



Deployment Golang

PART TIME PROGRAM

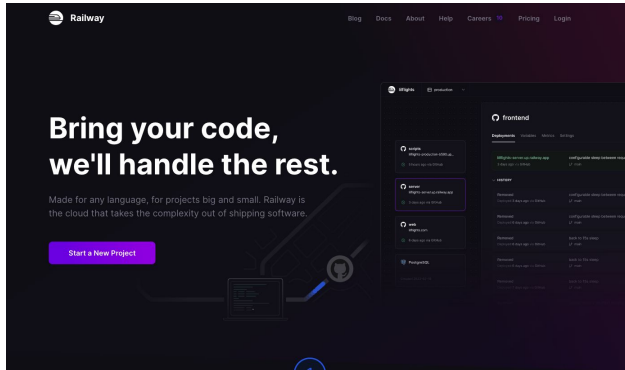


Deploy To Railway

Introduction

Railway adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu, Railway mendukung bahasa pemrograman seperti Go, Ruby, Node.js, Python, Java, PHP, dan lain-lain.

Railway termasuk ke dalam kriteria Platform As A Service (PaaS), sehingga bagi kita yang ingin melakukan deploy aplikasi ke Railway cukup hanya dengan melakukan konfigurasi aplikasi yang ingin di deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan, menjalankan, dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi.



Introduction

Langkah - langkah dalam menginstall Railway

Tahap 1 : Register di Railway.app

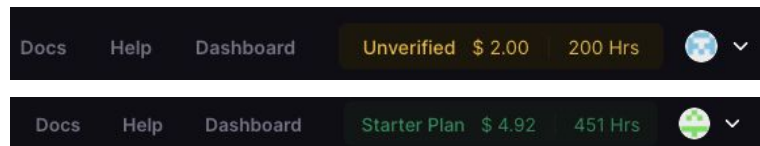
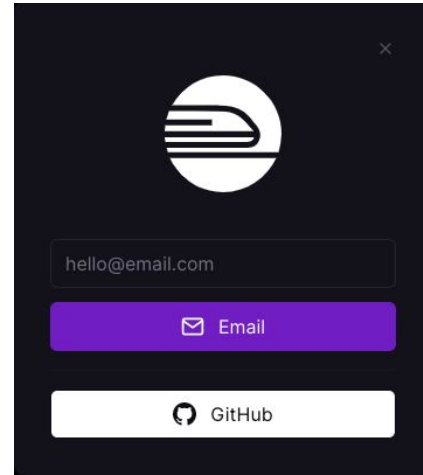
Teman teman bisa memulai dengan membuka website Railway dan melakukan register di sana. Register dilakukan dengan menekan tombol login dan bisa login dengan github.com

Tahap 2: Pairing Github.com account dengan Railway

Teman-teman yang login dengan github dapat memasukan username dan passwordnya setelah berhasil teman2 dapat melakukan konfirmasi authorisasi untuk menginstall railway pada akun github teman-teman

Tahap 3 : Lakukan verifikasi akun Railway

Ketika sudah selesai melakukan pairing github dengan Railway teman-teman dapat melakukan verifikasi akun Railwaynya dengan menyetujui beberapa ketentuan dari Railway.

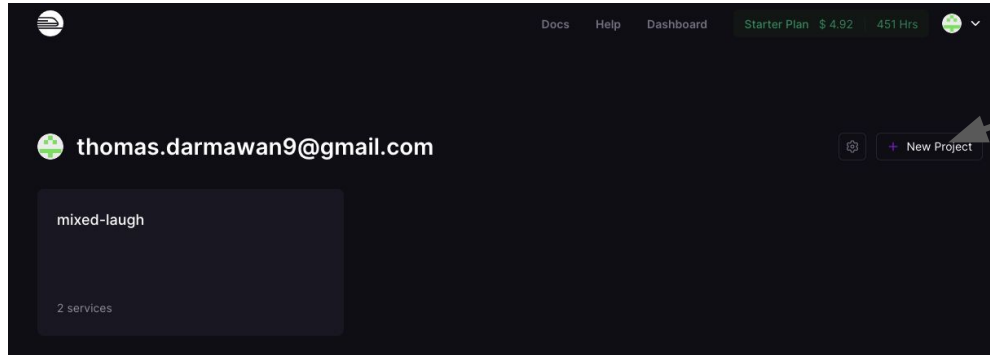


Setting Project To Deploy

Langkah - langkah dalam setting project untuk deploy di Railway

Tahap 1 : Upload project

Dari github yang sudah terhubung dengan Railway, buatlah project baru untuk di deploy dengan menekan tombol new project pada halaman dashboard

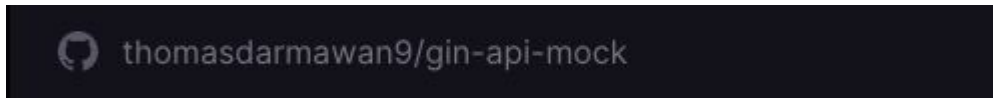
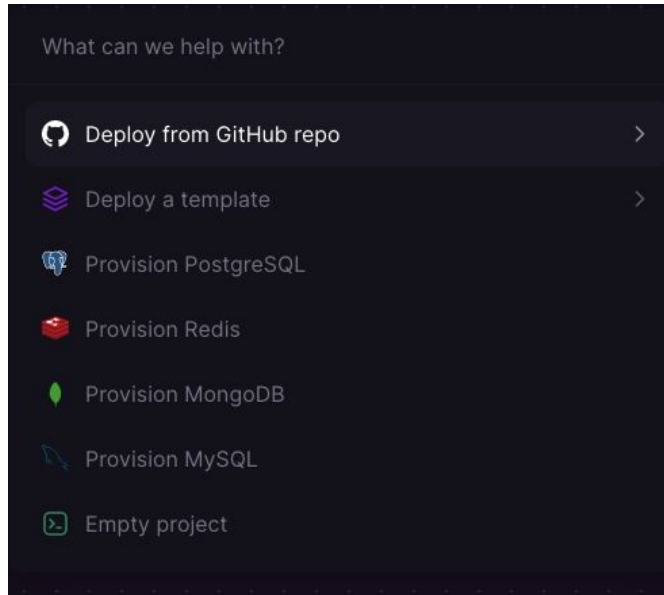


Tekan untuk membuat project baru

Setting Project To Deploy

Langkah - langkah dalam setting project untuk deploy di Railway

Setelah muncul halaman new project, teman - teman dapat memilih upload dari github repo lalu memilih project yang ingin di deploy ke Railway.

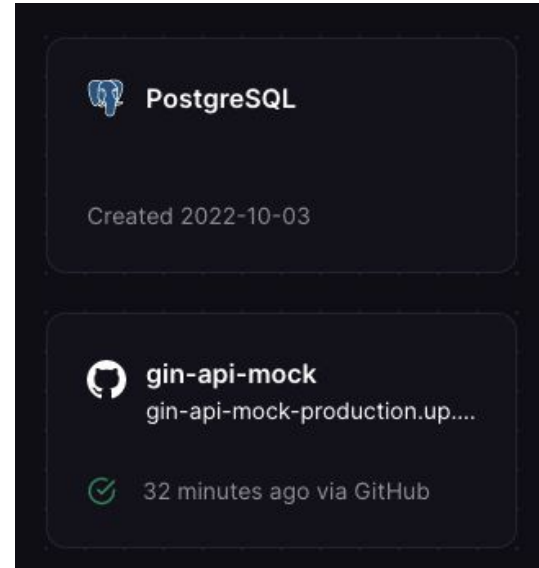
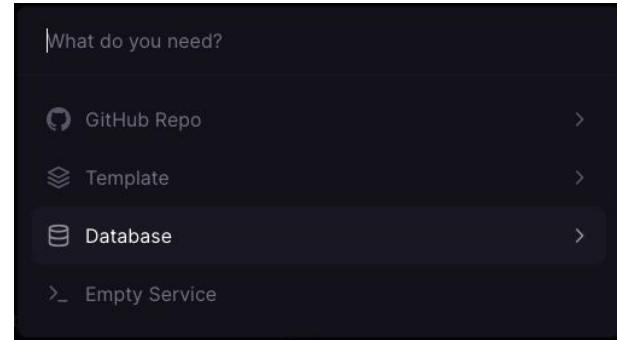
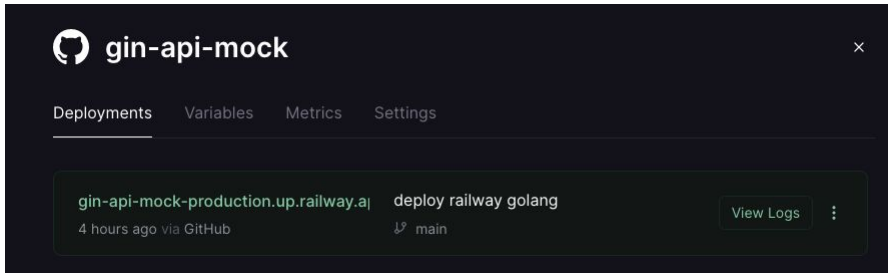


Setting Project To Deploy

Setelah project berhasil di upload dan dideploy maka hasilnya akan tampak di log deployment dengan warna hijau dan status sukses.

Berikutnya kita dapat melakukan setting postgres agar project Golang ini dapat terhubung ke database. Teman-teman dapat melakukan penambahan baru di dalam project Railway yang sama dengan memilih postgres.

Setelah terinstall database pada project tersebut maka akan tampil card database yang digunakan pada project tersebut seperti gambar berikut.



Setting Project To Deploy

Berikutnya kita dapat melakukan penyesuaian variable config DB pada codingan dengan config postgre pada Railway kita. Konfigurasi database di file db.go atau file yang berisi settingan database, teman-teman dapat gunakan **os.Getenv()** untuk mengambil value config yang ada pada file **.env**. Selain itu perlu ditambahkan `Run(":" + PORT)` agar port server sesuai pada port deployment Railway. Berikutnya pada file **.env** kita perlu sesuaikan dengan value variable yang ada pada postgre di Railway project kita seperti pada gambar berikut.

```
func main() {  
    database.StartDB()  
  
    var PORT = os.Getenv("PORT")  
  
    routers.StartServer().Run(":" + PORT)  
}
```

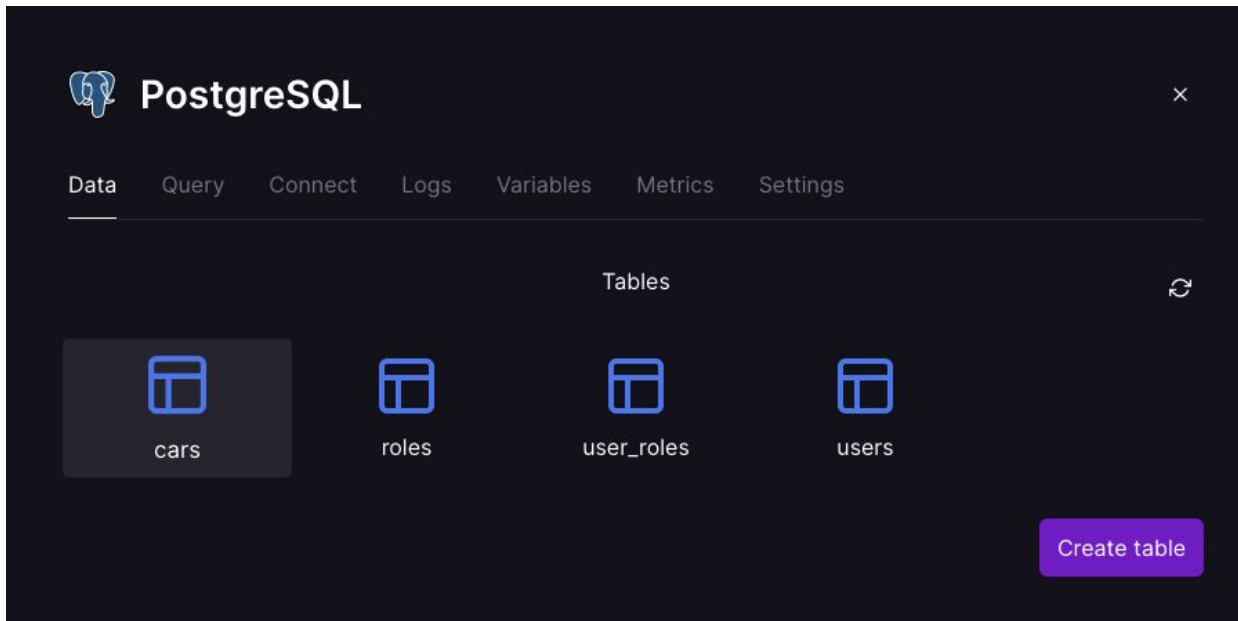
```
var (  
    host = os.Getenv("PGHOST")  
    user = os.Getenv("PGUSER")  
    password = os.Getenv("PGPASSWORD")  
    dbPort = os.Getenv("PGPORT")  
    dbname = os.Getenv("PGDATABASE")  
    db = *gorm.DB  
    err error  
)  
  
func StartDB() {  
    config := fmt.Sprintf("host=%s user=%s password=%s dbname=%s port=%s sslmode=disable",  
        host, user, password, dbname, dbPort)  
  
    db, err = gorm.Open(postgres.Open(config), &gorm.Config{})  
    if err != nil {  
        log.Fatal("error connecting to database :", err)  
    }  
  
    db.AutoMigrate(&models.Car{})  
}
```

Sesuaikan pada tab variables di postgre railway

```
1 DATABASE_URL=postgres://${{ PGUSER }}:${{ PGPASSWORD }}@${{ PGHOST }}:${{ PGPORT }}/${{ PGDATABASE }}  
2 PGDATABASE=railway  
3 PGHOST=containers-us-west-80.railway.app  
4 PGPASSWORD=dIoZwlyuHiVVbT9vABVe  
5 PGPORT=5583  
6 PGUSER=postgres
```

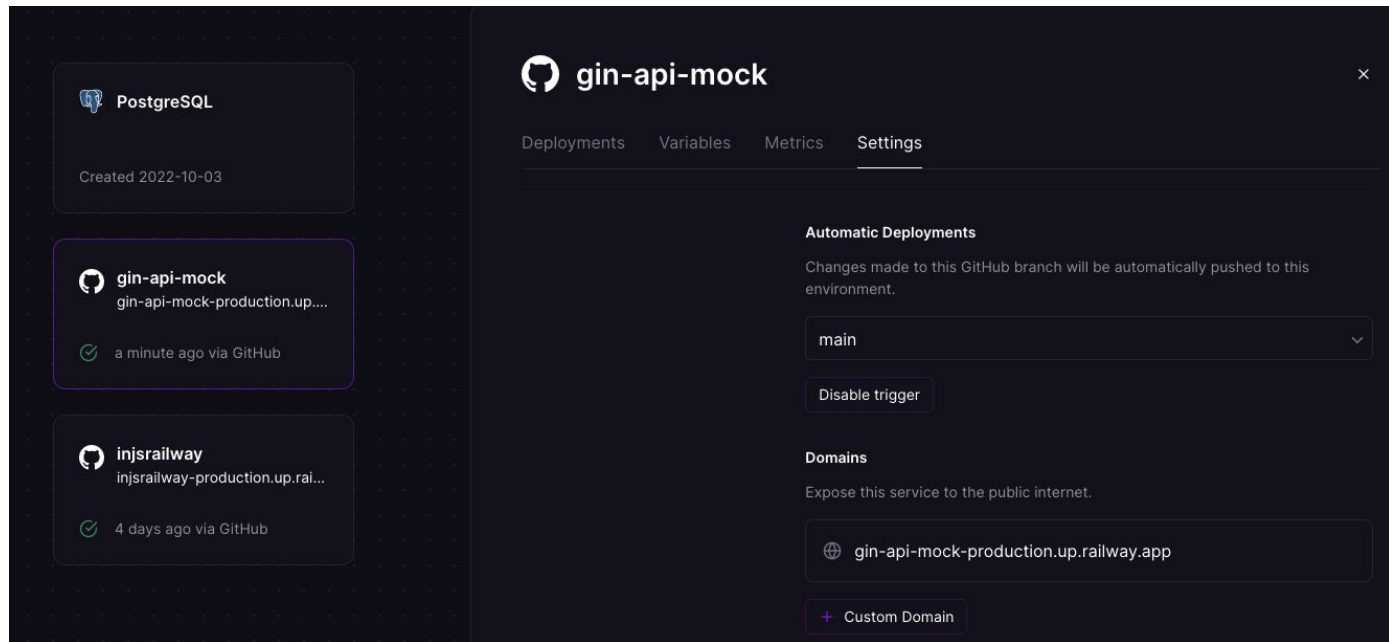

Setting Project To Deploy

Jika variable pada postgres di Railway dan config db di codingan sudah benar dan sama maka setelah dipush di git dan menunggu beberapa menit akan tampil beberapa table yang kita buat sebelumnya pada codingan agar melakukan migration secara otomatis.



Setting Project To Deploy

Untuk melihat nama domain dari project yang kita deploy di Railway kita dapat melihat pada tab setting di card project github kita yang kita deploy. Apabila belum ada domain pada project tersebut kita dapat klik tombol generate domain.

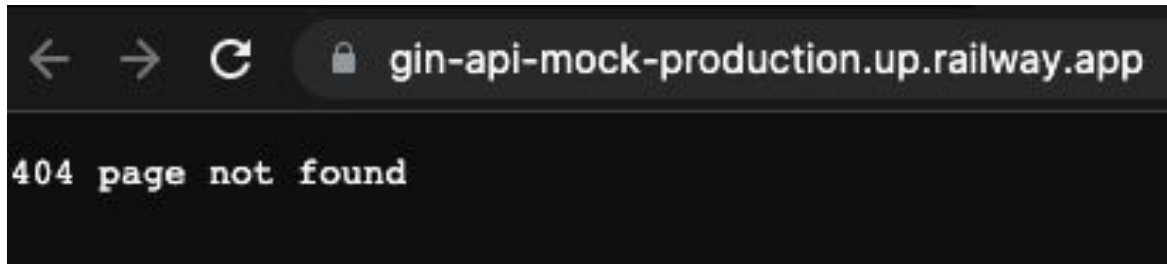


The screenshot displays the Railway dashboard interface. On the left, a sidebar lists three projects: PostgreSQL (created 2022-10-03), gin-api-mock (deployed a minute ago via GitHub), and injsrailway (deployed 4 days ago via GitHub). The main panel shows the 'Settings' tab for the 'gin-api-mock' project. Under 'Automatic Deployments', it indicates that changes to the GitHub branch will be automatically pushed to the environment, with a dropdown menu set to 'main' and a 'Disable trigger' button. Under 'Domains', it shows the current domain 'gin-api-mock-production.up.railway.app' and a '+ Custom Domain' button.



Setting Project To Deploy

Jika tampilan aplikasi kita pada browser sudah seperti pada gambar dibawah ini, maka itu menandakan bahwa kita telah berhasil mendeploy aplikasi kita pada Railway.



Setting Project To Deploy

Mari kita coba eksekusi endpoint create cars pada **Postman** dengan menggunakan URL yang sudah tergenerate oleh Railway.

Contohnya seperti pada gambar dibawah ini.

