *testing.B) { benchmarkFib(3, b) }
func BenchmarkFib10(b *testing.B) { benchmarkFib(10, b) }
func BenchmarkFib20(b *testing.B) { benchmarkFib(20, b) }
func BenchmarkFib40(b *testing.B) { benchmarkFib(40, b) }
```



# Benchmark

```
~ go test -bench=. -benchmem
```

BenchmarkFib1-8	615455739	2.03 ns/op	0 B/op	0 allocs/op
BenchmarkFib2-8	212591606	5.60 ns/op	0 B/op	0 allocs/op
BenchmarkFib3-8	127227314	9.35 ns/op	0 B/op	0 allocs/op
BenchmarkFib10-8	3549856	354 ns/op	0 B/op	0 allocs/op
BenchmarkFib20-8	28455	41759 ns/op	0 B/op	0 allocs/op
BenchmarkFib40-8	2	640903980 ns/op	0 B/op	0 allocs/op



# Go ile İlk Test ve Sonuçları

✗ Memory leak sorunu

✗ Trafikle beraber artan kaynak tüketimi

✗ Eş zamanlı isteklerde yaşanan peakler

✗ Zaman aşımı ve loglama sorunları

+ Düşük kaynak tüketimi

+ Hızlı scale olabilme

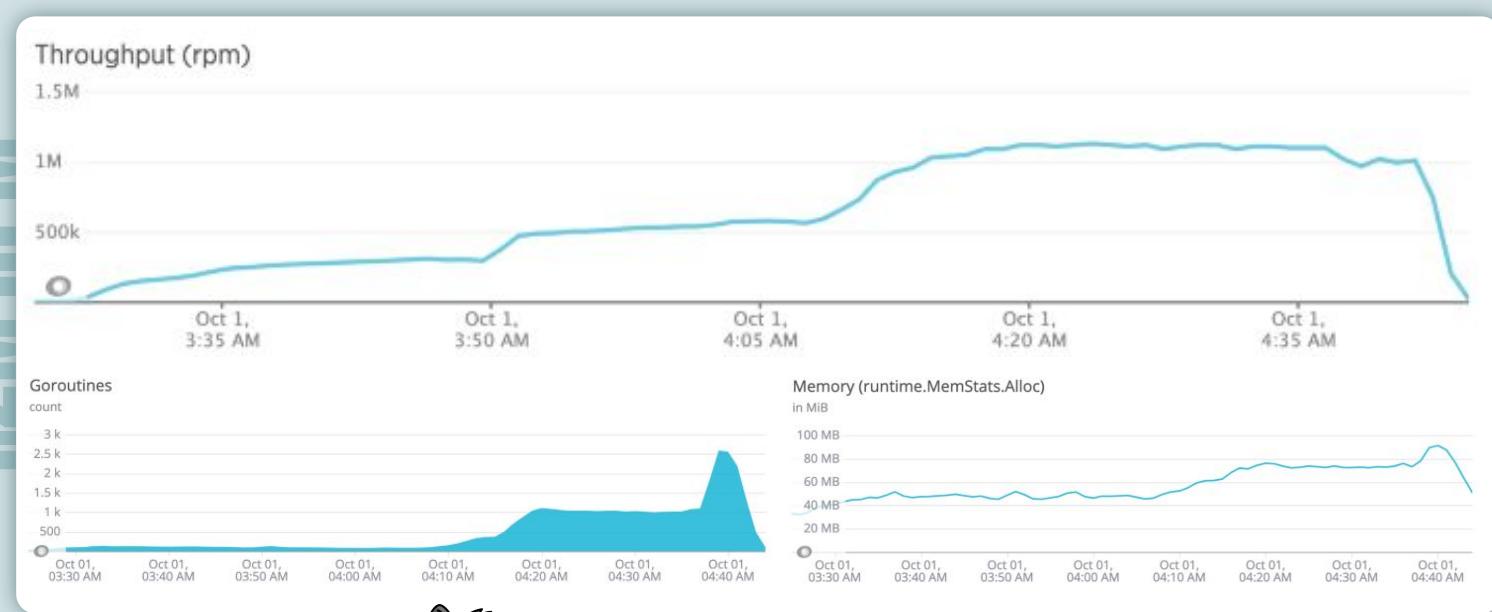
+ ~3sn ayağa kalkma süresi

+ Warmup gibi bir sorunun olmamısı



# Go ile İlk Test ve Sonuçları

1.2M RPM



Trendyol Group

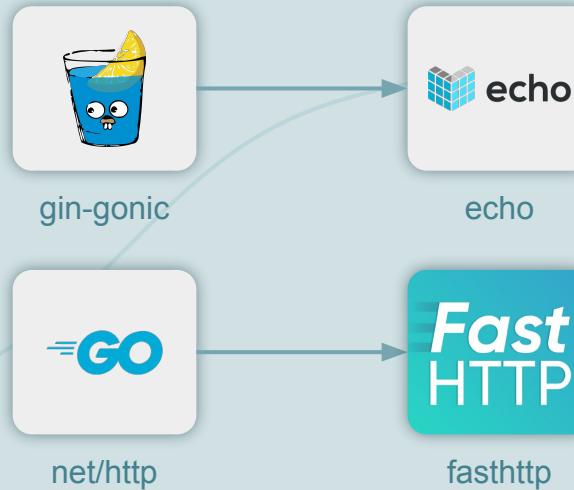
trendyol.com

trendyol  
tech

trendyol  
express

dolap

# Optimizasjon



# Optimizasyon

```
● ● ●  
package main  
  
import (  
    "bytes"  
    "compress/gzip"  
    "io"  
    "io/ioutil"  
    "sync"  
    "testing"  
)  
  
var writerGzipPool = sync.Pool{New: func() interface{} { return gzip.NewWriter(ioutil.Discard) }}  
  
func GetGzipWriter() *gzip.Writer { return writerGzipPool.Get().(*gzip.Writer) }  
  
func PutGzipWriter(buf *gzip.Writer) {  
    buf.Flush()  
    writerGzipPool.Put(buf)  
}  
  
func BenchmarkWriteGzipWithPool(b *testing.B) {  
    b.ReportAllocs()  
    for n := 0; n < b.N; n++ {  
        data := bytes.NewReader([]byte("Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque molestie."))  
        writer := GetGzipWriter()  
        _, _ = io.Copy(writer, data)  
        PutGzipWriter(writer)  
    }  
}  
  
func BenchmarkWriteGzipWithoutPool(b *testing.B) {  
    b.ReportAllocs()  
    for n := 0; n < b.N; n++ {  
        data := bytes.NewReader([]byte("Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque molestie."))  
        writer := gzip.NewWriter(ioutil.Discard)  
        _, _ = io.Copy(writer, data)  
    }  
}
```



Trendyol Group

trendyol.com

trendyol tech

trendyol express

dolap

# Optimizasyon

```
~ go test -bench=. -benchmem
```

BenchmarkWriteGzipWithPool-8	552501	2157 ns/op	139 B/op	2 allocs/op
BenchmarkWriteGzipWithoutPool-8	9798	122969 ns/op	807088 B/op	16 allocs/op



Trendyol Group

trendyol<sup>.com</sup>

trendyol<sup>tech</sup>

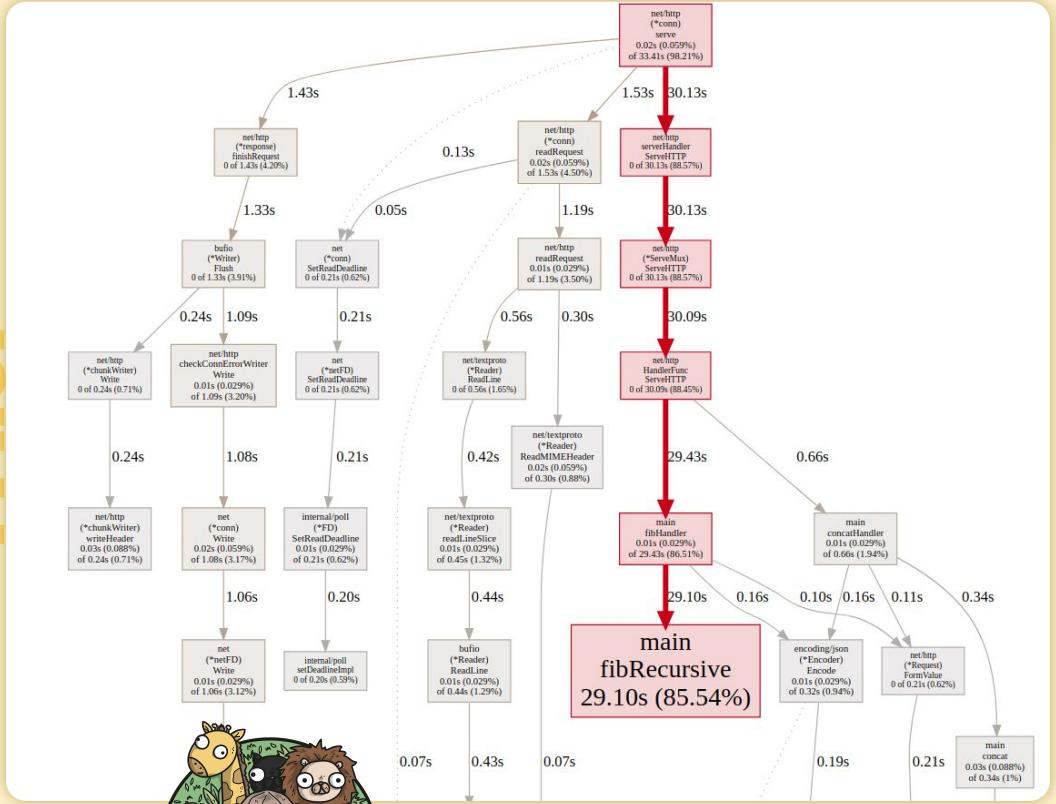
trendyol<sup>express</sup>

dolap

# Profiling, Monitoring ve Error Handling



# PPROF



Trendyol Group

trendyol.com  
trendyol  
tech

trendyol  
express

dolap

# GO TRACE



Trendyol Group

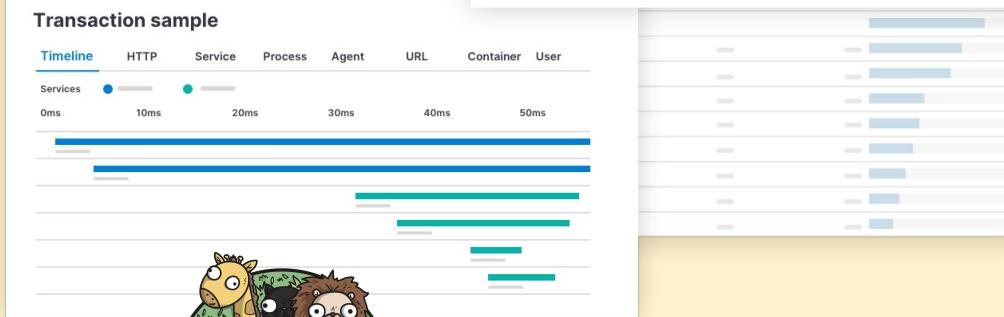
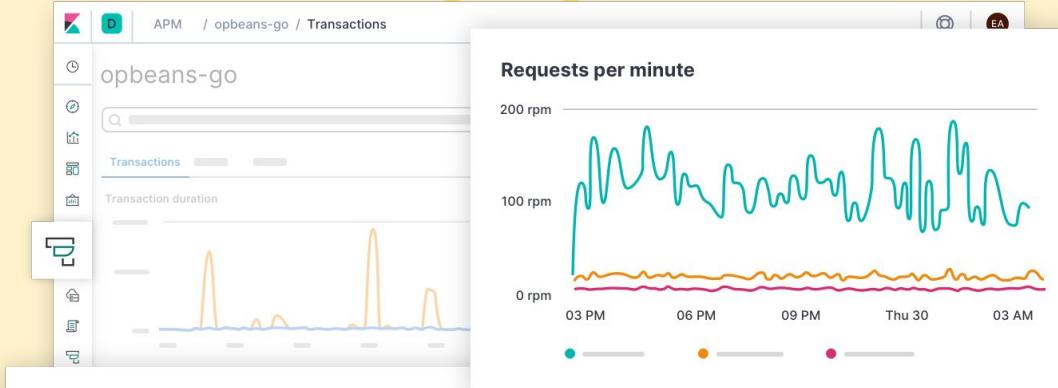
trendyol.com

trendyol  
tech

trendyol  
express

dolap

# ELASTIC APM



Trendyol Group

trendyol.com

trendyol tech

trendyol express

dolap



# Race Detection

```
~ go test -race

=====
WARNING: DATA RACE
Read by goroutine 5:
  main.func·001()
    race.go:14 +0x169

Previous write by goroutine 1:
  main.main()
    race.go:15 +0x174

Goroutine 5 (running) created at:
  time.goFunc()
    src/pkg/time/sleep.go:122 +0x56
  timerproc()
    src/pkg/runtime/ztime_linux_amd64.c:181 +0x189
=====
```



Trendyol Group

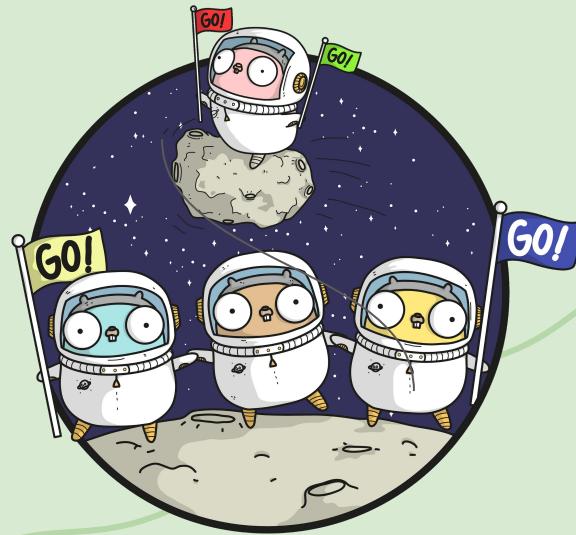
trendyol<sup>.com</sup>

trendyol  
tech

trendyol  
express

dolap

# 1.5M Throughput



1.57M

## 300Mi CPU

---

## 300Mb Memory



Trendyol Group

trendyol.com

trendyol tech

trendyol express

dolap

# Scale ve Kaynak Kazanımı

**%92,5**

Kaynak Kazanımı

**~3sn**

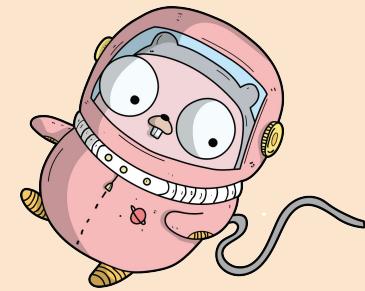
Ayağa Kalkma Süresi

**1.15dk**

Deployment Süresi

**43sn**

Hotfix Süresi



# Soru & Cevap



# Teşekkürler



[github.com/ashleymcnamara/gophers](https://github.com/ashleymcnamara/gophers)